

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

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique  
DEPARTEMENT D'automatique

**Mémoire de Fin d'Etude  
De MASTER ACADEMIQUE**  
Spécialité : **commande des systèmes**

*Présenté par*  
**Ahmed HAMOUANI**

Mémoire dirigé par **Pr. Moussa DIAF**

Thème

**Etude et réalisation d'une commande  
d'un système composé d'un robot, d'un  
plateau tournant et d'une caméra**

*Mémoire soutenu publiquement le 28 septembre 2017 devant le jury composé de :*

**Mem Nadia DJEGHALI**

MCA, UMMTO, Présidente

**M Moussa DIAF**

Professeur, UMMTO, Encadreur

**Mem Ouardia CHILALI**

MAA, UMMTO, Examinatrice

**M Takfarinas CHELLI**

MAA, UMMTO, Examineur

---

## *Avant-propos*

*Ce travail a été réalisé au Laboratoire Vision Artificielle et Automatique des Systèmes (LVAAS) du département Automatique (LVAAS), FGEI, UMMTO. Il fait suite au travail d'un mémoire de Magister soutenu et réalisé dans ce même laboratoire. Nous avons apporté beaucoup d'améliorations par rapport à la commande du moteur de la table, à la carte de commande, au système de vision et à la commande du robot.*

*Nous tenons à remercier vivement Monsieur DIAF Moussa, Professeur pour nous avoir proposé ce sujet et pour nous avoir guidé et dirigé durant tout ce travail.*

*Nos vifs remerciements vont aussi à Messieurs YOUNSI Marzouk, YESLI Samir et IRATNI Salim, doctorants au LVAAS pour nous avoir aidés et encouragés durant tout ce projet.*

*Nous ne manquerons pas de remercier madame DJEGHALI Nadia, MCA au département Automatique, FGEI, UMMTO, pour nous avoir fait honneur en acceptant de présider le jury de ce mémoire.*

*Nous exprimons notre reconnaissance à madame CHILALI Ouardia, MAA au département Automatique, FGEI, UMMTO, pour avoir accepté de faire partie du jury d'examen de notre mémoire.*

*Nos sincères remerciements vont aussi à Monsieur CHELLI Takfarinas, MAA au département Automatique, FGEI, UMMTO pour avoir accepté de faire partie du jury d'examen de notre mémoire.*

## *Dédicaces*

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.*

*A ma chère sœur, Imane, pour ses encouragements permanents, et son soutien moral.*

*A mes chers frères, Khaled et Anis , pour leur appui et leur encouragement.*

*A toute ma famille pour leur soutien tout au long de mon parcours universitaire.*

*A mes chères amies, Sabrina, Lila , Thiziri, Ryma, Kenza, qui n'ont cessé de m'encourager et de me motiver tout au long mon parcours universitaire.*

*A mes chers amis, Bilal, Rabah, Malek, Moh, Yahia, qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible.*

*Merci d'être toujours là pour moi.*

# Sommaire

**Introduction générale** ..... 1

## **Premier chapitre : Commande du plateau tournant**

1.1. Introduction..... 4

1.2. Description du plateau tournant..... 5

    1.2.1. Composants du plateau tournant ..... 6

    1.2.2. Configuration matérielle ..... 6

1.3 Identification du système ..... 8

    1.3.1. Méthode d'identification..... 9

    1.3.2. Identification du système réel..... 10

1.4. Commande du plateau tournant..... 15

    1.4.1. Passage au système discret ..... 15

    1.4.2. Synthèse du correcteur..... 16

        1.4.2.1. Choix du correcteur..... 17

        1.4.2.2. *Implémentation du correcteur*..... 22

1.5. Conclusion ..... 24

## **Deuxième chapitre : Commande du bras manipulateur**

2.1 Introduction ..... 26

2.2 Description du bras manipulateur..... 27

2.3 Modélisation ..... 28

    2.3.2 Modèle géométrique direct ..... 29

    2.3.3. Modèle géométrique inverse ..... 34

2.4. Commande du bras manipulateur..... 37

2.4.1. Algorithme de commande.....	38
2.4.2. Implémentation du programme et testes pratiques.....	39
2.5. Conclusion .....	41

### **Troisième Chapitre : système de vision artificielle**

3.1. Introduction.....	42
3.2. Détection de présence .....	43
3.3. Localisation de l'objet.....	47
3.3.1. Position de l'objet par rapport au repère fixe.....	47
3.3.2. Position de l'objet dans l'image.....	49
3.4. Reconnaissance de forme.....	51
3.4.1. Détection des contours .....	51
3.4.2. Méthode de classification .....	52
3.5. Conclusion .....	58

<b>Conclusion générale</b> .....	59
----------------------------------	----

### **Annexes**

### **Bibliographie**

## Table des figures

<b>Fig.i.1</b> : Synoptique du système réel. ....	2
<b>Fig.1.1</b> : Schéma explicatif des composants du plateau tournant. ....	5
<b>Fig.1.2</b> : réducteur de type roue et vis sans fin.....	6
<b>Fig.1.3</b> : Réponse du moteur. ....	7
<b>Fig.1.4</b> : Circuit de filtrage.....	7
<b>Fig.1.5</b> : Réponse du moteur avant et après filtrage.....	8
<b>Fig.1.6</b> : Schéma d'identification. ....	9
<b>Fig.1.7</b> : Outil d'identification MATLAB. ....	10
<b>Fig.1.8</b> : Schéma d'acquisition de valeurs pour l'identification.....	11
<b>Fig.1.9</b> : Introduction des données. ....	12
<b>Fig.1.10</b> : Choix de l'ordre de la fonction de transfert. ....	12
<b>Fig.1.11</b> : Estimation de la fonction de transfert.....	13
<b>Fig.1.12</b> : Schéma de simulation pour visualiser les deux réponses (réel et estimée). .....	14
<b>Fig.1.13</b> Réponse réel et estimée. ....	14
<b>Fig.1.14</b> Réponses indicielles de $g(s)$ et $g(z)$ .....	16
<b>Fig.1.15</b> Implémentation du correcteur dans la boucle de l'asservissement.....	17
<b>Fig.1.16</b> Réponses des correcteurs proportionnels.....	18
<b>Fig.1.17</b> Choix des performances désirées.....	19
<b>Fig.1.18</b> Schéma de simulation pour l'autoréglage du correcteur. ....	20
<b>Fig.1.19</b> Calcul des paramètres du correcteur.....	20
<b>Fig.1.20</b> Réponse du système corrige.....	21
<b>Fig.1.21</b> : réponse du système corrige avec une perturbation de type échelon. ....	21
<b>Fig.1.22</b> : réponse du système corrige avec une perturbation de type rampe. ....	22
<b>Fig.1.23</b> réponse du système discret corrige.....	23
<b>Fig.1.24</b> Réponse du système réel corrigé.....	23
<b>Fig.1.25</b> Organigramme de l'implémentation du correcteur .....	24
<b>Fig.2.1</b> : Composants du bras manipulateur. ....	27
<b>Fig.2.2</b> : Paramètres structuraux du bars manipulateur.....	28
<b>Fig.2.3</b> : Représentation des repères sur le robot. ....	30
<b>Fig.2.4</b> : repère associé a l'articulation des la pince. ....	32
<b>Fig.2.5</b> : espace de travail du robot.....	38
<b>Fig.2.6</b> : représentation des angles commandés par les moteurs. ....	38
<b>Fig.2.7</b> : Organigramme de la commande du robot. ....	39
<b>Fig.2.8</b> : nuage de points marqué par le robot. ....	41
<b>Fig.3.1</b> : Les quatre classes d'objets.....	44
<b>Fig.3.2</b> : Images acquises des quatre objets.....	45
<b>Fig.3.3</b> : Images aux niveaux de gris. ....	45
<b>Fig.3.4</b> : Images aux niveaux de gris filtrées.....	45

<b>Fig.3.5</b> : Histogrammes des images en présence des objets. ....	46
<b>Fig.3.6</b> : Histogramme de l'image sans objet.....	46
<b>Fig.3.7</b> : Organigramme de la détection de présence. ....	47
<b>Fig.3.8</b> : Position de la caméra par rapport au repère fixe. ....	48
<b>Fig.3.9</b> : position du champ de vision sur le plateau tournant. ....	49
<b>Fig.3.10</b> : Surface couverte par un pixel sur le plateau.....	49
<b>Fig.3.11</b> : Résultat de la binarisation des images. ....	51
<b>Fig.3.12</b> : résultat de la détection des contours.....	52
<b>Fig.3.13</b> : Contours et centres de gravité représentés sur les images acquises. ....	53
<b>Fig.3.14</b> : les signatures polaires des quatre classes d'objets.....	54
<b>Fig.3.15</b> : Signatures polaires des différents échantillons de chaque classe. ....	55
<b>Fig.3.16</b> : Signatures polaires moyennes des quatre classes d'objets ....	56
<b>Fig.3.17</b> : organigramme général de l'application.....	57

## Table des tableaux

<b>Tab.2.1</b> : tableau des paramètres de Denavit-Hartenberg. ....	31
<b>Tab.2.2</b> : paramètres de Denavit-Hartenberg du dernier repère. ....	32
<b>Tab.2.3</b> : tableau des positions limites des articulations.....	37

## Abréviations

2D : deux dimensions.

3D : trois dimensions.

cm : centimètre.

mm : millimètre.

ALU(UAL) : Unité arithmétique et logique.

DSP (PSN) : Processeur de signal numérique.

MGD : model géométrique directe.

MGI : model géométrique inverse.

## **Introduction générale**

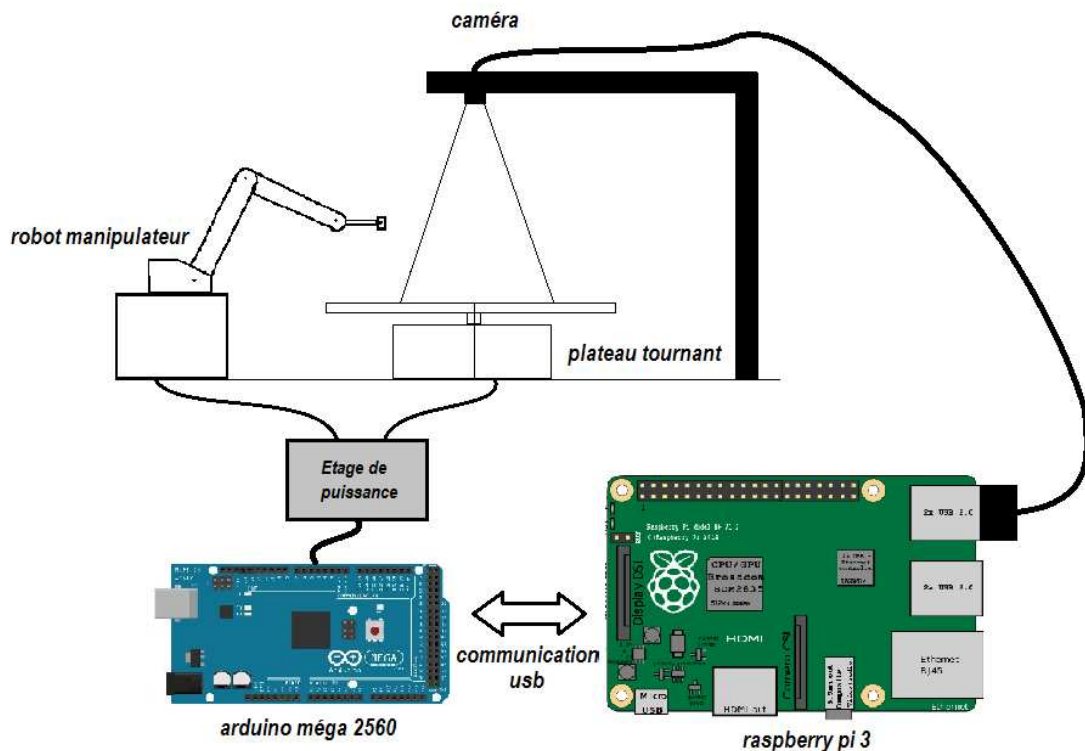
L'un des plus grands rêves de l'homme est de pouvoir faire exécuter par des acteurs autres que lui-même des tâches ou activités qu'il considère comme aliénantes, dangereuses ou simplement ennuyeuses et non gratifiantes, chose qui a commencé à se réaliser avec l'émergence de la robotique au cours des années 1960.

Depuis leur apparition, les robots industriels se sont concentrés à réaliser des tâches classiques qui peuvent toujours se ramener à des interactions précises avec leur environnement. De ce fait, une modification de ce dernier peut entraîner de graves conséquences sur le déroulement de la tâche à accomplir. C'est ce qui a poussé les chercheurs à équiper le robot de capteurs de vision extéroceptifs, afin de lui permettre d'appréhender son environnement et de réagir à d'éventuels changements de ce dernier. Ce qui a permis d'améliorer de manière significative leurs performances et d'aider à développer de nouvelles applications d'une part et l'émergence d'un nouveau domaine de recherche sous le nom d'asservissement visuel, d'une autre part.

Le travail réalisé dans ce mémoire s'inscrit dans le domaine d'asservissement visuel. En effet, le travail se limite à l'étude, sur le plan théorique et pratique, de la commande d'un système comportant un plateau tournant, un robot manipulateur et une caméra. Le fonctionnement de cet ensemble est piloté par un ordinateur (Raspberry pi 3) et une carte d'acquisition et de commande (Arduino Méga 2560). La figure i.1 représente la configuration du système réel. L'objectif consiste à détecter, localiser et reconnaître en temps réel des objets arrivant sur le plateau tournant, puis de les ramener en face du robot à l'aide du plateau pour que le robot réalise par la suite la tâche de saisie et de déplacement des objets vers des emplacements correspondants aux types d'objets reconnus.

Les objets à poser sur le plateau sont classés suivant leurs formes géométriques en quatre classes :

- classe des objets carrés
- classe des objets circulaires
- classe des objets triangulaires
- classe des objets hexagonaux



**Fig.i.1** Synoptique du système réel.

L'objectif principal de cette étude est de développer un programme de commande qui permet à l'ordinateur de gérer à travers la carte d'acquisition et de commande le fonctionnement du plateau tournant, du système de vision artificielle et du bras manipulateur.

Pour ce faire, trois parties ont été effectuées.

Dans la première, nous avons présenté le plateau tournant, puis nous avons identifié le modèle mathématique de ce dernier et enfin le calcul et

l'implémentation de la loi de commande assurant un fonctionnement performant du plateau tournant.

En second lieu, nous sommes intéressés à la commande du bras manipulateur où nous avons brièvement présenté la configuration de notre robot puis nous avons procédé à une modélisation géométrique directe et inverse et enfin l'élaboration et l'implémentation de l'algorithme de commande permettant au robot de réaliser sa tâche, la saisie et de déplacement des objets présents sur le plateau.

En dernier lieu la partie vision artificielle est réalisée. Nous avons élaboré l'algorithme permettant au système le prétraitement de l'image et la détection de présence d'un objet sur le plateau tournant, puis la localisation de l'objet en se basant sur le principe des moments géométriques dans l'image et enfin la reconnaissance de forme en utilisant la méthode de la distance euclidienne minimale appliquée sur la signature polaire de l'objet en question.

Le mémoire est organisé comme suit :

- Le premier chapitre est dédié à la description détaillée et à la commande du plateau tournant.
- Le deuxième chapitre est consacré à la modélisation et au développement du programme de commande du bras manipulateur.
- Le troisième et dernier chapitre expose la partie relative à la vision artificielle ou son réalisées la détection de présence, la localisation et la reconnaissance d'objet.

Le mémoire se termine par une conclusion générale sous forme d'un bilan global sur l'ensemble des parties réalisées tout en présentant certaines perspectives qui peuvent être envisagées.

# **Chapitre 1**

## ***Commande du plateau tournant***

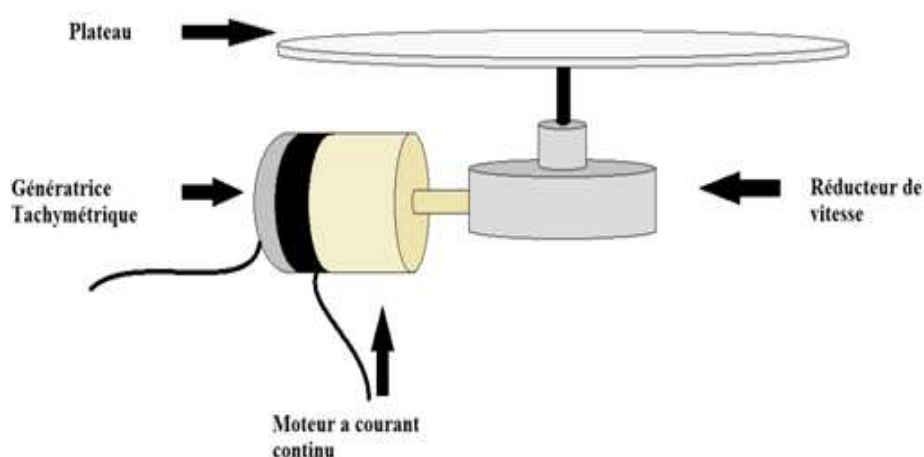
### **1.1. Introduction**

Durant ces dernières années, la commande des machines électriques a subi des progrès significatifs, ce qui a permis le développement de solutions numériques efficaces avec une possibilité d'implanter des algorithmes plus complexes. Les premières implantations de commande de machines électriques furent réalisées avec des solutions analogiques. Ces solutions assuraient la réalisation de contrôles ayant une large bande passante et quasiment en temps réel. Cependant, elles manquaient de fiabilité vu leur sensibilité aux perturbations et aux variations de paramètres de contrôle liées aux contraintes thermiques des circuits analogiques de contrôle. Par la suite, les solutions numériques se sont naturellement imposées afin de remédier à ces inconvénients.

De nos jours, les implantations numériques sont en majorité basées sur les microprocesseurs et les DSP (Digital Signal Processor). Ces solutions numériques sont équipées d'unités arithmétiques et logiques ALU (Arithmetic Logic Unit) dédiées à la réalisation des calculs arithmétiques et logiques des algorithmes de commandes. Elles

intègrent également des périphériques tels que les convertisseurs analogiques/numériques et les « timers » bien adaptés aux besoins de commande de machines électriques. Leur utilisation a permis de résoudre les problèmes liés à l'utilisation des commandes analogiques. Par ailleurs, elles présentaient un grand intérêt économique et une meilleure flexibilité de conception. Il est à noter aussi que ces solutions numériques sont des solutions purement logicielles. En effet, les conceptions qui leur sont associées sont conçues en utilisant des architectures prédéfinies. Le concepteur ne peut donc pas agir sur la partie matérielle de la conception et n'a accès qu'à la partie logicielle. D'un autre côté, malgré les avantages offerts par ces solutions numériques, certains avantages offerts par les implantations analogiques sont perdus. Cela est principalement dû au fait que la discrétisation et la quantification des algorithmes de commande à implanter, ainsi que les délais de temps de calcul détériorent les performances de contrôle en termes de rapidité de correction et de résolution de contrôle [1].

## 1.2. Description du plateau tournant



**Fig.1.1** : Schéma explicatif des composants du plateau tournant.

### 1.2.1. Composants du plateau tournant

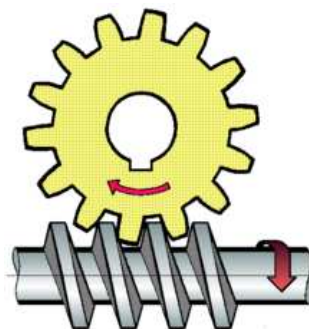
Le système est composé de quatre parties :

- un moteur à courant continu à aimant permanent dont la tension et la vitesse nominale sont respectivement 12 volts et 2500 tr/min.
- un capteur de vitesse qui est une génératrice tachymétrique linéaire qui produit une tension proportionnelle à la vitesse de rotation du moteur dont la relation est :

$$U_c = K_c * \Omega_m \quad (1.1)$$

Où  $K_c = 0.0013$  volts/tours/minutes (sensibilité du capteur)

- un réducteur de vitesse de type roue et vis sans fin (comme le montre la figure ci-dessous) dont le rapport de réduction est  $K_r = 1/69$ . Ce dernier réduit la vitesse du moteur de 69 fois mais augmente aussi la puissance avec le même rapport ce qui permet au moteur de déplacer des charges importantes et éviter de le faire tourner à basse vitesse.



