

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 : automatique et système

Présenté par
OUAZANI Yazid

Mémoire dirigé par le Pr. Moussa DIAF et co-dirigé par Mr. Salim IRATNI

Thème

**Etude et réalisation d'un système de
détection et de suivi d'une personne en
mouvement**

Mémoire soutenu publiquement le 26 Septembre 2018 devant le jury composé de :

M Prénom NOM

MCB, ALIBEY Mohammed, Président

M Prénom NOM

professeur, DIAF Moussa, Rapporteur

M Prénom NOM

MCB, BOUCHEBBAT Rochedi, Examineur

M Prénom NOM

doctorant, IRATNI Salim, Co-Rapporteur

Avant-propos

Ce travail a été réalisé au laboratoire Vision Artificielle et Automatique des Systèmes (LVAAS) du département Automatique (LVAAS), FG EI, UMMTO.

Il fait suite au travail d'un mémoire de master réalisé par monsieur IRATNI Salim soutenu et réalisé dans ce même laboratoire.

Nous avons apporté des modifications en dans ce qui concerne la commande des moteurs de la camera et la méthode de détection et de suivi d'une personne.

Nous tenons à remercier vivement monsieur DIAF Moussa, professeur pour nous avoir proposé ce sujet et pour nous avoir guidé et dirigé durant ce travail.

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

Nous ne manquerons pas de remercier Monsieur ALIBEY Mohamed, MCB au département Automatique, FG EI , UMMTO, pour nous avoir fait honneur en acceptant de présider le jury de ce mémoire.

Nos sincères remerciements vont aussi a Monsieur BOUCHEBBAT Rochedi, MCB au département Automatique, FG EI, UMMTO pour avoir accepté de faire partie du jury d'examen de notre mémoire.

Introduction générale 1

Chapitre I : Généralité sur l'asservissement visuel

1.1 Introduction	3
1.2 Historique de l'asservissement visuel	4
1.3 Définition et principe de l'asservissement visuel	5
1.3.1 Définition	5
1.3.2 Principe de l'asservissement visuel	5
1.4 Les fondamentaux de l'asservissement visuel	6
1.4.1 Schéma fonctionnel de l'asservissement visuel	7
1.5 classification	7
1.5.1 En fonction de la position de(s) caméra(s)	8
1.5.2 En fonction du type de commande	8
1.5.3 En fonction du type de mesure	10
1.5.4 En fonction de la rapidité	12
1.6 Applications de l'asservissement visuel	13
1.7 Description générale de l'application de ce projet	13
1.8 Conclusion	14

Chapitre II : La détection et le suivie d'un objet

2.1 Introduction	15
2.2 Les différentes méthodes de détection d'objet ou de mouvement	15
2.2.1 méthode de détection par réseau de neurone	16
2.2.2 Méthode de Viola et Jones	18
2.2.3 Les différentes méthodes de détection de mouvement	25
2.2.3.2 Détection sans modélisation de l'arrière-plan	25

2.2.3.2	Modélisation de l'arrière-plan	26
2.2.3.3	Modélisation semi-locale de l'arrière-plan	27
2.2.3.4	Modélisation global de l'arrière-plan	27
2.3	Les différentes méthode suivi de l'objet	27
2.3.1	Classification des algorithmes de suivi	27
2.3.2	Structure général des algorithmes de suivi	28
2.3.3	Phase d'ajustement	29
2.3.4	Phase d'estimation	30
2.3.5	Suivi de trajectoire d'objet en mouvement	30
2.3.6	Suivi de mouvement par filtre de Kalman	30
2.5	Conclusion	31

Chapitre III : Etude et réalisation d'un système de détection et de suivi d'une personne

3.1	Introduction	32
3.2	Détection et le suivi d'une personne	33
3.2.1	régulateur PID	40
3.3	Partie hardware du projet	44
3.3.1	Les servomoteurs	45
3.3.2	Composition des servomoteurs	45
3.3.3	Asservissement angulaire d'organe final	46
3.3.4	Modification d'un servomoteur asservi en position à un servomoteur asservi en vitesse	47
3.3.5	La carte ARDUINO UNO	52
3.3.6	Branchement des deux servomoteurs à ARDUINO	54
3.4	tests et résultats	54
3.5	Conclusion	58

Conclusion générale 60

Fig.1.1 Schéma de principe de l'asservissement visuel	6
Fig.1.2 Perception pour agir	6
Fig.1.3 Schéma bloc de l'asservissement	7
Fig.1.4 Configuration de la camera	8
Fig.1.5 Schéma bloc de l'asservissement direct	9
Fig.1.6 Schéma bloc de l'asservissement indirect	9
Fig.1.7 Schéma bloc de l'asservissement visuel 3.D	10
Fig.1.8 Schéma bloc de l'asservissement visuel 2.D	11
Fig.1.9 Schéma bloc de l'asservissement visuel 2D $\frac{1}{2}$	11
Fig.1.10 Schéma bloc de l'asservissement visuel d2D $\frac{1}{2}$	12
Fig.1.11 Schéma de principe du système a commandé	13
Fig.2.1 détection d'objet par réseaux de neurones	18
Fig.2.2 Exemple de caractéristique de pseudo-haar	19
Fig 2.3 : La valeur de l'image intégrale au point (x,y).	21
Fig 2.4 : Calcul de la somme du rectangle D avec l'image intégrale	22
Fig 2.5 : image intégrale.	22
Fig 2.6 : illustration de l'architecture de la cascade	24
Fig.2.7 Méthode de détection de mouvement	25
Fig.2.6 Exemple de champ de mouvement	26
Fig.3.1 détection de visage par la méthode de VIOLA & JONES	34
Fig.3.2 détection du haut de corps par la méthode de VIOLA & JONES	35
Fig.3.3 éliminations de faux positive	36
Fig.3.4 Détection de personne	37
Fig.3.5 détection de coin par valeur propre minimale	37
Fig.3.6 masque a la silhouette d'un humain	38
Fig.3.7 élimination des points caractéristiques de l'arrière-plan	38
Fig.3.8 Organigramme du programme	39
Fig .3.9 boucle d'asservissement visuel	40
Fig.3.10 Structure en série d'un PID	41
Fig.3.11 Structure en parallèle d'un PID	41
Fig.3.12 Structure en mixte d'un PID	41
Fig.3.13 fonction de saturation	43
Fig.3.14 Servomoteur	45
Fig.3.15 Les composition d'un servomoteur	46
Fig.3.16 rapport largeur d'impulsion/angle de sortie	47
Fig.3.17 servomoteur mg996	48
Fig.3.18 le rapport entre/sortir du servomoteur utilisé	49
Fig.3.19 réponse indicielle mesurée	50
Fig.3.20 réponse indicielle de fonction de transfert estimé	50
Fig.3.21 les deux réponses	51
Fig.3.22 La sortie V en fonction de l'entré a	51
Fig.3.23 Schéma bloc d'un asservissement avec correcteur	52
Fig.3.24 Description de la carte arduino	53

Fig.3.25 les différentes valeurs des gains d'un PID	48
---	----

2D : deux dimensions

3D : Trois dimensions

Introduction générale

L'apparition des systèmes automatiques et l'évolution des nouvelles technologies d'automatisation d'informations et de communication constituent une révolution pour l'être humain, d'où leurs généralisations leur diversités et indispensabilités dans la vie quotidienne, et afin de nous faciliter les tâches quotidiennes ceci-dit, l'homme a pu franchir des barrières dans différents domaines tel que la robotique, le traitement d'images... etc.

Avec la popularisation des images numériques et leur accessibilité, l'analyse du mouvement dans les vidéos s'est avérée être un outil primordial pour de multiples applications tel que la vidéo surveillance, l'imagerie médicale, la robotique... etc. Et cela grâce à l'asservissement visuel.

L'émergence de ce dernier a permis en intégrant des systèmes avec capteurs de vision aux robots industriels d'appréhender et de réagir à d'éventuels changements de son environnement, ceci a permis d'améliorer de manière significative leur performances.

Durant ces trente dernières années, La détection de mouvement et le suivi d'objet en mouvement ont fait un objet de recherche, ces derniers restent toujours des domaines où de nouvelles approches sont à prospector. Mais le problème qui se pose à l'heure actuelle est qu'il n'existe pas un algorithme abouti qui s'adapte à n'importe quelle situation.

Le travail réalisé dans ce mémoire s'inscrit dans le domaine du suivi par détection d'une personne et d'asservissement visuel. L'objectif est de réaliser un asservissement visuel 2D en utilisant la configuration d'une caméra embarquée et la technique Look and Move, le travail se limite sur le plan théorique et pratique de la commande d'un système articulé a deux axes qui pivote la caméra embarquée sur l'organe terminal de ce système. Le fonctionnement de ce dernier consiste au suivi par détection d'une personne en temps réel, de telle manière à toujours suivre la personne détectée et de la placer au milieu du champ de vision.

Introduction générale

Pour réaliser la commande en position de cette caméra, trois parties ont été effectuées, dont la première est la liaison entre la carte ARDUINO et le logiciel Matlab, puis nous avons programmé le programme du tracking (la détection et suivi d'une personne) en temps réel grâce au logiciel Matlab, et enfin avec le logiciel ARDUINO nous avons programmé un programme pour commander directement deux servomoteurs débridés et récupérer les données nécessaires pour donner à ce dernier la position nécessaire aux moteurs pour qu'ils puissent tourner avec exactitude et minimiser au mieux l'erreur entre le centre de la personne détectée et le centre du champ de vision grâce à un PID.

Le mémoire est organisé comme suit :

Le premier chapitre présente des généralités sur l'asservissement visuel et les lois de commandes liées à ce dernier.

Le deuxième chapitre est dédié à la détection de mouvement.

Le troisième chapitre est consacré à expliquer les différentes étapes de la réalisation du système de suivi, en divisant ce dernier en trois étapes : partie software, partie hardware, des tests et résultats.

Chapitre I

Généralités sur l'asservissement visuel

I.1 Introduction :

De nos jours, l'asservissement a pris une place indéniable dans l'industrie et les nouvelles technologies en général, car il permet de minimiser les erreurs qui s'ajoutent à la consigne.

Dans notre cas l'objet de notre étude est l'un des types d'asservissement : l'asservissement visuel, qui utilise des capteurs de vision (caméra, capteur ultra son, télémètre laser et radar) pour évaluer l'erreur et y remédier.

Ce domaine a ouvert les portes à de nouvelles branches permettant de rendre le robot, la machine etc. Plus performant en termes de précision, de rapidité et de stabilité.

En effet, en l'absence d'asservissement visuel, le robot peut effectuer que des tâches programmées et n'anticipe aucune perturbation externe, donc l'environnement doit être invariant, avec son intégration, des tâches non connues au préalable peuvent être acceptées (envisagées). Par exemple les technologies embarquées dans les véhicules récents leur permettent d'opérer d'une manière autonome.

Si on veut stationner le véhicule, l'asservissement visuel rentre en jeu pour éviter toute collision en apportant des corrections aux consignes.

Parmi les capteurs embarqués la caméra représente un des principaux capteurs car elle permet de connaître l'environnement, d'évaluer les distances et cela grâce à des traitements adéquats

1.2 Historique de l'asservissement visuel :

Au tout début l'asservissement visuel à été consacré à des robots manipulateurs. Ensuite, il a été appliqué à plusieurs domaines de la robotique, par exemple les robots d'aide aux personnes handicapées, les robots médicaux qui sont utilisés comme aide aux chirurgiens (télé-chirurgie) ou la réalisation automatique d'échographie.

Au dernier quart du 20eme siècle l'asservissement visuel a connu un développement considérable qui s'est poursuivi jusqu'à nos jours. On peut citer quelques principales dates qui ont marquées l'histoire de l'asservissement visuel :

- En 1973, Shirai et Inoue : Tâche de saisie d'un objet cubique, la caméra est fixe et observe la scène, période de l'asservissement : 10s
- En 1979, Hill et Parks : Tâche de suivi, de saisie et de déplacement par rapport à une scène. Projection d'un marqueur lumineux pour reconstruire la profondeur et l'orientation. Etude des effets dynamiques
- 1979, Prajoux : Utilisation d'un observateur pour anticiper le mouvement de la cible. Tâche de suivi de cible à 2 degrés de liberté. Temps d'établissement de 1 s
- 1984, Weiss : Premier asservissement visuel 2D
- 1990, Venkatesan et Archibald : Projection de marqueurs lumineux à l'aide de 2 scanners LASER. Commande par vision suivant 5 degrés de liberté. Bande passante de la boucle fermée : 4.5 Hz
- 1991, Papanikolopoulos : Estimation de la mesure par flot optique. Tâche de suivi de cible
- 1992, Espiau, Chaumette et Rives: Formalisation de la commande 2D par l'utilisation de la notion de fonction de tâche. Preuve de stabilité. Validation expérimentale : tâche de suivi de cible

- 1996, Wilson : Asservissement 3D, filtrage de Kalman de la mesure. Tâche de suivi de cible à 5 degrés de liberté. Fréquence de l'asservissement : 60 Hz
- 1996, Corke : Prise en compte du modèle dynamique du manipulateur pour la synthèse du correcteur. Tâche de suivi de cible à 2 degrés de liberté. Fréquence de l'asservissement : 60 Hz
- 1998, EAVR : Modèle dynamique d'un asservissement visuel à 6 DDL. Asservissement visuel à 120 Hz d'un robot à 6 DDL. Tâche de suivi de cible
- 1999, Malis et Chaumette : Asservissement visuel 2D 1/2. Tâche de suivi de cible
- 1999, Nakabo : Première utilisation d'une vision chip. Tâche de suivi de cible avec une caméra orientable (2 degrés de liberté). Asservissement visuel à 1000 Hz

1.3 Définition et principe de l'asservissement visuel

I.3.1 Définition

L'asservissement visuel est un type d'asservissement qui se base sur une ou plusieurs images issues d'un ou plusieurs capteurs visuels (caméras) dans le but d'asservir des grandeurs avec lesquels un robot évolue dans un environnement connu ou inconnu.

1.3.2 Principe de l'asservissement visuel

Après sélection d'un ensemble de mesures efficaces obtenues par les caméras, on peut contrôler les degrés de liberté du robot, et ainsi mettre au point une loi de commande pour se rapprocher au mieux des consignes (S^*) en annulant le vecteur d'erreur (erreur entre consigne et mesure) à partir de la différence entre S et S^* qui est la consigne de vitesse à envoyer à l'engin. L'engin réagira donc à ces commandes selon sa dynamique, et l'image donnée par la caméra sera donc modifiée en conséquence [3].

Cette erreur est notée par $e(t)$, signifiant l'écart entre la valeur mesurée et désirée.

Valeur désirée $\Rightarrow (S^*)$.

Valeur mesurée =>(S(r(t)).

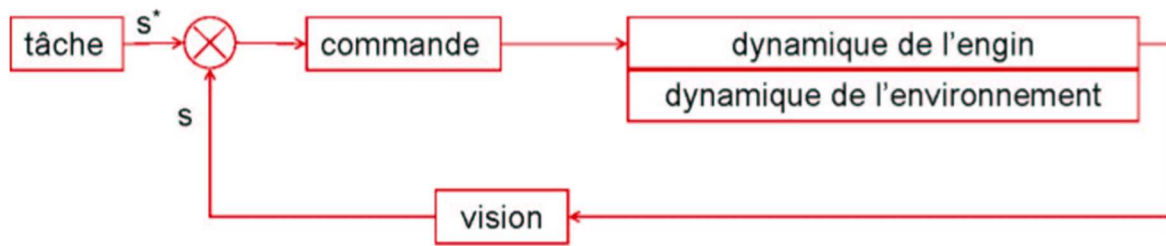


FIG. 1.1 – Schéma de principe de l'asservissement visuel

I.4 Les fondamentaux de l'asservissement visuel

Dans l'exemple du robot évoluant dans l'environnement, la vision de ce dernier est fondamentale car elle renvoie l'état actualisé en prenant en compte les changements éventuels en s'adaptant en conséquence [1].

Donc le mouvement se résume au respect de la consigne en supprimant les erreurs qui s'ajoutent à l'asservissement visuel.

L'asservissement visuel consiste à intégrer le sens de vision dans une boucle fermée comme nous le montre la figure suivante

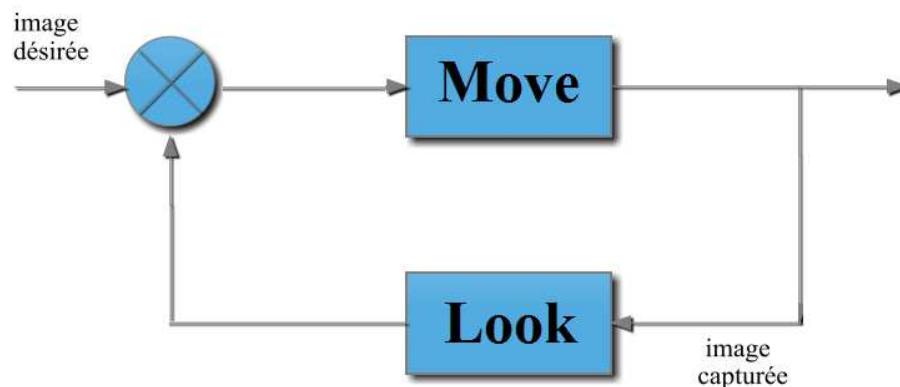


Fig.1.2 Perception pour agir

La vision est en réalité un ensemble de mesures sur une cible, elle nous renvoie par exemple les coordonnées d'un point donné dans l'image.

Le robot utilise ces informations pour effectuer une tâche, ce qui nous amène à la conclusion qu'une vision pertinente de l'environnement est indispensable de ce fait cela revient à modéliser l'information visuelle adéquate.

Or la quantité d'informations captées est considérable ce qui nous amène à la problématique de la gestion intelligente du robot pour ce flot d'informations et à les utiliser en conséquence pour l'amélioration du comportement.

1.4.1 Schéma fonctionnel de l'asservissement visuel

La boucle d'asservissement visuel peut être représentée par le schéma bloc de la figure (1.3).

