

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'Etudes
de MASTER ACADEMIQUE**
Spécialité : **commande des systèmes**

Présenté par
Salim IRATNI

Mémoire dirigé par **Pr. Moussa DIAF**

Thème

**Asservissement visuel : étude et
réalisation d'un dispositif de suivi d'un
objet en mouvement**

Mémoire soutenu publiquement le 17 septembre 2014 devant le jury composé de :

Mme Fadhila BOUDJEMAA

MAA, UMMTO, Présidente

M Moussa DIAF

Professeur, UMMTO, Rapporteur

M Takfarinas CHILLI

MAB, UMMTO, Examineur

Mme Ourida MEGHERBI

MAA, UMMTO, Examinatrice

Avant-propos

Ce mémoire a été effectué au laboratoire de recherche « LVAAS, Laboratoire de Vision Artificielle et Automatique des Systèmes » du département Automatique, FGEI, UMMTO.

Nos vifs remerciements vont tout d'abord à Monsieur DIAF Moussa, professeur à l'UMMTO pour nous avoir proposé le thème de ce mémoire et m'avoir dirigé, aidé et conseillé tout au long de notre travail.

Nous ne manquerons pas de remercier Madame Boudjemaa Fadhila MAA 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 Megherbi Ourida, 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, MAB au département Automatique, FGEI, UMMTO pour avoir accepté de faire partie du jury d'examen de notre mémoire.

Aussi nous ne manquerons pas de présenter nos profonds remerciements à Monsieur LALAMI Salim et à Monsieur DOUALI Faride de nous avoir conçu bénévolement la conception mécanique de ce projet.

Sommaire

Introduction générale	1
------------------------------	---

Premier chapitre : Généralités sur l'asservissement visuel

1.1 Introduction	4
1.2 Capteurs et asservissement visuel	5
1.3 Type de commande	6
1.4 Principe de l'asservissement Visuel	7
1.4.1 Asservissement visuel basé image	8
1.4.1 Asservissement visuel basé espace	8
1.4.1 Asservissement visuel hybride	9
1.5 Les fondamentaux de l'asservissement visuel	9
1.6 Loi de commande	12
1.7 Applications de l'asservissement visuel	16
1.8 Objectifs de notre projet	17
1.9 Conclusion	18

Deuxième chapitre : Détection de mouvement

2.1 Introduction	19
2.2 Terminologie en détection de mouvement	19
2.2.1 Détection de mouvement	20
2.2.2 Estimation de mouvement	20
2.2.3 Modélisation de l'arrière-plan	20
2.2.4 Soustraction de l'arrière-plan	21
2.2.5 Segmentation de mouvement	22
2.3 Méthodes de détection de mouvement	22
2.3.1 Détection sans modélisation de l'arrière-plan	22
2.3.2 Modélisation locale de l'arrière-plan	22
2.3.3 Modélisation semi-locale de l'arrière-plan	23
2.3.4 Modélisation globale de l'arrière-plan	23
2.4 Détection sans modélisation	23
2.4.1 Dérivée temporelle	24
2.4.2 Entropie spatio-temporelle	26

2.4.3 Flot optique	27
2.5 Détection par modélisation locale	28
2.5.1 Modélisation par image	28
2.5.2 Modélisation statistique	29
2.5.3 Modélisation prédictive	29
2.6 Conclusion	30
Troisième chapitre : Etude et réalisation d'un système de détection et de suivi d'objets en mouvements	
3.1 Introduction	31
3.2 Détection d'un objet en mouvement	32
3.3 Réalisation électrique	35
3.3.1 Moteur à courant continu	35
3.3.2 Le circuit de puissance	44
3.3.3 Carte ARDUINO UNO	47
3.3.4 Adaptation de la camera en mode nocturne	48
3.3.5 Alimentation	48
3.3.6 Branchement ARDUINO	49
3.4 Conception de la partie mécanique	50
3.5 Commande des moteurs	51
3.5.1 Calibration du pixel fournie par la camera	52
3.5.2 Loi de commande	53
3.5.3 Exemple illustratif	56
3.6 Partie software du projet	58
3.7 Conclusion	62
Conclusion générale	63
Annexes	
Bibliographie	