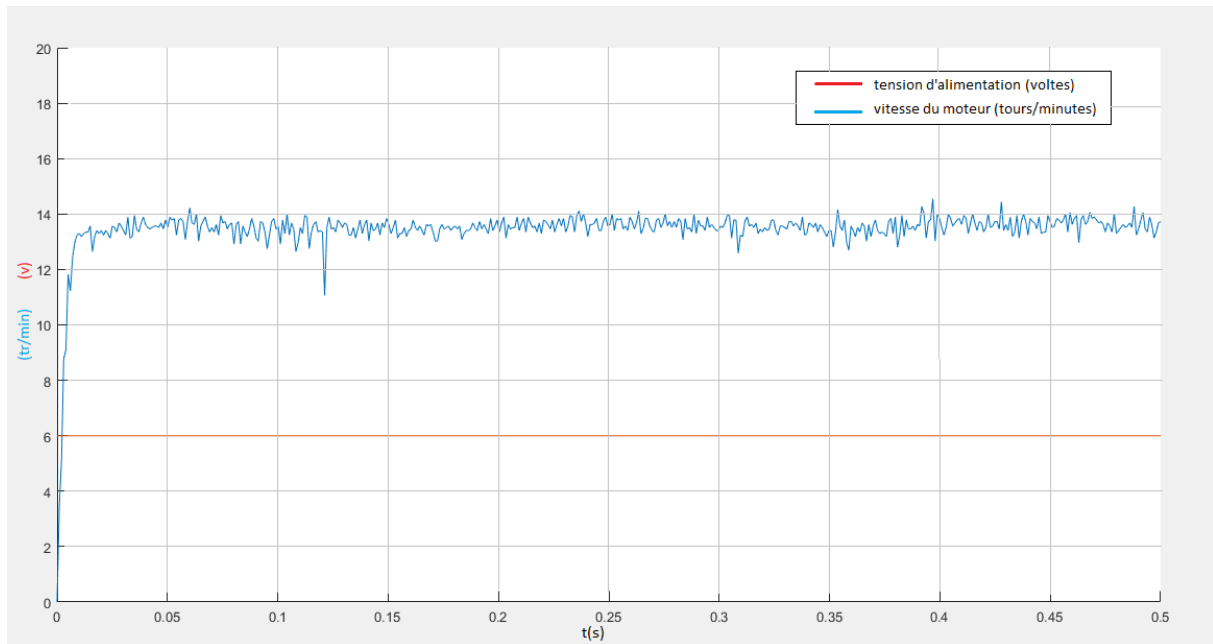
**Fig.1.2** : réducteur de type roue et vis sans fin

- un plateau monté sur le réducteur dont le diamètre est de 48 centimètres et d'épaisseur 0.4 centimètres

### 1.2.2. Configuration matérielle

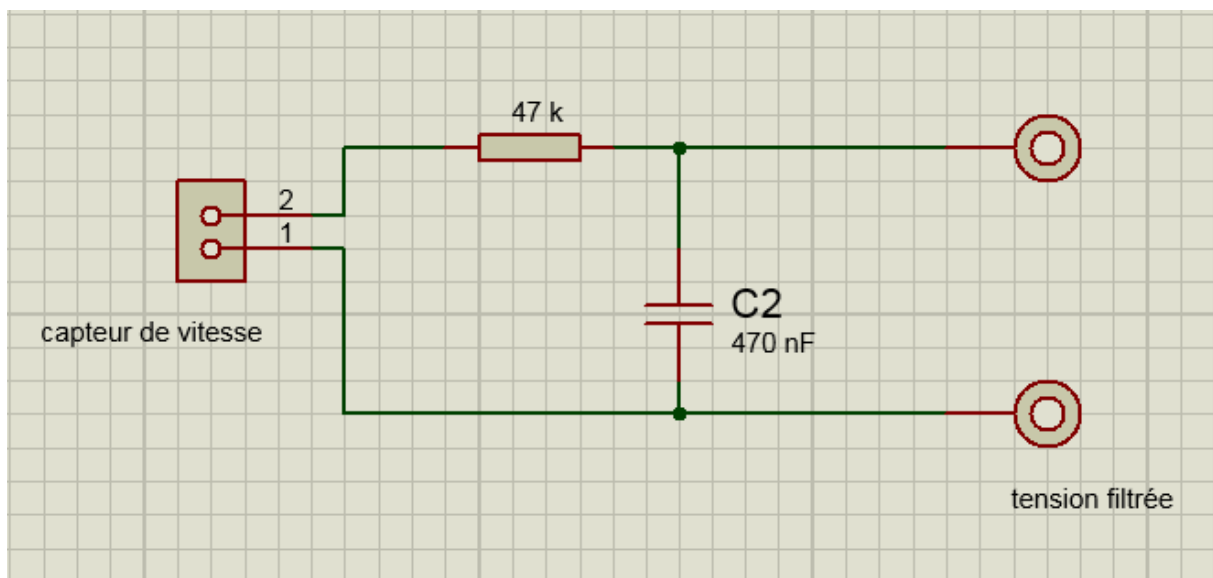
Pour la commande du plateau nous avons réalisé le circuit de commande (étage de puissance) à base de circuit intégré L293D qui permet d'alimenter le moteur avec une tension entre 0 et 12 volts

En branchant le capteur sur un circuit d'acquisition et en alimentant le moteur avec une tension d'amplitude de 6 volts on obtient la repense suivante :



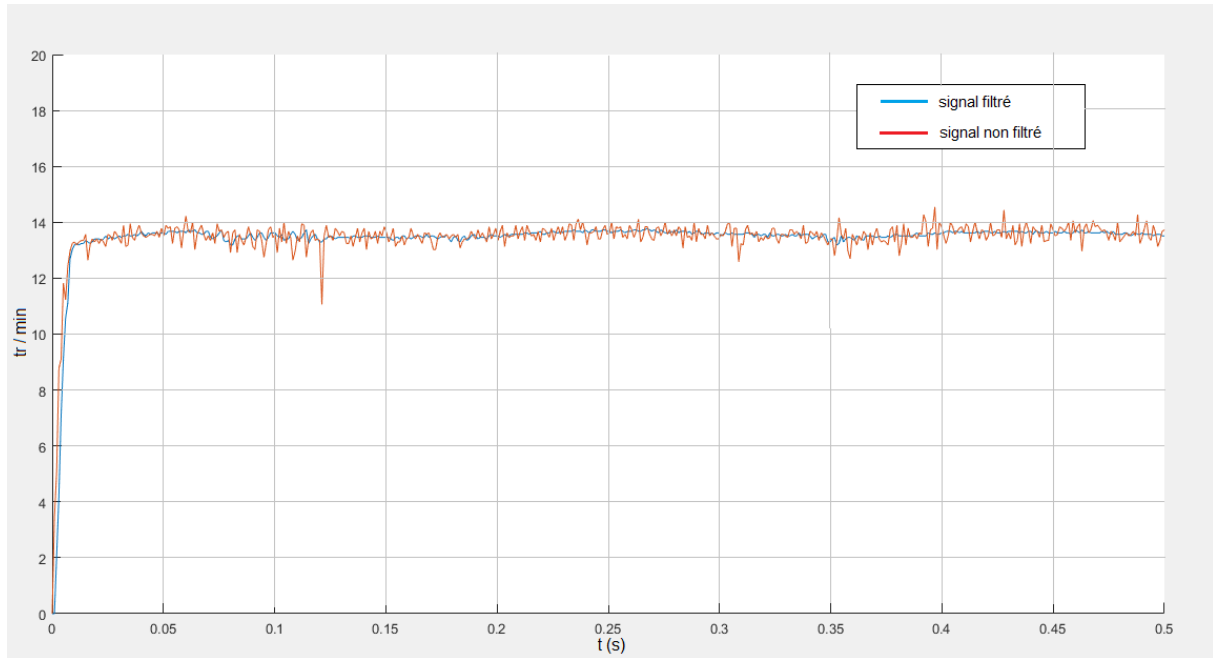
**Fig.1.3** : Réponse du moteur.

Nous remarquons qu'un bruit se superpose sur le signal issu du capteur sous forme d'ondulation de haute fréquence, et pour atténuer ces oscillations nous avons rajouté un filtre passe-bas conformément à la figure 1.4 suivante :



**Fig.1.4** : Circuit de filtrage.

Ce filtre a une fréquence de coupure d'environ 7 hertz, ce qui permet d'atténuer toutes les fréquences supérieures à cette fréquence. Ainsi la réponse du capteur est donnée sur la figure 1.5 suivante :



**Fig.1.5** : Réponse du moteur avant et après filtrage.

### 1.3 Identification du système

Pour commander un moteur à courant continu on doit avoir son modèle mathématique pour pouvoir élaborer une loi de commande qui va assurer le fonctionnement désiré. Pour cela, une modélisation est nécessaire pour obtenir une relation mathématique liant la tension de commande à la vitesse de rotation du moteur. Les lois de la physique nous permettent de donner différentes équations régissant le fonctionnement de la machine à courant continu. On les rappelle brièvement:

$$U = R * I + L \frac{dI}{dt} + E \quad (1.2)$$

$$J \frac{d\Omega}{dt} = C - C_r \quad (1.3)$$

$$C = K * I \quad (1.4)$$

$$E = K * \Omega \quad (1.5)$$

où  $R$  est la résistance d'induit,  $U$  la tension d'alimentation,  $I$  le courant de l'inducteur,  $L$  l'inductance,  $J$  le moment d'inertie de la machine par rapport à l'axe de rotation du rotor et  $C_r$ , le couple résistant ramené au rotor,  $K$ , le coefficient électromécanique et  $\Omega$ , la vitesse de rotation du moteur.

Mais dans notre cas la modélisation est une tâche très compliquée à réaliser à cause de l'absence d'information sur les paramètres du moteur (R, L, K, etc..) Pour cela, une identification s'impose.

### 1.3.1. Méthode d'identification

L'identification consiste à chercher les paramètres du modèle mathématiques du système à partir de données expérimentales et de connaissances disponibles a priori. Ces paramètres peuvent avoir une signification physique comme dans les modèles de connaissance (issues de lois de la mécanique, d'électricité, etc.), ou ne pas en avoir comme c'est le cas pour les modèles de comportement. Dans les deux cas, ils doivent fournir une approximation fidèle des comportements du système physique, dans la mesure où leurs paramètres sont ajustés à partir de données expérimentales. L'objectif recherché est de rendre identiques les réponses du processus et du modèle, pour des séquences d'entrée données.

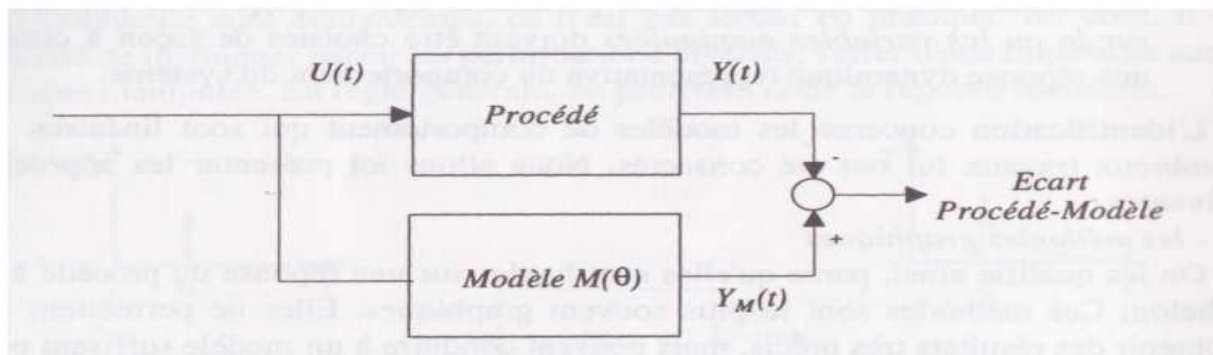


Fig.1.6 : Schéma d'identification.

Dans notre cas nous allons réaliser l'identification à l'aide de la méthode des moindres carrés. Le principe général de cette méthode est de choisir le jeu de paramètres d'un modèle que l'on définira de telle sorte qu'il minimise la somme des carrés de la différence entre les valeurs prédites par le modèle et les valeurs observées.

Pour l'identification des paramètres, on dispose d'un vecteur de mesures :

$$Y = [Y_1 Y_2 \dots Y_N]^T$$

Et on veut construire un vecteur estimé, via le modèle et ses paramètres :

$$\hat{Y} = [\hat{Y}_1 \hat{Y}_2 \dots \hat{Y}_N]^T$$

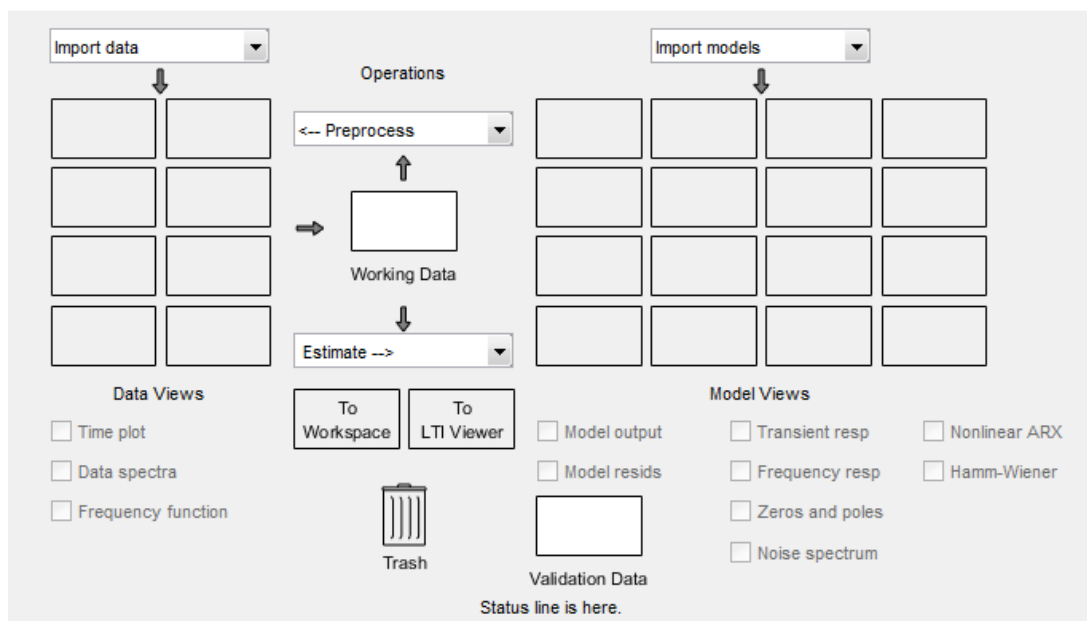
Il convient donc d'établir le modèle permettant de calculer la variable  $\hat{Y}$ . On cherche ce modèle sous forme discrète:

$$\hat{Y}(k+1) = -(a_1 \hat{Y}(k) + a_2 Y(k-1) + \dots + a_N Y(k-N-1)) + (b_1 u(k) + b_2 u(k-1) + \dots + b_M u(k-M-1)) \quad (1.6)$$

Nous supposons qu'une séquence  $(u(1), u(2), \dots, u(N))$  est appliquée en entrée et que la séquence des sorties correspondantes est observée  $(y(1), y(2), \dots, y(N))$ .

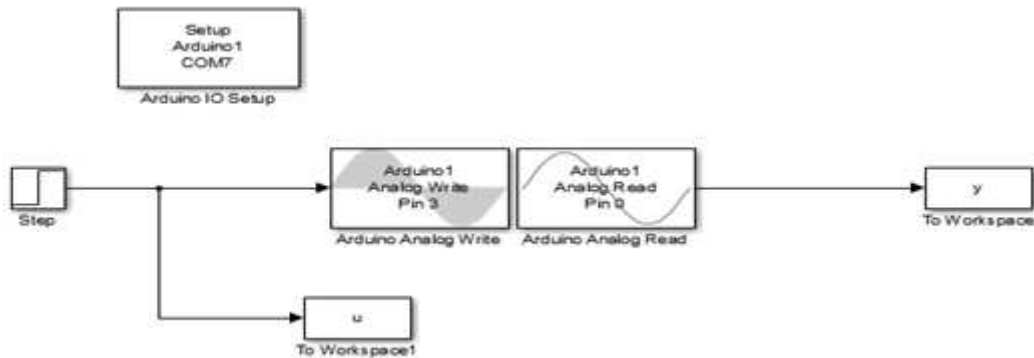
### 1.3.2. Identification du système réel

Pour réaliser cette identification on utilise l'outil Identification de MATLAB, et pour procéder, dans l'onglet « APPS » de la fenêtre MATLAB on lance l'application « System Identification » qui est représentée dans la figure suivante :



**Fig.1.7** : Outil d'identification MATLAB.

Pour l'acquisition des données nous avons réalisé une liaison entre SIMULINK et une carte d'acquisition comme le montre la figure 1.8 suivante :



**Fig.1.8** : Schéma d'acquisition de valeurs pour l'identification.

Les données acquises sont deux vecteurs  $\mathbf{U}$  et  $\mathbf{Y}$  qui représentent respectivement le vecteur d'entrée et le vecteur de sortie.

D'après la réponse représentée auparavant (fig.1.6), le système présente une réponse apériodique avec un démarrage en 's', ce qui correspond à un système du deuxième ordre avec deux pôles réels négatifs et sans retard.

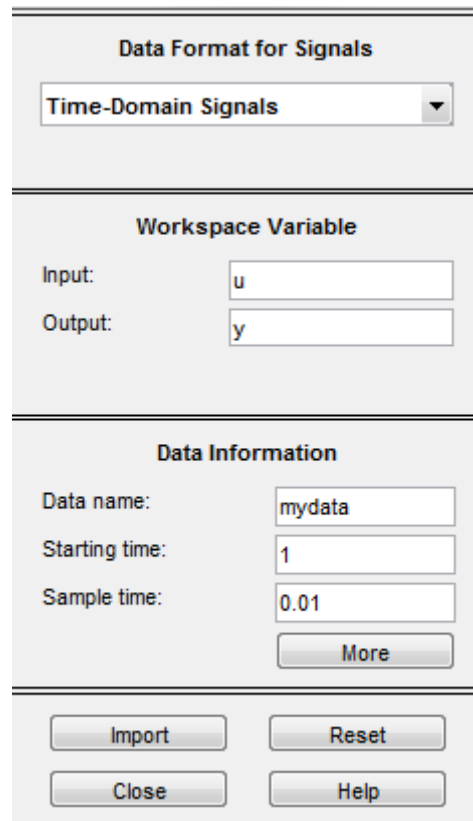
Les étapes de l'identification sont les suivantes :

- Importation des données

La fonction « Import Data » nous permet d'introduire les vecteurs de données  $\mathbf{U}$  et  $\mathbf{Y}$  qui correspondent respectivement à l'entrée et à la sortie du système comme le montre la figure (1.9).

- Le choix de l'ordre et du type de la fonction de transfert :

La fenêtre sur la figure (1.10) nous permet de choisir l'ordre du système (nombre de pôles et de zéros) ainsi que le type de la fonction de transfert (continue ou discrète).



**Data Format for Signals**

Time-Domain Signals

**Workspace Variable**

Input: u

Output: y

**Data Information**

Data name: mydata

Starting time: 1

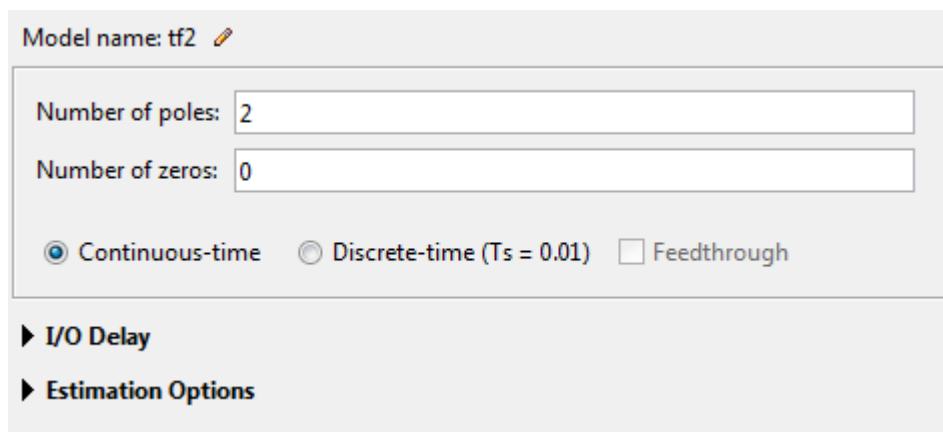
Sample time: 0.01

More

Import Reset

Close Help

**Fig.1.9** : Introduction des données.



Model name: tf2

Number of poles: 2

Number of zeros: 0

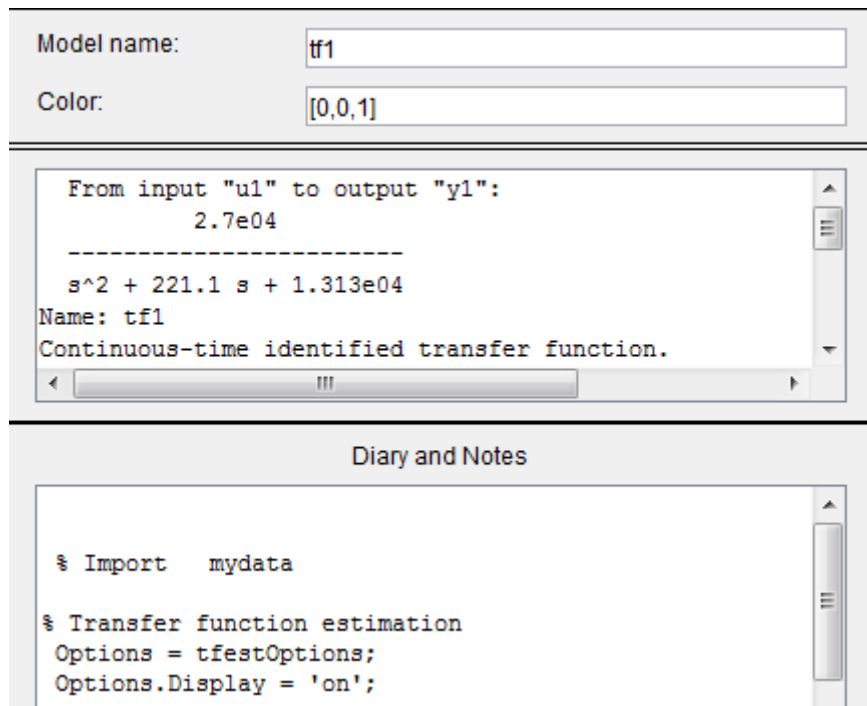
Continuous-time  Discrete-time ( $T_s = 0.01$ )  Feedthrough

► I/O Delay

► Estimation Options

**Fig.1.10** : Choix de l'ordre de la fonction de transfert.

La figure suivante nous montre le résultat de l'identification sous forme de fonction de transfert continue du deuxième ordre.