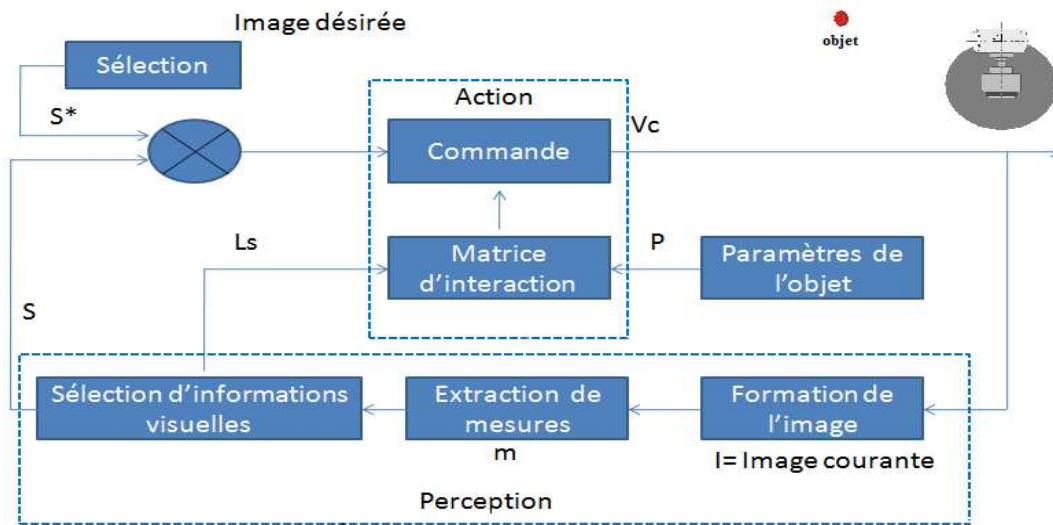


Fig.1.3 Schéma bloc de l'asservissement visuel

Comme il est indiqué sur la figure précédente. La boucle d'asservissement visuel est composée essentiellement de deux parties :

- Partie **Look** (Perception)
- Partie **Move** (Action)

1.5 Classification

On peut distinguer plusieurs cas de classifications :

- En fonction de la position de(s) caméra(s)
- En fonction du type de commande
- En fonction du type de mesure
- En fonction de la rapidité

1.5.1 En fonction de la position de(s) caméra(s)

La configuration d'un robot caméra peut être configurée en trois cas, selon la commande du robot et la manière d'acquérir les informations visuelles concernant l'environnement [4].

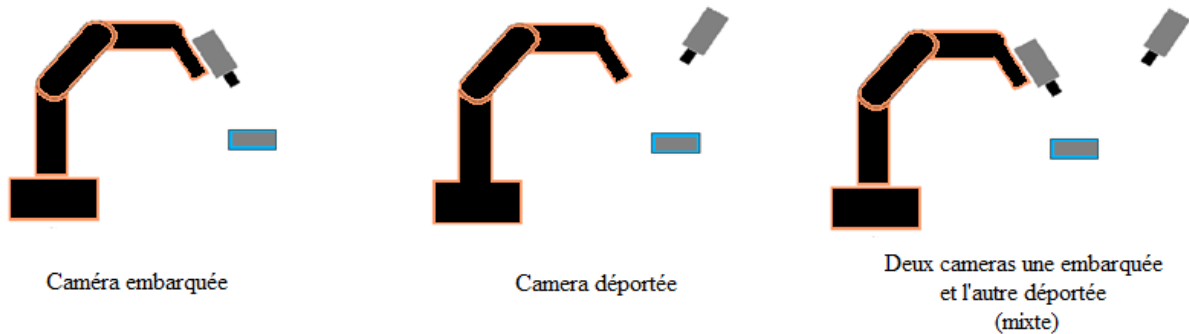


Fig.1.4 Configuration de la caméra

- **La configuration caméra embarquée** : la caméra est montée sur l'organe terminal.
- **La configuration caméra déportée** : la caméra est externe, sans aucune liaison mécanique avec le robot.
- **La configuration mixte** : où l'on utilise une caméra portée par l'organe terminal et une autre portée par un autre robot.

1.5.2 En fonction du type de commande

L'asservissement visuel en fonction du type de commandes se décompose en 2 :

1.5.2.1 Asservissement visuel direct :

Les positions de l'organe terminal du robot sont asservies par le variateur qui est commandée en vitesse ou en couple. Le variateur est commandé par les consignes du correcteur.

Donc dans ce cas l'asservissement de l'organe terminal est obtenu grâce à l'information retournée par le capteur visuel.

Ce cas est utilisé lorsque on a besoin d'un asservissement visuel en temps réel [1].

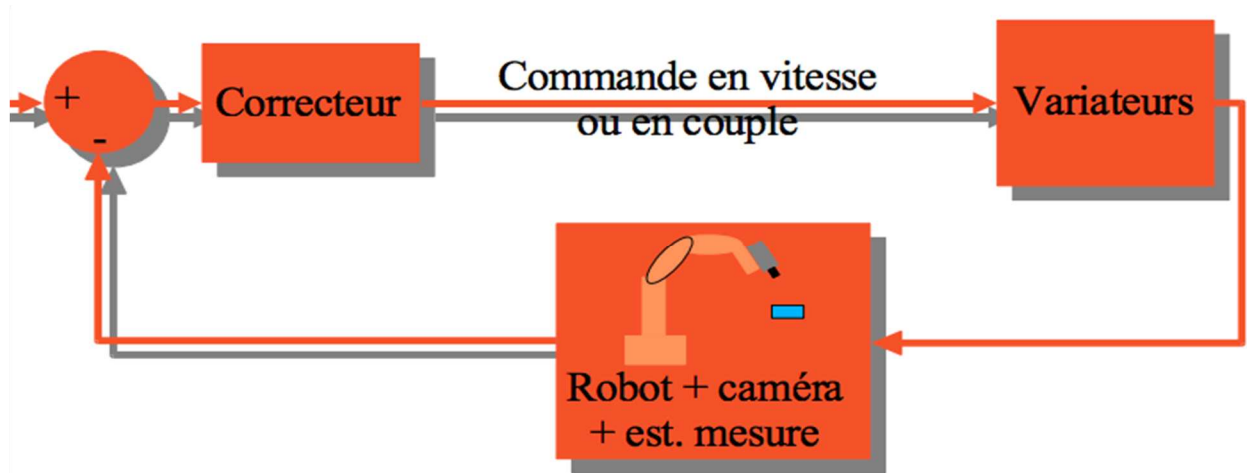


Fig.1.5 Schéma de l'asservissement visuel direct.

I.5.2.2 asservissement visuel indirect :

Contrairement à l'asservissement direct, l'organe terminal est asservi directement par un contrôleur. Les consignes de vitesse fournies par le contrôleur sont transformées en consigne de position articulaire par le générateur de trajectoire [4].

En général cette architecture est souvent utilisée lorsqu'on veut que l'asservissement soit précis (période d'échantillonnage élevée).

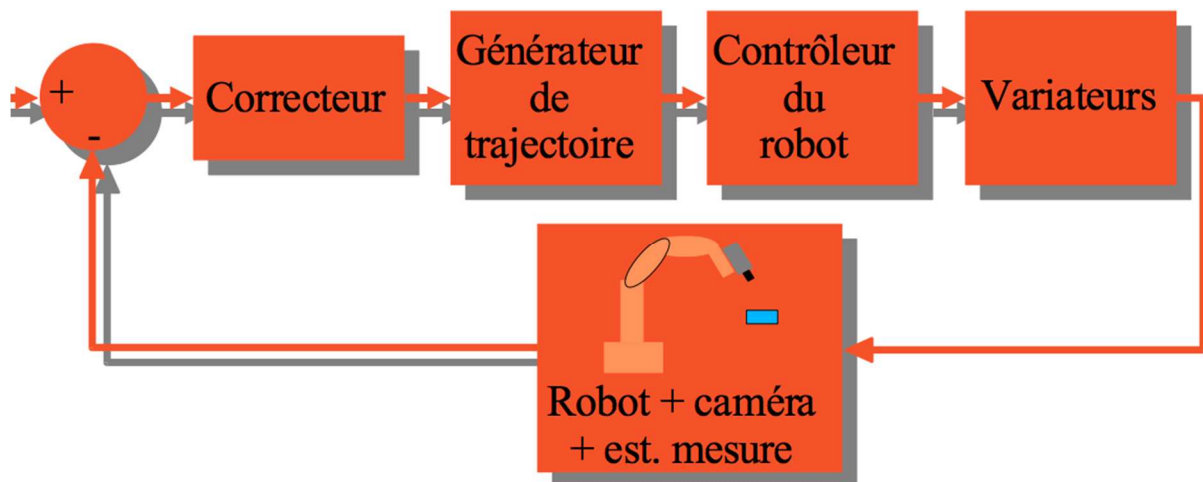


Fig.1.6 Schéma de l'asservissement visuel indirect.

1.5.3 En fonction de type de mesure :

Dans ce type on distingue quatre sous-catégories

1.5.3.1 Asservissements visuels 3D (position based)

Implique la reconstruction 3D. Ici la situation mono-caméra est la plus courante. Il est primordial de connaître préalablement la forme géométrique de la scène pour la reconstruction [2], et connaître les coordonnées de l'objet d'intérêt, ainsi la consigne et la mesure sont les coordonnées de position dans l'espace.

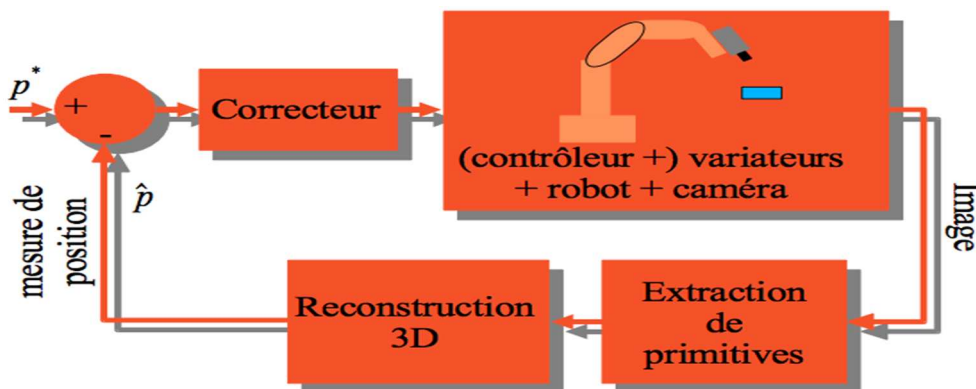


Fig.1.7 Schéma de l'asservissement visuel 3D.

1.5.3.2 asservissements visuels 2D (image based)

Dans ce deuxième type, il n'est pas nécessaire de connaître la forme géométrique de l'objet d'intérêt, une connaissance basique de la géométrie et la mesure sont des coordonnées de type primitives visuelles (points, droites, moments).

La consigne est alors exprimée comme la différence entre un motif courant et un motif désiré dans l'espace [1].

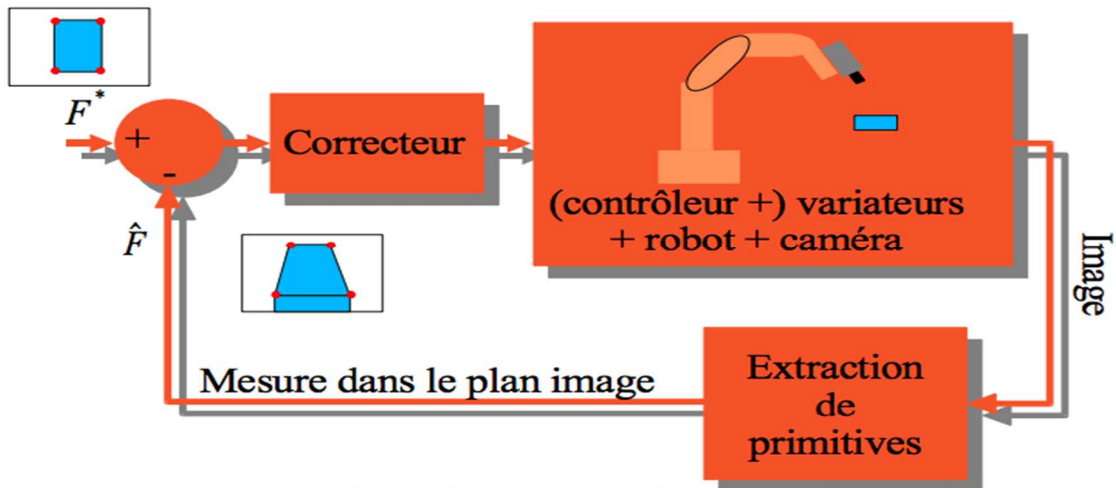


Fig.1.8 Schéma de l'asservissement visuel 2D.

1.5.3.3 Asservissement visuel 2D ½

Ce type est un hybride entre asservissement visuel 3D et asservissement visuel 2D. On a alors deux boucles d'asservissement découplées.

L'orientation est asservie de manière 3D et la vitesse de manière 2D. [4]

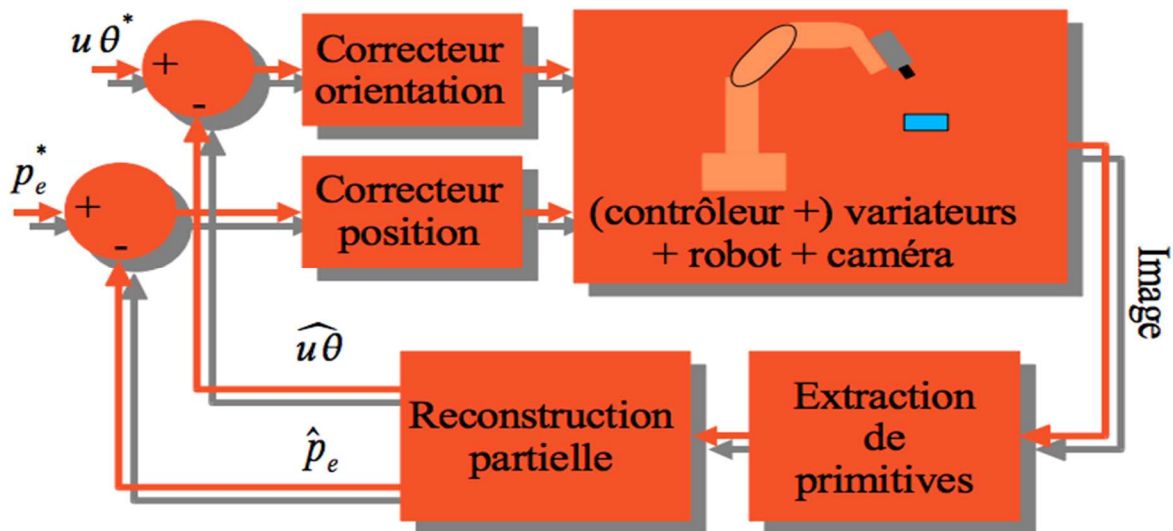


Fig.1.9 Schéma de l'asservissement visuel 2D1/2.

1.5.3.4 asservissement visuel dérivée $d2D/dt$

Après le dérivé des primitives (points, droites, moments) on obtient un champ de vitesse dans le plan image qui est utilisé pour asservir l'objet d'intérêt.

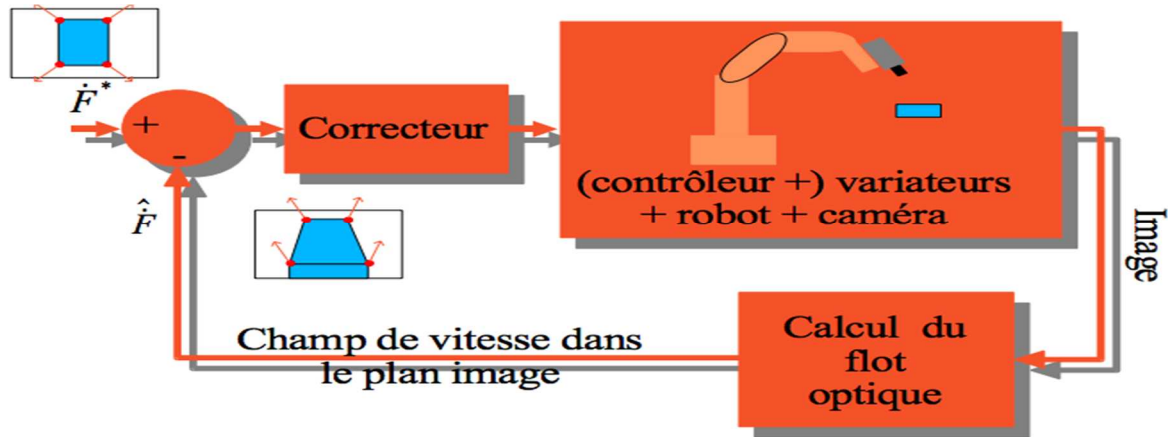


Fig.1.10 Schéma de l'asservissement visuel $d2D/dt$

1.5.4. En fonction de la rapidité

On distingue deux types d'asservissements :

1.5.4.1 Asservissement visuel rapide (dynamique)

L'asservissement visuel dynamique est utilisé pour améliorer les performances et cela à partir d'une limite prédéfinie à partir de 50HZ.

Il se base sur le principe de contenir le modèle dynamique du manipulateur dans la synthèse du correcteur, on peut utiliser la commande de type direct.[1]

1.5.4.2 Asservissement visuel lent (cinématique)

Le principe est d'obtenir à partir de la fonctions de transfert du robot entre la consigne de vitesse et la mesure de vitesse un gain.

Du fait que c'est une approximation elle est valable pour des déplacements relativement lents, et cela avec un type de commande indirecte.

I.6 Applications de l'asservissement visuel

En particulier l'asservissement visuel est utilisé pour réaliser des opérations de positionnement afin de déplacer un robot à partir d'une position initiale à une position finale (consigne mesurée par l'utilisation d'une caméra). Il peut très bien être utilisé dans différents domaines comme la réalisation des travaux de soudure par un robot, la position peut être réalisée avec la précision voulue en agissant sur les imprévues comme le décalage, 1mm près par rapport à la référence.

Ou bien même le suivi d'un objet en mouvement en utilisant les images fournies par la caméra pour ramener le robot à un état d'équilibre tout en évitant les obstacles rencontrés en chemin.

I.7 Description générale de l'application de ce projet

Dans le cadre de notre mémoire de fin d'études, on propose d'améliorer une application qui s'inscrit dans trois domaines (le domaine de la commande visuelle, de la robotique, et de la vision artificielle).

En fait, il consiste à étudier, sur le plan théorique et pratique la commande d'un système mécanique réalisé par Salim IRATNI pour son projet de fin d'études en 2014, composé d'une caméra numérique, un système articulé à deux axes (deux degrés de liberté) l'un vertical et l'autre horizontal, et un support de caméras en tant qu'organe terminal.

Une carte de développement (c'est une carte d'acquisition et de commande) qui est ARDUINO UNO est utilisée pour le pilotage de cet ensemble par un ordinateur.

Notre application consiste à détecter une personne dans le champ de vision de la caméra sélectionnée et de la suivre avec un asservissement visuel et faire en sorte que cette personne détectée reste au milieu du champ de vision du capteur visuel.