Table des figures

Fig.1.1 Configurations de la caméra	5
Fig.1.2 Boucle d'asservissement visuel direct	6
Fig.1.3 Boucle d'asservissement visuel indirect	6
Fig.1.4 Représentation du problème de l'asservissement visuel	7
Fig.1.5 Schéma de l'asservissement visuel 2D	8
Fig.1.6 Schéma de l'asservissement visuel 3D	9
Fig.1.7 Schéma d'un asservissement 2D1/2	9
Fig.1.8 Perception pour agir	10
Fig.1.9 Schéma bloc de l'asservissement visuel	11
Fig.1.10 Diagramme explicatif de la partie Perception et Action.	11
Fig.1.11 Modèle de la projection perspective	14
Fig.1.12 Schéma de principe du système à commander	17
Fig.2.1 Détection de mouvement	20
Fig.2.2 Estimation de mouvement par flot optique	20
Fig.2.3 Modélisation d'arrière-plan	21
Fig.2.4 Détection d'un objet en mouvement par soustraction d'AP	21
Fig.2.5 Segmentation par Camshift	22
Fig.2.6 Méthodes de détection de mouvement	23
Fig.2.7 Détection de mouvement par dérivée temporelle	25
Fig.2.8 Détection de mouvement par lissage temporel de la séquence	26
Fig.2.9 Détection de mouvement par Entropie spatio-temporelle	27
Fig.2.10 Modélisation d'arrière-plan par une image dépourvue d'objet	29
Fig.3.1 Etape de soustraction	32
Fig.3.2 Etape de binarisation	33
Fig.3.3 Etape de traitement morphologique	33
Fig.3.4 Calcul du centre de gravité	33
Fig.3.5 Algorithme pour la détection de mouvement	34
Fig.3.6 Moteur à courant continu avec réducteur	35
Fig.3.7 Schéma du circuit RC	36
Fig.3.8 Diagramme de Bode du filtre passe-bas	37
Fig.3.9 Tension générée par le capteur de vitesse avant et après filtrage	38
Fig.3.10 Schéma fonctionnel du moteur avec réducteur	40
Fig.3.11 Tension d'alimentation et vitesse mesurée de la table	41
Fig.3.12 Réponse indicielle mesurée	43
Fig.3.13 Réponse indicielle mesurée	44
Fig.3.14 Circuit intégré L293D	44
Fig.3.15 L293D pins	45
Fig.3.16 Description de la carte ARDUINO UNO	48

Fig.3.17 Camera adapté en mode nocturne	48
Fig.3.18 Alimentation externe	49
Fig.3.19 Branchement ARDUINO avec le L293D	49
Fig.3.20 Branchement ARDUINI avec le joystick	50
Fig.3.21 Schéma basique du système mécanique	50
Fig.3.22 Simulation 3D de la solution proposé par le bureau d'étude	51
Fig.3.23 Boucle d'asservissement visuel	52
Fig.3.24 Méthode de détermination des paramètres du champ de vision	52
Fig.3.25 Angles du champ de vision	53
Fig.3.26 Signal d'entre	53
Fig.3.27 Tracé des points du Tab3.2	54
Fig.3.28 Tracé de la droit $\nabla\alpha(T)$	55
Fig.3.29 Exemple de commande d'un traitement	56
Fig.3.30 Algorithme pour générer un signal de commande	57
Fig.3.31 Interface du projet	59
Fig.3.32 Parties conçu dans ce projet	61

Table des tableaux

Tab.3.1 Configurations de fonctionnement du L293D	46
Tab.3.2 Tableau de mesure entre la longueur de l'imputation et l'angle	53

Table des abréviations

2D : deux dimensions

3D : trois dimensions

AP : arrière-plan

DSTEI : difference-based spatial temporal entropy image

STEI : spatial temporal entropy image

Tps : traitement par seconde

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, ce qu'il a commencé à se réaliser avec l'émergence de la robotique au cours des années 1960.