**Fig.1.11** : Estimation de la fonction de transfert.

D'où la fonction de transfert est donnée comme suit :

$$G(s) = \frac{27000}{s^2 + 221.1s + 13130} \quad (1.6)$$

$$G(s) = \frac{2.0564}{0.00007616s^2 + 0.01684s + 1} \quad (1.7)$$

La fonction de transfert est sous la forme :

$$G(s) = \frac{k}{\frac{1}{w_n^2}s^2 + \frac{2\xi}{w_n}s + 1} \quad (1.8)$$

Avec :

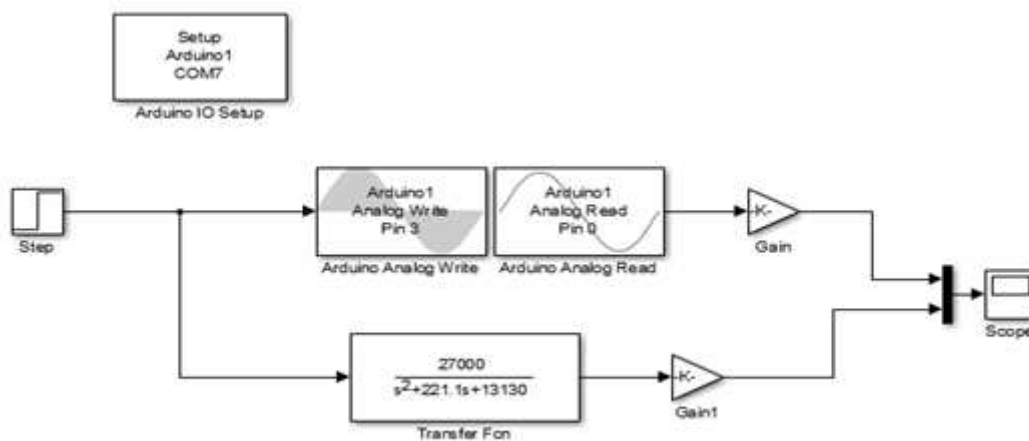
$K = 2.0564$  est le gain statique

$w_n = 114.5862$  est la pulsation naturelle

$\xi = 0.9648$  est le coefficient d'amortissement

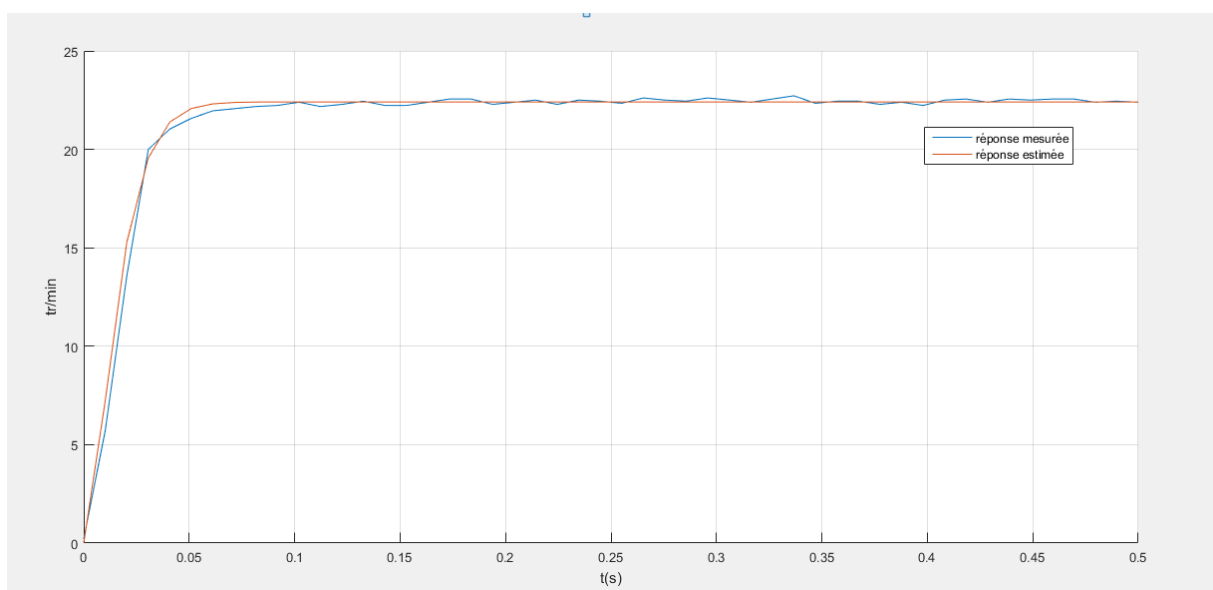
La fonction de transfert est calculée avec les données brutes sans conversion des valeurs de l'entrée et de la sortie (l'entrée (0v - 12 v) est codée de (0 - 255) et la sortie (0v - 5v) est codée de (0 - 1023)) pour faciliter l'implémentation du régulateur.

Pour vérifier la fiabilité du modèle identifié, nous avons réalisé le schéma de simulation représenté dans la figure suivante, qui va permettre de visualiser la réponse du système réel et la réponse estimée en temps réel.



**Fig.1.12** : Schéma de simulation pour visualiser les deux réponses (réel et estimée).

La figure 1.13 suivante représente la réponse du modèle identifié et la réponse mesurée en tr/min



**Fig.1.13** Réponse réel et estimée.

## 1.4. Commande du plateau tournant

Dans cette partie nous nous intéressons à la réalisation et l'implémentation de la loi de commande du moteur en boucle fermée pour réaliser un asservissement en vitesse.

L'objectif de cet asservissement est d'assurer une poursuite parfaite de la consigne en gardant toujours les caractéristiques de notre système (temps de réponse, allure de la réponse ...) et en assurant aussi une erreur statique nulle.

### 1.4.1. Passage au système discret

L'implémentation de l'algorithme de commande nécessite le passage vers le système discret (échantillonné) et pour cela nous devons déterminer la période d'échantillonnage adéquate pour ce transfert.

La condition que doit satisfaire le pas d'échantillonnage  $\Delta$  pour un système du second ordre est :

$$\frac{0.25}{w_n} < \Delta < \frac{1.5}{w_n} \quad (1.9)$$

où  $w_n$  est la pulsation naturelle (propre) du système

D'où :

$$\frac{0.25}{114.5862} < \Delta < \frac{1.5}{114.5862}$$

$$0.002 < \Delta < 0.013 \quad (1.10)$$

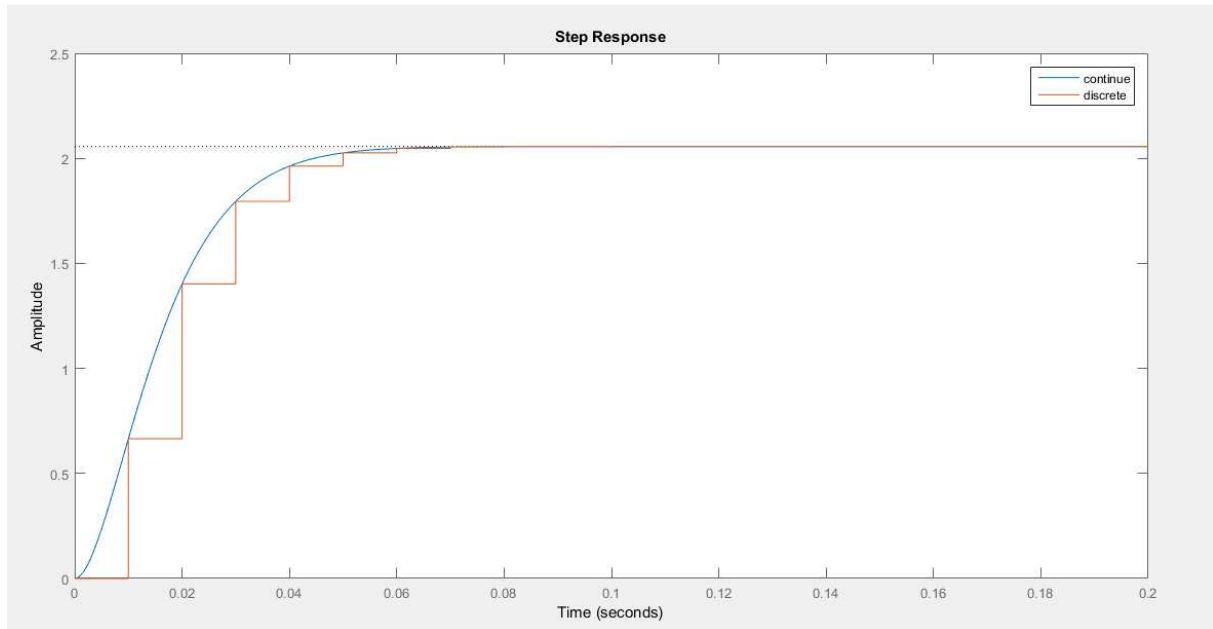
Dans notre cas nous avons fixé la période d' d'échantillonnage à

$$\Delta = 0.01 \text{ s} = 10 \text{ ms}$$

Après une discrétisation de la fonction de transfert continue, nous obtenons la fonction de transfert discrète suivante :

$$G(z) = \frac{0.6651 z + 0.3165}{z^2 - 0.6322 z + 0.1096} \quad (1.11)$$

La réponse indicielle des deux fonctions (continue et discrète) sont données sur la figure 1.14 suivante :



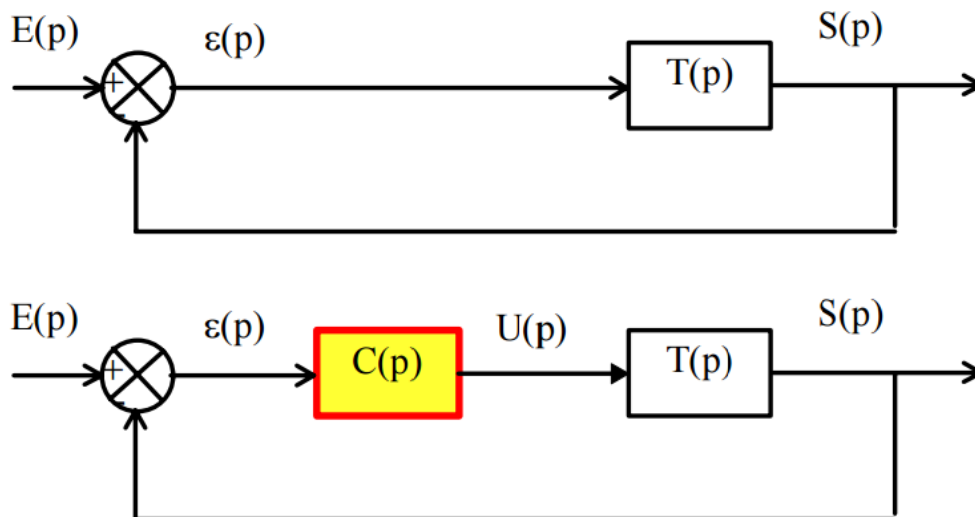
**Fig.1.14** Réponses indicielles de  $g(s)$  et  $g(z)$ .

### 1.4.2. Synthèse du correcteur

Les systèmes asservis peuvent présenter des défauts, une précision insuffisante, une stabilité trop relative (voire une instabilité), un temps de réaction trop lent, un dépassement trop important, au regard d'un cahier des charges. Il est donc souvent nécessaire d'intégrer dans le système asservis un réseau correcteur dont l'objectif est d'améliorer un ou plusieurs de ces différents paramètres sans le faire au détriment des autres.

Les correcteurs doivent permettre de réaliser le meilleur compromis entre précision, stabilité et rapidité du système étudié [4].

La figure suivante montre l'implémentation d'un correcteur dans une boucle d'asservissement :



**Fig.1.15** Implémentation du correcteur dans la boucle de l'asservissement.

#### 1.4.2.1. Choix du correcteur

Dans cette partie nous s'intéressons au choix du correcteur à introduire dans la boucle de l'asservissement et, pour cela, nous allons effectuer une série de tests afin de déterminer le correcteur adéquat pour satisfaire le cahier de charges.

##### Le correcteur proportionnel

Le correcteur proportionnel est le plus simple des correcteurs. Il ne permet pas toujours d'obtenir des performances très élevées mais il peut suffire dans certains cas si le cahier des charges n'est pas trop contraignant ou si le système a un comportement très simple. Expérimentalement, son réglage peut se faire directement en partant d'un gain faible et en augmentant petit à petit jusqu'à atteindre un comportement satisfaisant. Pour un gain trop élevé, le système deviendra instable ce qui se manifestera d'abord par des oscillations de plus en plus importantes.

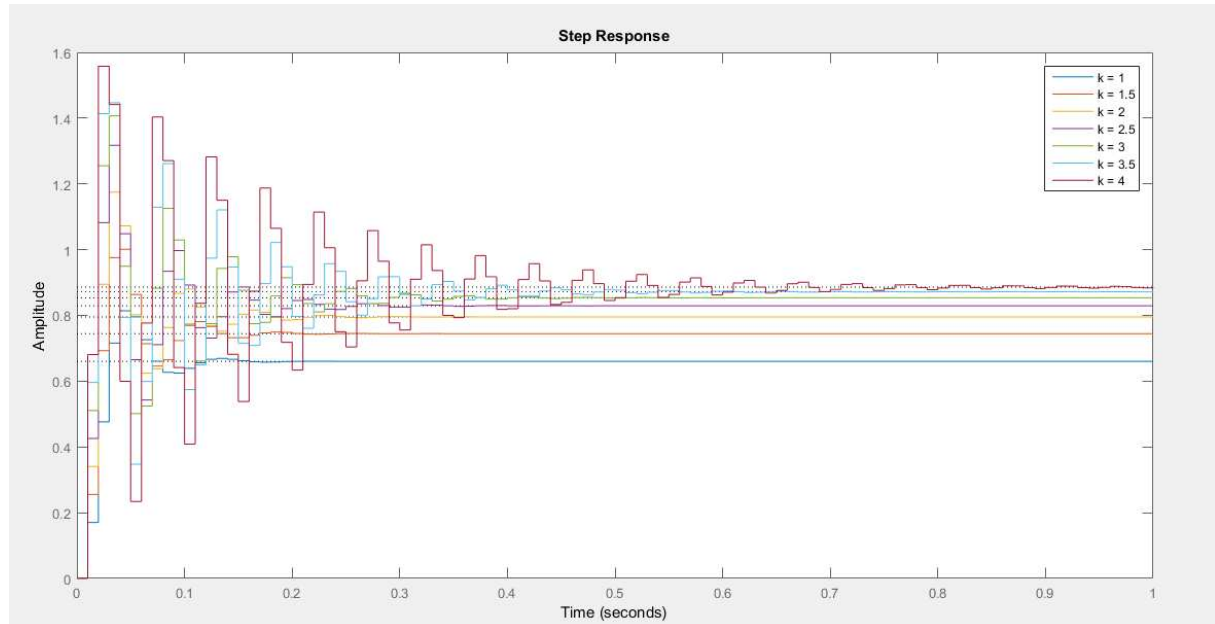
Dans le cas d'un correcteur proportionnel, la loi de commande corrigée  $u(t)$  est proportionnelle à l'écart  $\varepsilon(t)$  :

$$U(t) = K_p * \varepsilon(t) \quad (1.12)$$

La fonction de transfert du correcteur est donc :

$$C(s) = \frac{U(s)}{\varepsilon(s)} = K_p \quad (1.13)$$

La figure 1.16 suivante représente la réponse indicielle du système corrigé avec un correcteur proportionnel dont le gain vari de 1 à 4 avec un pas de 0.5 :



**Fig.1.16** Réponses des correcteurs proportionnels.

Nous remarquons que l'effet d'une augmentation du gain entraîne une diminution de l'erreur statique, rend le système plus rapide mais augmente l'instabilité du système.

### Le correcteur Proportionnel Intégral PI

L'intérêt de la correction de type intégrale est de permettre une erreur statique nulle. En effet, si le système comporte un intégrateur et se stabilise à un point d'équilibre tous les signaux sont constants. Or, pour que la sortie d'un intégrateur soit constante, il est nécessaire que son entrée soit nulle. Si l'entrée de l'intégrateur est reliée à l'erreur de régulation, l'effet du terme intégral sera bien d'annuler cette erreur. Notons que l'erreur est annulée même en présence d'une perturbation.

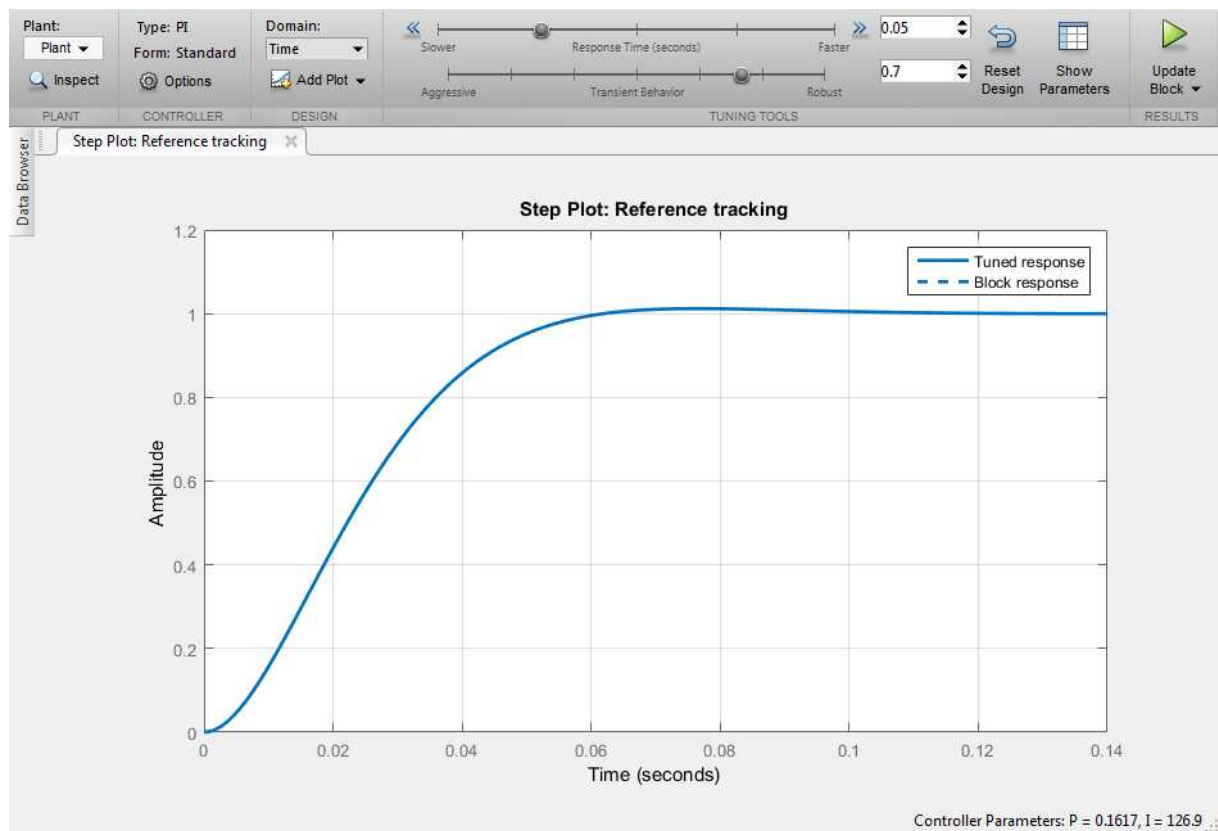
Le correcteur Intégrateur est en général associé au correcteur proportionnel et la loi de commande corrigée est de la forme :

$$U(t) = K_p \left( \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(u) du \right) \quad (1.14)$$

La fonction de transfert du correcteur est donc :

$$C(s) = K_p \frac{1+T_i s}{T_i s} \quad (1.15)$$

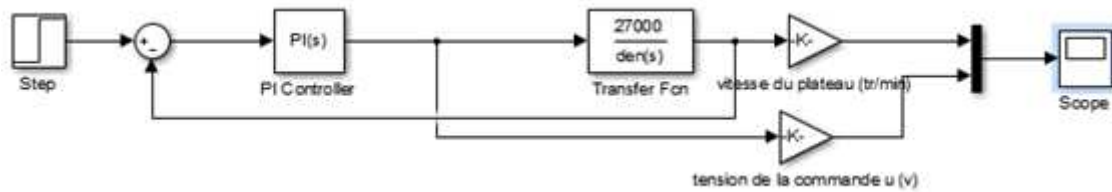
L'objectif de cet asservissement est d'assurer au system en boucle fermée un une erreur statique nulle, un temps de réponse  $t_r = 0.05$  s et un coefficient d'amortissement  $h = 0.7$  comme le montre la figure suivante :



**Fig.1.17** Choix des performances désirées.

Et pour cela les paramètres du régulateur seront calculés à l'aide de l'outil MATLAB.

La figure suivante montre le schéma de simulation pour l'auto-régulation des paramètres du correcteur PI



**Fig.1.18** Schéma de simulation pour l'autoréglage du correcteur.

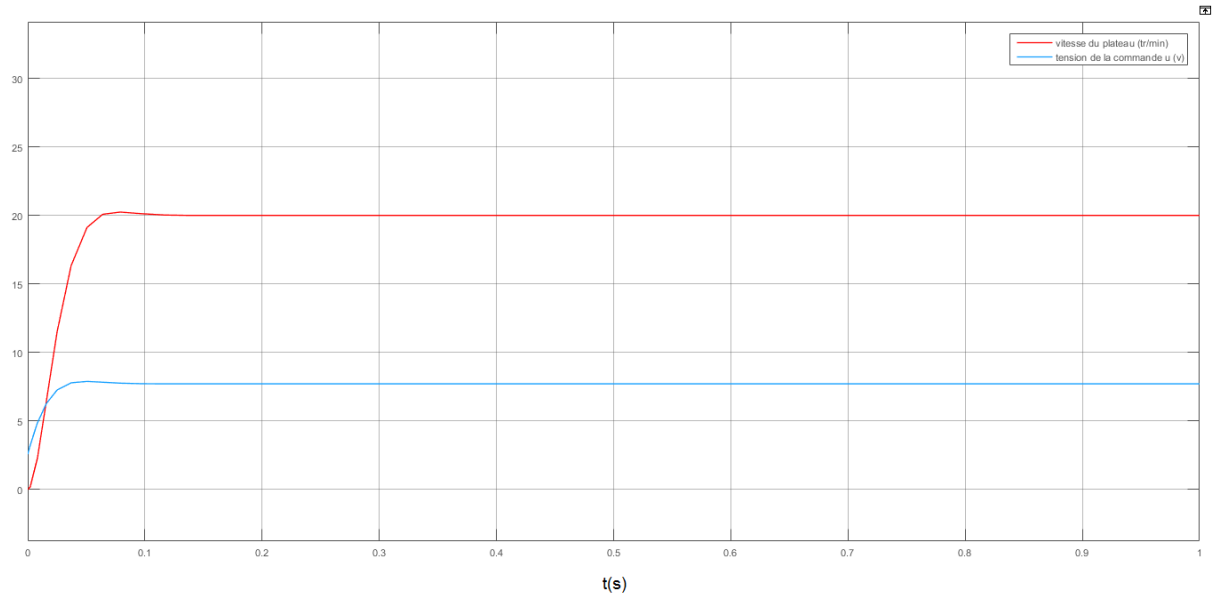
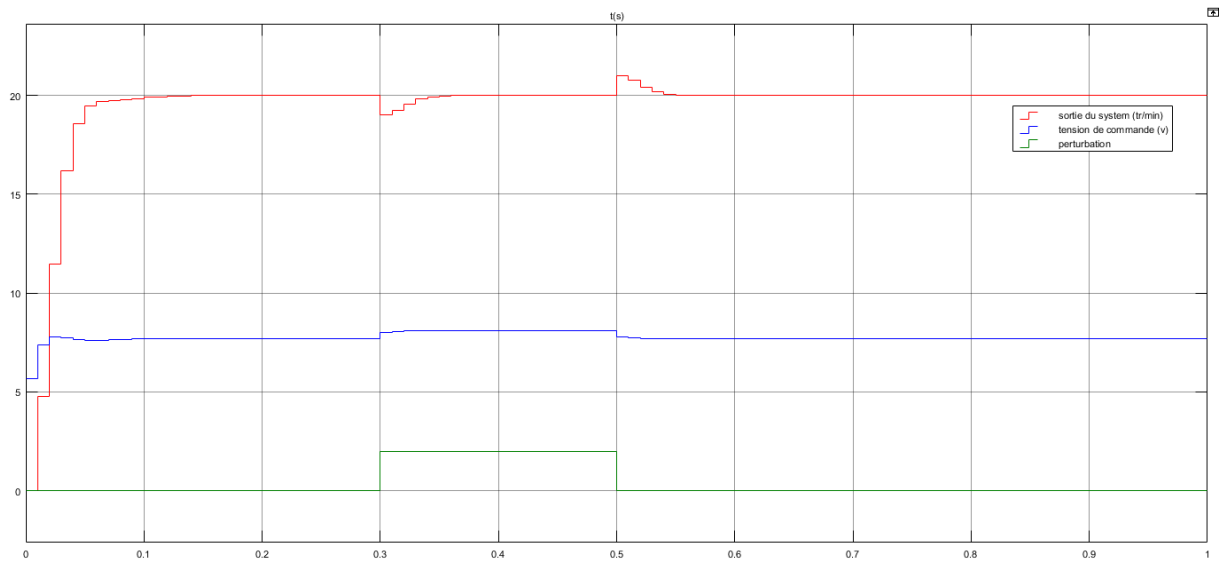
Les paramètres obtenus lors du réglage sont donnés dans la figure suivante :