Le schéma de principe du système à commander est représenté comme suit :

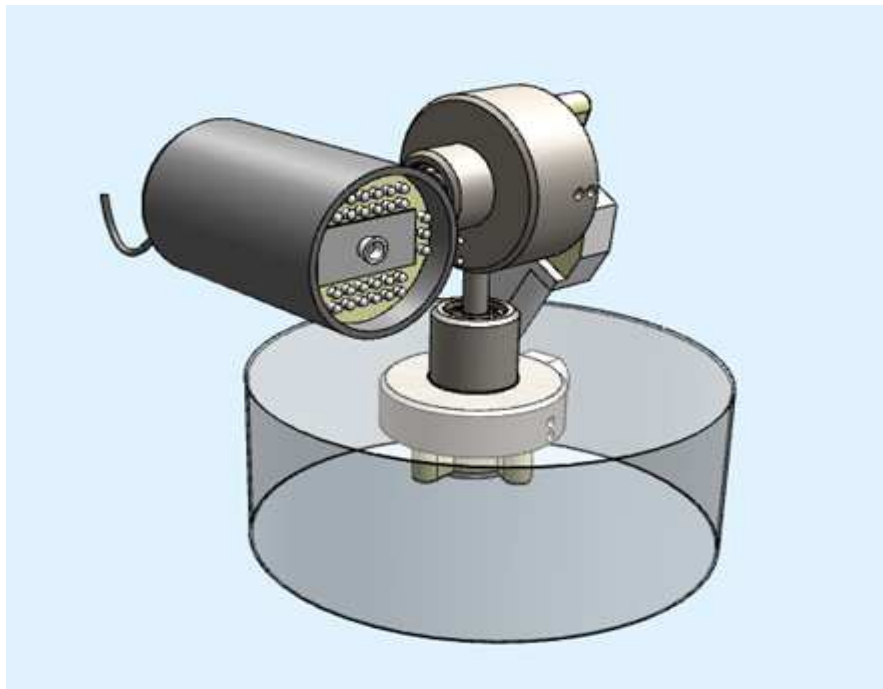


Fig1.11 Schéma de principe du système à commander

I.9 Conclusion

L'asservissement visuel est un asservissement qui inspire les chercheurs de par ses plusieurs applications. Etant récent il utilise des concepts poussés comme la vision 3D, en termes de technologie de navigation, de poursuite d'objet (ou autre), d'évitement d'obstacles.

Dans le sujet de notre étude qui consiste à détecter une personne dans le champ de vision de notre camera sélectionner, on a décidé de développer une application qui va gérer le fonctionnement d'une poursuite de cette personne en mouvement.

Chapitre II

La détection et le suivi d'un objet

2.1 Introduction

Pour permettre un suivi de trajectoire d'un objet une étape préliminaire est nécessaire qui consiste à détecter l'objet à suivre en utilisant les pixels des images prises par plusieurs cameras, ces derniers sont utilisés pour déterminer les objets mobiles à suivre.

Depuis que les ordinateurs ont fait leur preuve en termes de traitements en temps réel de données vidéo, la recherche dans l'analyse de séquences vidéo est devenue très active, surtout dans les années 90, de nombreuses études ont été réalisées dans ce domaine.

2.2 Les différentes méthodes de détection d'objet et de mouvement.

Il existe plusieurs méthodes pour la détection d'objets et de mouvements, dans ce chapitre on présentera quelques-unes de ses méthodes en général en mettant en avant plan la méthode utilisée dans notre travail qui est la méthode de Viola & Jones.

2.2.1 Méthode de détection par réseau de neurone

Les réseaux de neurones impulsionnels sont considérés comme une méthode de nouvelle génération.

Les réseaux de neurones artificiels sont utiles de par ses propriétés très intéressantes comme l'intégration du temps au niveau du neurone qui leur permet de traiter des problèmes dynamiques comme la détection d'objet où le temps est une composante essentielle.

Plusieurs approches peuvent être envisagées pour la détection d'objet en utilisant les neurones, on citera quelques-unes de ces approches dans ce qui va suivre.

2.2.1.1 Approche de détection de contour

Un contour se matérialise par une rupture d'intensité dans l'image suivant une direction donnée. [5]

Deux étapes sont nécessaires pour la détection d'un contour :

La première permet de localiser les contours à partir d'un calcul de Gradient ou de Laplacien dans des directions privilégiées.

La seconde étape va permettre d'isoler les contours du reste de l'image à partir d'un seuillage judicieux.

L'application de détecteurs de contours sous la forme de filtres dérivateurs permet d'obtenir les contours des objets présents dans la scène.

2.2.1.1.1 Approche par convolution

Avec cette approche on a deux façons de détecter les contours.

2.2.1.1.1.1 Détection des contours par dérivée première

Les filtres utilisés ici sont les filtres de dérivée première (filtres étroits) et l'on cherche alors le maximum de leur réponse, le prototype de ce filtre est le filtre de gradient mais la dérivation accentue le bruit (pixels parasites de répartition aléatoire) sachant que le gradient est un vecteur caractérisé par son amplitude et sa direction, en un pixel d'une image numérique. [5]

La dérivation au premier ordre du gradient est définie comme suit ;

$$\nabla I(x, y) = (\partial I / \partial x) \hat{x} + (\partial I / \partial y) \hat{y} \quad (\text{II.1})$$

2.2.1.1.1.2 Détection des contours par dérivée seconde

Ces filtres larges recherchent les zéros de la dérivée seconde plus précisément, du Laplacien qui est une dérivation au deuxième ordre. [5]

On définit comme étant ;

$$\Delta I(x, y) = (\partial^2 I(x, y) / \partial x^2) + (\partial^2 I(x, y) / \partial y^2) \quad (\text{II.2})$$

La description des opérateurs du type laplacien.

2.2.1.1.2 Approche par filtrage optimal

Les qualités attendues d'un filtre de détection de contours est un filtre de réponse impulsionnelle qui permet une bonne détection des contours c'est-à-dire une réponse forte même à de faibles contours, une bonne localisation de ceux-ci et une faible multiplicité des maxima dus au bruit. [5]

Cette approche nous a permis d'avoir de bon détecteur grâce à une meilleure compréhension des conditions de détections de contours.

2.2.1.2 Exemple d'application

Un détecteur d'objets performant et rapide en utilisant des technologies d'apprentissage profonde (Deep Learning) pour des applications allant de la vidéosurveillance à la conduite autonome et l'analyse de manière plus précise les objets.

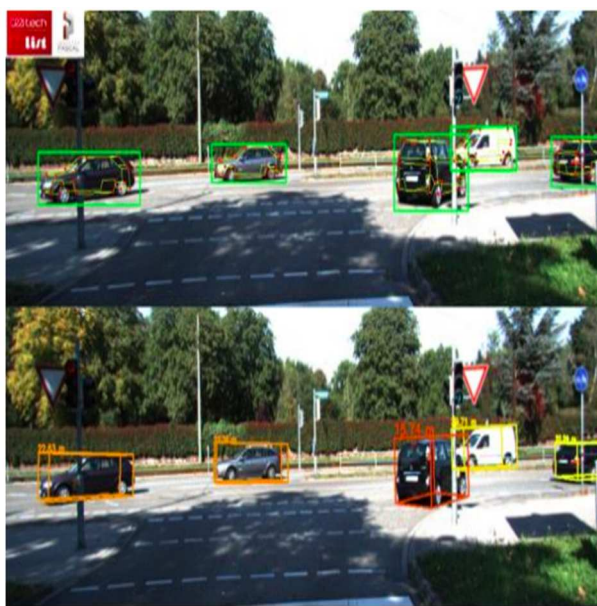


Fig 2.1 détection d'objet par réseaux de neurones

2.2.2 Méthode de Viola et Jones

En 2001, Paul Viola et Michael Jones ont proposé une méthode qui consiste à la détection d'un objet pour une image numérique. Elle fait partie des premières méthodes capables de détecter efficacement et en temps réel des objets dans une image.

Inventée à l'origine pour la détection d'un ou plusieurs visages, elle a pu également servir à détecter n'importe quel objet dans une image numérique comme des avions et des voitures...etc.

Cette dernière est l'une des méthodes les plus utilisées et connues, en particulier pour la détection de visages et de personnes. [6]

Cette méthode combine quatre principes clés qui sont : Les caractéristiques rectangulaires appelées pseudo-haar, L'approche d'image intégrale pour la détection de ces caractéristiques,

La méthode d'apprentissage adaptative AdaBoost cherchant à minimiser l'erreur de classification, et l'algorithme en cascade de classifieur, permettant de détecter la présence de visage sur une image.

2.2.2.1. Fonctionnement

La méthode de Viola et Jones n'a pas besoin d'étudier chaque pixel d'une image pour l'analyser et ce grâce à l'image intégral qui permet de définir plusieurs zones rectangulaires au sein d'une image. L'intérêt de cette notion est le fait d'offrir la possibilité d'accéder aux valeurs des autres zones à gauche et au-dessus de la zone à laquelle nous sommes.

Ces zones permettent de créer des caractéristique pseudo-Haar, qui sont en fait des masques permettant de déterminer plusieurs patterns [6].



Figure 2.2 Exemple de caractéristiques pseudo-Haar.

On arrive à reconnaître des visages en détectant des motifs est cela est rendu possible par la variation de l'intensité de la lumière entre les yeux et les pommettes (caractéristique n°3), et l'intensité de la lumière entre les yeux et le nez (caractéristique n°2).

La méthode de Viola et Jones repose sur l'utilisation des images intégrales et les caractéristiques pseudo-Haar, plus deux mécanismes : le classifieur et le boosting.

Le classifieur permet de déterminer l'ensemble des zones rentrant sous la coupe d'une caractéristique pseudo-Haar, en déterminant les seuils pouvant déterminer les exemples positifs des négatifs. Ceci demande donc une phase d'apprentissage, qui permet de définir les seuils utilisés. Au final, un classifieur est une association entre une caractéristique pseudo-Haar et un seuil. C'est ce qu'on appelle un classifieur faible [6].

Les caractéristiques sont calculées en soustrayant la somme des pixels noirs

à la somme des pixels blancs.

2.2.2.2 Principe

Le principe de cette méthode consiste à parcourir l'ensemble d'une image en calculant certaines caractéristiques dans des zones rectangulaires qui se chevauchent à l'aide d'une fenêtre de détection de taille initiale de 24px par 24px (dans l'algorithme original) [8], et ainsi détecter un visage s'il est présent (et ceci en décalant la fenêtre d'un pixel). Une fois l'image parcourue entièrement, la taille de la fenêtre est augmentée, qui se fait par un facteur multiplicatif de 1.25 et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image.

Cette méthode utilise des caractéristiques très simples, mais très nombreuses.

Il existe d'autres méthodes, mais celle de Viola et Jones est la plus performante à l'heure actuelle. Ce qui la différencie des autres est notamment [7] :

- L'utilisation **d'images intégrales** qui permet de calculer plus rapidement les caractéristiques.
- La **sélection par boosting** des caractéristiques.
- La combinaison **en cascade de classifieurs** boostés, apportant un gain net de temps d'exécution.

a) Image intégrale :

L'image intégrale est une matrice de la même dimension que l'image a traitée, ou l'amplitude de chaque élément de coordonnée (x,y) dans l'image intégrale représente la somme des amplitudes de tous les pixels au-dessus et à gauche de l'image y compris le pixel de coordonnée (x,y) d'où le nom image intégrale. Cette image sera donc une représentation intermédiaire dans le traitement pour diminuer le temps de calcul et cela pour calculer d'une manière efficace et rapide les caractéristique pseudo-Haar (figures 2.5).

Soit I_n l'image intégrale de l'image initiale i et $i(x, y)$ la valeur de l'image intégrale au point (x, y) :

On peut également définir l'image intégrale ii par :

$$In(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (II.3)$$

Où $In(x, y)$ est l'image intégrale et $i(x, y)$ l'image originale. Comme nous utilisons cette nouvelle représentation pour diminuer le temps de calcul, on peut expliquer ses avantages. D'abord, elle peut être calculée par un moyen efficace en utilisant la paire des récurrences suivantes :

$$S(x, y) = S(x, y - 1) + i(x, y) \quad (II.4)$$

$$In(x, y) = In(x - 1, y) + s(x, y) \quad (II.5)$$

D'où $s(x, y)$, s est la somme cumulative [9]

$$\forall x, s(x, -1) = 0, \text{ et } \forall y, In(-1, y) = 0 \quad (II.6)$$

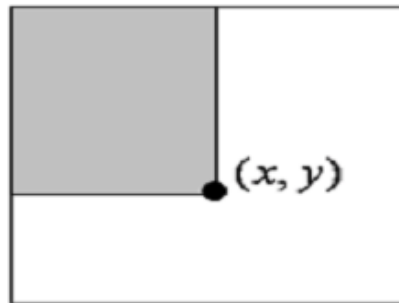


Figure 2.3 : La valeur de l'image intégrale au point (x, y) .

L'utilisation de l'image intégrale réduit considérablement le nombre des opérations de somme. La somme de tous les éléments d'un rectangle peut être calculée en quatre opérations (voir la figure 2.6). Donc la différence entre les deux sommes rectangulaires peut être calculée en huit opérations, la caractéristique à deux rectangles définis ci-dessus, implique des sommes rectangulaires adjacentes qui peuvent être calculés en six opérations, huit dans le cas d'une caractéristique à trois rectangles, et neuf pour la caractéristique à quatre rectangles.

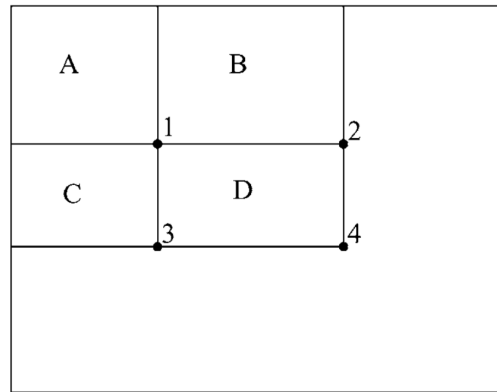


Figure 2.4 : Calcul de la somme du rectangle D avec l'image intégrale

La somme des pixels dans le rectangle D peut être calculée avec quatre opérations. La valeur de l'image intégrale au point 1 est la somme des pixels dans le rectangle A. La valeur à la position 2 est A + B, à la position 3 est A + C, et à la position 4 est A + B + C + D. La somme dans D peut être calculée comme $4 + 1 - (2 + 3)$. [10]

Parce que la somme d'intensités dans une région rectangulaire $((x1; y1) ; (x2; y2))$ est:

$$r = In(x1, y1) - In(x1, y2) - In(x2, y1) + In(x2, y2) \quad (II.7)$$

Exemple :

1	1	1
1	1	1
1	1	1

Image initiale

1	2	3
2	4	6
3	6	9

image intégrale

Figure 2.5 : image intégrale.

Par conséquent, la différence entre deux rectangles adjacents est obtenue à travers six références à l'image intégrale $In(u, v)$.

b) Sélection de caractéristiques par boosting :

Le boosting est un principe qui consiste à construire un classifieur « fort » à partir d'une combinaison pondérée de classifieur « faibles », c'est-à-dire donnant en moyenne une réponse meilleure qu'un tirage aléatoire.

La valeur seuil de la caractéristique doit être trouvée pour l'apprentissage du classifieur faible qui va permettre de mieux séparer les exemples positifs des négatifs. Dans ce cas, le classifieur se réduit alors à un couple (caractéristique, seuil). [12]

La méthode d'AdaBoost (Adaptive Boosting) est un algorithme de boosting qui permet de combiner plusieurs hypothèses pour créer une autre hypothèse plus performante (crée un ensemble dont la performance de chaque élément où ce dernier est *boostée*).

L'ensemble composé des hypothèses est défini comme suit :

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (\text{II.8})$$

Avec α_t le poids qui est attribué à l'hypothèse h de l'ensemble. Les poids α_t et les hypothèses h_t doivent être entraînés pendant la procédure du Boosting.

Les exemples d'apprentissage pondérés permettent une sélection itérative des coefficients α_t et des hypothèse h_t lors du Boosting. À chaque itération, les poids des exemples d'apprentissage sont recalculés de manière à attribuer une grande pondération aux exemples d'apprentissage mal classifiés et une faible pondération aux autres.

Adaboost :

Pour la détection de visage, l'algorithme AdaBoost a été proposé avec une architecture de cascade sur laquelle on peut appliquer la détection de visage. Pour les caractéristiques de l'algorithme, un classifieur faible est établi par une caractéristique rectangulaire, qui est l'équivalent d'une caractéristique faible,

Cet algorithme a été proposé par Viola et Jones, son résultat est le plus abouti connu à ce jour sur la détection de visage, c'est l'algorithme AdaBoost. Il est

rapide quand il s'agit d'appliquer un nombre important de caractéristique de rectangle a une petite région d'une fenêtre candidate, le classifieur fort est formé par quelques-unes qui sont choisies et combinées.

c) Cascade :

Principe

Une cascade de classifieurs est un arbre de décision dégénéré où chaque étape est entraînée pour détecter un maximum d'objets intéressants tout en excluant une certaine fraction des objets non-intéressants.

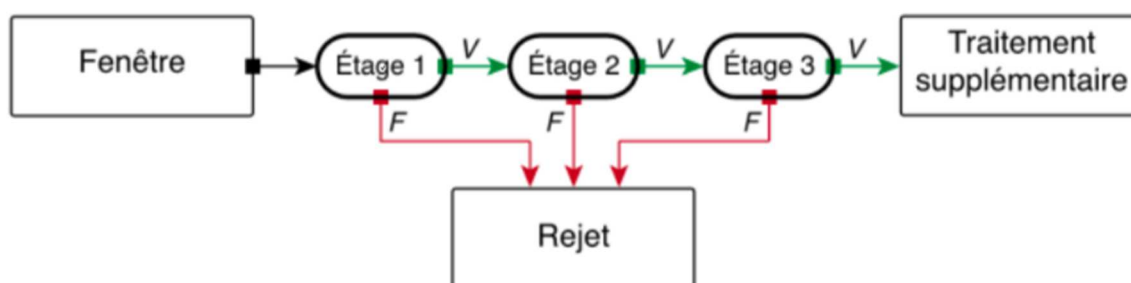


fig 2.6 : illustration de l'architecture de la cascade

Les fenêtres sont traitées séquentiellement par les classifieurs, qui prennent :

- une décision d'acceptation ; la fenêtre contient l'objet et l'exemple est alors passé au classifieur suivant.