Depuis leur apparition, les robots industriels réalisent 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 aller vers des systèmes avec capteurs de vision, afin de lui permettre d'appréhender son environnement et de réagir à d'éventuels changements de ce dernier. Ceci 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'autre part. L'asservissement visuel correspond à un asservissement d'un robot où les grandeurs asservies sont mesurées à partir 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. La réalisation d'un asservissement visuel implique la mise en œuvre d'au moins trois sous-systèmes : un capteur visuel, un dispositif d'acquisition/traitement, et un système articulé. Le capteur visuel est souvent une caméra, mais peut également être tout type de dispositif imageur (scanner, IRM, caméra linéaire, etc.). Le dispositif d'acquisition/traitement est un système capable de traiter l'information visuelle numérisée, autrement dit un ordinateur. Souvent, c'est le même système qui est chargé du traitement de l'image et de la commande. Cette commande qui résulte du retour visuel est envoyée au système articulé.

Un autre domaine aussi qui a fait un objet de recherche durant ces trente dernières années et qui reste toujours un domaine où de nouvelles approches sont à prospecter est le domaine de détection de mouvements. Ce dernier généralement passe par une étape de modélisation d'arrière-plan pour lequel plusieurs techniques ont été mises en point.

Le travail réalisé dans ce mémoire s'inscrit dans le domaine de détection de mouvements et d'asservissement visuel. L'objectif consiste à utiliser la configuration caméra déportée et la technique Look and Move statique pour réaliser un asservissement visuel 2D d'un système articulé. En effet, le travail se limite à l'étude, sur le plan théorique et pratique, de la commande d'un système comportant une caméra, un système articulé qui pivote la caméra sur axes et un ordinateur connecté à une carte d'acquisition et de commande qui est une ARDUINO UNO. Le fonctionnement de cet ensemble consiste à détecter en temps réel, un objet en mouvement dans le champ de vision de la caméra et guider cette dernière en temps réel de telle manière à ce que cet objet soit toujours suivi par l'organe visuel et de le placer au milieu du champ de vision. Ainsi, une conception mécanique est nécessaire dans ce projet. Pour réaliser la commande en position de cette caméra, trois parties ont été effectuées, dans la première, nous avons calculé la fonction de transfert des moteurs sur la base des considérations mécaniques et électriques. Puis, nous avons calculé une loi de commande assurant un fonctionnement performant des moteurs. La réalisation pratique d'une commande de ce dernier est abordée dans la dernière partie. En second lieu, nous nous sommes intéressés à la partie vision artificielle où nous avons réalisé le programme permettant l'acquisition, en temps réel, des images délivrées par la caméra pour détecter un objet en mouvement.

Le mémoire est organisé comme suit :

Le premier chapitre présente des généralités sur l'asservissement visuel et les lois de commande liées à ce dernier ainsi que le cahier des charges de notre travail.

Le second chapitre est dédié à la détection de mouvements.

Le troisième chapitre expose la partie étude et réalisation du dispositif.

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.

Premier chapitre

Généralités sur l'asservissement visuel

1.1 Introduction

L'asservissement visuel est un asservissement dont la boucle de réaction est constituée d'un système de vision artificielle. Ainsi, en robotique, l'asservissement des articulations, 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 nouveau domaine permet aux robots de d'exécuté de nouvelles tâches comme celles non connues au préalable mais qui 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 un environnement par rapport à cette caméra.

1.2 Capteurs et asservissement visuel

Rendre les robots autonomes nécessite un bon choix de capteurs leur fournissant des informations sur leurs états et leur environnement. L'utilisation de caméras s'avère extrêmement utile pour réaliser des tâches de localisation et d'exploration de l'environnement. Nous les retrouvons dans les robots manipulateurs, la robotique médicale utilisée surtout en chirurgie, en micro-robotique, dans les robots d'aide aux personnes handicapées etc. Les capteurs de vision nous fournissent des images 2D ou 3D. Différentes configurations peuvent être envisagées pour commander un robot par asservissement visuel.

- **Classification des asservissements visuels**

Dans les asservissements visuels, on peut distinguer les trois cas selon la configuration robot-caméra, la commande du robot et la manière d'acquérir les informations concernant l'environnement. Ainsi, trois types de configuration robot-caméra peuvent être distingués. Dans la première, la configuration caméra embarquée, la caméra est montée sur l'organe terminal. Dans la configuration caméra déportée, la caméra est externe autrement dit, sans aucune liaison mécanique avec le robot. Dans la configuration mixte, on utilise une caméra portée par l'organe terminal et une autre portée par un autre robot (Fig.1.1).