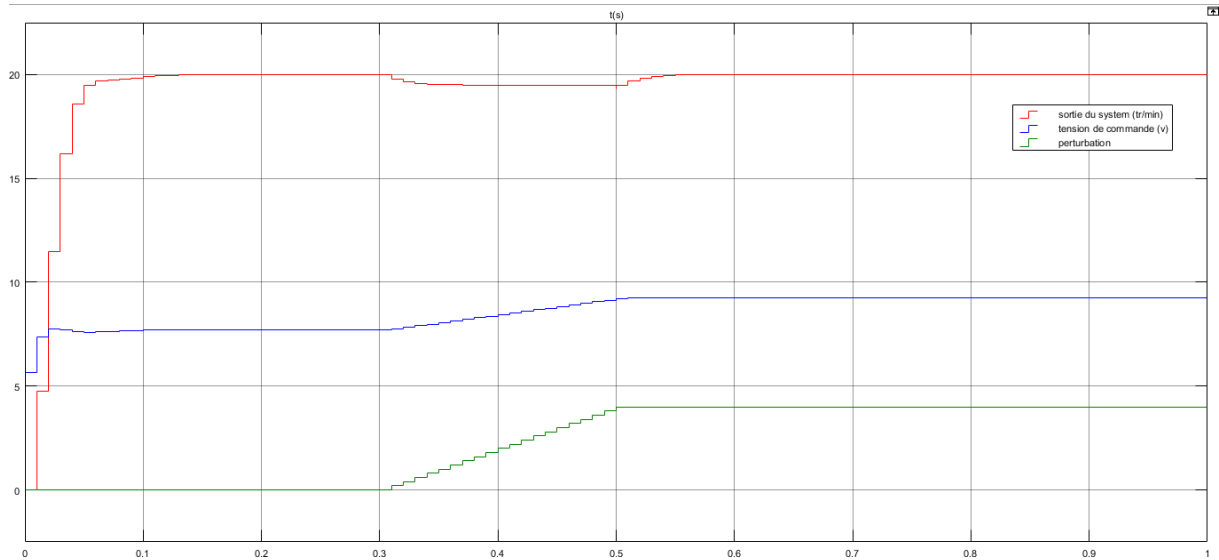
**Fig.1.19** Calcul des paramètres du correcteur.

Donc la fonction de transfert du correcteur PI est donnée comme suit :

$$C(s) = \frac{0.1617s + 20.53}{s} \quad (1.16)$$

La figure suivante représente la réponse du système à une consigne de 20 tr/min ainsi que l'allure de la commande :

**Fig.1.20** Réponse du système corrigé.**Fig.1.21** : réponse du système corrigé avec une perturbation de type échelon.



**Fig.1.22** : réponse du système corrigé avec une perturbation de type rampe.

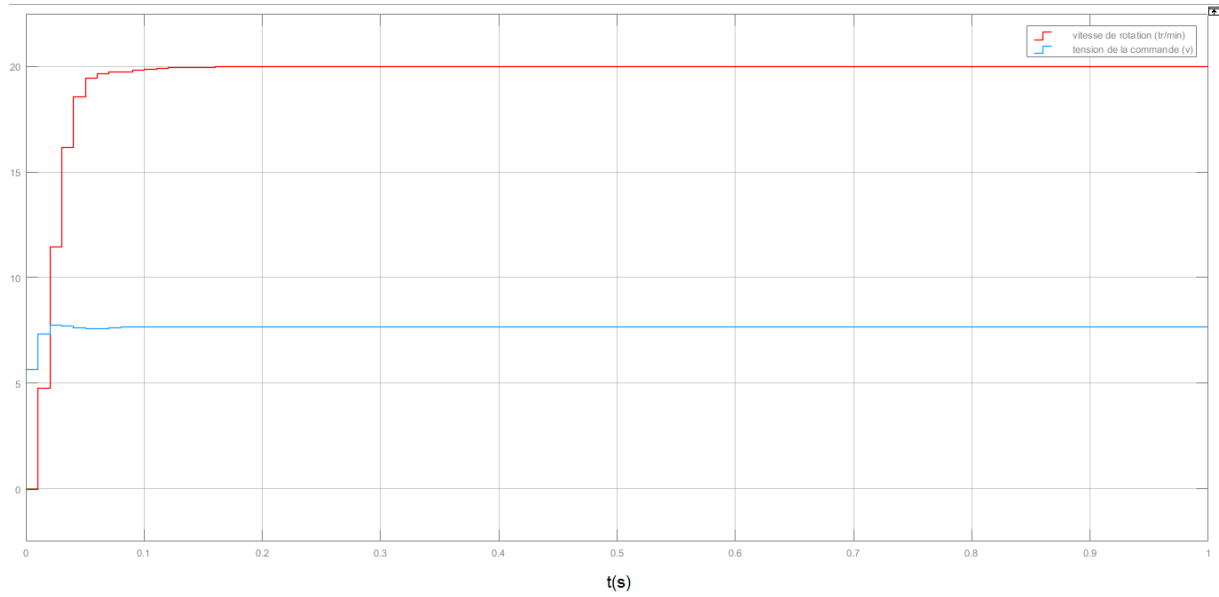
On remarque que le correcteur proportionnel intégral (PI) satisfait largement le cahier des charges, même en présence de perturbations le système asservi arrive à compenser l'erreur et garder la sortie égale à la consigne.

#### 1.4.2.2. Implémentation du correcteur

En utilisant l'outil MATLAB pour l'auto-réglage du correcteur nous obtenons la fonction de transfert du correcteur discret suivante :

$$C(z) = \frac{0.3581z - 0.1663}{z - 1} \quad (1.17)$$

La figure suivante représente la réponse du système corrigé à une consigne de 20 t r/min :



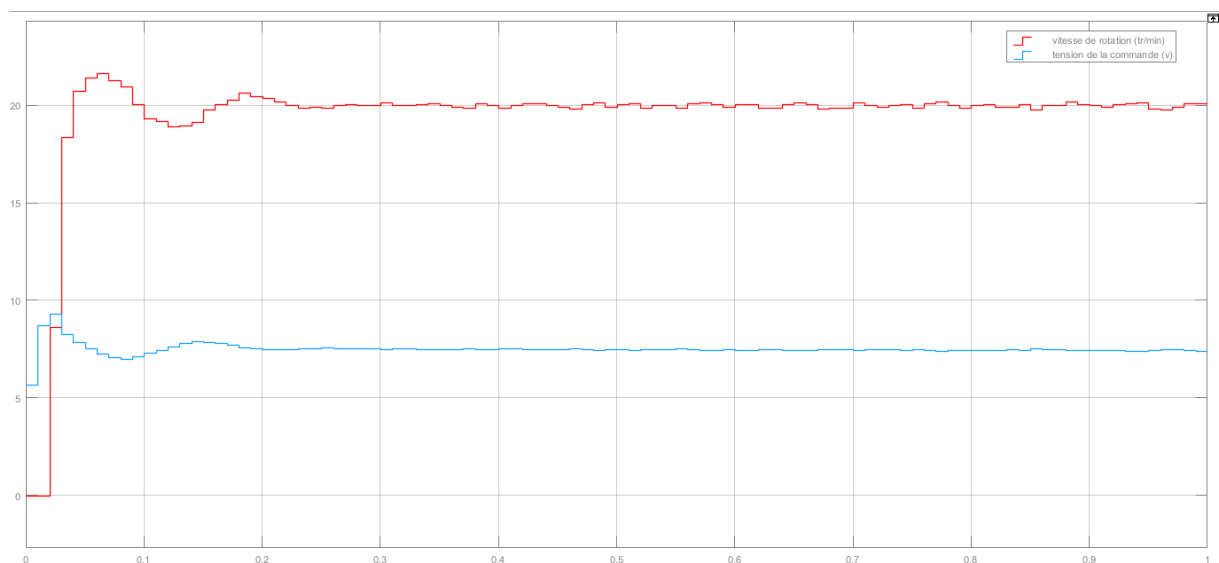
**Fig.1.23** Réponse du système discret corrigé

L'équation de récurrence liant la commande à l'erreur est donnée par :

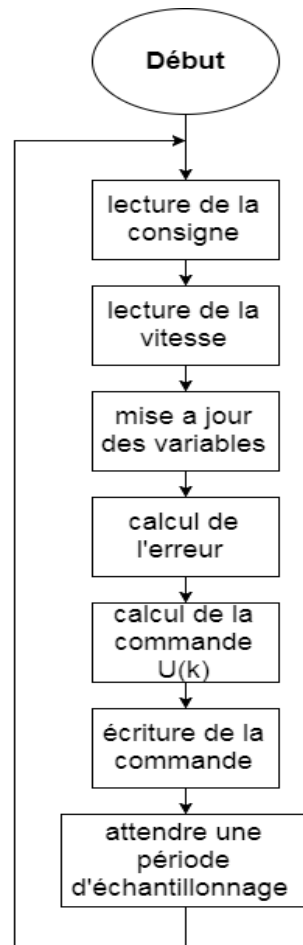
$$U(k) = U(k - 1) + 0.3581 \varepsilon(k) - 0.1663\varepsilon(k - 1) \quad (1.18)$$

L'organigramme qui permet de programmer cette loi de commande pour obtenir la tension  $U(k)$ , est donné à la figure (1.25).

Après l'implémentation pratique de la commande du moteur, la figure suivante représente la réponse du système réel à une consigne de 20 tr/min



**Fig.1.24** Réponse du système réel corrigé



**Fig.1.25** Organigramme de l'implémentation du correcteur

A partir de cette figure (1.24) nous remarquons que le système présente quelques oscillations en régime transitoire provoquées par les imperfections dans le système d'acquisition et probablement à des bruits superposés aux signaux de la commande et du capteur.

## 1.5. Conclusion

Dans ce chapitre dédié à la commande numérique du plateau tournant, nous avons effectué une étude et une réalisation de la commande du moteur à courant continu en passant par plusieurs étapes.

Après avoir décrit les composants et la configuration matériel du plateau tournant, l'étape de l'identification est ensuite réalisée en utilisant la méthode des moindres carrés, qui nous a permis d'avoir un modèle mathématique (sous forme de fonction de transfert) qui reproduit, de

manière fidèle, le comportement de notre système. Ensuite vient l'étape de la commande dans laquelle nous avons effectué plusieurs tests afin de déterminer le correcteur adéquat pour satisfaire le cahier des charges imposé auparavant. En dernière étape, vient l'implémentation de la loi de commande et l'interprétation des résultats pratiques.

## **Chapitre 2**

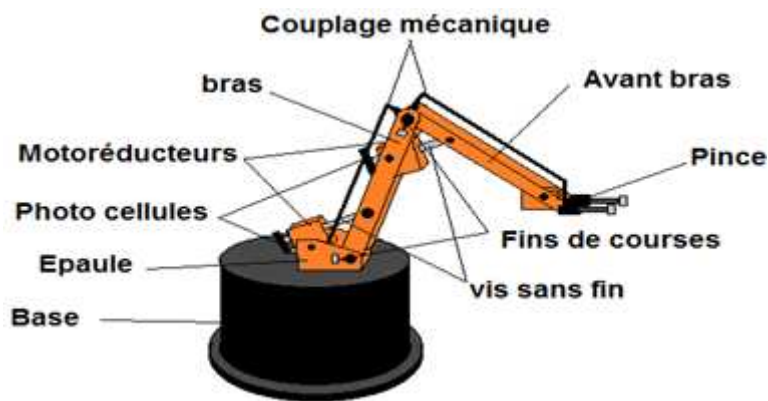
### ***Commande du bras manipulateur***

#### **2.1 Introduction**

Les robots sont actuellement présents dans pratiquement toutes les entreprises industrielles. Ils sont utilisés pour effectuer rapidement des tâches répétitives comme dans les chaînes de fabrication et de montage. Ils sont aussi présents dans des environnements difficilement supportables par l'homme là où les conditions de température ou de pression sont trop élevées, en milieu radioactif, etc. Initialement conçus aveugles, les mouvements des robots étaient calculés et parfois avec beaucoup de précision selon la tâche à exécuter. Ensuite, pour approcher l'homme dans sa perception, d'autres capteurs sont venus compléter l'aspect sensoriel des robots afin de les rendre plus souples dans leur fonctionnement et leur déplacement. Dans ce chapitre, notre objectif est de mettre en œuvre la commande d'un bras manipulateur lui permettant de réaliser sa tâche de saisie, de déplacement de tout objet détecté, localisé et reconnu sur le plateau tournant par un système de vision artificielle. Pour ce faire, une carte de commande est conçue à cet effet et un programme gérant l'ensemble des actions a été mis en œuvre.

## 2.2 Description du bras manipulateur

Le robot utilisé dans notre application est un bras manipulateur à trois degrés de liberté, il est composé d'une base, d'une épaule, d'un bras et d'un avant bras montés de bout en bout. L'organe de préhension est une pince à serrage parallèle. Chaque partie du bras manipulateurs est connectée aux autres par une articulation. La rotation de chaque articulation du bras est assurée par un motoréducteur et une vis sans fin. Un encodeur incrémental pour la mesure de l'angle de rotation est placé sur l'arbre du motoréducteur de chacune des articulations. Ce dispositif correspond à un générateur d'impulsions réalisé par une photocellule traversée par un disque perforé porté par la vis sans fin. Ces dispositifs génèrent une impulsion pour un angle de valeur  $a_1$  dans le cas de l'articulation de la base, de  $a_2$  dans le cas de l'articulation de l'épaule et de  $a_3$  dans le cas de l'articulation du coude ( $a_1$ ,  $a_2$  et  $a_3$  sont des paramètres qui seront déterminés ultérieurement). Deux butées de contact et deux fins de courses sont disponibles au niveau de la pince et de chaque articulation pour définir les positions limites du mouvement dans les deux sens par la mise à zéro des fins de courses (fig.2.1).



**Fig.2.1** : Composants du bras manipulateur.

Le rôle du couplage mécanique reliant la troisième articulation à l'épaule d'un côté et à la base de la pince de l'autre côté est de maintenir la pince à la position horizontale par rapport au plan de la base, quel que soit le

mouvement effectué. Par conséquent l'articulation de la pince n'est pas contrôlée. Elle est automatiquement commandée par ce couplage.

La figure 2.2 suivante montre les paramètres structuraux du bras manipulateur.

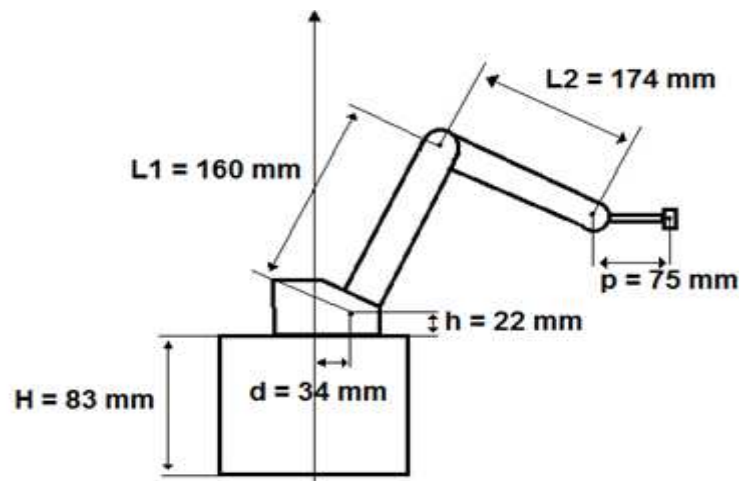


Fig.2.2 : Paramètres structuraux du bras manipulateur.

## 2.3 Modélisation

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). On distingue :

- les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
- les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement.
- les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou

forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations. [7]

Dans les commandes cinématiques et dynamiques où l'on veut contrôler la vitesse lors des déplacements, et même avoir des accélérations importantes, on est obligé d'avoir recours aux modèles cinématique et dynamique qui font intervenir les masses, les forces d'inertie etc.

Dans notre cas, nous avons opté pour une modélisation géométrique où nous avons fait intervenir que les longueurs des segments et les variables articulaires. Ce type de modélisation convient à la commande en position ou point à point où la vitesse de déplacement est fixée uniquement par le temps de réponse des dispositifs intervenant dans la structure et la commande du système mécanique articulé.

### 2.3.2 Modèle géométrique direct

Le bras manipulateur est représenté par des segments rigides successifs, connectés les uns aux autres par des articulations. En partant de la base pour arriver à l'organe terminal, on affecte à chaque segment rigide un repère. En utilisant les transformations homogènes, on calcule pour chaque repère  $R_i$  la matrice de transformation  $A_{i-1 i}$ , permettant le passage du repère  $R_{i-1}$  au repère  $R_i$ .

En ce qui nous concerne, nous utiliserons la méthode de Denavit-Hartenberg. Dans la modélisation géométrique utilisant les transformations homogènes, la structure mécanique articulée sera modélisée par une matrice  $M_{0n}$ .

Soient  $S_0$  la base du système et  $S_n$  le dernier segment.  $M_{0n}$  est le produit des matrices successives  $A_{i-1 i}$ , donnant chacune la position et l'orientation du repère  $R_i$  attaché à un segment  $S_i$  par rapport au précédent.

$$M_{0n} = \prod_{i=1}^{i=n} A_{i-1 i} \quad (2.1)$$

La matrice  $M$  donne la situation du dernier segment (support de l'effecteur) par rapport à  $R_0$ . [8]



Le tableau suivant représente les paramètres de Denavit-Hartenberg pour chaque repère :

**Tab.2.1** : tableau des paramètres de Denavit-Hartenberg.

$i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	H	0	0
2	0	H	d	$\pi/2$
3	$\theta_3$	0	L1	0
4	$\theta_4$	0	L2	0

La matrice homogène de Denavit-Hartenberg est donnée par :

$$A_{i-1 i} = R_{(z_{i-1}, \theta_i)} T_{(z_{i-1}, d_i)} T_{(x_i, a_i)} R_{(x_i, \alpha_i)} \quad (2.2)$$

Où :

- $R_{(z_{i-1}, \theta_i)}$  est la matrice de rotation autour de Z d'un angle  $\theta$
- $T_{(z_{i-1}, d_i)}$  est la matrice de translation le long de Z d'une longueur d
- $R_{(x_i, \alpha_i)}$  est la matrice de rotation autour de X d'un angle  $\alpha$
- $T_{(x_i, a_i)}$  est la matrice de translation le long de X d'une longueur a

$$A_{i-1 i} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{i-1 i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Où  $c\theta_i$  et  $c\alpha_i$  sont respectivement  $\cos(\theta_i)$  et  $\cos(\alpha_i)$  et  $s\theta_i$  et  $s\alpha_i$  sont respectivement  $\sin(\theta_i)$  et  $\sin(\alpha_i)$

Dans notre cas, les matrices de transformation homogène de chaque repère sont :

$$A_{01} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_{23} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L1c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & L1s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_{34} = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & L2c\theta_4 \\ s\theta_4 & c\theta_4 & 0 & L2s\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

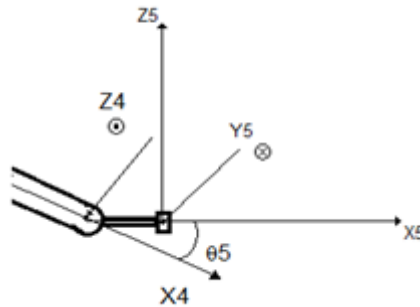
La matrice de transformation entre la base et la dernière articulation est donnée comme suit :

$$M_{04} = \begin{bmatrix} c(\theta_3 + \theta_4)c\theta_1 & -s(\theta_3 + \theta_4)c\theta_1 & s\theta_1 & c\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ c(\theta_3 + \theta_4)s\theta_1 & -s(\theta_3 + \theta_4)s\theta_1 & -c\theta_1 & s\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ s(\theta_3 + \theta_4) & c(\theta_3 + \theta_4) & 0 & H + h + L1s\theta_3 + L2s(\theta_3 + \theta_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

La dernière colonne de cette matrice donne les coordonnées de l'origine du dernier repère (articulation de la pince)

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} c\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ s\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ H + h + L1s\theta_3 + L2s(\theta_3 + \theta_4) \end{pmatrix} \quad (2.5)$$

Pour obtenir les coordonnées du centre de l'organe terminal, nous associons un repère à ce dernier où l'axe  $y$  sera orienté dans le sens de la fermeture de la pince comme le montre la figure suivante :



**Fig.2.4** : repère associé à l'articulation des la pince.

Les paramètres de Denavit-Hartenberg de ce repère sont les suivants :

**Tab.2.2** : paramètres de Denavit-Hartenberg du dernier repère.

$i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
5	$\theta_5$	P	0	$-\Pi/2$

Où :

$$\theta_5 = -(\theta_3 + \theta_4) \quad (2.6)$$

D'où la matrice de transformation du dernier repère vers l'organe terminale est la suivante :

$$E = \begin{bmatrix} c(\theta_3 + \theta_4) & 0 & s(\theta_3 + \theta_4) & pc(\theta_3 + \theta_4) \\ -s(\theta_3 + \theta_4) & 0 & c(\theta_3 + \theta_4) & -ps(\theta_3 + \theta_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

La matrice de transformation homogène entre la base et l'organe terminal est donnée sous la forme suivante :

$$T = M_{04} * E \quad (2.8)$$

$$T = \begin{bmatrix} c(\theta_3 + \theta_4)c\theta_1 & -s(\theta_3 + \theta_4)c\theta_1 & s\theta_1 & c\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ c(\theta_3 + \theta_4)s\theta_1 & -s(\theta_3 + \theta_4)s\theta_1 & -c\theta_1 & s\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4)) \\ s(\theta_3 + \theta_4) & c(\theta_3 + \theta_4) & 0 & H + h + L1s\theta_3 + L2s(\theta_3 + \theta_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c(\theta_3 + \theta_4) & 0 & s(\theta_3 + \theta_4) & pc(\theta_3 + \theta_4) \\ -s(\theta_3 + \theta_4) & 0 & c(\theta_3 + \theta_4) & -ps(\theta_3 + \theta_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & c\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4) + p) \\ s\theta_1 & c\theta_1 & 0 & s\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4) + p) \\ 0 & 0 & 1 & H + h + L1s\theta_3 + L2s(\theta_3 + \theta_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Ainsi, les coordonnées de l'organe terminal par rapport à la base sont :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4) + p) \\ s\theta_1(d + L1c\theta_3 + L2c(\theta_3 + \theta_4) + p) \\ H + h + L1s\theta_3 + L2s(\theta_3 + \theta_4) \end{pmatrix} \quad (2.10)$$

### 2.3.3. Modèle géométrique inverse

Nous avons vu que le modèle direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des dos coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). On peut distinguer trois méthodes de calcul du MGI :

- la méthode de Paul,
- la méthode de Pieper
- la méthode générale de Raghavan et Roth.