- une décision de rejet ; la fenêtre ne contient pas l'objet et dans ce cas l'exemple est définitivement écarté. [12]

2.2.3 Les différentes méthodes de détection de mouvement

Pour un algorithme de suivi de mouvement, la détection de ce dernier est une étape initiale. Cette étape consiste à détecter les zones fixes et mobiles dans une scène. Le mouvement d'objet cause des différences temporelles entre images, donc sa détection s'appuie sur l'étude de la variation temporelle. [1]

On distingue plusieurs méthodes de détection de mouvement qui sont représentées dans le schéma de la figure (2.7) :

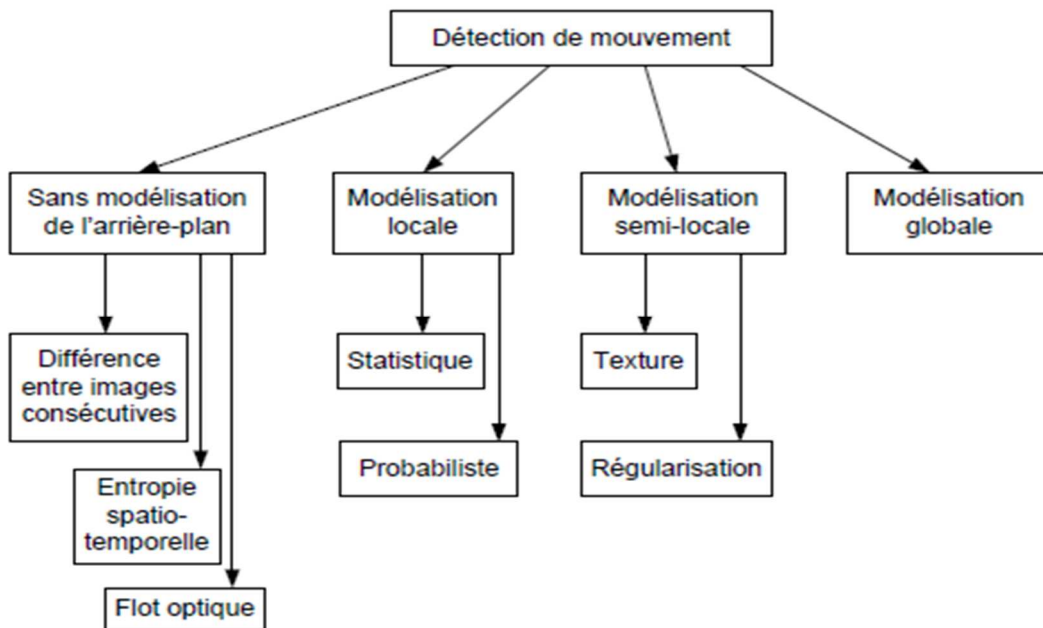


fig.2.7 Méthode de détection de mouvement

2.2.3.1 Détection sans modélisation de l'arrière-plan

Ces méthodes consistent à détecter le mouvement par le calcul d'une quantité mathématique en tout point d'une image en fonction de l'intensité ou de la couleur de l'ensemble des pixels qui est censé refléter l'ensemble du mouvement visible dans la scène sans prendre en considération les informations de l'arrière-plan. On peut citer trois approches, la dérivée temporelle, le flot optique, la soustraction de l'arrière-plan.

La dérivée temporelle

La méthode de la dérivée temporelle permet de mesurer les changements d'apparence de pixel entre deux trames consécutives. Cette approche est attribuée pour la première fois à Jain_et_Nagel. L'estimation de la dérivée temporelle à l'instant t est donnée par :

$$\forall (x, y) \in E \quad \forall t > 0 \quad I_t(x, y, t) \approx |I(x, y, t) - I(x, y, t - \Delta t)| \quad (\text{II.9})$$

Cette méthode s'avère peu robuste face à quelques phénomènes tels que les arrêts brusques (brefs) d'un objet en mouvement, les mouvements lents.

Le flot optique

La projection de mouvement réel sur le plan image définit un champ de vecteur à deux dimensions qui est le flot optique. En temps de calcul cette méthode est très coûteuse donc il est nécessaire d'adopter des algorithmes plus rapides. [2]

On peut citer celui de Lucas Kanade qui est fondé sur l'hypothèse de conservation de l'intensité lumineuse de la scène.

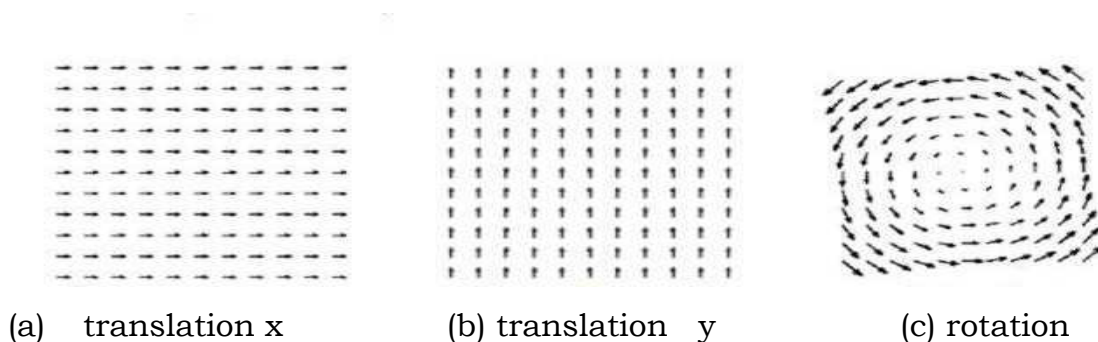


Fig.2.8 Exemple de champ de mouvement

La soustraction de l'arrière-plan.

La soustraction pixel par pixel de l'image courante à l'image du fond induit à détecter la région de mouvement. La méthode de soustraction de l'arrière-plan est très avantageuse, quelle que soit la façon utilisée pour la créer car elle détecte l'objet complètement. Cette dernière est utilisée dans plusieurs applications.

2.2.3.2 Modélisation local de l'arrière-plan

Tout pixel de l'image est associé à une fonction ou à une valeur, permettant ainsi la modélisation de l'apparence de l'arrière-plan de ce pixel.

Seules les observations qui ont eu lieu en un point résulte le modèle d'apparence de l'arrière-plan en ce point. Les autres pixels de l'image n'interviennent pas. La modélisation peut être statique ou probabiliste. [1]

2.2.3.3 modélisations semi-locale de l'arrière-plan

La seule différence qui caractérise cette méthode à la méthode précédente est que la modélisation en un point dépend des observations qui ont eu lieu dans la région de l'image à laquelle elle appartient. [1]

2.2.3.4 modélisation globale de l'arrière-plan

Pour la construction d'un modèle de l'arrière-plan, à chaque instant ces méthodes utilisent l'ensemble des observations. [2]

2.3 Les différentes méthode de suivi de l'objet

Une fois l'objet détecté dans une image numérique ou une suite d'images séquentielles, arrive la phase de suivi. Le problème est de suivre cet objet en mouvement qui peut avoir n'importe quelle forme ou texture.

Dans ce qui suit on va présenter une classification des algorithmes de suivi qui ont été présentés ces dernières années. Ces méthodes sont utilisées par exemple pour la surveillance vidéo.

2.3.1 Classification des algorithmes de suivi

2.3.1.1 Algorithmes orientés mouvement

Ils consistent à regrouper des régions homogènes au sens du mouvement. Ils sont rapides mais cela ne veut pas dire que toutes ces régions ont une signification physique globale, du fait qu'elles n'ont pas été obtenues à partir d'une représentation d'objet mais plutôt à partir d'un critère d'homogénéité, toutes ces régions ont une signification physique globale. [13]

2.3.1.2 Algorithmes orientés modèle

La complexité calculatoire est importante vu qu'il est primordial de déterminer les rotations, translations et autres déformations, sachant que les régions suivies sont aussi importantes.

L'algorithme orienté modèle se base sur le suivi de régions correspondant aux projections 2D, on les représente par des modèles déformables soit en 3D ou 2D. [13]

2.3.1.3 Les méthodes basées sur les frontières

Ces méthodes sont indépendantes du type de mouvement ainsi que de l'objet, elles sont efficaces pour estimer les contours de l'objet précis, donc plus adaptées au suivi d'objets déformés

Leur principe est de déplacer d'un contour initial vers le contour désiré, elles sont utilisées beaucoup plus pour des mouvements lents. [13]

2.3.1.4 Les méthodes basées sur les régions

Contrairement aux méthodes précédentes celles-ci ont des difficultés à suivre avec précision les contours, elles sont plus robustes car elles utilisent une donnée plus globale et moins dépendante de l'initialisation.

Il est même possible de se passer de toute configuration puisque cette division révèle les objets en mouvement dans l'image.

2.3.2 Structure général des algorithmes de suivi

Le suivi d'un objet d'un instant t à $t+1$, est généralement composé de trois étapes

- Prédiction du mouvement des primitives de l'instant t à l'instant $t + 1$.
- Ajustement de la segmentation prédite.
- Estimation du mouvement des régions suivies de t vers $t + 1$.

2.3.2.1 Phase de prédiction

Elle fait la liaison de segmentation de l'image à l'instant t à celle de l'image suivante ($t + 1$). On peut envisager quatre modèles de prédiction.

2.3.2.1.1 Prédiction de mouvement nul

On considère que la primitive est fixe. Sa position prédite est donc identique à celle qu'elle avait à l'instant précédent

2.3.2.1.2 Prédiction à court-terme simple

L'unique information prise en compte est le mouvement entre les deux images précédentes. [13]

2.3.2.1.3 Prédiction à court-terme lissée

On utilise alors un modèle d'évolution, tel que le filtre de Kalman. Le mouvement considéré est celui fourni par les équations de filtrage (ou lissage) de ce filtre.

2.3.2.1.4 Prédiction à long-terme

Dans ce cas, l'information prise en compte est le mouvement entre toutes les images précédentes. Les valeurs prédites sont alors issues de l'équation de prédiction du filtre de Kalman. [13]

2.3.3 Phase d'ajustement

Une fois la première phase réalisée on dispose d'une segmentation prédite mais cette dernière n'est généralement pas parfaite.

C'est pour cela que l'ajustement des frontières et des régions de cette segmentation est une étape importante.

Il existe deux types de méthode d'ajustement qui sont :

2.3.3.1 Ajustement basé contour

Ces méthodes utilisent des modèles de contours déformables afin d'ajuster les frontières des régions prédites

2.3.3.2 Ajustement basé région

Cette approche utilise des modèles de régions déformables. Elle peut prendre en considération à la fois des informations spatiales dans l'image

courante (gradient spatial, texture ... etc.), et des informations de mouvement entre les images courantes et précédentes.

2.3.4 Phase d'estimation

Cette phase est effectuée directement après la phase d'ajustement pour permettre de savoir quel a été le mouvement réel des régions suivies entre les images et aux instants t et $t+1$.

Pour cette estimation on utilise l'une des trois grandes méthodes qui sont :

- ° Les méthodes par mise en correspondance.
- ° Les méthodes par transformées.
- ° Les méthodes différentielles.

2.3.5 Suivi de trajectoire d'objet en mouvement

Le fonctionnement de ce principe réside dans le fait de considérer comme point chaque objet à suivre qui représente son point de gravité, dans ce cas c'est l'interpolation entre les centres de gravité de l'objet détecté qui est la trajectoire d'objet en mouvement, il y a toujours une erreur quand on segmente un objet dans une image ou quand on mesure sa position.

On diminue les erreurs pour calculer la trajectoire de l'objet.

2.3.6 Suivi de mouvement par filtre de Kalman

Le filtre de Kalman est une méthode de prédiction qui intéresse beaucoup de chercheurs.

C'est une approche statistique qui se présente sous forme d'un ensemble d'équations différentielles, il permet une estimation optimale de processus aléatoire, il estime des variables d'état d'un système (par exemple une tension et une intensité), il s'applique aux signaux non stationnaires, on peut l'utiliser par exemple pour coupler des informations venant de sources différentes afin de retrouver une trajectoire. [13]

2.5. Conclusion

Dans ce chapitre on a vu différentes méthodes de détection de mouvement dans laquelle on a considéré que les séquences vidéo sont des images successives, qui sont répertoriées en deux catégories, les images courantes et les images passées, dans ce cas de figure on va construire un modèle qui va représenter les images passées et les confronter à celles courantes afin de savoir si la détection et le suivi de l'objet sont possibles.

Chapitre III

Etude et réalisation d'un système de détection et le suivi d'une personne

3.1 Introduction

Dans ce chapitre nous allons présenter le travail effectué dans le cadre de notre mémoire de fin d'étude, qui consiste à adapter un robot articulé à deux degrés de liberté qui a pour objectif la détection et le suivi d'une personne en temps réel. Le système est composé d'une caméra et d'un ensemble mécanique articulé qui garantit l'orientation de cette dernière et il est commandé par l'asservissement visuel qui a pour objectif de garder une personne détectée au centre du champ de vision de la caméra. Le tout est piloté par un microordinateur via une carte d'acquisitions de type Arduino. Dans la première section de ce chapitre, nous allons présenter la partie software, quant à la deuxième elle sera portée sur la partie hardware.

3.2 Détection et suivi d'une personne

Les systèmes de vision artificielle suscitent des données récoltées à partir des images, ces dernières dans notre projet sont acquises à partir d'une caméra USB de résolution 640×360 pixels et une vitesse d'acquisition de 30 FPS et a 3 canaux du domaine RGB.

Le travail effectué dans ce mémoire a été développé, simulé et programmé avec le logiciel MATLAB, qui nous offre une souplesse d'utilisation ainsi que la présence des méthodes mathématiques dont notre algorithme final a besoin et des méthodes de traitement d'images préprogrammées et surtout la présence de l'algorithme de VIOLA & JONES utiliser pour la détection de personne dans notre travail. La version de MATLAB sur laquelle notre projet a été développer est MATLAB « R2016b » implémenté une machine de calcule de possesseur INTEL CORE i7 3120U et une capacité mémoire RAM de 8GO.

Les images issues des caméras sont des informations brutes, une étape de traitement est indispensable, la complexité de ce traitement dépend de la détection souhaitée, dans notre cas nous allons faire de la détection et suivi d'une personne dans une séquence d'images prises et traitées en temps réel. Pour cela, nous allons utiliser une des méthodes qui ont été présentées dans le chapitre précédent à savoir, la méthode de VIOLA et JONES, cette méthode basée sur un classificateur supervisé. L'entraînement de la méthode de VIOLA & JONES n'est pas l'objet d'intérêt notre travail. MATLAB contient des bases entraînées pour divers détection (visages, haut de corps, personnes, yeux, bouches, voiture, avion...)

Sachant que y'a une fonction d'entraînement pour tout objet qu'on souhaite détecter en injectant à cette dernière une base de l'objet a détecté.

On n'a pas eu besoin de faire un entraînement dans notre cas vu que MATLAB a des bases entraînées sur ce qu'on a besoin de détecter.

Au début on a commencé par tester la détection de personnes proposée par MATLAB. Cette méthode a dû être abandonnée pour sa non-stabilité et sa détection erronée, on peut distinguer la détection de personne là ou y a absence de personnes mais aussi l'absence de détection en cas de présence de personne dans le champ visuel.

Dans ce travail, on a opté pour l'utilisation de la stratégie présentée ci-dessous car elle est caractérisée par une meilleure stabilité et robustesse comparé aux méthodes préprogrammées par MATLAB.

La détection de visage par la méthode de VIOLA & JONES est plus stable, elle est efficace malgré de fausses détections qui s'ajoutent parfois simultanément à cette dernière (fig 3.1)

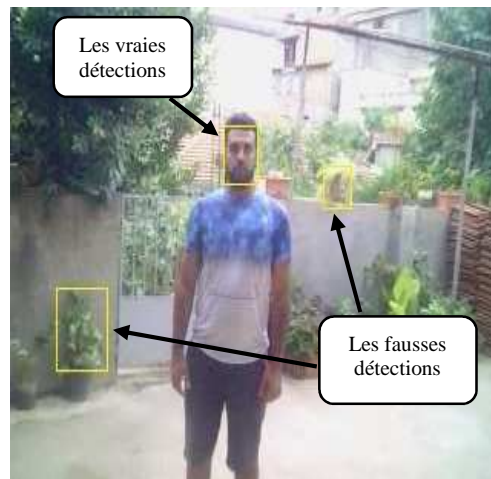


fig (3.1) détection de visage par la méthode de VIOLA & JONES

La méthode utilisée pour avoir la figure (3.1) détecte seulement les visages de face, en utilisant une autre base d'apprentissage qui détecte les visages de profil on retrouve pratiquement les mêmes défauts et avantage que la méthode de détection de visages de face, pour cela l'utilisation des deux détections serait intéressante pour détecter le visage de la personne qu'elle soit de face ou de profil.

Pour rendre la méthode plus stable et plus robuste, on a introduit la détection du haut du corps, elle aussi elle détecte bien les hauts du corps, mais on retrouve aussi les fausses détections qui sont bien plus importantes que celles de la détection de visage.

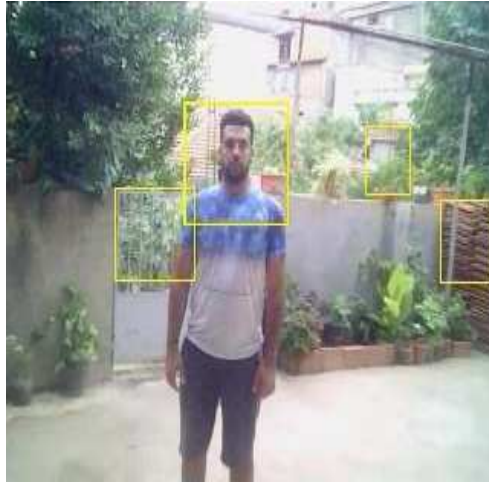


fig (3.2) détection du haut de corps par la méthode de VIOLA & JONES

Le défaut majeure qu'on a pu remarquer de par l'application de ces méthodes est la détection de faux positif (détection de personnes qui n'existe pas par exemple : le cadre d'une porte est considéré comme le haut d'un corps, un pot de fleur est considéré comme visage), pour y remédier on a programmé une fonction qui fait l'intersection entre la détection du haut du corps et la détection du visage, obligation de détection de visage dans la zone du haut du corps, vu que y'a pas de haut du corps sans visage pour une personne.