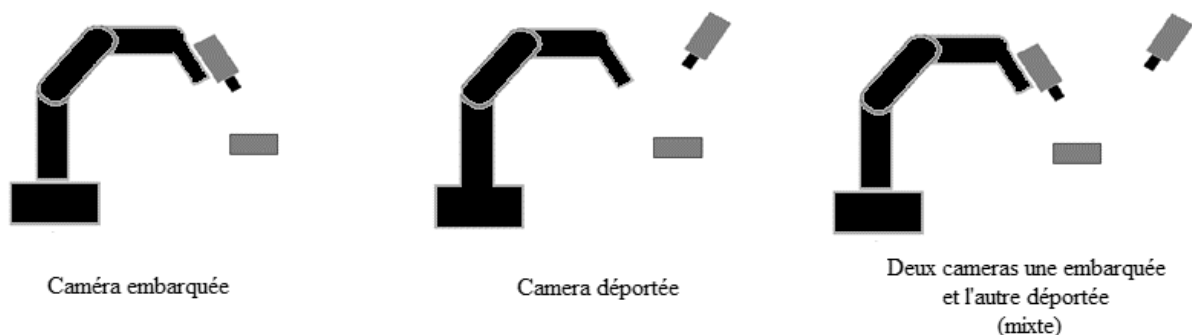


Fig.1.1 Configurations de la caméra

1.3 Types de commande

Les asservissements visuels sont sous deux types l'asservissement visuel direct et l'asservissement visuel indirect.

Dans l'asservissement visuel direct, la commande du robot est réalisée par des commandes générées par le contrôleur de l'asservissement visuel, appliquées directement sur les actionneurs du robot (Fig.1.2). Ce type de commandes est utilisé lorsque la période d'échantillonnage de l'asservissement visuel est grande, cas où l'asservissement visuel est dit lent.

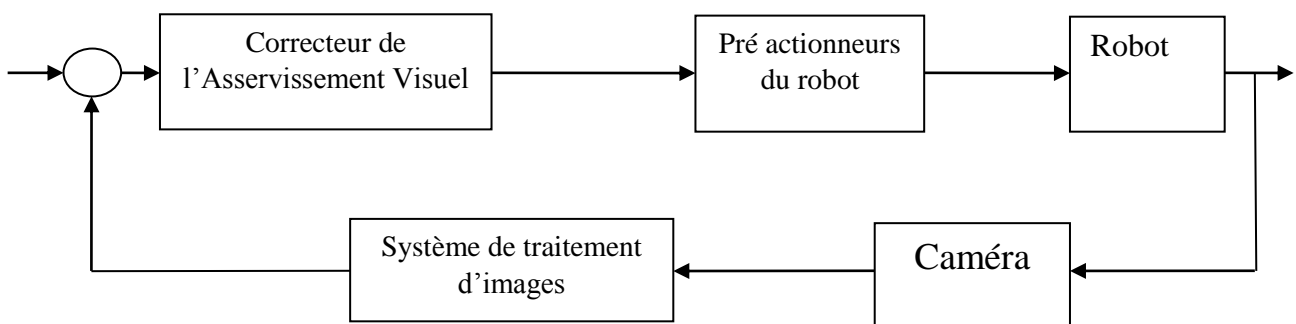


Fig.1.2 Boucle d'asservissement visuel direct

Dans le cas de l'asservissement visuel indirect, les commandes générées par le contrôleur de l'asservissement visuel sont appliquées à un contrôleur intermédiaire (contrôleur du robot) pour générer les commandes articulaires du robot (Fig.1.3). Ce type de commande est utilisé pour les asservissements visuels rapides.