Lorsqu'il n'est pas possible de trouver une forme explicite du modèle géométrique inverse, on peut calculer une solution particulière par des procédures numériques. [7]

Dans notre cas la méthode adoptée est celle de Paul, dont le principe est le suivant :

Considérons un robot manipulateur dont la matrice de transformation homogène a pour expression :

$$M_{0n} = A_{01} \cdot A_{12} \dots \cdot A_{n-1 n} \quad (2.11)$$

Soit  $U_0$  la situation désirée telle que :

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & x \\ s_y & n_y & a_y & y \\ s_z & n_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On cherche à résoudre le système d'équations suivant :

$$U_0 = A_{01} \cdot A_{12} \dots \cdot A_{n-1 n} \quad (2.12)$$

Pour trouver les solutions de l'équation, Paul a proposé une méthode qui consiste à multiplier à gauche successivement les deux membres de l'équation par les matrices  $A_{n n-1}$  (où  $A_{n n-1}$  est l'inverse de la matrice  $A_{n-1 n}$ ) pour  $n$  variant de 1 à  $n-1$  (ou bien jusqu'à avoir la solution). Ces opérations

permettent d'isoler et d'identifier l'une après l'autre les variables articulaires que l'on cherche [7]

Dans notre cas, nous avons:

$$A_{10} \cdot U_0 = A_{10} \cdot A_{01} \cdot A_{12} \cdot A_{23} \cdot A_{34} \cdot E$$

$$A_{10} \cdot U_0 = A_{12} \cdot A_{23} \cdot A_{34} \cdot E$$

$$A_{10} \cdot U_0 = \begin{bmatrix} s_x c \theta_1 + s_y s \theta_1 & n_x c \theta_1 + n_y s \theta_1 & a_x c \theta_1 + a_y s \theta_1 & x c \theta_1 + y s \theta_1 \\ s_y c \theta_1 - s_x s \theta_1 & n_y c \theta_1 - n_x s \theta_1 & a_y c \theta_1 - a_x s \theta_1 & y c \theta_1 - x s \theta_1 \\ s_z & n_z & a_z & z - H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

$$A_{12} \cdot A_{23} \cdot A_{34} \cdot E = \begin{bmatrix} 1 & 0 & 0 & d + p + L2c(\theta_3 + \theta_4) + L1c\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h + L2s(\theta_3 + \theta_4) + L1s\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Par identification des deux dernières colonnes, nous obtenons les équations suivantes :

$$x c \theta_1 + y s \theta_1 = d + p + L2c(\theta_3 + \theta_4) + L1c\theta_3 \quad (1)$$

$$y c \theta_1 - x s \theta_1 = 0 \quad (2)$$

$$z - H = h + L2s(\theta_3 + \theta_4) + L1s\theta_3 \quad (3)$$

De l'équation (2), nous avons:

$$y c \theta_1 = x s \theta_1$$

$$\frac{s \theta_1}{c \theta_1} = \frac{y}{x}$$

d'où :

$$\theta_1 = \arctan\left(\frac{y}{x}\right) \quad (2.15)$$

Et de (1) et (3), nous avons:

$$x c \theta_1 + y s \theta_1 - d - p = L2c(\theta_3 + \theta_4) + L1c\theta_3$$

$$z - H - h = L2s(\theta_3 + \theta_4) + L1s\theta_3$$

Si l'on pose :

$$A = x c \theta_1 + y s \theta_1 - d - p$$

$$B = z - H - h$$

Nous obtenons:

$$A = L2c(\theta_3 + \theta_4) + L1c\theta_3 \quad (4)$$

$$B = L2s(\theta_3 + \theta_4) + L1s\theta_3 \quad (5)$$

### Calcul de $\theta_4$

Nous avons:

$$A^2 + B^2 = (L2c(\theta_3 + \theta_4) + L1c\theta_3)^2 + (L2s(\theta_3 + \theta_4) + L1s\theta_3)^2$$

$$A^2 + B^2 = L2^2(c^2(\theta_3 + \theta_4) + s^2(\theta_3 + \theta_4)) + L1^2(c^2\theta_3 + s^2\theta_3) + 2L1L2(c(\theta_3 + \theta_4)c\theta_3 + s(\theta_3 + \theta_4)s\theta_3)$$

$$c^2\alpha + s^2\alpha = 1$$

$$c(\alpha + \beta)c\beta + s(\alpha + \beta)s\beta = c\alpha$$

D'où :

$$A^2 + B^2 = L1^2 + L2^2 + 2L1L2c\theta_4$$

Donc

$$c\theta_4 = \frac{A^2 + B^2 - L1^2 - L2^2}{2L1L2}$$

$$\theta_4 = \arccos\left(\frac{A^2 + B^2 - L1^2 - L2^2}{2L1L2}\right)$$

Nous remarquons que nous avons deux valeurs possibles pour  $\theta_4$  car :

$$c(\theta_4) = c(-\theta_4) = a$$

D'où  $\arccos(a) = \theta_4$  ou  $-\theta_4$

Et nous avons  $\theta_4 < 0$  quelle que soit la configuration du bras. Ainsi,

$$\theta_4 = -\arccos\left(\frac{A^2 + B^2 - L1^2 - L2^2}{2L1L2}\right) \quad (2.16)$$

Calcul de  $\theta_3$  :

Reprenons les équations (4) et (5)

$$A = L2c(\theta_3 + \theta_4) + L1c\theta_3$$

$$B = L2s(\theta_3 + \theta_4) + L1s\theta_3$$

Nous avons :

$$c(\alpha + \beta) = c(\alpha)c(\beta) - s(\alpha)s(\beta)$$

$$s(\alpha + \beta) = c(\alpha)c(\beta) + s(\alpha)s(\beta)$$

D'où :

$$A = (L1 + L2c\theta_4)c\theta_3 - (L2s\theta_4)s\theta_3$$

$$B = (L2c\theta_4)c\theta_3 + (L1 + L2s\theta_4)s\theta_3$$

Nous avons un système de deux équations à deux inconnus ( $c\theta_3, s\theta_3$ ). La solution est donnée comme suit :

$$s\theta_3 = \frac{B(L1 + L2c\theta_4) - AL2s\theta_4}{(L1 + L2c\theta_4)^2 + (L2s\theta_4)^2}$$

$$c\theta_3 = \frac{A(L1 + L2c\theta_4) + BL2s\theta_4}{(L1 + L2c\theta_4)^2 + (L2s\theta_4)^2}$$

Et nous avons:

$$\tan(\theta_3) = \frac{s\theta_3}{c\theta_3} = \frac{B(L1 + L2c\theta_4) - AL2s\theta_4}{A(L1 + L2c\theta_4) + BL2s\theta_4}$$

D'où :

$$\theta_3 = \arctan\left(\frac{B(L1+L2c\theta_4)-AL2s\theta_4}{A(L1+L2c\theta_4)+BL2s\theta_4}\right) \quad (2.17)$$

Le modèle géométrique inverse est donc donné comme suit :

$$\begin{pmatrix} \theta_1 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{y}{x}\right) \\ \arctan\left(\frac{B(L1+L2c\theta_4)-AL2s\theta_4}{A(L1+L2c\theta_4)+BL2s\theta_4}\right) \\ -\arccos\left(\frac{A^2+B^2-L1^2-L2^2}{2L1L2}\right) \end{pmatrix} \quad (2.18)$$

Avec :

$$A = xc\theta_1 + ys\theta_1 - d - p$$

$$B = z - H - h$$

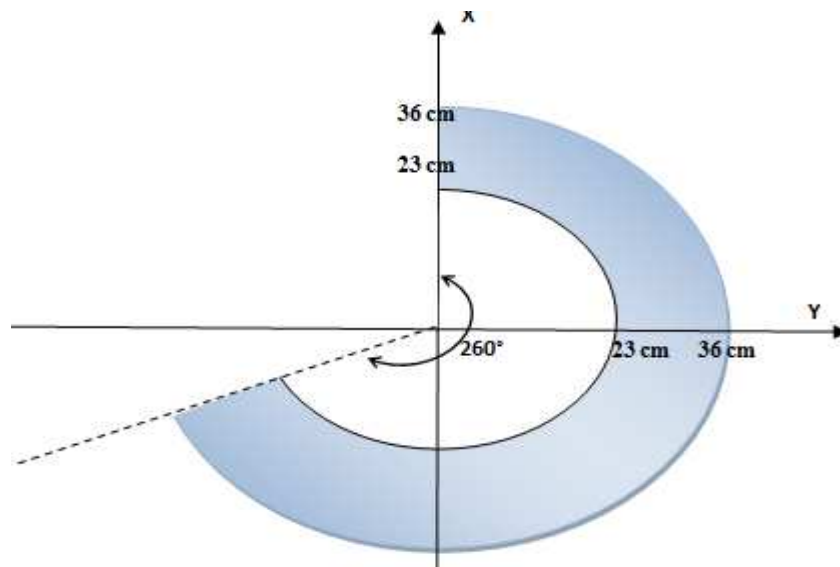
## 2.4. Commande du bras manipulateur

Avant d'entamer l'étape de la commande du bras manipulateur qui est composée de deux parties (l'élaboration de l'algorithme de commande et l'implémentation du programme), nous devons d'abord définir la configuration initiale, les limites des articulations et l'espace de travail du robot. Le tableau suivant représente les positions limites associés aux fins de courses :

**Tab.2. 3** : tableau des positions limites des articulations.

Articulation	Fins de courses	Positions limites
La base	Fcb1 (état initial)	Les deux butées sont placées de telle manière à permettre à la base une course de $\alpha_1 = 260^\circ$ .
	Fcb2	
L'épaule	Fce1 (état initial)	Le bras fait un angle de $\alpha_2 = 79.1^\circ$ avec l'horizontale
	Fce2	Le bras fait un angle de $\alpha_2 = 128^\circ$ avec l'horizontale
Le coude	Fcc1 (état initial)	L'avant bras fait un angle de $\alpha_3 = 50.5^\circ$ avec Le bras
	Fcc2	L'avant bras fait un angle de $\alpha_3 = 100^\circ$ avec Le bras
La pince	Fcp1 (état initial)	La pince complètement ouverte (4cm)
	Fcp2	Pince fermée

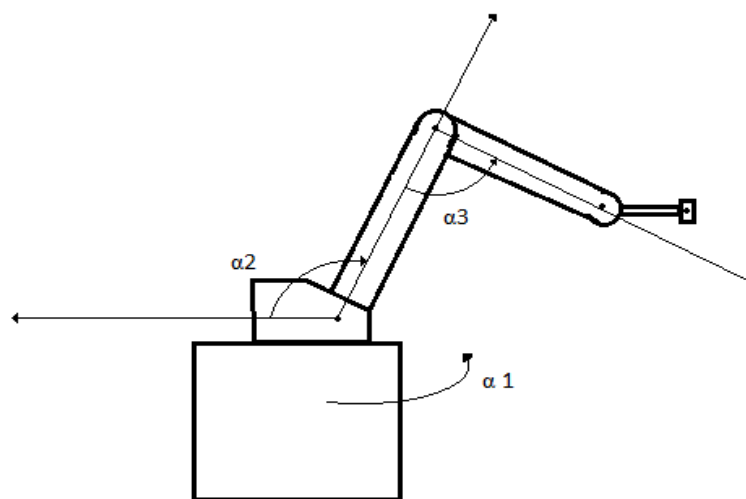
La figure suivante représente l'espace de travail du robot calculé à partir des positions limites sur une hauteur  $z = 12.5$  cm (la hauteur du plateau tournant) :



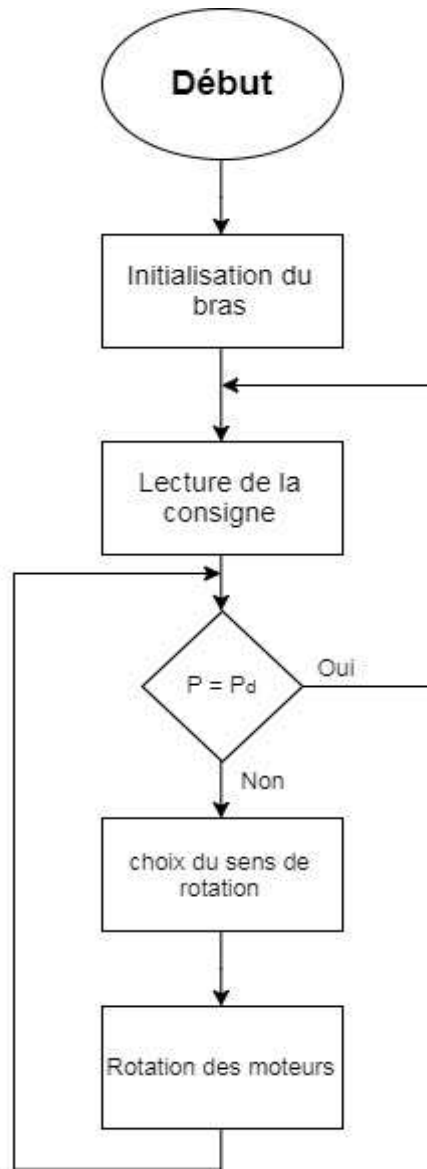
**Fig.2.5** : espace de travail du robot.

#### 2.4.1. Algorithme de commande

La commande du robot consiste à générer les signaux de commande des moteurs du robot pour permettre à ce dernier de réaliser sa tâche, le déplacement de chaque moteur est réalisé en suivant l'organigramme de la figure (2.7) ci-après.



**Fig.2.6** : représentation des angles commandés par les moteurs.



**Fig.2.7** : Organigramme de la commande du robot.

Ici  $P$  est la position du robot et  $P_d$  la position désirée.

### 2.4.2. Implémentation du programme et testes pratiques

Les motoréducteurs placés sur le robot nous permettent de contrôler 3 angles ( $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ ) comme le montre la figure 2.6 :

Les formules qui relient les angles  $\theta_i$  aux angles  $\alpha_i$  sont les suivantes :

$$\theta_1 = \alpha_1$$

$$\theta_3 = \Pi - \alpha_2$$

$$\theta_4 = -\Pi + \alpha_3$$

Les paramètres  $a_1$ ,  $a_2$  et  $a_3$  sont donné comme suit :

$$\begin{pmatrix} a1 \\ a2 \\ a3 \end{pmatrix} = \begin{pmatrix} 0.5934 \\ (0.1369 + 0.00062 * (\alpha_2 - \alpha_{20})) \\ (0.1445 + 0.00046 * (\alpha_3 - \alpha_{30})) \end{pmatrix} \quad (2.19)$$

Où  $\alpha_{20}$  et  $\alpha_{30}$  sont les positions initiales de  $\alpha_2$  et  $\alpha_3$

Soit  $N_i$  le nombre d'impulsions générer par l'angle  $\alpha_i - \alpha_{i0}$  (la variation de  $\alpha_i$ )

$$\alpha_i - \alpha_{i0} = N_i * a_i \quad (2.20)$$

Dans le cas de l'articulation de la base :

$$\alpha_1 = N_1 * 0.5934 \quad (2.21)$$

Dans le cas de l'articulation de l'épaule :

$$\alpha_2 - \alpha_{20} = N_2 * (0.1369 + 0.00062 * (\alpha_2 - \alpha_{20})) \quad (2.22)$$

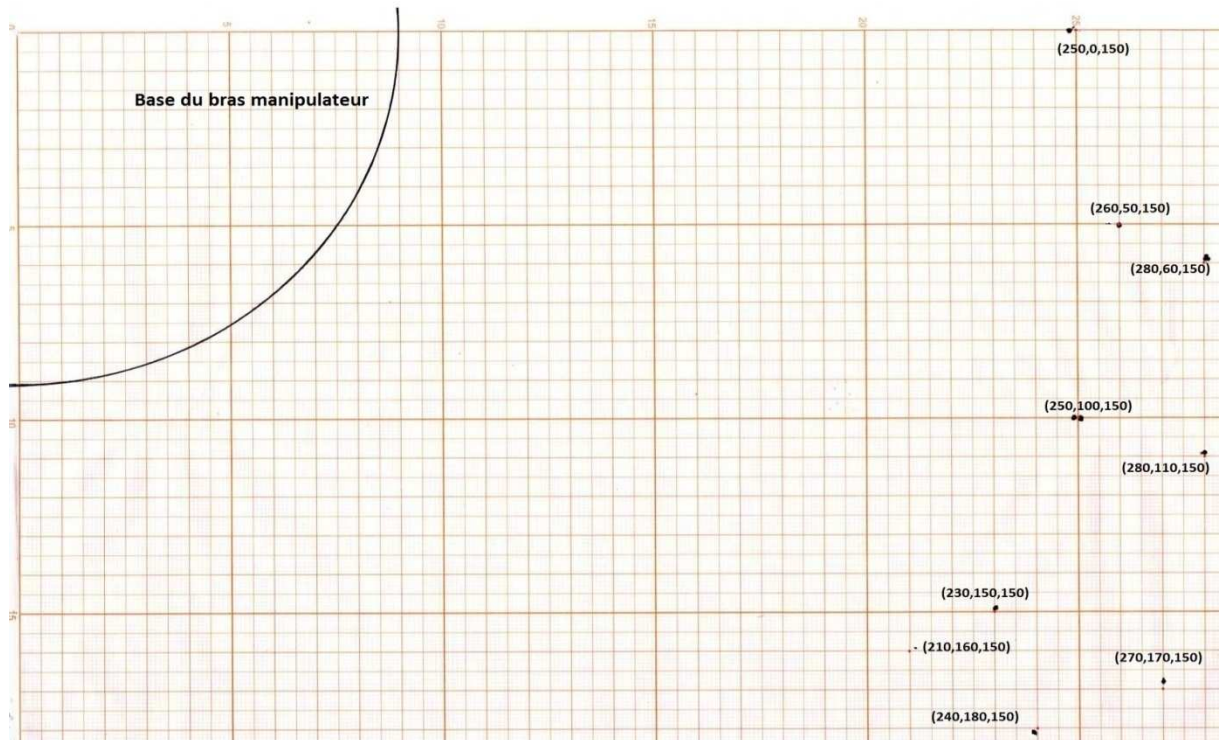
Dans le cas de l'articulation du coude :

$$\alpha_3 - \alpha_{30} = N_3 * (0.1445 + 0.00046 * (\alpha_3 - \alpha_{30})) \quad (2.23)$$

Après avoir programmé l'algorithme de commande, une série de tests constitué de 8 points différents dont la hauteur  $z$  est fixée a 12.5 cm, nous avons constaté que l'erreur maximale est de :

- 1 mm sur l'axe des  $x$
- 1 mm sur l'axe des  $y$
- 1 mm sur l'axe des  $z$

La figure suivante représente le nuage de point marqué par le bras manipulateur :



**Fig.2.8** : nuage de points marqué par le robot.

## 2.5. Conclusion

Dans ce chapitre dédié à l'étude et réalisation de la commande du bras manipulateur pour permettre à ce dernier d'effectuer sa tâche de saisie d'objets, nous l'avons divisée en trois grandes parties qui sont, la description du robot, la modélisation géométrique et la commande.

Dans la partie description nous avons décrit la configuration matérielle (type de moteurs, degré de liberté, fins de courses et positions limites) ainsi que les paramètres structuraux du robot.

Dans la deuxième partie qui est la modélisation géométrique, nous avons calculé le modèle géométrique direct ainsi que le modèle géométrique inverse qui va nous permettre de calculer les positions articulaires du robot afin d'obtenir une position désirée de l'organe terminal.

La dernière partie est consacrée à l'implémentation de la commande et les tests pratiques.

## Chapitre 3

### *Systeme de vision artificielle*

#### **3.1. Introduction**

L'asservissement visuel en robotique correspond à un asservissement d'un robot où les grandeurs asservies sont extraites d'une ou de plusieurs images fournies par un ou plusieurs capteurs de vision fixant une zone de l'environnement du robot par rapport à laquelle une tâche doit être effectuée. Ce domaine permet de rendre les robots plus performants et de développer de nouvelles tâches robotiques. En effet, en l'absence de capteurs donnant la position par rapport aux objets environnants, le robot ne peut réaliser que des tâches préprogrammées et ceci, dans un environnement invariant. Par contre avec l'intégration de capteurs de vision, des tâches non connues au préalable peuvent être envisagées même dans un environnement variant. Par exemple, s'il s'agit de réaliser des travaux de soudure au dixième de millimètre près et que l'objet à souder est décalé de 1 mm par rapport à la référence, tous les points de soudure seront décalés de cette même distance. Par contre, si on intègre au robot de soudure des capteurs de vision, même avec un décalage de l'objet à souder, la soudure peut être réalisée avec la précision voulue. Parmi les capteurs de vision les plus utilisés, on peut citer les capteurs à ultrasons et les caméras. En ce qui concerne la caméra, capteur proche de la vision humaine, l'image est acquise avec ses propres

caractéristiques de forme, de couleur et de texture. Un traitement adéquat de l'image permet, par exemple, de déterminer la position de l'objet dans son environnement par rapport à cette caméra [5].