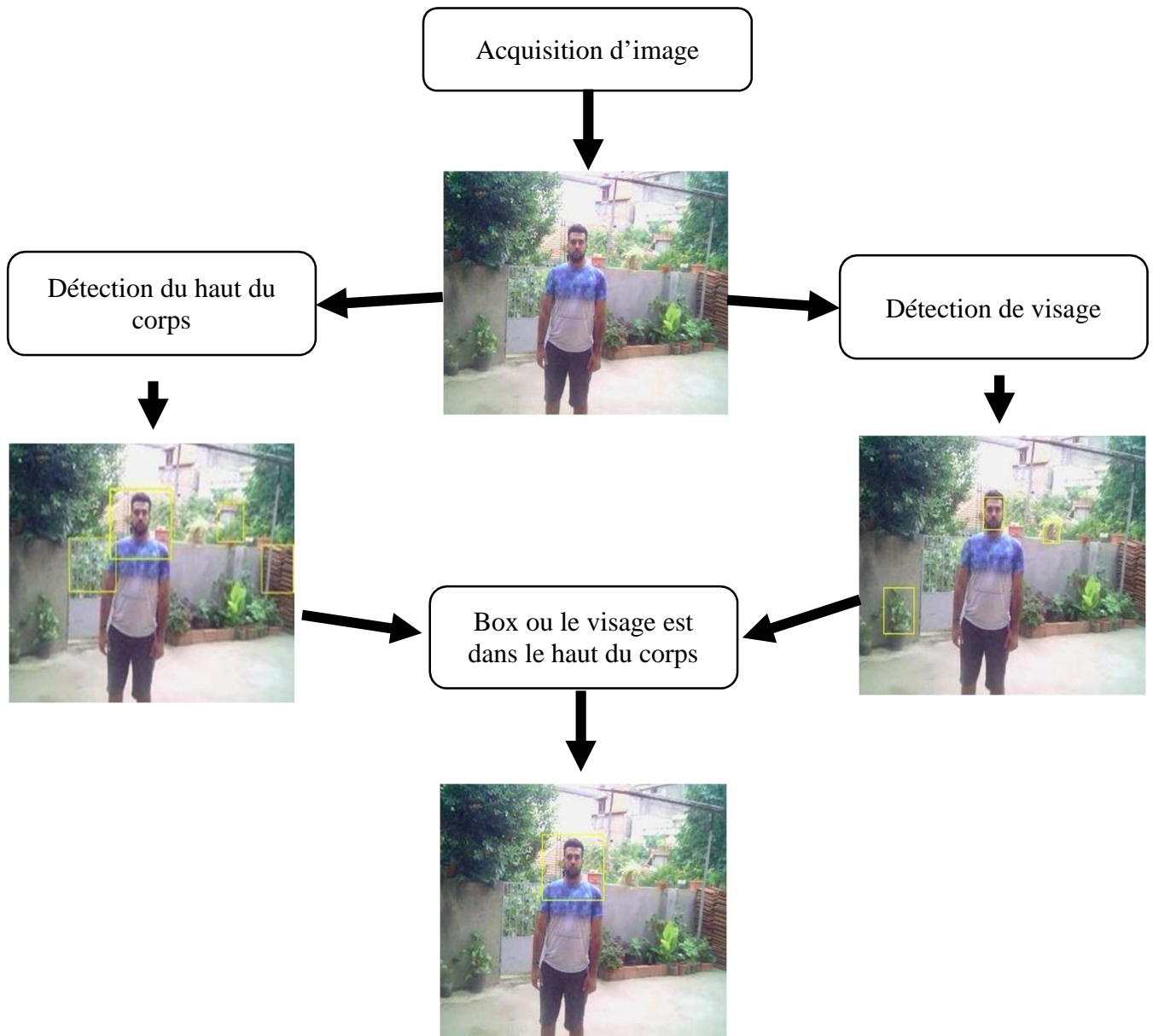


fig (3.3) éliminations de faux positive

Ensuite on a estimé que la longueur du cadre qui entoure la zone du haut du corps fait $\frac{1}{3}$ la taille d'une personne, donc on a multiplié le cadre par 3 pour détecter toute la personne

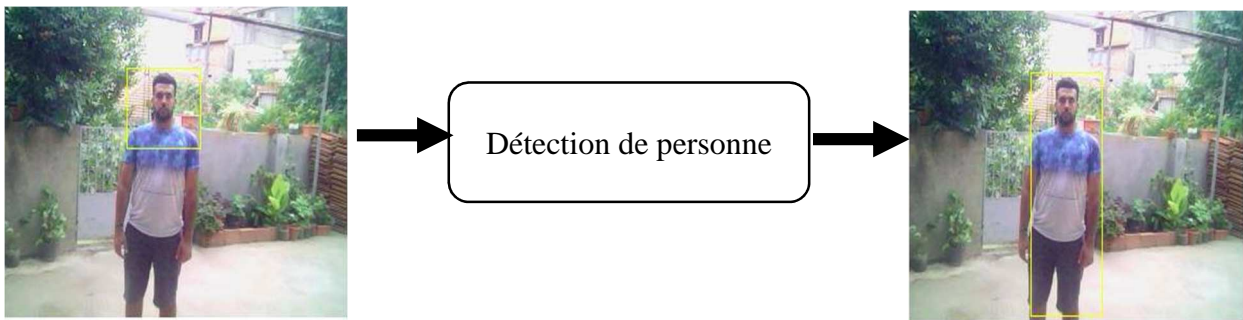


fig (3.4) détection de personne

Une fois la personne détectée on passe à l'étape de suivi, pour cela on applique l'algorithme de détection de coin par valeur propre minimale pour la zone où la personne a été détectée (fig3.5).



fig (3.5) détection de coin par valeur propre minimale

Cet algorithme est appliqué sur l'intégralité de la boîte enveloppante. Mais vu que le rectangle contient des parties de l'arrière-plan, On applique un masque de la forme d'une silhouette d'un humain (fig 3.6) pour détecter des points caractéristiques que sur la personne et éliminer les points détectés sur l'arrière-plan (fig.3.7

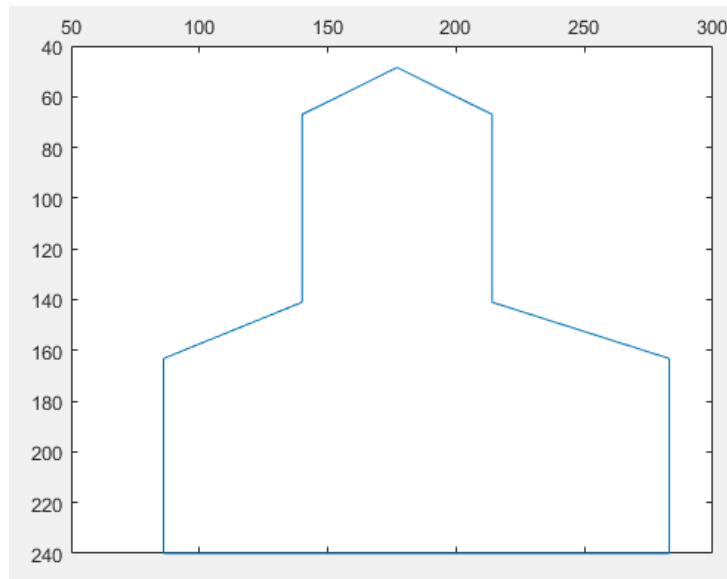


fig (3.6) masque a la silhouette d'un humain

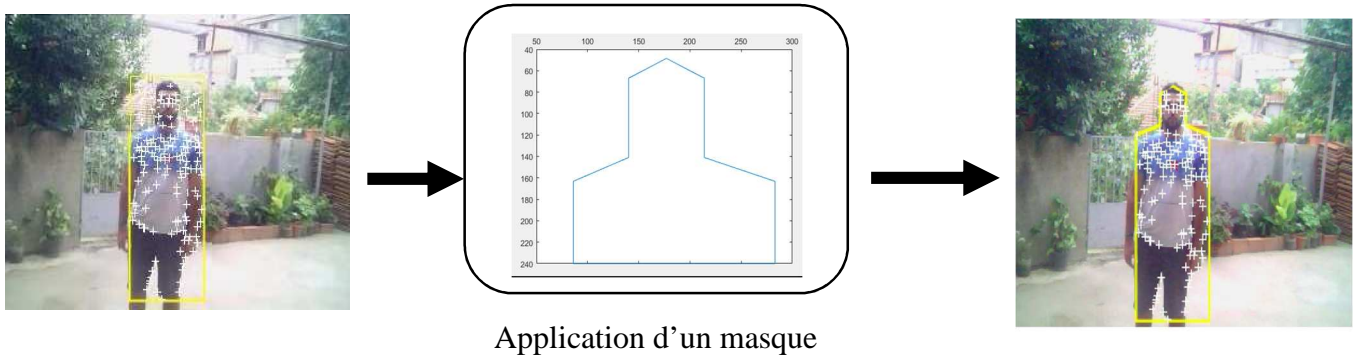


fig (3.7) élimination des points caractéristiques de l'arrière-plan

Dans le cas où le nombre de points caractéristiques est inférieurs à dix, l'algorithme cherche de nouveaux points. Le suivi utilise un algorithme d'estimation qui est la transformation géométrique basé sur l'algorithme kanade-Lucase-Tomasi [13]. C'est une méthode différentielle utilisée dans l'estimation du flot optique dans le but est de résoudre le problème du coût élevé en temps de calcul. L'algorithme kanade-Lucase-Tomasi utilise les informations d'intensité spatiale pour diriger la recherche de la position qui donne la meilleure correspondance. Il est plus rapide que les techniques traditionnelles par le fait qu'il examine beaucoup moins de correspondances potentielles entre les images. Les points détectés sont enveloppés par un

polygone dont le centre de gravité est un point qui modélise la personne, les coordonnées du centre de gravité sont injectées comme mesure d'un régulateur PID (fig3.9) où la consigne de ce dernier sont les coordonnées du centre de l'image.

L'organigramme suivant résume notre stratégie (3.8).

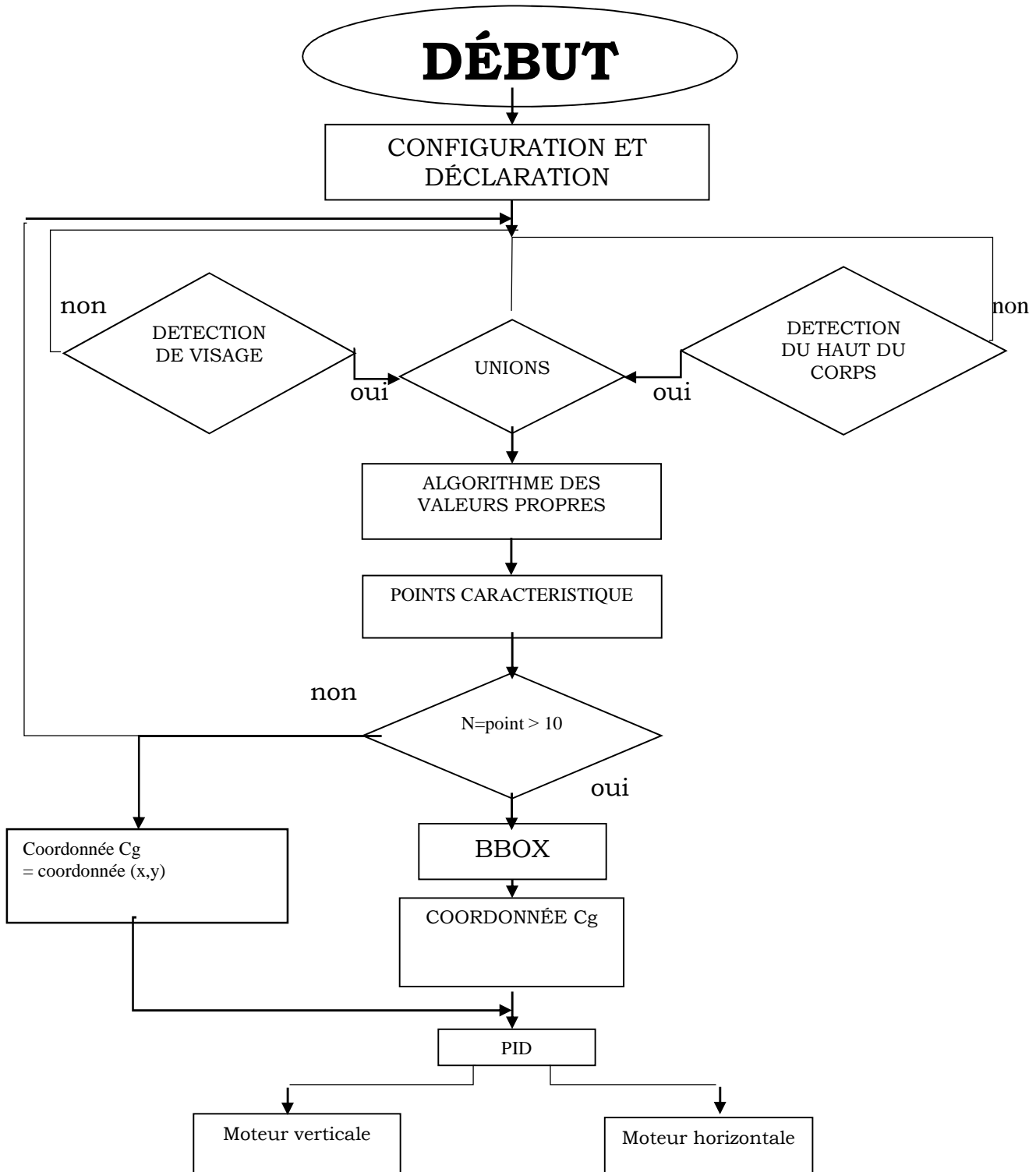


fig (3.8) Organigramme du programme

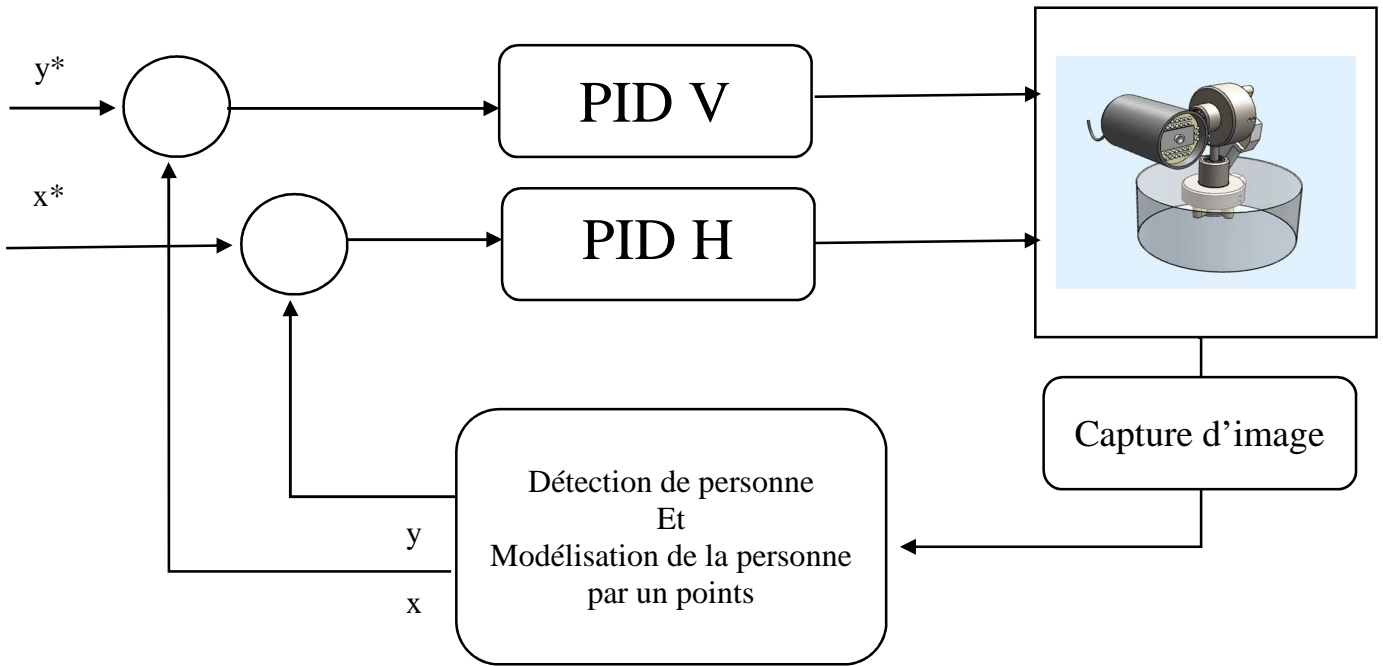


fig (3.9) boucle d'asservissement visuel

3.2.1 Régulateur PID :

Le régulateur PID (Proportionnel Intégral Dérivé) appelé aussi correcteur PID est un organe de contrôle permettant d'améliorer les performances d'un asservissement c'est-à-dire un système effectuant une régulation en boucle fermée.

Principe :

C'est un algorithme de calcul, il délivre un signal de commande à partir de l'erreur observée de la différence entre la consigne et la mesure. Dans notre cas la mesure c'est le centre de gravité de la personne détectée et la consigne le centre de vision.

Le PID utilise trois actions mathématiques : action proportionnelle, action intégrale, action dérivée.

Architecture :

Il existe trois structures pour combiner les trois actions. Série, parallèle, mixte. La différence entre ces trois structures consiste en l'effet des coefficients de réglage sur le comportement du régulateur.

$$C(s) = K\left(1 + \frac{1}{sTi}\right)(1 + sTd) \quad (3.1)$$

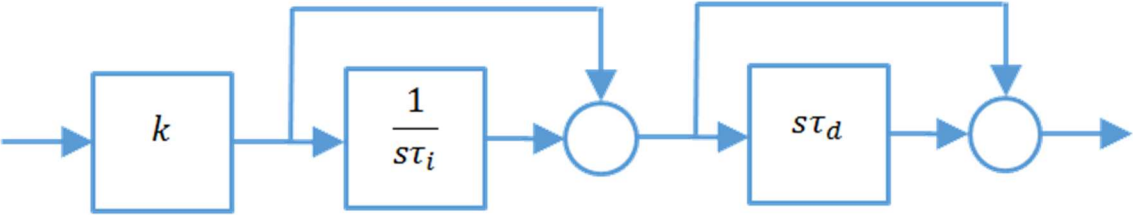


fig.(3.10) structure série d'un PID

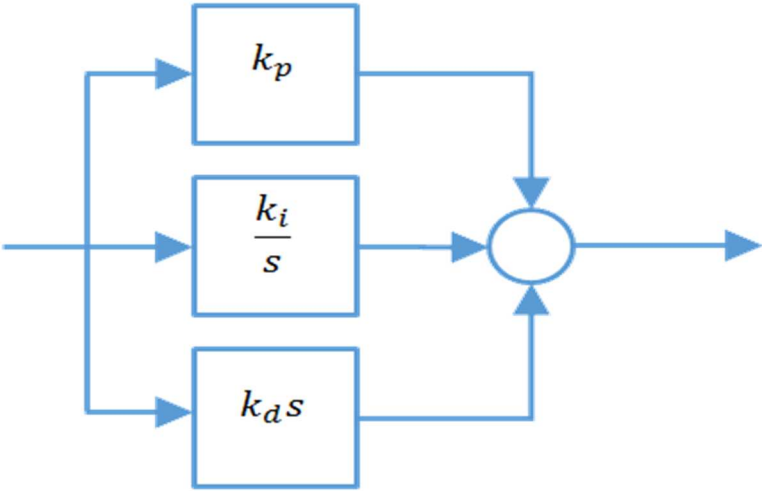


fig (3.11) structure parallèle d'un PID

$$C(s) = k_p + \frac{k_i}{s} + k_d s \tag{3.2}$$

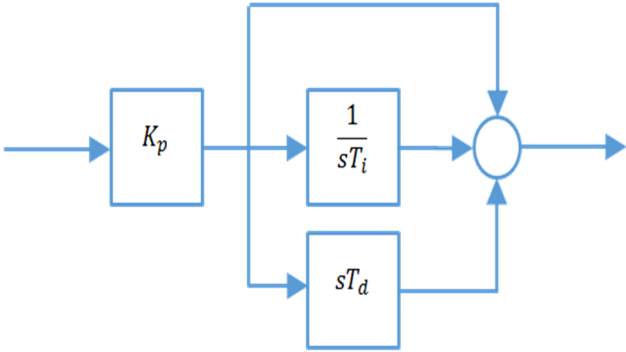


fig (3.12) structure mixte d'un PID

$$C(s) = Kp(1 + Ti \frac{1}{s} + Td s) \quad (3.3)$$

Il est difficile de recommander une forme particulière pour le PID car elles représentent chacune des avantages et des inconvénients. En réalité toutes ces formes sont similaires, il existe des formules pour passer d'une forme à une autre.