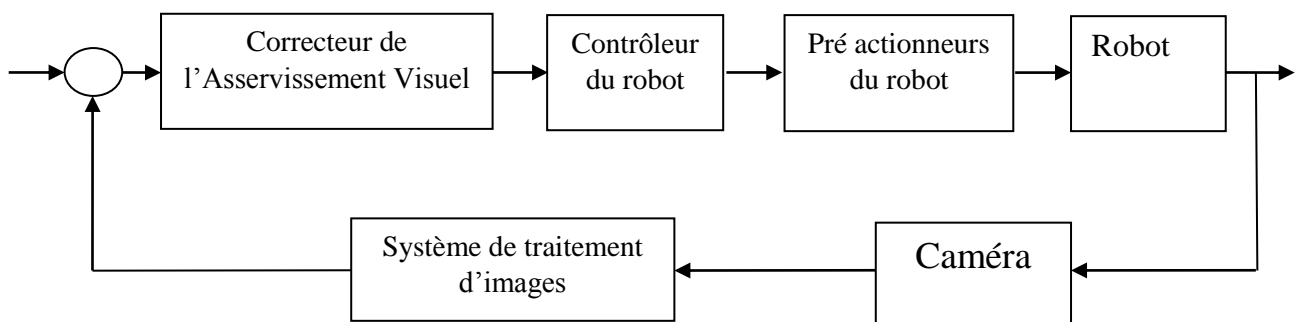


Fig.1.3 Boucle d'asservissement visuel indirect

1.4 Principe de l'asservissement Visuel

Les techniques de l'asservissement visuel consistent à utiliser les informations fournies par une caméra afin de contrôler le mouvement d'un système robotique. Pour ce faire, il s'agit de sélectionner au mieux un ensemble de mesures, permettant de contrôler les degrés de liberté souhaités et d'élaborer une loi de commande afin que ces mesures atteignent une valeur désirée. Le principe de la commande se ramène à annuler le vecteur d'erreur entre mesure et consigne, c'est-à-dire, l'erreur entre les valeurs courante et désirée d'informations visuelles choisies. On peut définir cette erreur, notée $e(t)$, comme étant la différence entre descripteur $s(r(t))$ mesuré à la position r et le descripteur de consigne s^* (Fig.1.4).

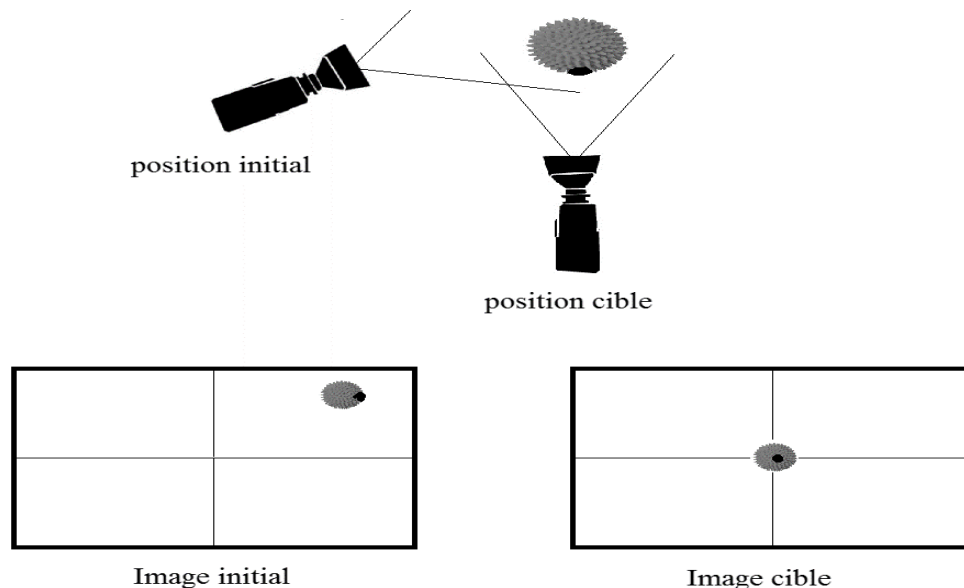


Fig.1.4 Représentation du problème de l'asservissement visuel. (Le but est de déterminer la loi de commande qui va permettre d'amener l'image d'une position initiale vers la position cible)

Nous distinguons trois grandes familles d'asservissement visuel :

- Asservissement visuel basé image : Asservissement Visuel 2D
- Asservissement visuel basé espace : Asservissement Visuel 3D
- Asservissement visuel hybride : Asservissement Visuel 2D1/2

1.4.1 Asservissement visuel basé image

Dans le cas de l'asservissement visuel 2D, comme son nom l'indique, la loi de commande est élaborée à partir des informations visuelles exclusivement extraites de l'image 2D. La consigne est alors exprimée comme la différence entre un motif courant et un motif désiré dans l'espace image. Ces primitives peuvent être des points, des segments de droites, des droites, des cercles, ou tout autre type ou combinaison d'informations géométriques extraites de l'image. Le schéma de cet asservissement est donné à la figure 1.5.