Dans ce chapitre, nous allons étudier et réaliser la partie du mémoire relative à la détection, localisation et reconnaissance d'objets en vue de permettre au robot de réaliser sa tâche de saisie et de déplacement d'un objet arrivant sur le plateau tournant. La détection d'un objet sur la ce dernier conduit à l'arrêt immédiat de cette table. Cette détection est effective dès que l'objet entre dans le champ de vision de la caméra. La localisation permet de déterminer la position de l'objet par rapport à un repère fixe lié au robot. La reconnaissance permet d'identifier le type de l'objet à manipuler. L'ensemble de ces opérations est piloté par un système de vision artificielle dans lequel interviennent plusieurs techniques de traitement d'images et de reconnaissance de formes. A cet effet, les objets à reconnaître sont réduits à de simples objets facilement reconnaissables par leur couleur et leurs formes géométriques les plus classiques.

### 3.2. Détection de présence

Les objets utilisés dans le cadre de cette réalisation sont des objets de forme géométrique simple facilement reconnaissables qui sont divisés en quatre classe différentes : carrée, triangulaire, circulaire et hexagonale comme le montre la figure 3.1 suivante :



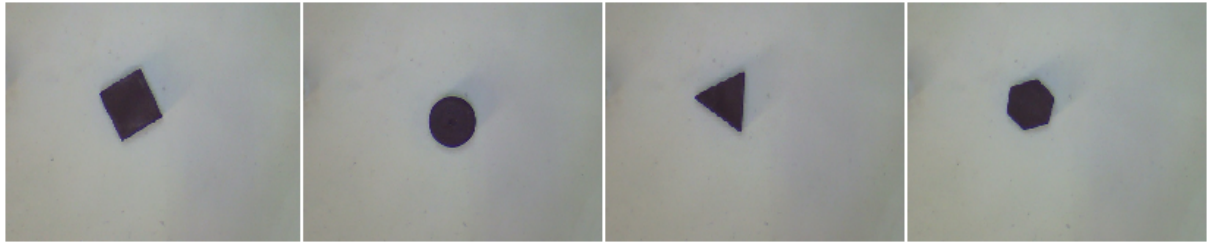
**Fig.3.1** : Les quatre classes d'objets.

L'objectif de cette partie est de détecter la présence d'un objet dans le champ de vision de la caméra et, pour réaliser cette tâche, nous avons suivi les étapes suivantes :

- Acquisition de l'image et prétraitements

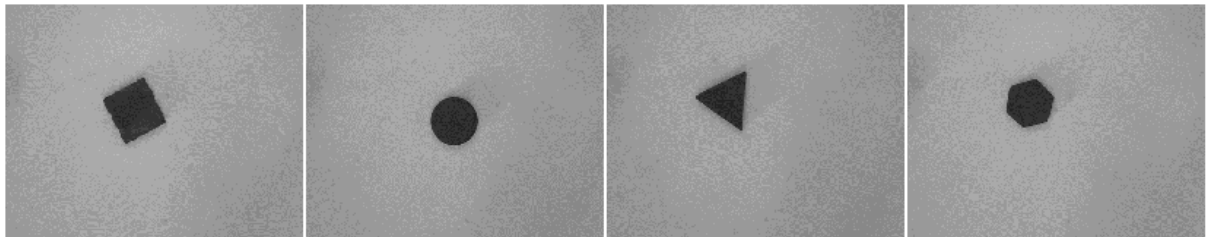
L'image acquise de la caméra est une image couleur de dimension 480x640x3 (image RGB), des opérations de prétraitement ont été effectuées sur l'image afin d'améliorer la perception des objets.

La figure 3.2 suivante représente les images acquises par la caméra des quatre objets :



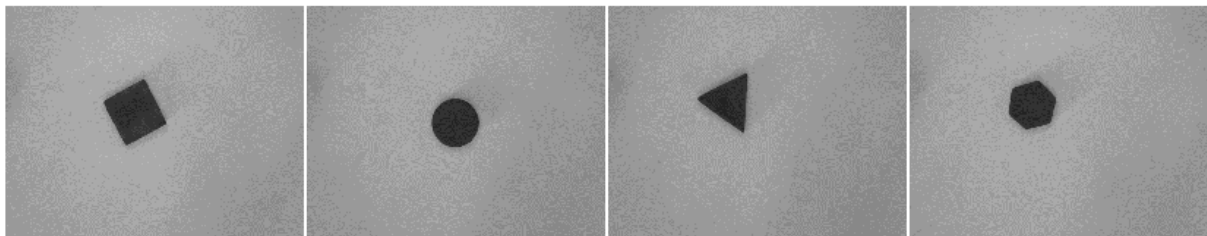
**Fig.3.2 :** Images acquises des quatre objets.

La première opération de prétraitement effectuée sur les images est la transformation de couleurs en niveaux de gris comme le montre la figure 3.3 suivante :



**Fig.3.3 :** Images aux niveaux de gris.

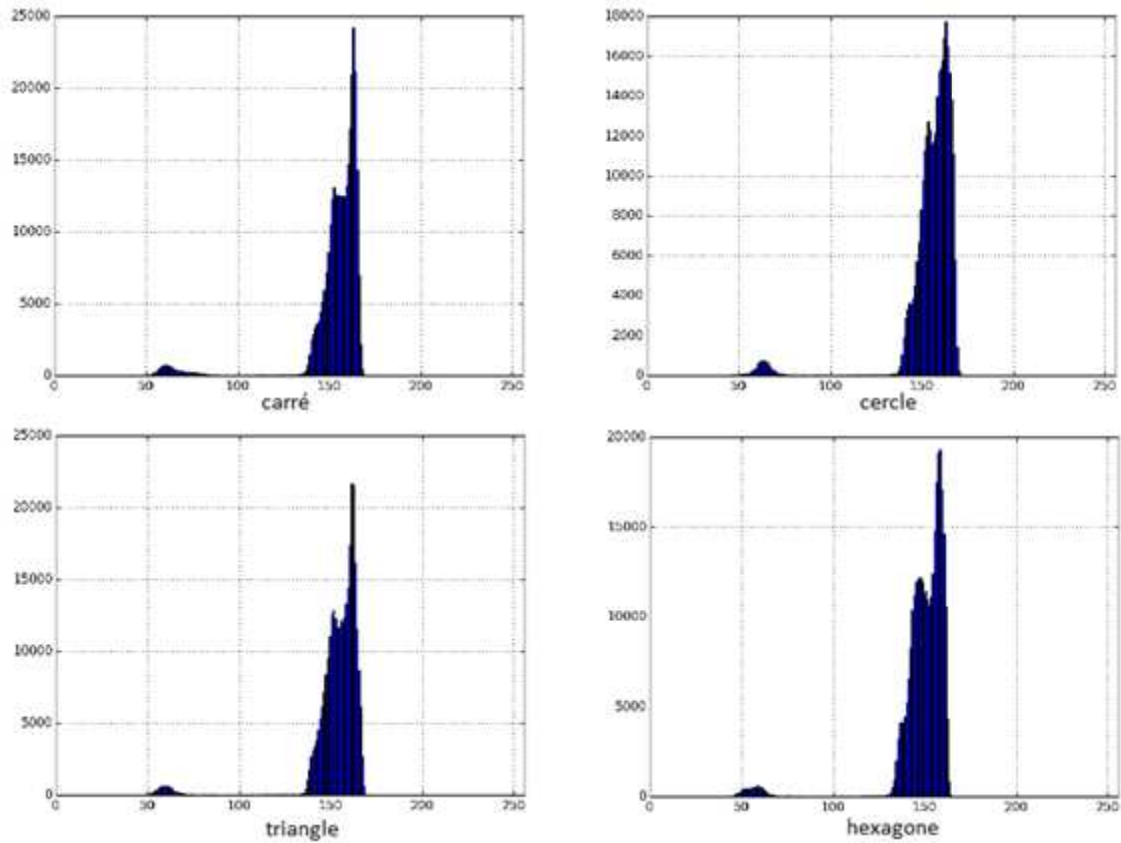
Après la transformation en niveaux de gris, une opération de filtrage est effectuée sur l'image avec un filtre médian afin d'éliminer les bruits qui affectent les pixels sans dégrader les contours de l'image comme le montre la figure 3.4 suivante :



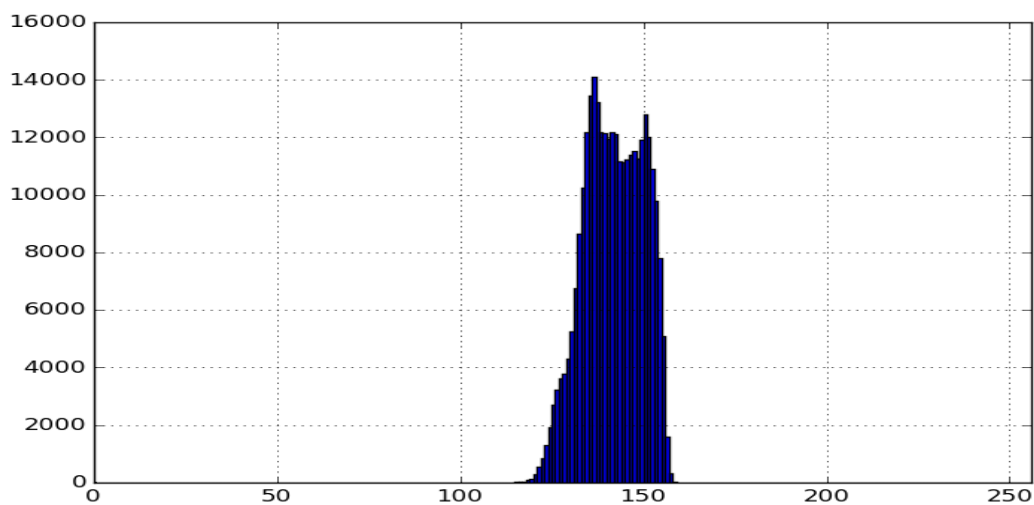
**Fig.3.4 :** Images aux niveaux de gris filtrées.

Test de présence d'un objet

En analysant les histogrammes représentés si dessous figure 3.5, on peut distinguer deux modes différents, l'un des modes représente l'objet et l'autre représente le l'arrière plan, contrairement a la figure 3.6 ou un seul mode est présent et qui représente l'arrière plan de l'image



**Fig.3.5** : Histogrammes des images en présence des objets.

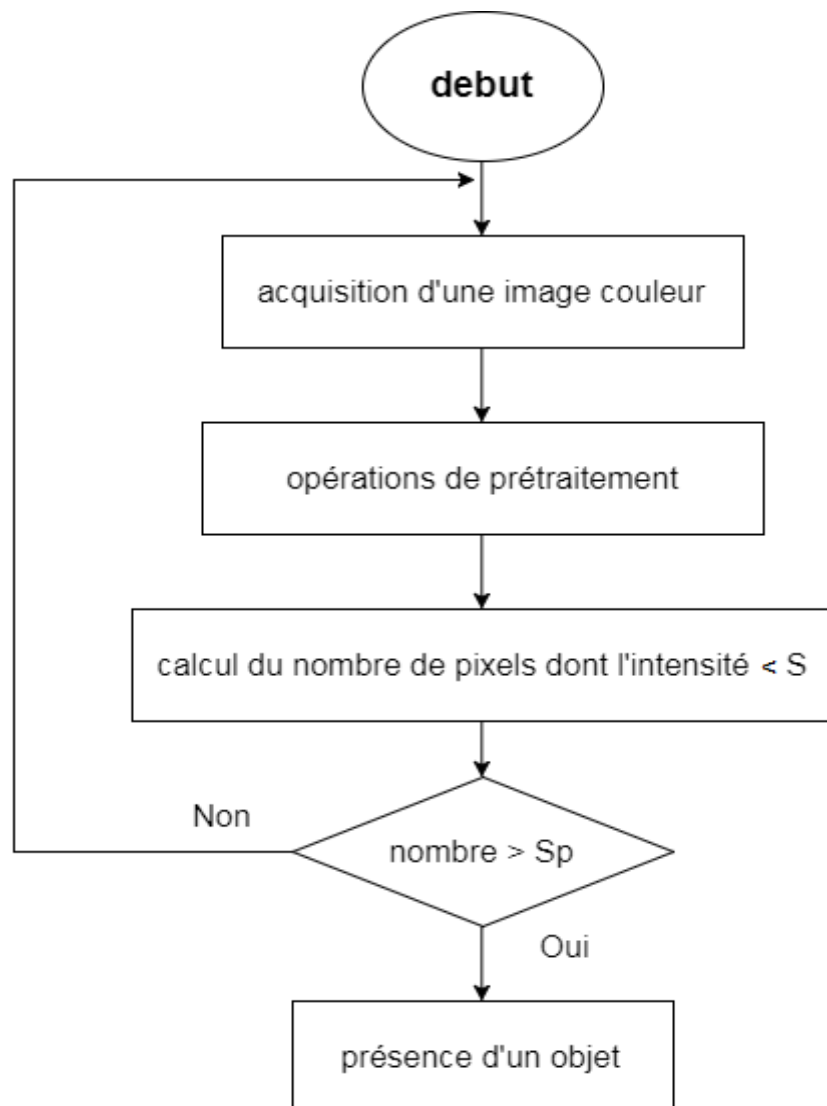


**Fig.3.6** : Histogramme de l'image sans objet.

En partons de ces histogrammes en peut déterminer un seuil « S » qui sera pris entre les deux modes et qui va nous permettre de partager les niveaux de gris appartenant à l'objet et les niveaux de gris appartenant à l'arrière plan et qui va servir aussi dans les étapes suivantes.

Le principe de la méthode de détection est de calculer le nombre de pixel dont l'intensité est inférieure au seuil S, un objet est présent si ce nombre est supérieur a un seuil  $S_p$  qui est déterminé expérimentalement et qui représente le nombre minimal de pixels sombres constituant un objet.

L'organigramme suivant représente la tâche de détection de présence des objets :

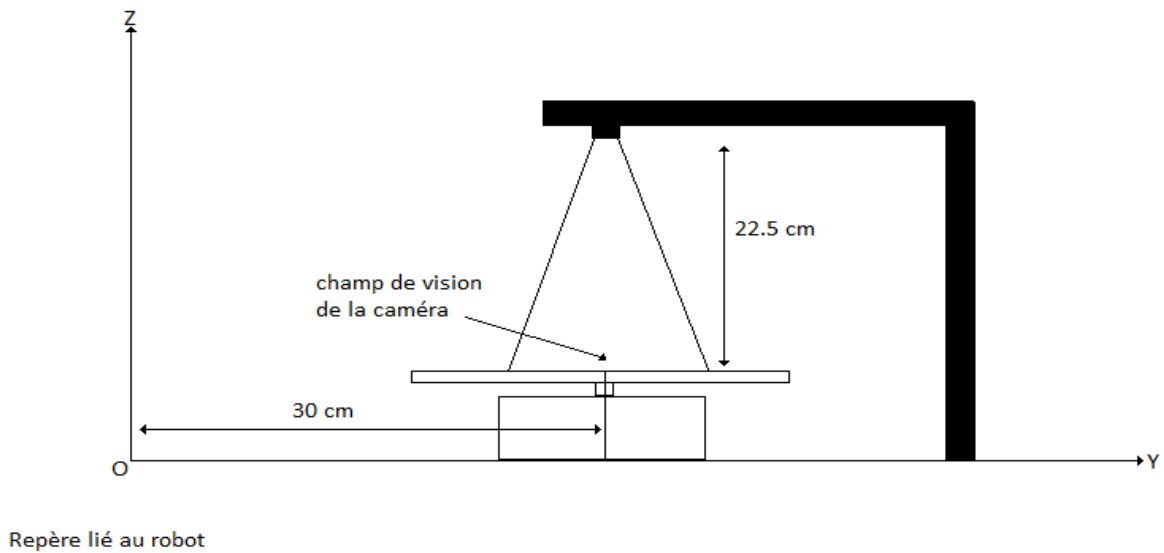


**Fig.3.7** : Organigramme de la détection de présence.

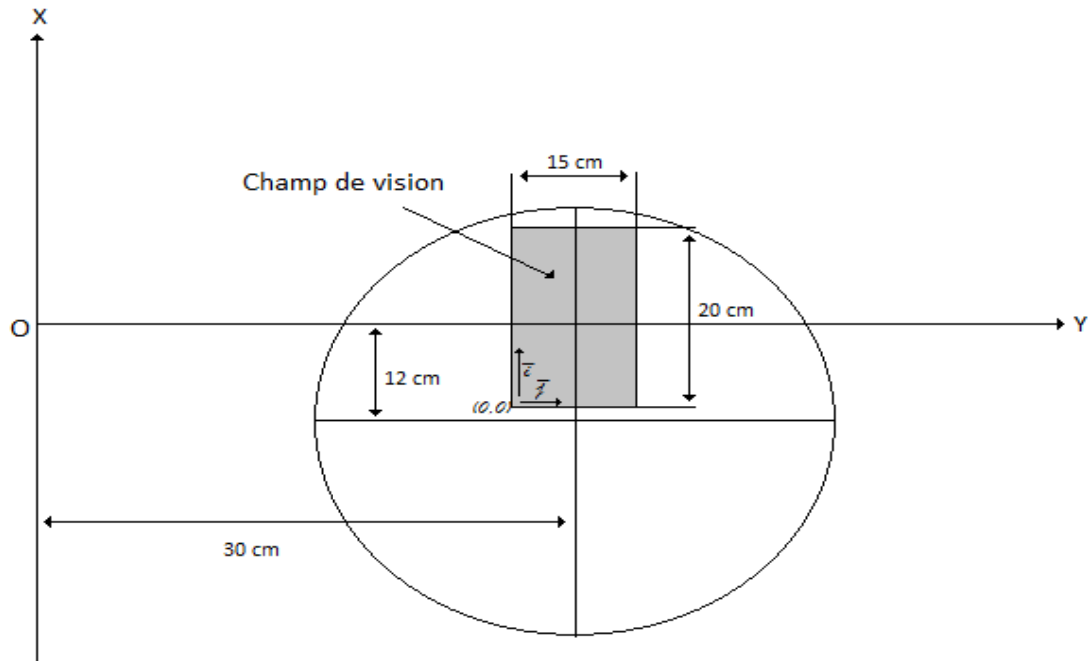
### 3.3. Localisation de l'objet

#### 3.3.1. Position de l'objet par rapport au repère fixe

L'objectif dans cette partie est de déterminer la position de l'objet détecté auparavant par rapport au repère fixe lié au robot manipulateur. Les figures 3.8 et 3.9 suivantes représentent la position du champ de vision sur le plateau tournant et par rapport au repère fixe.

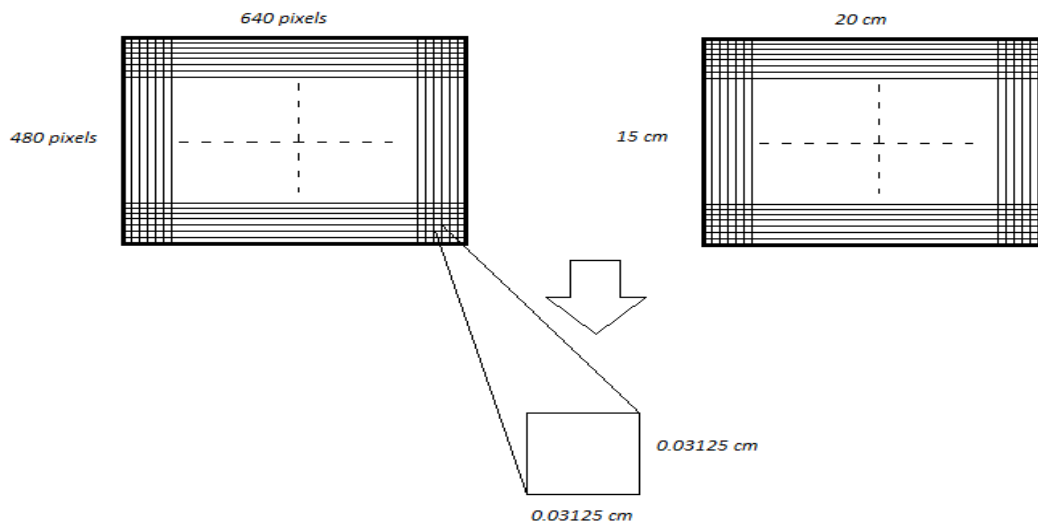


**Fig.3.8** : Position de la caméra par rapport au repère fixe.



**Fig.3.9** : position du champ de vision sur le plateau tournant.

L'image est de dimension 640x480 (pixel) et couvre une zone de 20\*15 (cm) sur le plateau, d'où chaque pixel couvre une zone de 0.03125x0.03125 (cm) comme le montre la figure 3.10 suivante :



**Fig.3.10** : Surface couverte par un pixel sur le plateau.

Chaque objet présent dans le champ de vision est localisé par son centre de gravité dont les coordonnées sont  $(X_c, Y_c)$ , d'où la position  $(X_i, Y_i)$  de ce dernier dans l'image (en centimètres) est donnée par les formules suivantes :

$$X_i = 0.03125 X_c \quad (3.1)$$

$$Y_i = 0.03125 Y_c \quad (3.2)$$

D'après la figure 3.9, la position de l'origine de l'image dans le repère lié au robot manipulateur est :

$$(X_0, Y_0) = (-10, 22.5) \quad (3.3)$$

La hauteur du plateau est constante  $h = 12.5$  cm

Ainsi, la position de l'objet par rapport à l'origine du robot est donnée par les formules suivantes :

$$\begin{aligned} X_p &= X_0 + 0.03125 X_c \\ Y_p &= Y_0 + 0.03125 Y_c \\ Z_p &= 12.5 \\ (X_p, Y_p, Z_p) &= (-10 + 0.03125 X_c, 22.5 + 0.03125 Y_c, 12.5) \end{aligned} \quad (3.4)$$

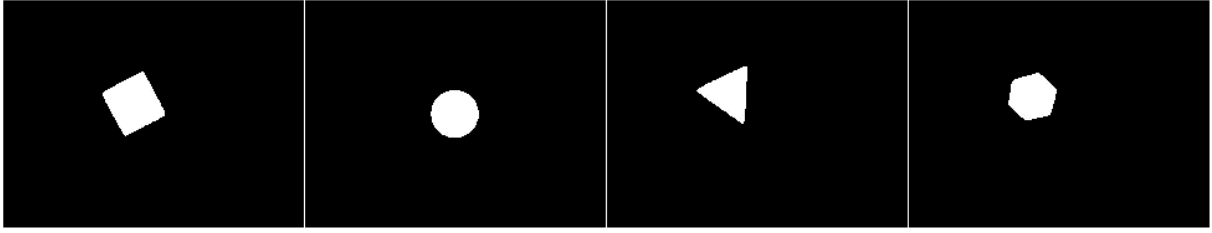
### 3.3.2. Position de l'objet dans l'image

Avant d'entamer le calcul des coordonnées du centre de l'objet, une étape de traitement d'images doit être effectuée sur l'image (après prétraitement) qui est la binarisation. Cette opération consiste donc à réaliser une opération qui transforme l'image numérique acquise et filtrée en une image binaire dont la surface représentant l'objet est blanche et le reste de la surface du plateau est noir [5]. Pour ce faire la méthode d'Otsu [10] est appliquée sur l'image. Cette méthode repose sur la sélection d'un minimum entre deux modes pour le choix du seuil, dans notre cas le seuil « S » est déjà calculé (partie détection)

Les nouvelles valeurs de l'image sont données donc par la formule suivante :

$$I(i, j) = \begin{cases} 0 & \text{si } I(i, j) > S \\ 1 & \text{si } I(i, j) \leq S \end{cases} \quad (3.5)$$

La figure 3.11 suivante représente le résultat de la binarisation des images des quatre objets :



**Fig.3.11** : Résultat de la binarisation des images.

Dans cette partie, nous cherchons à déterminer le centre de gravité de l'objet en question. Le calcul du centre de l'objet a été effectué en appliquant la procédure suivante sur les images binaire :

Soit  $\mu_{nm}$  le moment géométrique d'ordre « n+m » dont la formule est donnée comme suit [11]:

$$\mu_{nm} = \sum_{i=0}^{Nl-1} \sum_{j=0}^{Nc-1} i^n * j^m * I(j, i) \quad (3.6)$$

Où  $I(j, i)$  est l'image binaire calculée auparavant.