Vu que l'action dérivée est non causale, l'action dérivée est estimée avec un filtrage ou le terme $Td s$ est remplacé par $\frac{Td s}{1 + \frac{Td}{N} s}$. Donc les correcteurs précédents deviennent

$$\text{En série} \quad C(s) = K(1 + \frac{1}{sTi})(1 + \frac{Td s}{1 + \frac{Td}{N} s}) \quad (3.4)$$

$$\text{En parallèle} \quad C(s) = kp + \frac{ki}{s} + \frac{Td s}{1 + \frac{Td}{N} s} \quad (3.5)$$

$$\text{Mixte} \quad C(s) = Kp(1 + Ti \frac{1}{s} + \frac{Td s}{1 + \frac{Td}{N} s}) \quad (3.6)$$

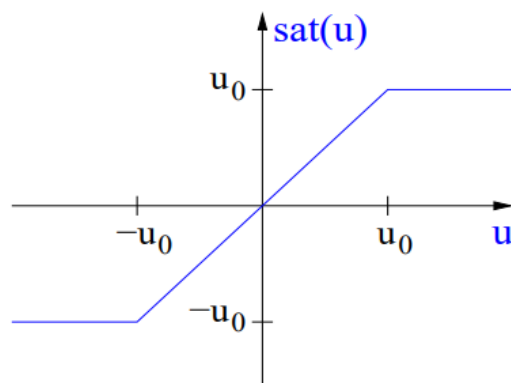
L'écriture discrète du PID discret mixte sera donc comme suit sachant que $C(s) = \frac{E(s)}{Y(s)}$.

$$y(k) = e_k + \frac{Te}{Ti} \sum_{i=0}^k e_i + \frac{Td}{Te} (e_k - e_{k-1}) \quad (3.7)$$

Le programme a implémenté sur un calculateur est exprimé ci-dessous.

```
# def Te, Ki, Kp, Kd, Cog;
i = 0;
e0 = e = 0;
while (1)
{
    # y = mesure de la sortie
    e = cog - y ;
    // calcul de la partie intégral
    i = i+Ki*Te*e ;
    // calcul de la commande
    u = Kp * (e + i + Kd/Te*(e-e0)) ;
    // enregistrement d'erreur
    e0 = e ;
}
```

Vu que l'action intégrale en numérique est la somme de toutes les erreurs précédentes, donc en cas de blocage du terminal, un cumul d'erreurs doit être compensé, pour réduire la forte compensation un anti-saturation est utilisé (Anti-Windup en anglais) (fig3.13)



Fi

fig (3.13) fonction de saturation

Cette fonction peut être programmée ainsi :

$$\text{sat}(u) = \begin{cases} -u_0 & \text{si } u < -u_0 \\ u & \text{si } -u_0 \leq u \leq u_0 \\ u_0 & \text{si } u > u_0 \end{cases}$$

L'algorithme précédent devient :

```
# def Te, Ki, Kp, Kd, Cog ,l;  
i = 0;  
e0 = e = 0;  
while (1)  
{  
    # y = mesure de la sortie  
    e = cog - y ;  
    // calcul de la partie intégral  
    i = i+Ki*Te*e ;  
    // anti-saturation  
    if (i< -l)  
        { l = -l }  
    else if(i> l)  
        { i = l }  
    // calcul de la commande  
    u = Kp * (e + i + Kd/Te*(e-e0)) ;  
    // enregistrement d'erreur  
    e0 = e ;  
}
```

3.3 Partie hardware du projet

Les éléments qui composent cette partie sont deux servomoteurs modifiés qui articulent sur deux degrés de libertés et une caméra et un circuit de commande de type Arduino.

3.3.1 Les servomoteurs

Le servomoteur est un moteur souvent appelé <servo> ce nom provient du latin <servus> qui signifie esclave, donc on peut dire que c'est un moteur esclave muni d'un système asservi intégré lui permettant d'avoir une bonne puissance.

Il est capable de réaliser des déplacements en translation ou des rotations.

En général les servomoteurs sont utilisés pour atteindre une position prédéterminée et la maintenir. Ils sont utilisés dans des avions radio-contrôlés, pour positionner les surfaces de contrôle, exemple les gouvernails, les robots, des pinces de fonctionnement, des voitures jouets etc.

Ils sont également utilisés dans les applications industrielles et la robotique.



fig (3.14) Servomoteurs

3.3.2 Composition du servomoteur

Le servomoteur est composé de plusieurs éléments, parmi eux des éléments invisibles.

Les éléments visibles :

- ° trois fils qui se distingue par 3 couleurs (le noir ou marron pour la masse, le rouge pour l'alimentation positive entre 4,5V à 6V et enfin le jaune ou orange ou blanc pour l'entrée de signal de commande).
- ° l'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.

Les éléments invisibles :

- ° Un petit moteur a courant continue(CC).
- ° Un train d'engrenage qui permet d'augmenter le couple et de réduire la vitesse angulaire.
- ° Un potentiomètre pour contrôler la position de l'axe de sortie sur lequel est lié le palonnier (plateau de rotation).
- ° Une butée qui limite l'angle de rotation a 180°.
- ° Une carte électronique d'asservissement

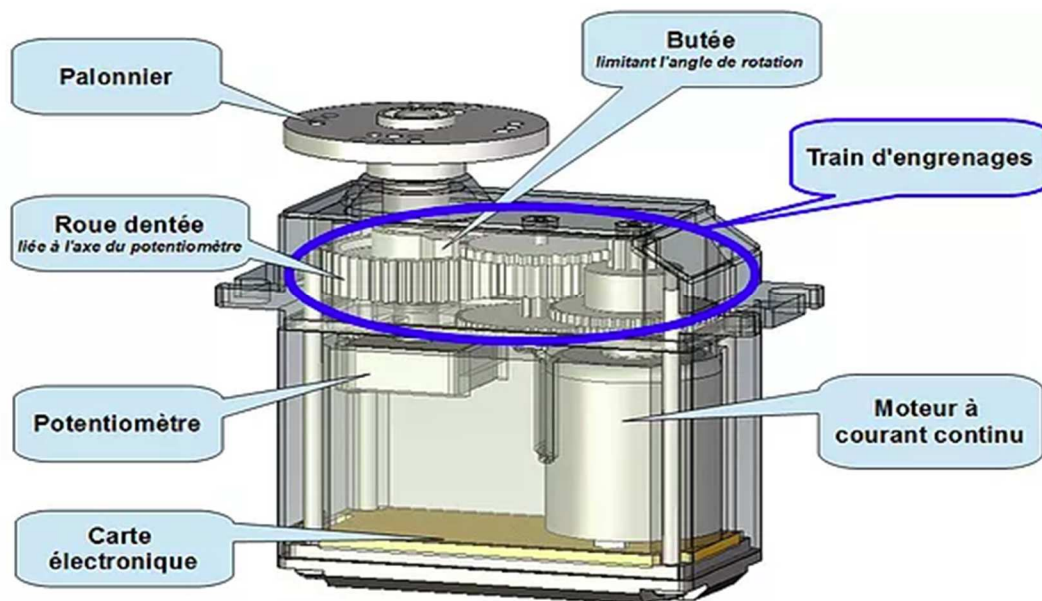


fig (3.15) les composant d'un servomoteur

3.3.3 Asservissement angulaire du servomoteur

Les servomoteurs sont commandés comme dit précédemment par l'intermédiaire d'un câble à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion PWM. Cela signifie que c'est la durée des impulsions qui

détermine l'angle de l'axe de sortie et donc la position du degré de liberté à commander. Le signal doit se répéter périodiquement, en général et dans notre cas toutes les 20 millisecondes, ce qui permet à l'électronique du servomoteur de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre qui représente le capteur de la chaîne de retour de l'asservissement. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle du correcteur numérique implémenté dans l'électronique du servomoteur est de commander le moteur en vitesse pour que la position de l'axe de sortie soit conforme à la consigne reçue.

Une fois que la mesure rejoint la consigne, l'erreur sera nulle et donc le correcteur commandera une vitesse nulle. La largeur d'impulsion pour un angle souhaité est représenté dans la figure ci-dessous (fig 3.16) où la largeur d'impulsion de 1ms correspond à un angle de 0° et la largeur de 2ms à un angle de 180° et toute largeur d'impulsion entre 1ms et 2ms va correspondre à un angle respectif entre 0° et 180° .

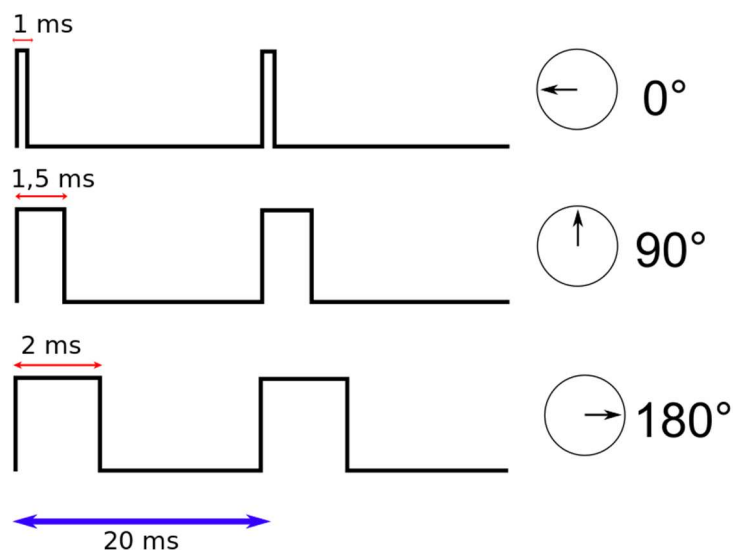


fig (3.16) rapport largeur d'impulsion/angle de sortie

3.3.4 Modification d'un servomoteur asservi en position à un servomoteur asservi en vitesse

Les servomoteurs achetés pour le projet sont des mg996 (fig 3.17)



fig (3.17) servomoteur mg996

Ce servomoteur est asservi en position, pour cela on a effectué une modification sur le produit acheté pour le transformer en un servomoteur asservi en vitesse. En premier lieu on a procédé à l'ablation de la butée qui limitait la rotation à moins d'un tour, pour que ce dernier se libère et puisse tourner au nombre de tours souhaités. Puit la suppression du potentiomètre qui indiquait la position de l'arbre. On a pu donc obtenir un asservissement en vitesse en boucle ouverte qui était à la base un asservissement en position en boucle fermée. Le potentiomètre est remplacé par 2 résistances identiques de 200Ω . Le correcteur dans ce cas comprendra que l'arbre est à un angle de 90°

Si on injecte un signal d'entrée PWM de largeur d'impulsion $\ll 1.5\text{ms}$ et bien évidemment inférieur a 1ms le contrôleur comprendra qu'il y a un grand écart à compenser donc le moteur tournera vite dans un sens. Et si le signal d'entrée PWM de largeur d'impulsion est proche de 1.5ms le contrôleur comprendra qu'il y a un petit écart à compenser donc le moteur tournera lentement dans un sens. Si on applique la même logique pour la largeur d'impulsion compris entre $]1.5\text{ms}, 2\text{ms}]$ le contrôleur va comprendre une erreur négative ou un dépassement. La compensation sera avec une vitesse négative, donc le moteur tournera dans l'autre sens.

Notre système articulé à deux degrés de libertés est actionné par des actionneurs rotatifs de type moteur à courant continu obtenu par modification des servomoteurs comme expliquer précédemment de telle manière à ce que ce dernier devienne comme un moteur à courant continu couplé à un correcteur numérique, cet actionneur qui a comme entrée un signal PWM de rapport cyclique $\alpha(t)$ est comme sortie une vitesse $v(t)$.

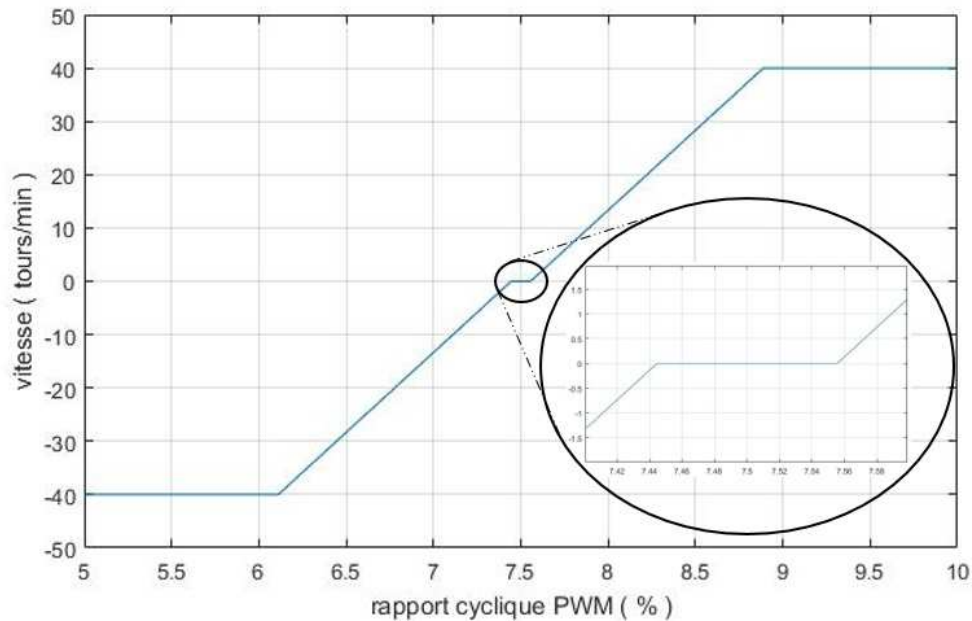


fig (3.18) les rapport entre/ sortir du servomoteur utilisé

Des mesures ont été faites pour la modélisation de notre nouveau système, la figure (3.18) représente la sortie v en fonction de l'entrée α , on remarque que le système est non linéaire vu la présence de la zone morte (fig3.17), dans le cadre de notre travail on a considéré que le système est linéaire en ignorant la zone morte pour avoir un modèle mathématique du système de type fonction de transfert, la transmittance de ce système a été estimée à l'aide d'un toolbox de Matlab (ident toolbox), pour cela ; le système a été excité par une entrée échelon et à l'aide d'un capteur de vitesse tachymétrique et d'un oscilloscope, les valeurs de vitesse ont été enregistrées dans un vecteur. Ce vecteur est injecté dans cet outil de modélisation que MatWorks a mis à notre disposition, cet outil nous a estimé la fonction de transfert, fig. (3.20)