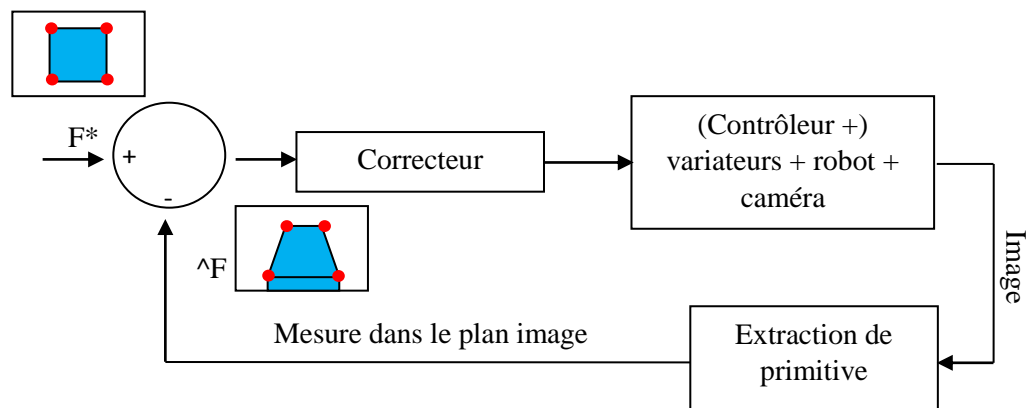


Fig.1.5 Schéma de l'asservissement visuel 2D

1.4.2 Asservissement visuel basé espace

En ce qui concerne l'asservissement visuel de type 3D (Fig.1.6), pendant longtemps, les chercheurs l'ont considéré comme nécessitant obligatoirement une étape de reconstruction 3D de la scène pour estimer la mesure 3D. Cette opération prend beaucoup de temps calcul et est très sensible aux erreurs de calibrage. Ainsi, par la suite, l'apparition des méthodes d'estimation de la pose 3D sans étape de reconstruction 3D ou de systèmes de vision distribués (2 caméras, voire plus), a apporté un deuxième souffle à l'asservissement visuel 3D. Ces méthodes sont capables d'estimer la pose 3D en utilisant une seule caméra et à des cadences élevées.

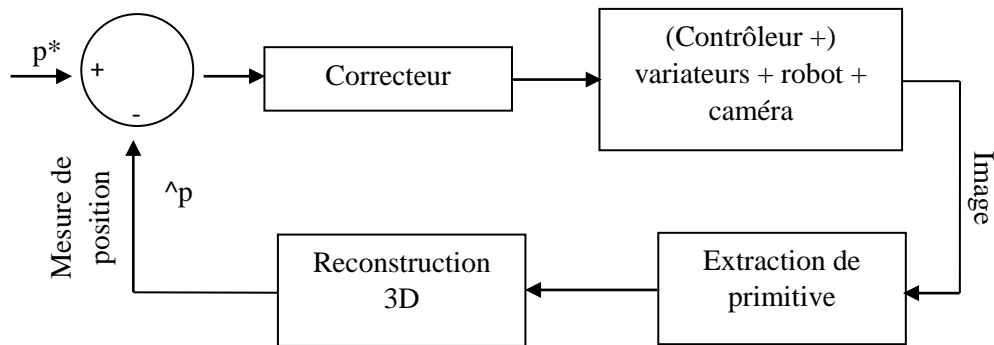
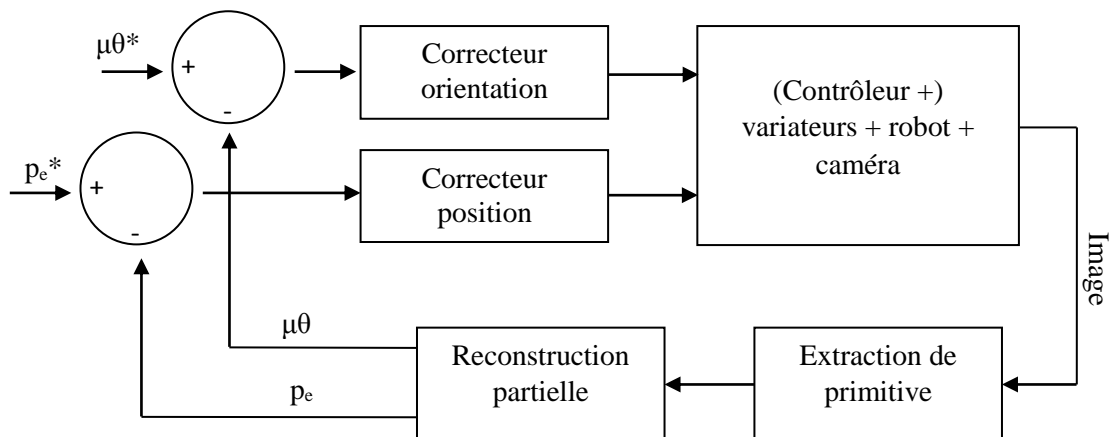