Nous avons  $\mu_{00}$ , le moment d'ordre « 0 » et appelé aussi superficie :

$$\mu_{00} = \sum_{i=0}^{Nc-1} \sum_{j=0}^{Nl-1} I(j, i) \quad (3.7)$$

Cette somme représente le nombre de pixels appartenant à l'objet.

On a aussi les moments d'ordre 1 ( $\mu_{10}$  et  $\mu_{01}$ )

$$\mu_{10} = \sum_{i=0}^{Nc-1} \sum_{j=0}^{Nl-1} i * I(j, i) \quad (3.8)$$

$$\mu_{01} = \sum_{i=0}^{Nc-1} \sum_{j=0}^{Nl-1} j * I(j, i) \quad (3.9)$$

Ces deux sommes représentent respectivement la somme des indices « i » et la somme des indices « j » appartenant à l'objet.

Donc les coordonnées du centre sont données comme suit :

$$X_c = \frac{\mu_{10}}{\mu_{00}} \quad (3.10)$$

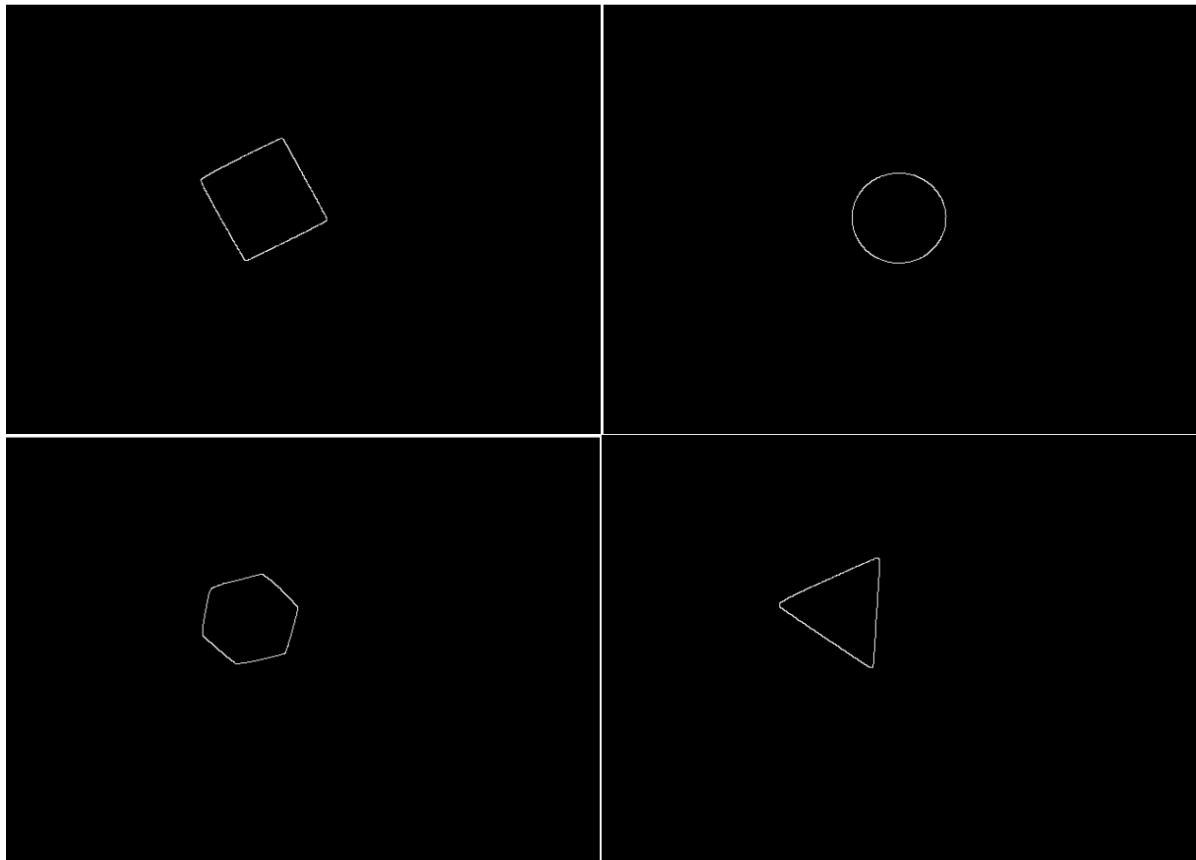
$$y_c = \frac{\mu_{01}}{\mu_{00}} \quad (3.11)$$

## 3.4. Reconnaissance de forme

La dernière partie de ce chapitre est consacrée à la reconnaissance de forme des objets détectés. Pour cela, deux étapes ont été effectuées afin de classer l'objet selon sa forme.

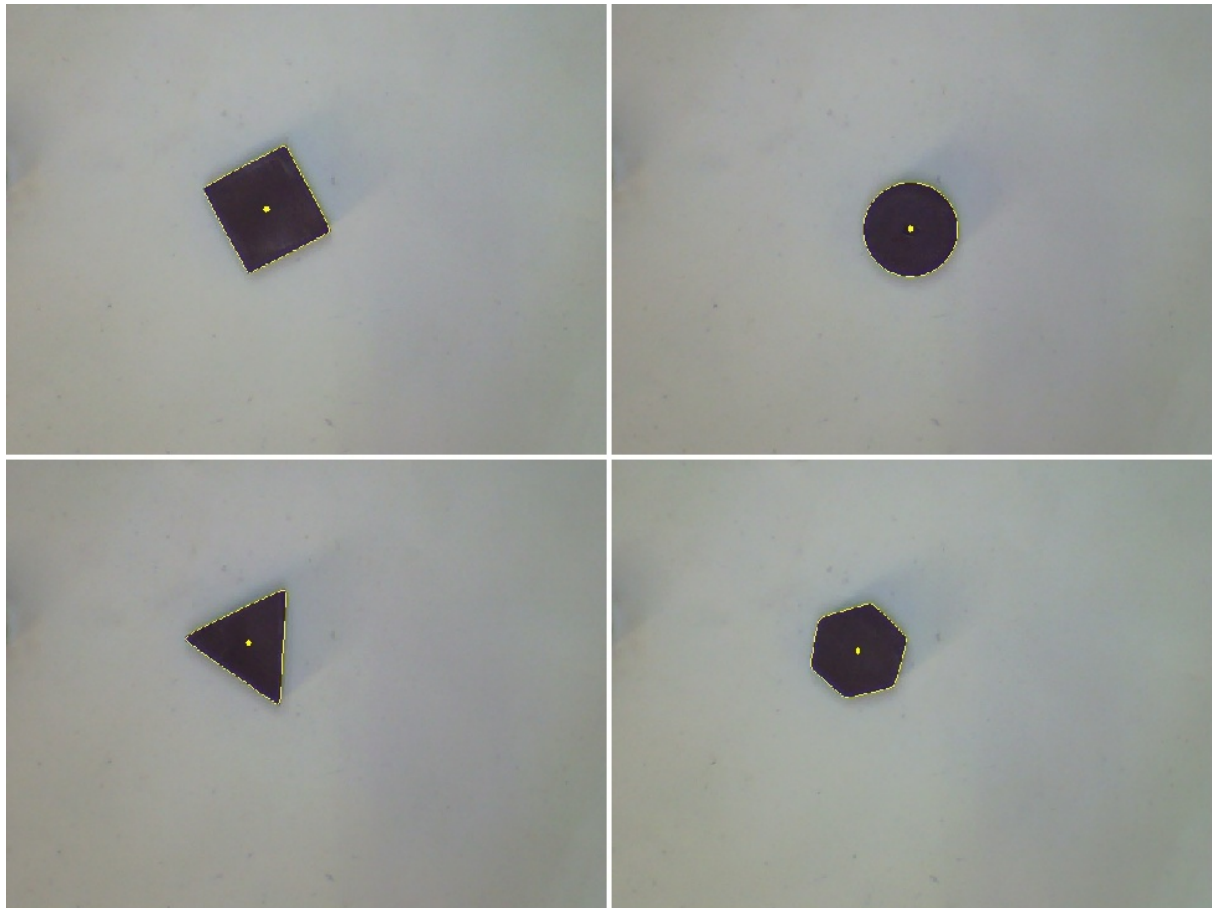
### 3.4.1. Détection des contours

Comme son nom l'indique, dans cette étape nous avons utilisé le filtre de Canny [12] pour détecter les contours des objets dans les images binaires. La figure 3.12 suivante illustre le résultat de la détection des contours avec cette méthode :



**Fig.3.12** : résultat de la détection des contours.

La figure 3.13 suivante représente les centres de gravité et les contours superposés sur les images acquises des quatre classes d'objets :



**Fig.3.13** : Contours et centres de gravité représentés sur les images acquises.

### 3.4.2. Méthode de classification

Dans cette étape nous calculons une caractéristique discriminante de chacune des classes d'objets. Cette caractéristique est la signature polaire de l'objet présent dans l'image en question. Elle correspond à la courbe des distances entre le centre de gravité et les pixels du contour à partir du pixel le plus loin du centre de gravité [5].

Le vecteur contenant la signature polaire est donnée par la formule (3.12).

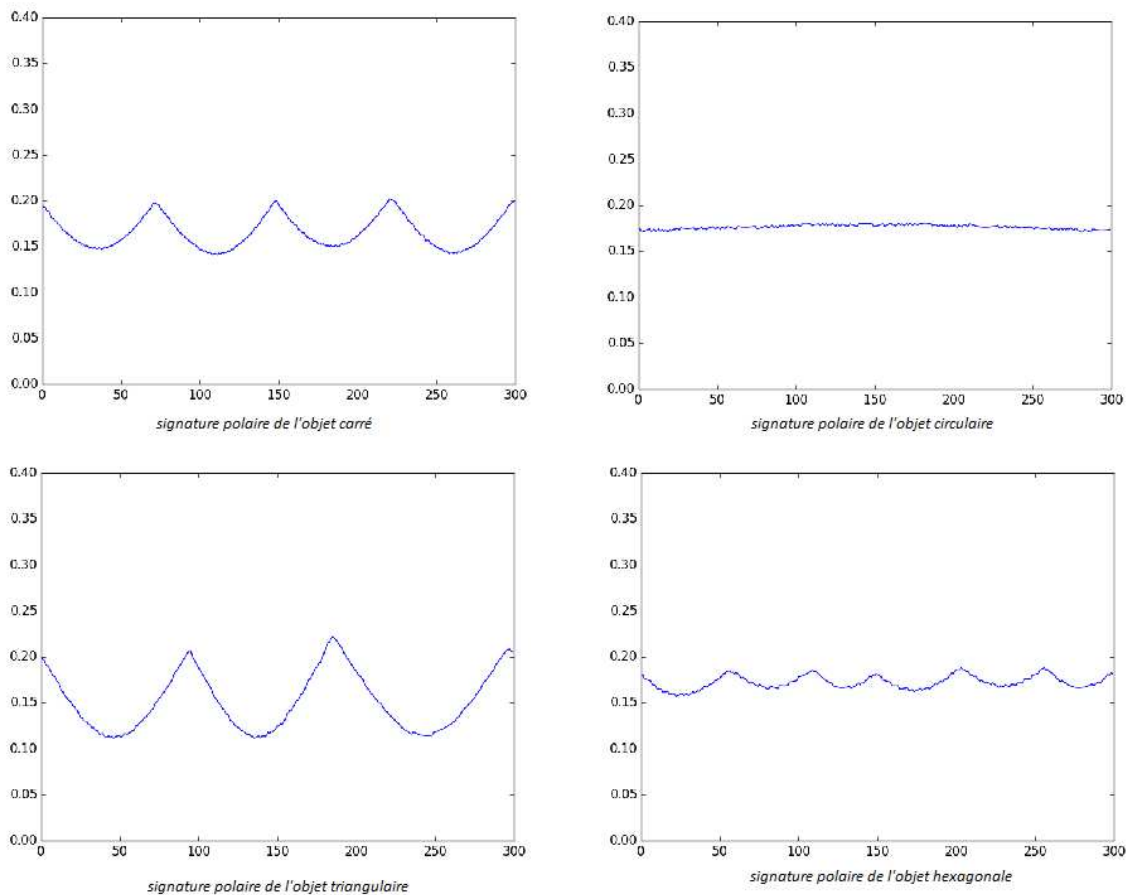
$$S_m = \sqrt{(i - i_c)^2 + (j - j_c)^2} \quad (3.12)$$

Où  $(i, j)$  sont les coordonnées des points du contour

La simplicité de formes géométriques utilisées nous permet d'avoir une invariance de la signature polaire par rapport à la rotation, et pour avoir une invariance par rapport à l'échelle nous avons effectué les deux opérations suivantes :

- Normalisation de la signature qui permet d'avoir une plage de variation fixe indépendante de la taille de l'objet
- Ré-échantillonnage de la signature polaire qui permet d'avoir un nombre de points  $N = 300$  invariant vis à vis la taille du contour.

La figure 3.14 suivante représente les signatures polaires des quatre classes d'objets normalisés et ré-échantillonnés :

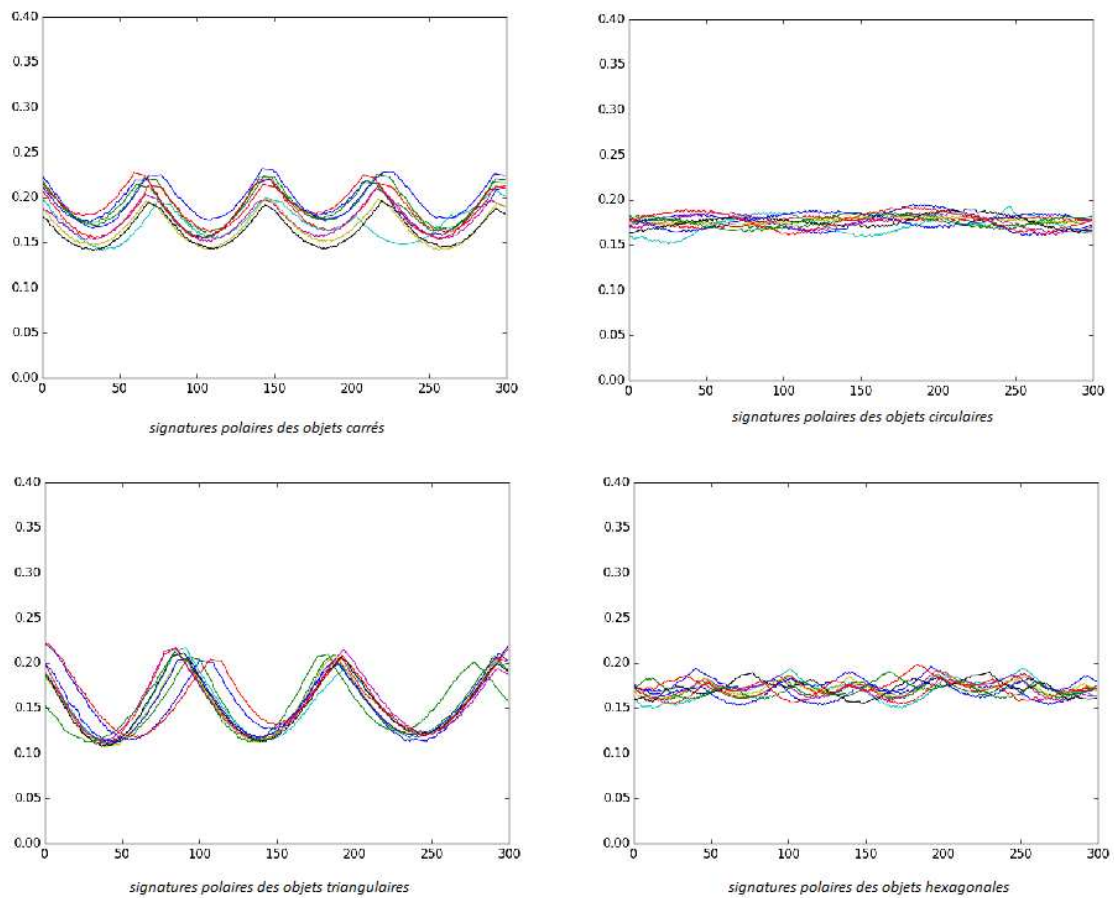


**Fig.3.14** : les signatures polaires des quatre classes d'objets.

Cette figure montre bien que la signature polaire permet de bien distinguer chacune des quatre formes en question.

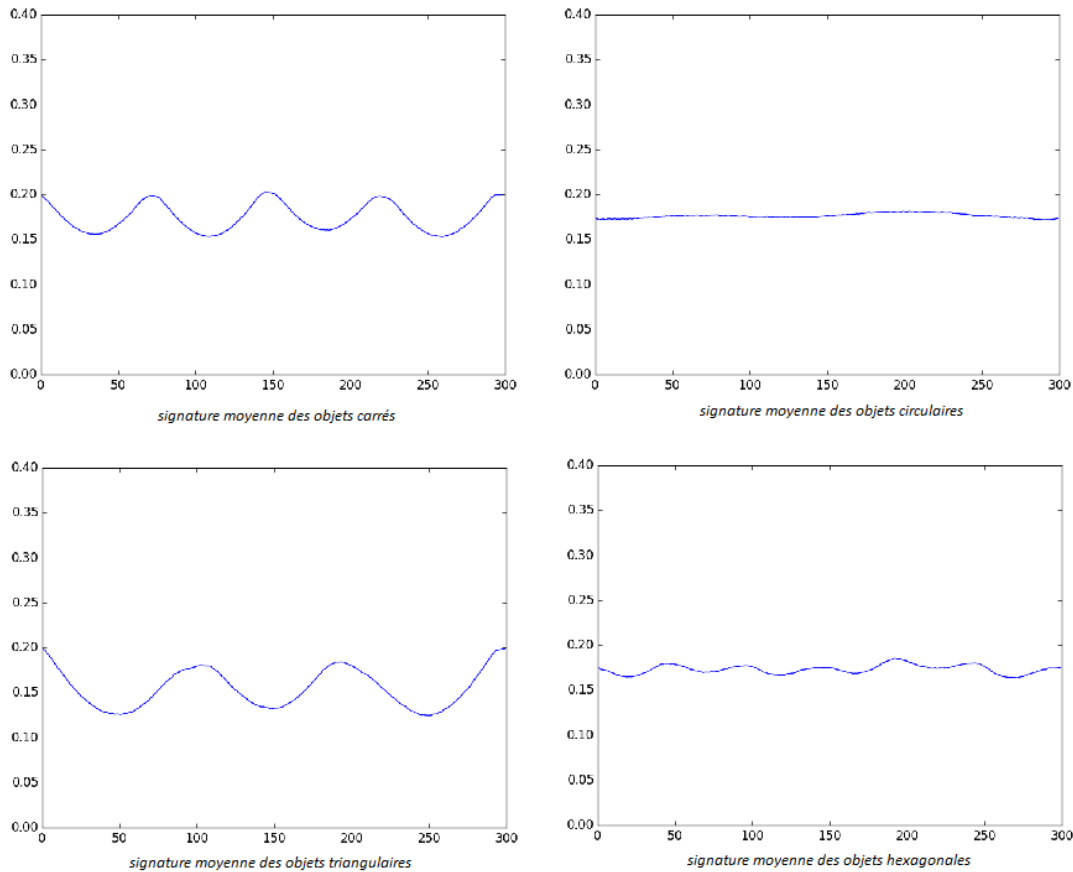
Pour pouvoir classer un objet détecté, nous devons comparer sa signature polaire par rapport à une base de données. Pour ce faire, nous avons réalisé une série tests sur un ensemble de 20 échantillons par classe d'objets dont

le résultat des signatures polaires de ces échantillons est représenté sur la figure 3.15 suivante :



**Fig.3.15 :** Signatures polaires des différents échantillons de chaque classe.

Après avoir effectué ces tests, une matrice d'apprentissage de taille 4x300 qui contient les courbes moyennes de chacune des classes d'objets est enregistrée afin de réaliser une base de données à laquelle la signature polaire sera comparée. La figure 3.16 suivante représente les signatures polaires moyennes de chaque classe :



**Fig.3.16** : Signatures polaires moyennes des quatre classes d'objets

La méthode utilisée pour la classification est la distance euclidienne minimale [13], son principe est comme suit :

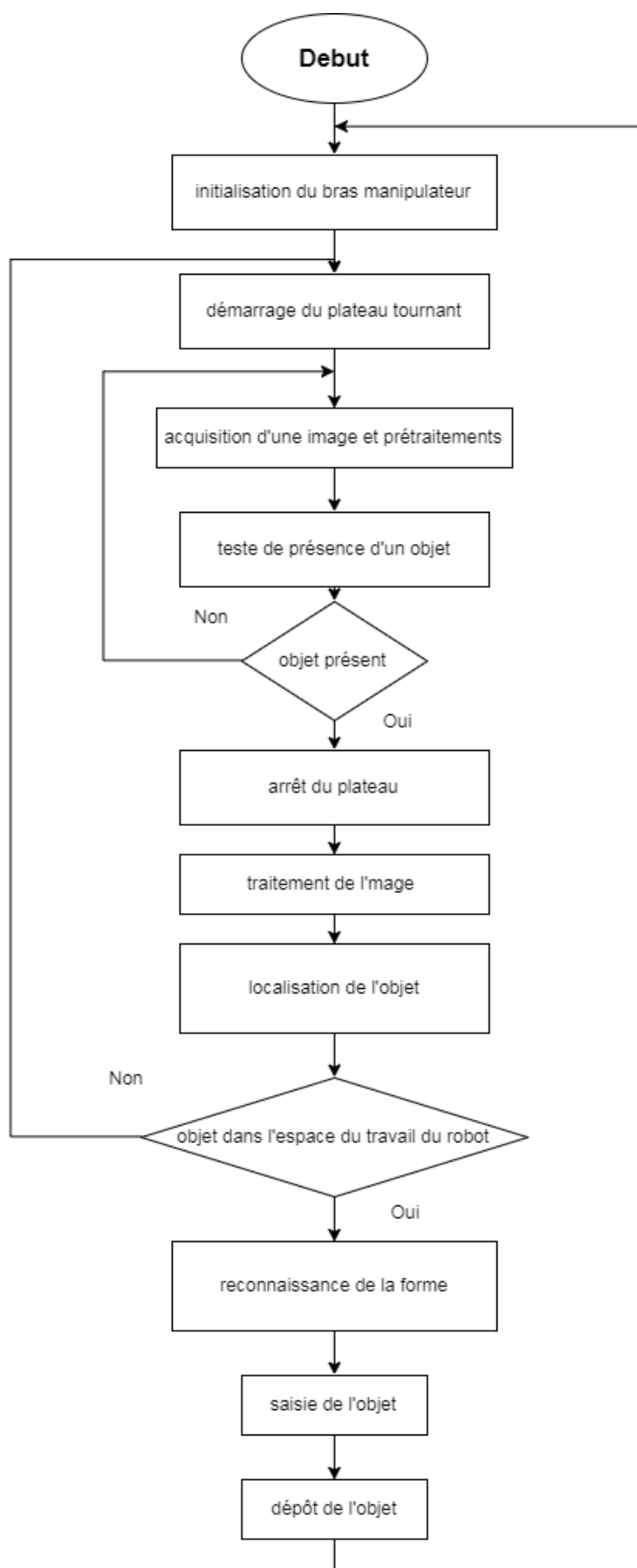
Soit  $S_m$  la signature polaire de l'objet détecté et  $S_{a1}$ ,  $S_{a2}$ ,  $S_{a3}$  et  $S_{a4}$  les signatures polaires moyennes enregistrées lors de l'apprentissage.