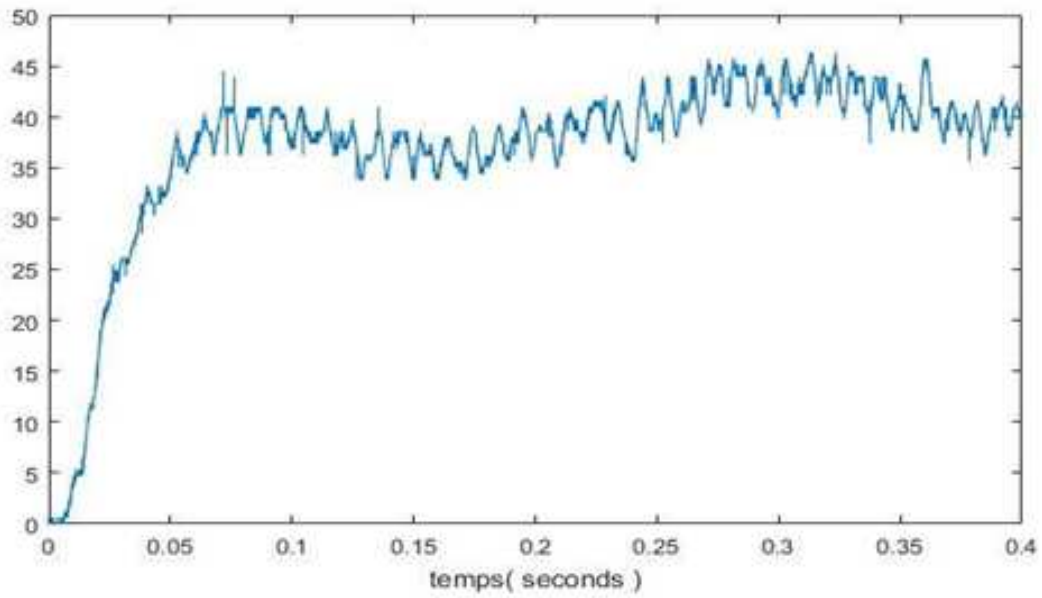


fig (3.19) réponse indicielle mesurée

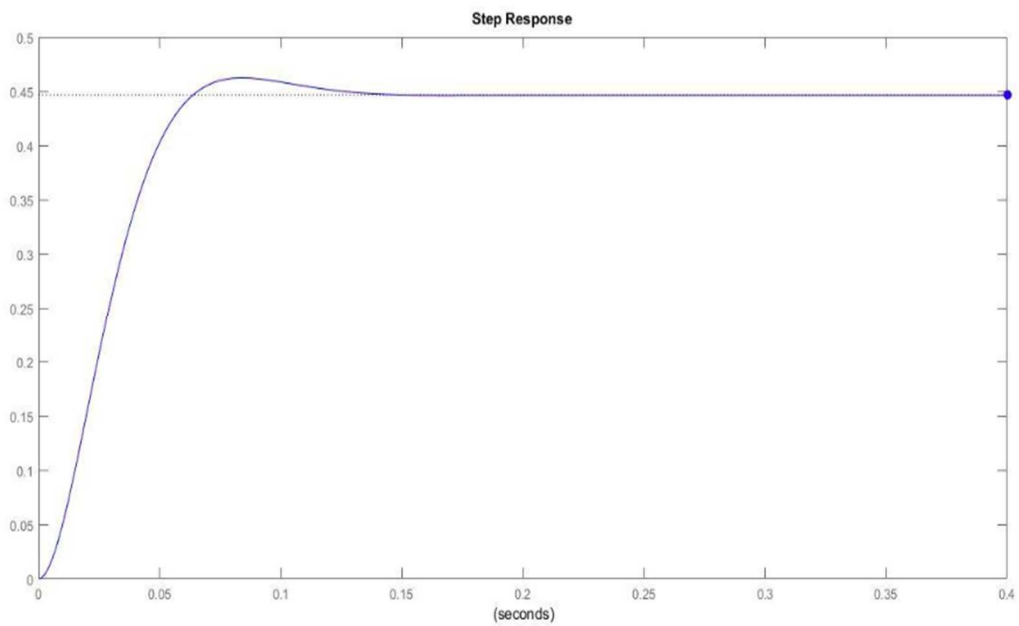


fig (3.20) réponse indicielle de la fonction de transfert estimée

$$g(s) = \frac{v(s)}{\alpha(s)} = \frac{1340}{s^2+80s+3000} \quad (3.9)$$

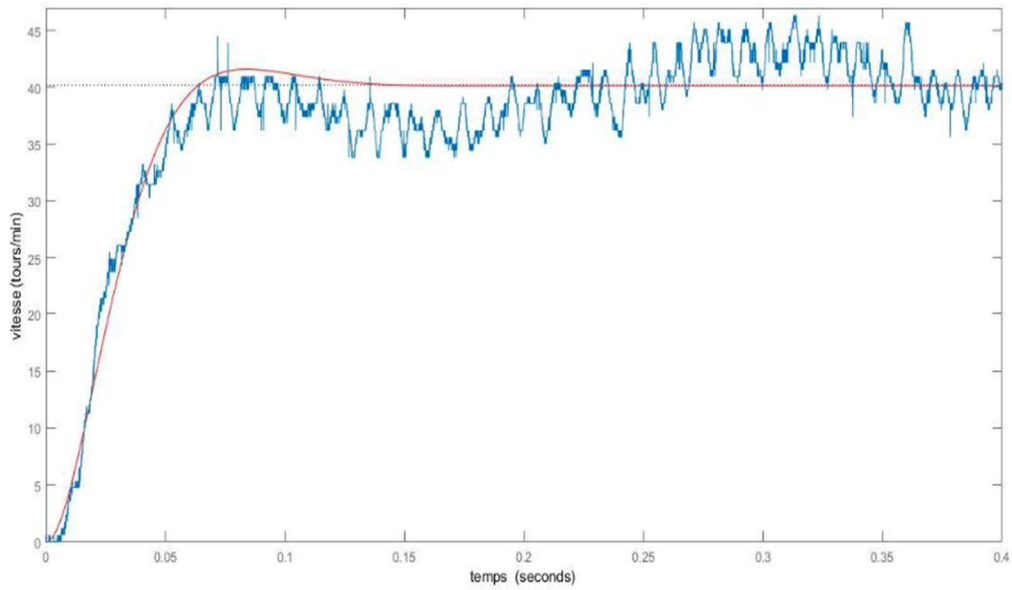


fig (3.21) les deux réponses

Vu que la position c'est l'intégral de la vitesse donc

$$H(s) = \frac{p(s)}{\alpha(s)} = \frac{g(s)}{s} = \frac{1}{s} \frac{v(s)}{\alpha(s)} = \frac{1340}{s^3 + 80s^2 + 3000s} \quad (3.10)$$

Telle que $p(s)$ est la position angulaire.

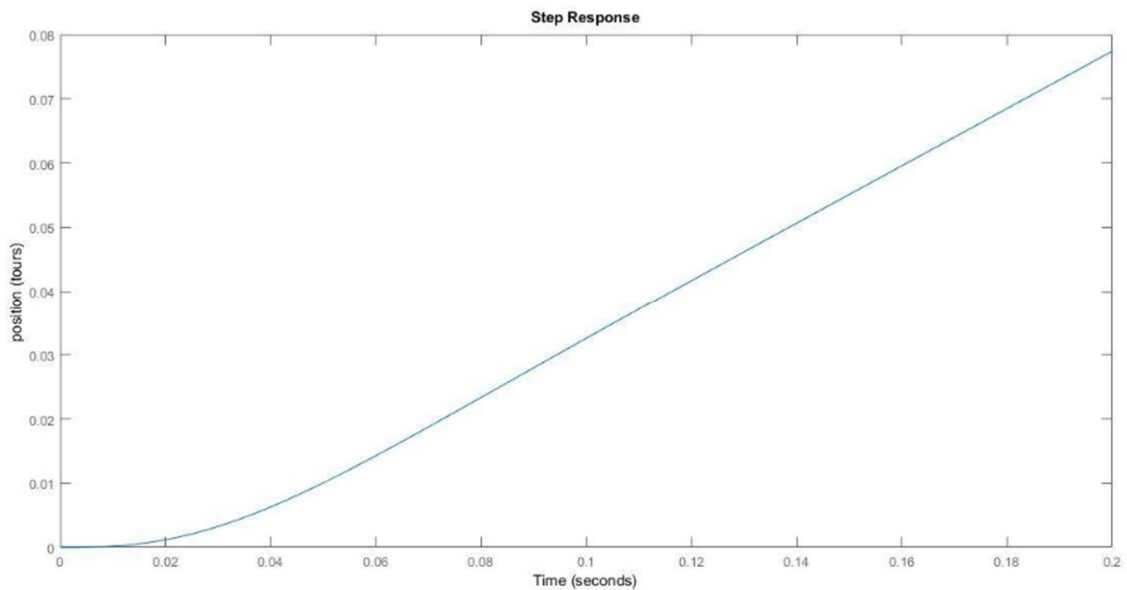


fig (3.22) sortie v en fonction de l'entrée α

3.3.4 Commande des moteurs

Afin d'avoir la personne détectée modélisée par son centre de gravité au centre de l'image aperçu, le système est contrôlé par un PID pour éliminer l'erreur statique, diminuer le dépassement, diminuer le temps de réponse et le temps de montée, et obtenir un système précis, rapide, stable et robuste. La commande de ce dernier est insérée dans la chaîne directe de l'asservissement en série avec le processus comme l'indique la figure suivante.

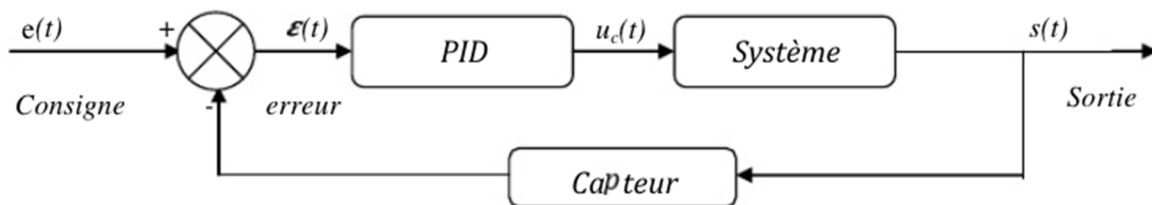


fig (3.23) Schéma bloc d'un asservissement avec correcteur (boucle fermée)

Ce correcteur élabore une commande $u(t)$ en fonction de trois actions proportionnelle, intégrale, dérivée.

3.3.5 La carte ARDUINO UNO

La carte ARDUINO UNO est une carte électronique programmable reposant sur un circuit intégré (microcontrôleur Atmega-328-p), ce dernier est associé à une interface USB de type C lui permettant de communiquer avec un ordinateur et lui servant d'alimentation, La carte peut être aussi alimentée via une prise Jack en la connectant à une alimentation externe de 5 V 500ma. La source d'alimentation est automatiquement sélectionnée

En général l'UNO est dotée de :

- 14 entrées/sorties numériques dont 6 fournissent la sortie PWM (pulse wide modulation), 6 entrées analogiques
- un cristal (horloge) à 16 MHz ou MIPS permettant au microcontrôleur d'exécuter jusqu'à 16 millions d'instructions par seconde
- une connexion USB et une prise jack d'alimentation qu'on a décrit précédemment
- un en-tête ICSP
- une fonction reset.

Caractéristiques techniques :

Microcontrôleur

ATmega328P

Tension de fonctionnement

5V

Tension d'entrée (recommandé)

7-12V

Tension d'entrée (limite)

6-20V

E / S numériques Pins

14 (dont 6 fournissent la sortie PWM*)

PWM numérique E / S Pins

6

Pins d'entrée analogique

6

DC Courant par I O Pin /

20 mA

Courant DC pour 3.3V Pin

50 mA

Mémoire flash

32 KB (ATmega328P) dont 0,5 KB utilisée par bootloader**

2 KB (ATmega328P)

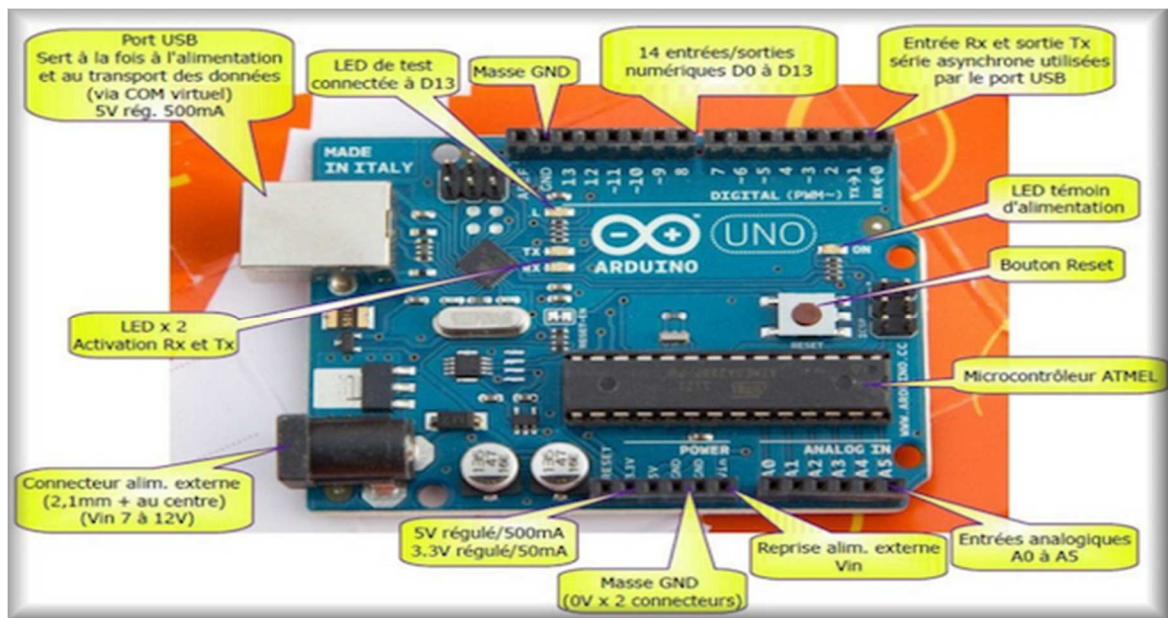


fig (3.24) Description de la carte Arduino

3.3.5 Branchement des servomoteurs à Arduino

C'est un branchement assez simple, dû au fait que le servomoteur a besoin de trois fils de connections pour fonctionner comme on l'a expliqué précédemment :

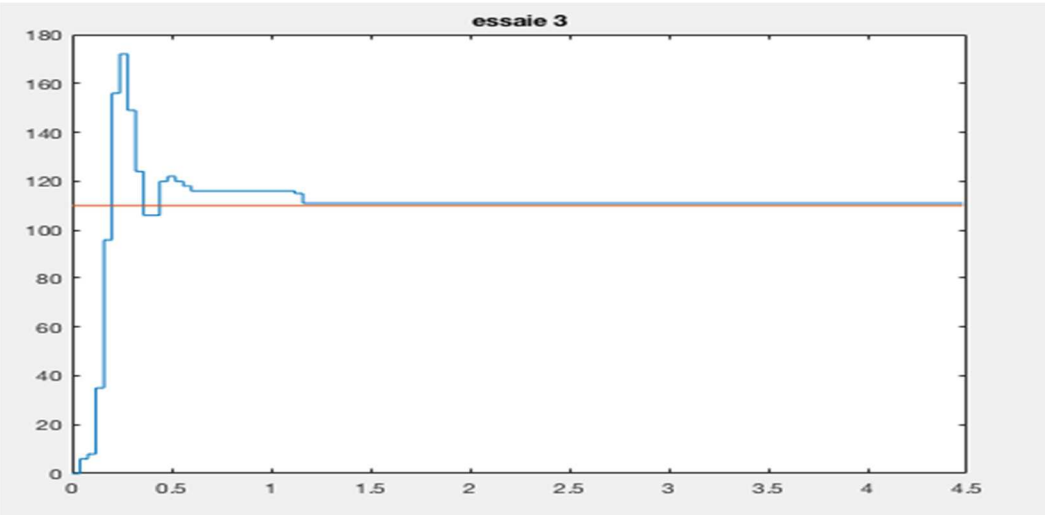
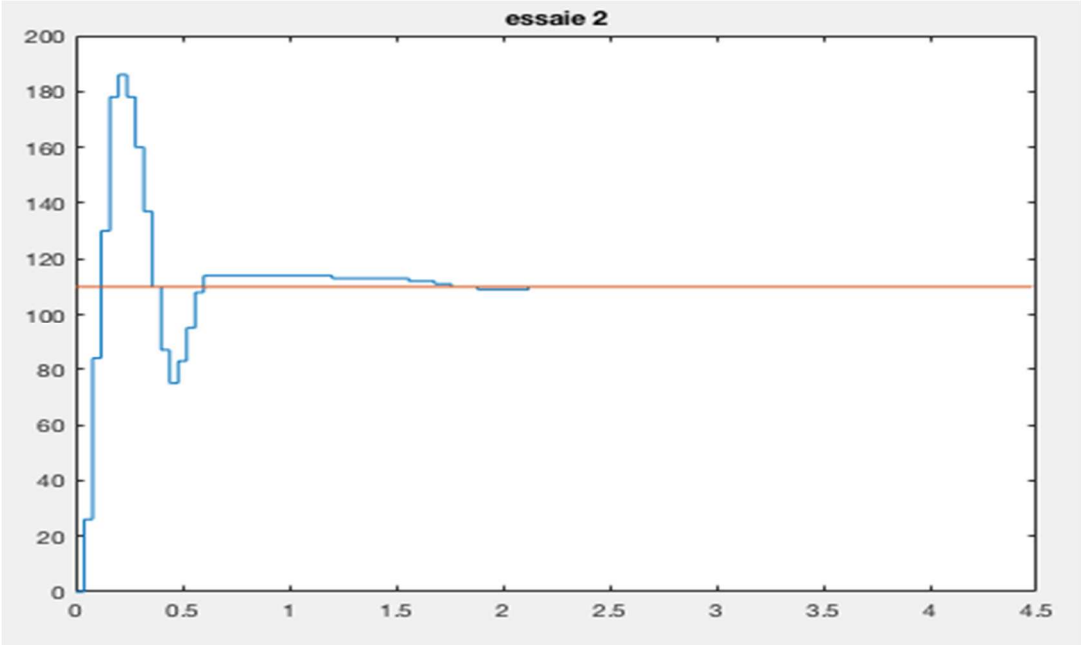
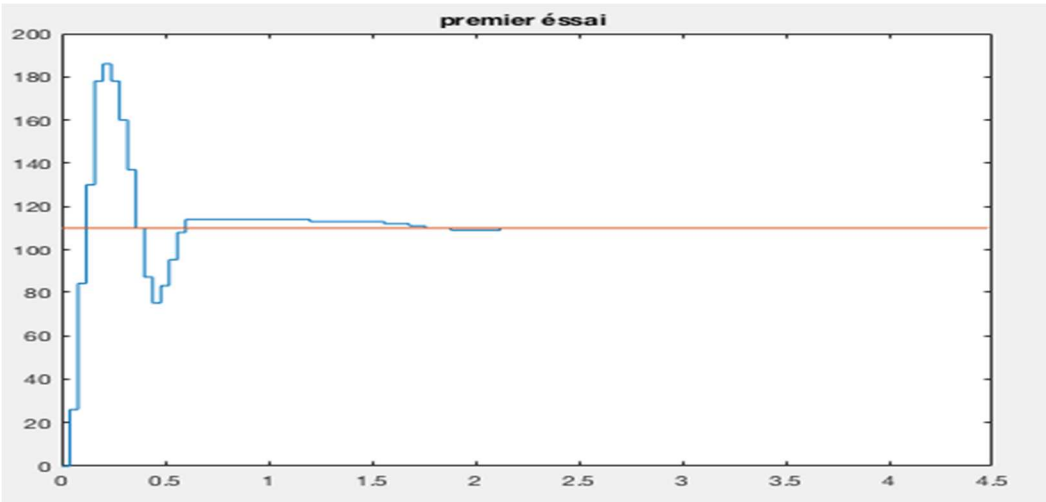
- ° 1 fil pour l'alimentation.
- ° 1 fil pour la masse.
- ° 1 fil pour entré du signal de commande.