La méthode consiste à calculer les distances euclidiennes entre  $S_m$  et les  $S_{ai}$  avec  $i = 1, 2, 3, 4$  selon la formule suivante :

$$E(i) = \sqrt{(S_m(0) - S_{ai}(0))^2 + \dots + (S_m(N-1) - S_{ai}(N-1))^2} \quad (3.13)$$

La classe de l'objet est celle qui correspond à la distance euclidienne  $E(i)$  minimale.

La figure 3.17 suivante représente l'organigramme général de notre réalisation :



**Fig.3.17** : organigramme général de l'application.

Tests pratiques :

Après l'écriture du programme correspondant à l'organigramme général de commande, nous avons réalisé une série de tests pratiques de ce programme : dix tests sur chaque classe d'objets. Nous avons effectué chaque test de saisie et de déplacement d'un objet de sa position initiale définie par les coordonnées de son centre de gravité  $(X_p, Y_p, Z_p)$ , vers son emplacement défini par les coordonnées  $(X_d, Y_d, Z_d)$  que doit avoir le centre de la pince lors du dépôt de l'objet à son emplacement, et On a obtenu les résultats suivants :

- Taux de détection des objets :
  - Le taux de détection sur la totalité de l'ensemble des tests est de 100%.

- Erreur maximale lors de la localisation des objets :

L'erreur maximale commise sur la position du centre de gravité, des objets est :

- 3 mm suivant l'axe X.
- 3 mm suivant l'axe Y.

Ces erreurs sont dues à l'apparition de la troisième dimension de l'objet (transformation de la scène 3D en une image 2D) lors de l'acquisition de l'image.

- Taux de reconnaissance :
  - 100% pour les pièces carrées.
  - 100% pour les pièces circulaires.
  - 100% pour les pièces triangulaires.
  - 80% pour les pièces hexagonales.

Ces tests ont été réalisés dans des conditions de luminosité optimale afin d'éliminer l'apparition des ombres sur l'arrière plan.

### **3.5. Conclusion**

Dans ce chapitre nous avons étudié et réalisé la partie vision artificielle de ce travail, dont l'objectif est de permettre au robot d'interagir avec son environnement et d'avoir des informations sur les variations et changements de ce dernier. Pour cela nous l'avons devisé en trois parties qui sont les suivantes :

- La détection de la présence d'un objet sur le plateau tournant qui utilise le nombre de pixels sombres comme indice de présence d'un objet,
- La localisation du centre gravité de l'objet en se basant sur la méthode des moments géométriques,
- Et enfin la reconnaissance de formes qui utilise le principe de classification par distance euclidienne minimale appliqué sur la caractéristique discriminante (signature polaire) calculée à partir du contour de l'objet et son centre de gravité.

## Conclusion générale

Notre travail consistait à étudier, sur le plan théorique et pratique, la commande d'un système composé d'un plateau tournant, d'un bras manipulateur et un système de vision artificielle. Le fonctionnement de cet ensemble piloté par un microcalculateur consiste à reconnaître, en temps réel, des objets arrivant sur le plateau tournant pour ensuite, les saisir et les déplacer à l'aide du bras manipulateur.

Pour atteindre cet objectif constitué d'une étude théorique et d'une réalisation pratique, nous avons subdivisé notre travail en trois étapes. La première est consacrée à la réalisation d'une boucle de commandes numérique en vitesse du moteur à courant continu assurant le mouvement de rotation de la table. La réalisation de la commande du robot permettant la saisie et le déplacement de l'objet vers l'emplacement correspondant à fait l'objet de la seconde partie. En dernière partie, La détection, la localisation et la reconnaissance des objets.

Pour la mise en oeuvre de la commande numérique du plateau tournant en premier lieu nous avons étudié les constituants de la table tournante puis nous avons réalisé les étapes permettant la commande de la table à savoir l'étape de l'identification, l'étape de calcul de la loi de commande, l'étape de simulation et validation ainsi que l'étape de réalisation de la loi de commande calculée. Dans l'étape de l'identification, nous avons déterminé la structure du modèle sous forme de fonction de transfert du second ordre en utilisant le principe des moindres carrés pour déterminer les paramètres optimaux impliquant la réponse indicielle la plus proche possible de celle relevée pratiquement. La loi de commande est obtenue à partir du calcul d'un régulateur numérique permettant de rendre le fonctionnement de la table performant, c'est dire avec un temps de réponse réduit et une

compensation de l'effet des perturbations. Après simulation et validation de cette loi de commande, nous avons donné l'algorithme permettant de réaliser cette loi de commande par un régulateur numérique.

En seconde partie de notre travail consacrée à la commande du robot manipulateur, après l'étude des caractéristiques et des constituants du robot, nous avons déterminé les modèles géométriques direct et inverse qui ont été utilisés dans l'élaboration du programme de commande du robot.

Dans la dernière partie dédiée à la vision artificielle, pour détecter les objets, la stratégie utilisée consiste à comparer la somme des niveaux de gris inférieure à un seuil et de comparer cette somme à seuil de présence. La localisation d'objet passe par la détermination du centre de gravité de l'objet en utilisant les moments géométriques dans l'image binaire à partir duquel les coordonnées du centre de gravité d'un objet peuvent être déterminés par la connaissance de ses coordonnées dans l'image. Enfin, la partie reconnaissance de forme des objets est réalisée suivant le schéma classique d'un processus de reconnaissance de formes. Après leurs acquisitions, les images ont subi une étape de prétraitement puis une étape de traitement où les signatures polaires sont extraites. Dans la partie classification, la méthode de la distance euclidienne minimale est utilisée pour déterminer la classe de l'objet à partir de cette signature polaire et d'une base d'apprentissage composée de signature polaires moyenne d'objets connus.

En perspective, il serait intéressant d'inclure un matériel professionnel incluant une caméra haute définition reliée à une carte d'acquisition permettant l'acquisition des images en temps réel et un robot industriel plus performant et plus rapide et un ensemble de capteurs. L'essai se fera avec différents types d'objets où interviennent la forme, la couleur et la texture.

# Annexes



**L293D**  
**L293DD**

## PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

### DESCRIPTION

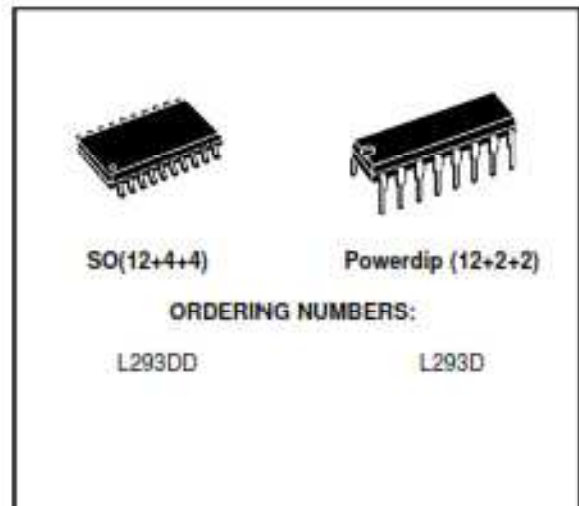
The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

### ABSOLUTE MAXIMUM RATINGS

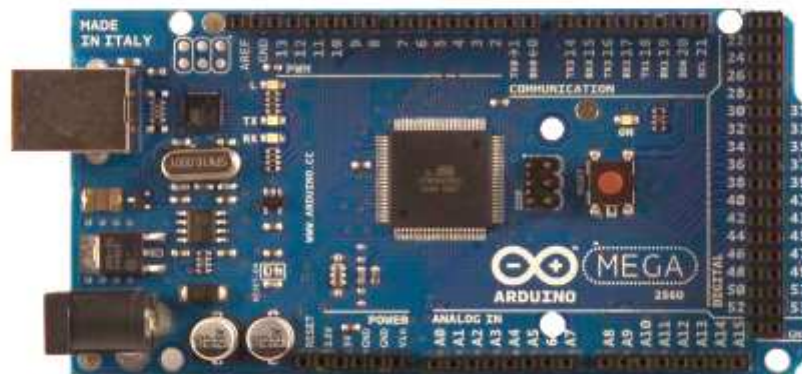
Symbol	Parameter	Value	Unit
$V_{S}$	Supply Voltage	36	V
$V_{SS}$	Logic Supply Voltage	36	V
$V_i$	Input Voltage	7	V
$V_{en}$	Enable Voltage	7	V
$I_{oi}$	Peak Output Current (100 $\mu$ s non repetitive)	1.2	A
$P_{tot}$	Total Power Dissipation at $T_{amb} = 90^{\circ}\text{C}$	4	W
$T_{stg}, T_j$	Storage and Junction Temperature	- 40 to 150	$^{\circ}\text{C}$



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

## Arduino MEGA 2560



### Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Technical Specification

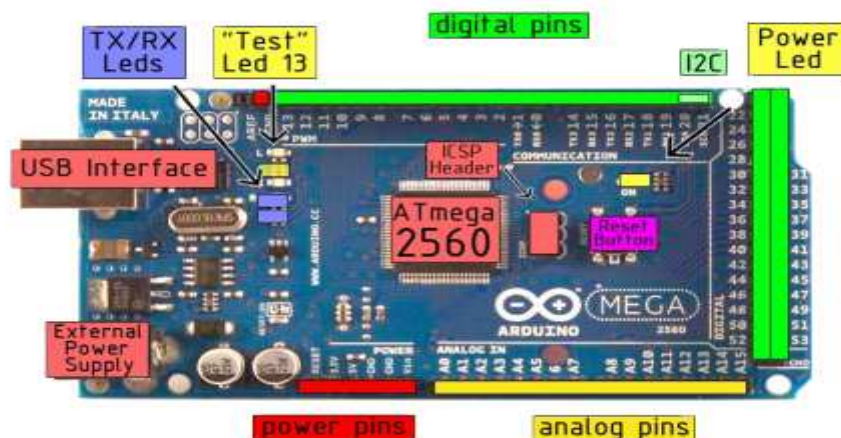


EAGLE files: [\\_arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

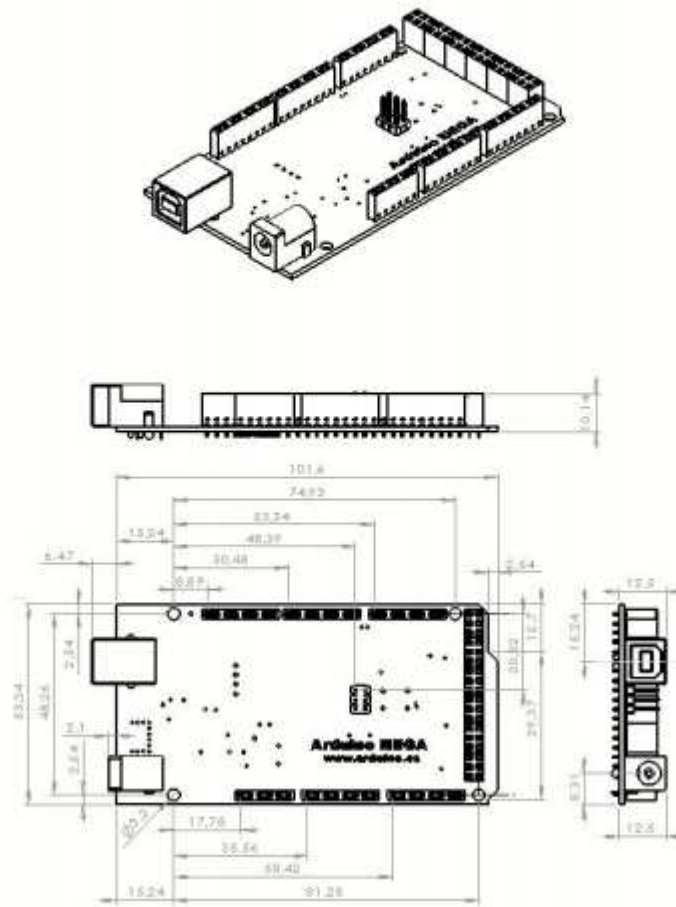
### Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

### the board



## Dimensioned Drawing



Annexe 2 : Arduino Méga 2560.



# Raspberry Pi

## Raspberry Pi 3 Model B



**Product Name**

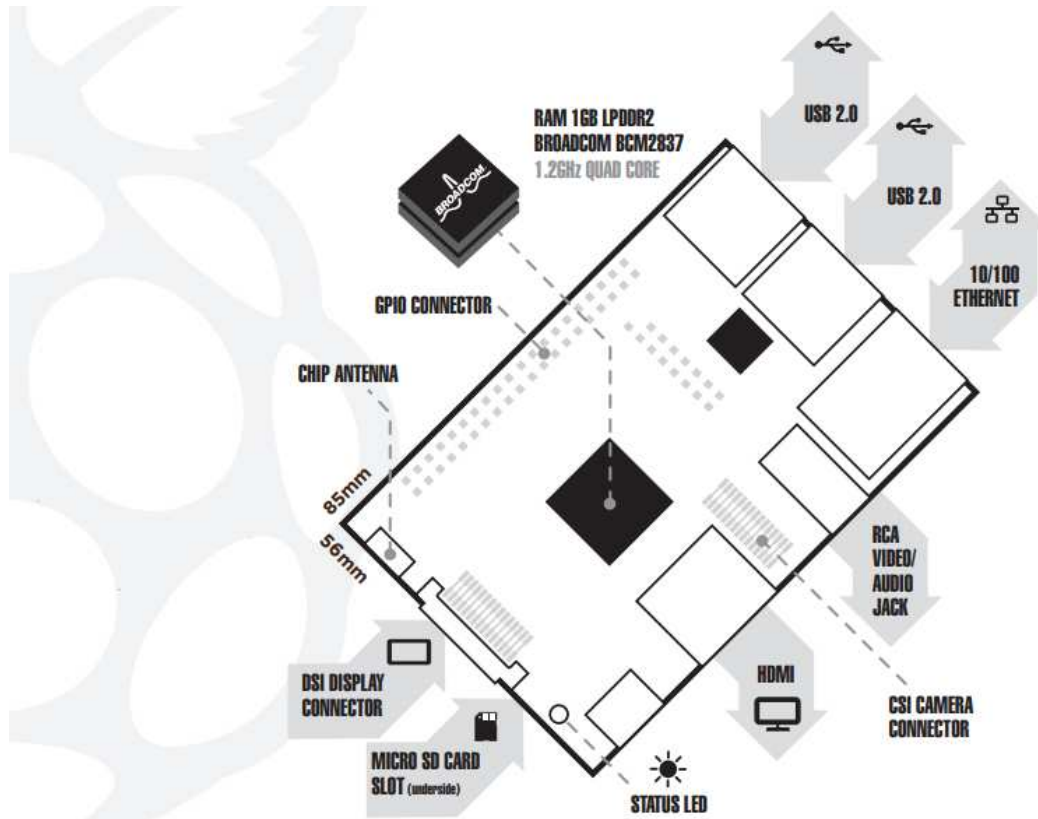
**Raspberry Pi 3**

**Product Description**

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

**RS Part Number**

**896-8660**



## Raspberry Pi

### Raspberry Pi 3 Model B

#### Specifications

##### Processor

Broadcom BCM2837 chipset,  
1.2GHz Quad-Core ARM Cortex-A53  
802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)

##### GPU

Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode.

Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

##### Memory

1GB LPDDR2

##### Operating System

Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT

##### Dimensions

85 x 56 x 17mm

##### Power

Micro USB socket 5V1, 2.5A

### Connectors:

<b>Ethernet</b>	10/100 BaseT Ethernet socket
<b>Video Output</b>	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
<b>Audio Output</b>	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
<b>GPIO Connector</b>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<b>Camera Connector</b>	15-pin MIPI Camera Serial Interface (CSI-2)
<b>Display Connector</b>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<b>Memory Card Slot</b>	Push/pull Micro SDIO

---

### Key Benefits

- Low cost
- 10x faster processing
- Consistent board format
- Added connectivity

---

### Key Applications

- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming

## Annexe 3 : Raspberry pi 3.

## Bibliographie

[1] Mohamed Wissem NAOUAR, Éric MONMASSON, Ilhem SLAMA BELKHODJA, Ahmad Ammar NAASSANI, (2009), "Introduction à la commande numérique des machines électriques". Editions T.I.

2] Lycée Gustave Eiffel de Dijon, (2017), "Modélisation du Moteur à Courant Continu", Classe préparatoire P.T.S.I.

[3] François Bailly, Julien Flamant, (2014), "Identification par la méthode des moindres carrés et par la méthode du modèle : principe et exemples", Récupéré le 23-03-2017 de :  
<https://jflamant.files.wordpress.com/2014/02/17-moindrescarrés.pdf>.

[4] Papanicola Robert, (2003), "Notion de correction des systèmes asservis", Récupéré le 08-04-2017 de  
[http://sciences-indus-cpge.papanicola.info/IMG/pdf/SA7-Correction\\_des\\_S-A.pdf](http://sciences-indus-cpge.papanicola.info/IMG/pdf/SA7-Correction_des_S-A.pdf).

[5] Triki, A, (2010), "contribution à l'étude et réalisation d'un asservissement comprenant une table tournante, un robot et un système de vision artificielle", mémoire de magister Université de Mouloud MAMMERI, Tizi Ouzou, Algérie.

[6] Wisama Khalil, (2002), "Commande des robots manipulateurs", Hermes Science – Lavoisier.

[7] Wisama Khalil, Etienne DOMBRE, (2012), "Bases de la modélisation et de la commande des robots-manipulateurs de type série", Récupéré le 06-03-2017 de  
[http://cours-online.gdr-robotique.org/Khalil-Dombre\\_Modelisation/Khalil-Dombre\\_Modelisation.pdf](http://cours-online.gdr-robotique.org/Khalil-Dombre_Modelisation/Khalil-Dombre_Modelisation.pdf).

[8] Hamdi Hocine, (2003), "Introduction a la robotique", Récupéré le 10-05-2017 de <https://www.fichier-pdf.fr/2014/06/04/hh-intro-robotique/>.

[9] Jacques Gangloff, (2012), " Robotique Manipulation et commande", Récupéré le 09-07-2017 de [http://icube-avr.unistra.fr/fr/images/c/c1/Cours\\_rob\\_4\\_2012.pdf](http://icube-avr.unistra.fr/fr/images/c/c1/Cours_rob_4_2012.pdf).

[10] Lionel Lacassagne, (2012), "Segmentation / Seuillage / binarisation", Récupéré le 24-05-2017 de [http://andresromero.github.io/lectures/TI\\_Chap1\\_Seuillage\\_2011\\_2012%20\(1\).pdf](http://andresromero.github.io/lectures/TI_Chap1_Seuillage_2011_2012%20(1).pdf).

[11] Philippe Foucher, (2005), "Reconnaissance de caractères: Descripteurs de formes et classification", Récupéré le 21-08-2017 de [http://www2.univ-paris8.fr/ingenierie-cognition/master-andi/etudiant/cours/informatique/cours\\_foucher/traitement\\_signal\\_05\\_06/rdf3.pdf](http://www2.univ-paris8.fr/ingenierie-cognition/master-andi/etudiant/cours/informatique/cours_foucher/traitement_signal_05_06/rdf3.pdf).

[12] Colin Leveger, (2016), "Détection des contours d'une image : le filtre de Canny", Récupéré le 15-07-2017 de <https://colinleverger.fr/assets/projects/CANNY-COLIN-LEVERGER.pdf>.

[13] Fabien Chevalier, Jérôme Le Bellac, (2013), "La classification", Récupéré le 03-08-2017 de

<https://perso.univ-rennes1.fr/valerie.monbet/ExposesM2/2013/Classification2.pdf>.

Cocquerez, J. P, Philipp-Foliguet, S, (1995), "Analyse d'images: filtrage et segmentation", Masson.

[15] ARDUINO MEGA, Datasheet, [www.arduino.cc](http://www.arduino.cc).

[16] Christian TAVERNIER, (2014), "Arduino : Maitrisez sa programmation et ses cartes d'interface (shields)" , DUNOD.

[17] Raspberry pi 3, Datasheet, [www.raspberrypi.org](http://www.raspberrypi.org).

[18] Simon MONK, (2014), "Programmez un raspberry pi : initiation avec python", DUNOD.

[19] Christian TAVERNIER, (2014), "Raspberry pi : Prise en main et premières réalisations", DUNOD.