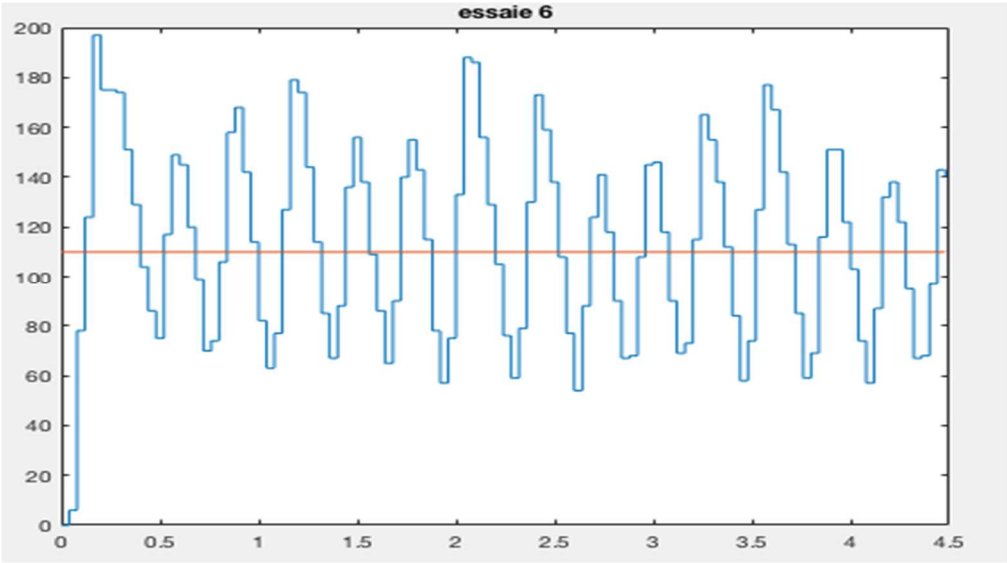
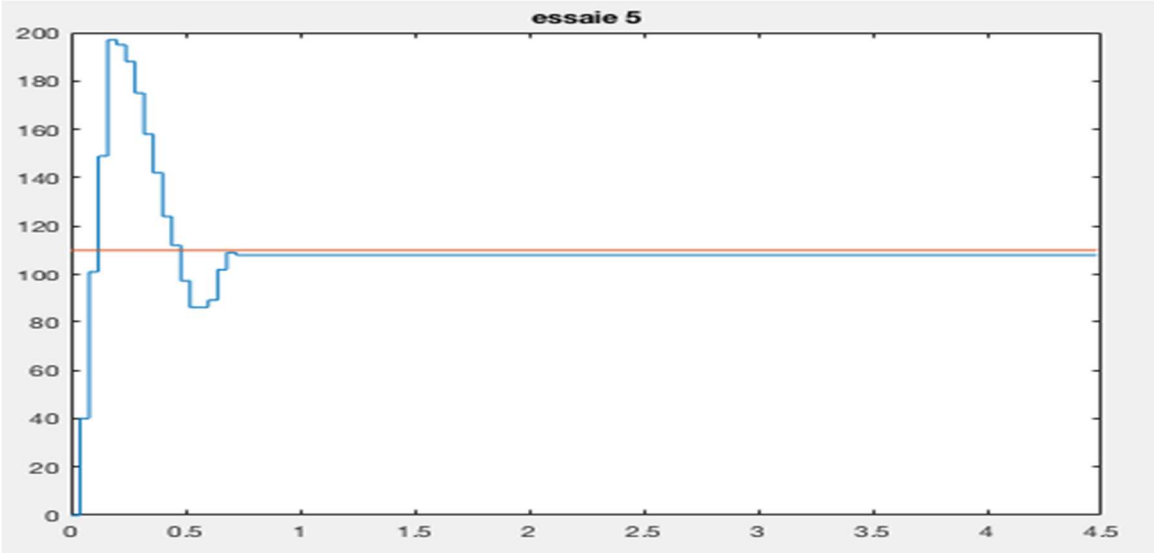
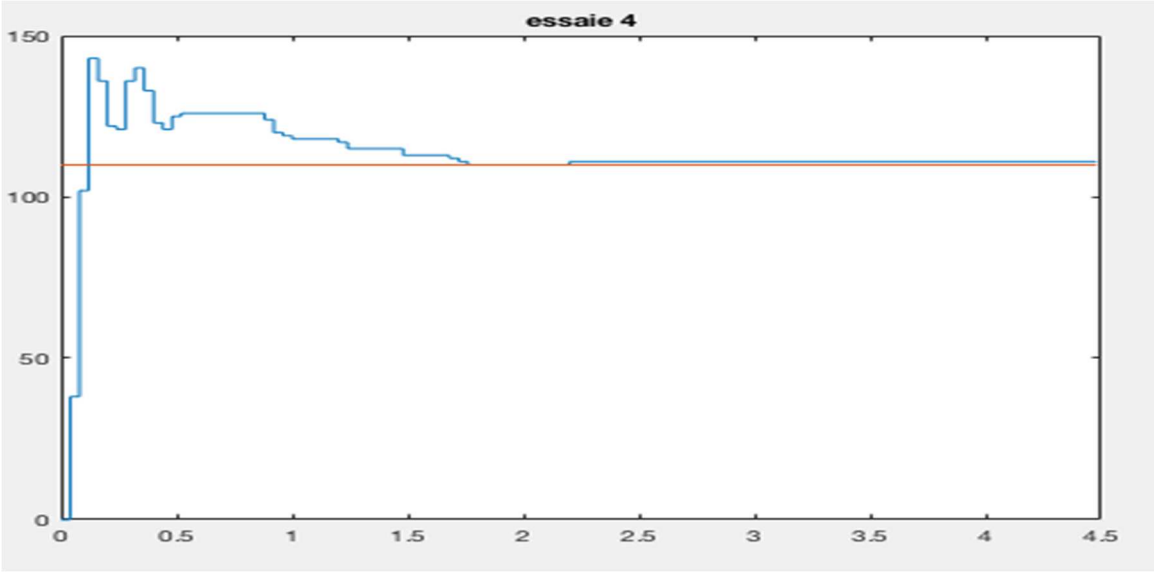
3.4 testes et resulta

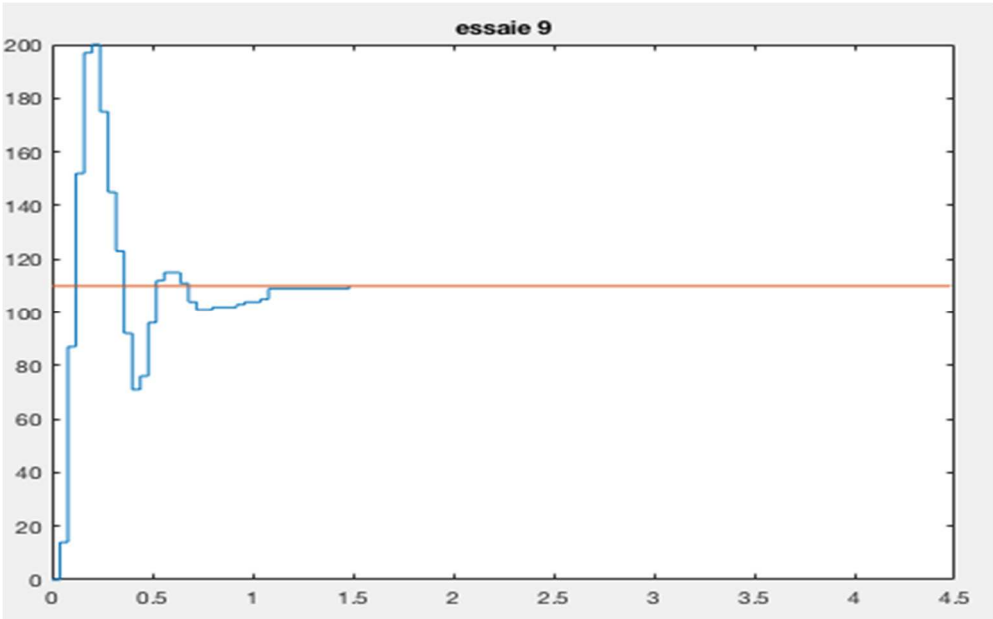
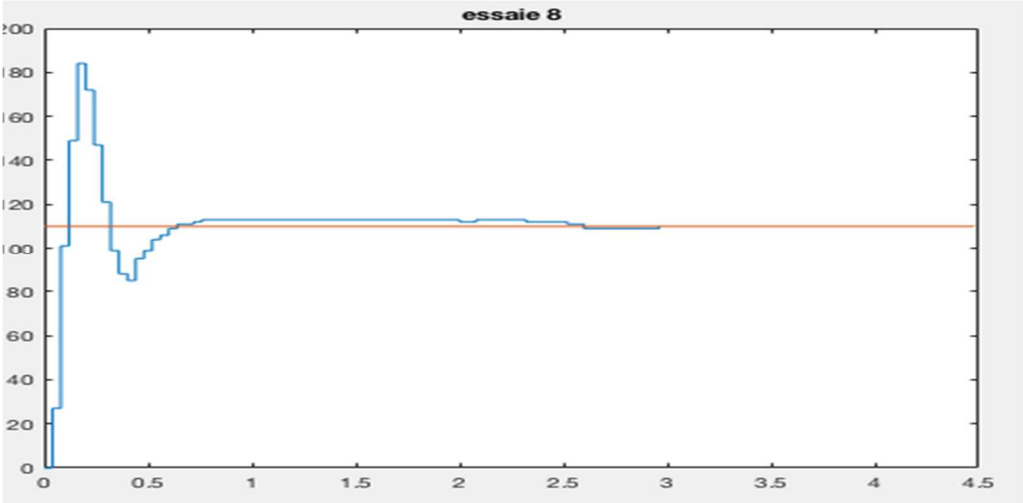
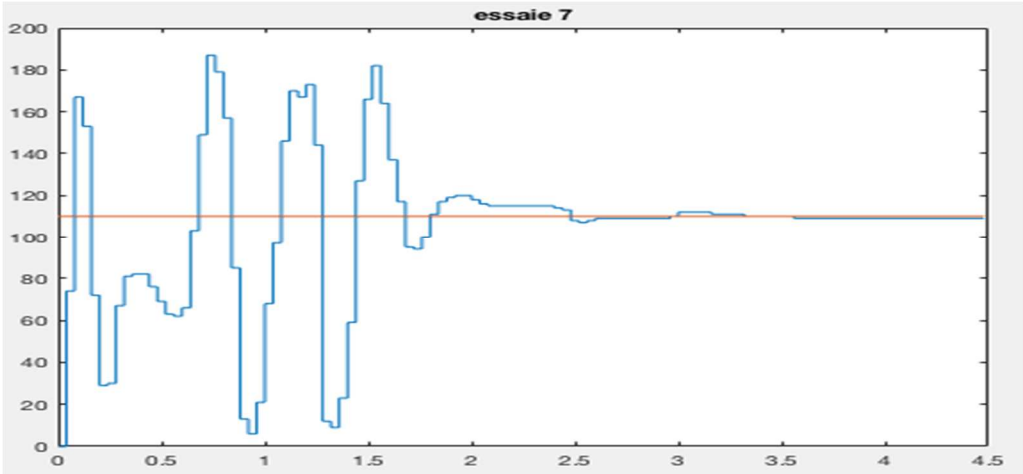
	P	I	D
Essaie 1	0.15	0.6	0.0035
Essaie 2	0.15	0.6	-
Essaie 3	0.15	0.6	0.005
Essaie 4	0.15	0.6	0.01
Essaie 5	0.15	0.15	0.0035
Essaie 6	0.5	0.6	0.0035
Essaie 7	0.15	0.4	-
Essaie 8	0.15	0.1	-
Essaie 9	0.2	0.1	-
Essaie 10	0.2	-	-
Essaie 11	0.05	-	-

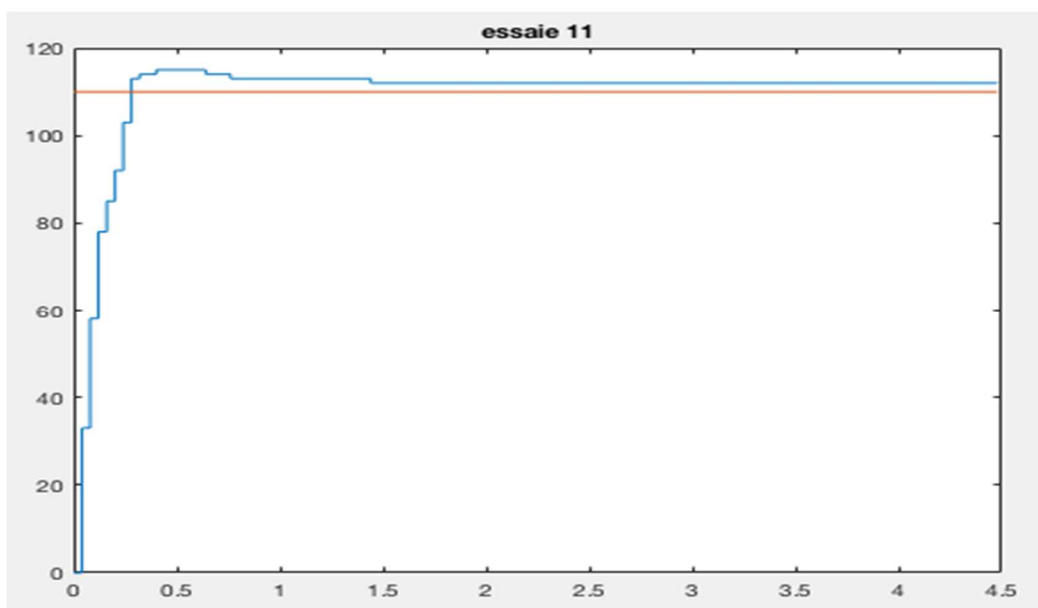
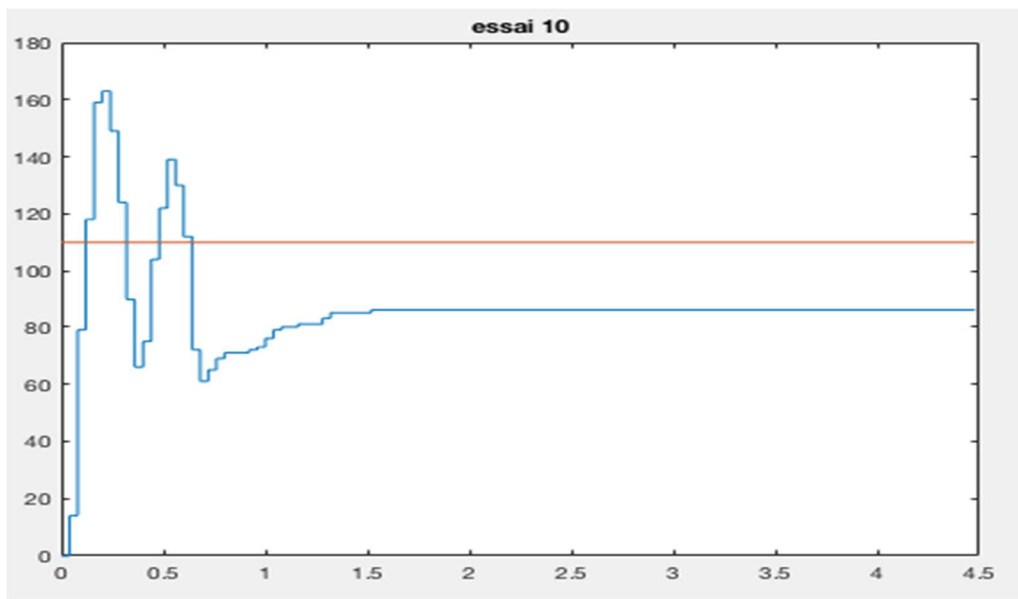
fig (3.25) Les différentes valeurs des gains d'un PID

Chapitre III : Etude et réalisation d'un système de détection et le suivi d'une personne









On voit clairement l'influence de chaque action:

- Le proportionnel est plus précis, plus rapide mais instable.
- L'intégrateur est précis mais instable, et lent
- Le dérivateur est stable, rapide m moins précis.

3.5 Conclusion

Après une description software du projet, où on a détaillé l'algorithme de détection et de suivi d'une personne. Nous sommes passés à la partie hardware, qui consiste à présenter et expliquer le fonctionnement du matériel (servomoteur, ARDUINO) ainsi que la liaison série entre Matlab et le logiciel ARDUINO. Enfin nous avons procédé à la présentation des tests et les résultats du système de PID.

Conclusion générale

Notre travail a consisté à étudier la détection et le suivi d'une personne (suivi par détection) tant sur le plan théorique que pratique en utilisant une caméra fixée à un système articulé à deux axes. Le fonctionnement de cet ensemble est piloté par une carte de développement ARDUINO qui consiste à commander ce système articulé monté sur deux servomoteurs dont l'angle de rotation est limité à 180° qui ont été modifiés de telle sorte à avoir deux moteurs à rotation continue dont l'objectif principal est de suivre, en temps réel, la personne détectée et qu'elle soit toujours dans le champ de vision de la caméra.

Pour atteindre cet objectif, nous avons subdivisé notre travail en trois étapes. La première étape est consacrée à la liaison du logiciel Matlab avec ARDUINO pour échanger les informations (envoyer le centre de la personne à suivre). Quant à la deuxième étape elle a été entamée en programmant grâce au logiciel Matlab, le programme de détection et de suivi d'une personne, qui se base sur deux algorithmes, l'algorithme de Viola et Jones pour la détection de la personne, et l'algorithme de Kanade-Lucas-Tomasi pour le suivi.

Enfin la dernière étape a été consacrée à programmer l'asservissement en position des moteurs sur le mouvement de rotation du système articulé en introduisant un PID pour qu'il puisse atteindre sa position avec exactitude.

Des essais ont été menés afin d'ajuster les valeurs du PID et des résultats satisfaisants ont été obtenus.

Bibliographie

- [1] Iratni Salim.2014. Asservissement visuel, étude et réalisation d'un dispositif de suivi d'un objet en mouvement. Université Mouloud Mammeri de Tizi Ouzou.
- [2] Asservissements visuels, Notes de cours. Jacques Gangloff, jacques@eavr.u-strasbg.fr 5 novembre 2004
- [3] Odile BOURQUARDEZ. (2004). Asservissement visuel et localisation d'un hélicoptère miniature. Campus universitaire de Beaulieu .35042 RENNES Cedex
- [4] Hocini OULHADJ. 2017. Etude et réalisation d'un système de détection et suivi d'un objet en mouvement. Université Mouloud Mammeri de Tizi Ouzou.
- [5] Yedjour Hayat. Détection de contours et suivi d'objet dans une séquence d'images par les réseaux de neurones impulsionnels. Université Mohamed Boudiaf (USTO) ORAN.
- [6] BERKANE Chahrazed. 2014. La détection des visages. Mémoire de fin d'étude en vue de l'obtention du diplôme Master en informatique, Université Larbi Ben M'hidi Oum El Bouaghi.
- [7] KH.Bouchra, « Mise au point d'une application de reconnaissance faciale », mémoire de Mastère, université Abou Beker Belkaid Telemcen, Novembre 2013. G.Pierre, université de Strasbourg France« TP la méthode Viola et Jones »,2013
- [8] M. Houda , « Reconnaissance et mappage des expression faciale vers un modèle virtuel», Mémoire de Mastère, université de sousse,2012.
- [9] TH. Ephraim,T.Himmelman, « Optimizing Viola-Jones face détection for use in webcams», conférence

Bibliographie

- [10] H. Vietman, « reconnaissance des visages sur le téléphone portable », mémoire de Mastère, institut de la francophonie pour l'informatique(IFI), Octobre 2010.
- [11] A. Negri Pablo, P. Lionel, C. Xavier, « Cascade de classifieur génératifs et discriminants pour la détection de véhicules », Article, institut des système intelligents et robotique, université Pierre et Marie Paris 6
- [12] MAHI Abdelhakim. 2018.Détection de visage par l'algorithme de boosting. Mémoire d'un master, Université Aboubakr belkaid –Tlemcen.
- [13] MEDJAHED Fatiha. 2012, Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images. Mémoire en vue de l'obtention du diplôme de Magister. Université des Sciences et de la Technologie d'Oran.
- [14] ARDUINO UNO. Datasheet, www.arduino.cc