Fig.1.6 Schéma de l'asservissement visuel 3D

1.4.3 Asservissement visuel hybride

Dans la nouvelle approche d'asservissement caractérisé de $2D_{1/2}$ développée par E. Malis dans le cadre de sa thèse [13], on utilise, à la fois, des informations directement exprimées dans l'image 2D pour l'asservissement de la position de l'organe terminal et des informations exprimées dans le repère de la caméra 3D pour asservir l'orientation de l'organe terminal. Les deux boucles d'asservissement sont découplées (Fig.1.7).

Fig.1.7 Schéma d'un asservissement $2D_{1/2}$

1.5 Les fondamentaux de l'asservissement visuel

Pour un robot qui suit une cible, la faculté de perception visuelle permet à un robot de s'informer sur l'état de son environnement et de s'adapter aux possibles variations de ce dernier. Le mouvement d'un robot équipé d'une caméra peut être commandé à partir de sa perception visuelle en

utilisant la technique d'asservissement visuel. L'asservissement visuel consiste à utiliser la perception pour l'action dans une boucle fermée, comme le montre la figure 1.8

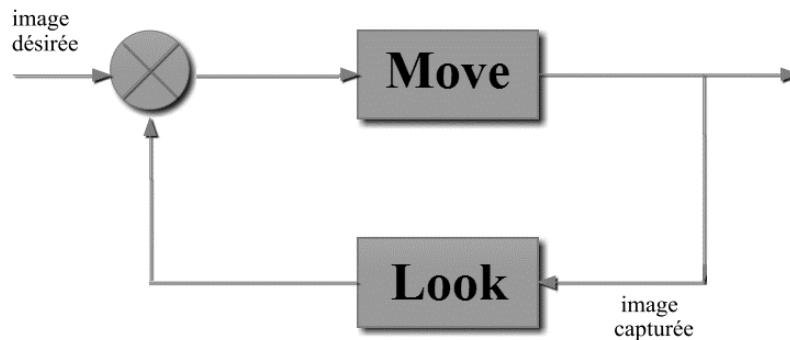


Fig.1.8 Perception pour agir

La perception visuelle d'un objet peut être définie comme l'abstraction d'un ensemble de mesures visuelles sur l'objet cible. Par exemple, si nous voulons essayer de déterminer les coordonnées d'un point de l'image, la perception se traduit par la position de l'objet par rapport au robot. L'action, quant à elle, peut être vue comme un mouvement de robot pour effectuer une tâche. La perception pertinente de l'environnement du robot est un facteur déterministe. Atteindre cette perception pertinente, revient à une modélisation d'informations visuelles adéquates, c'est-à-dire, permettant au robot de s'approcher d'un comportement idéal pendant une tâche. La question qui se pose est comment exploiter judicieusement la grande quantité d'informations qu'ils captent afin d'améliorer le comportement d'un robot commandé par asservissement visuel ?

- **Schéma fonctionnel de l'asservissement visuel**

La boucle d'asservissement visuel peut être représentée par le schéma bloc de la figure 1.9

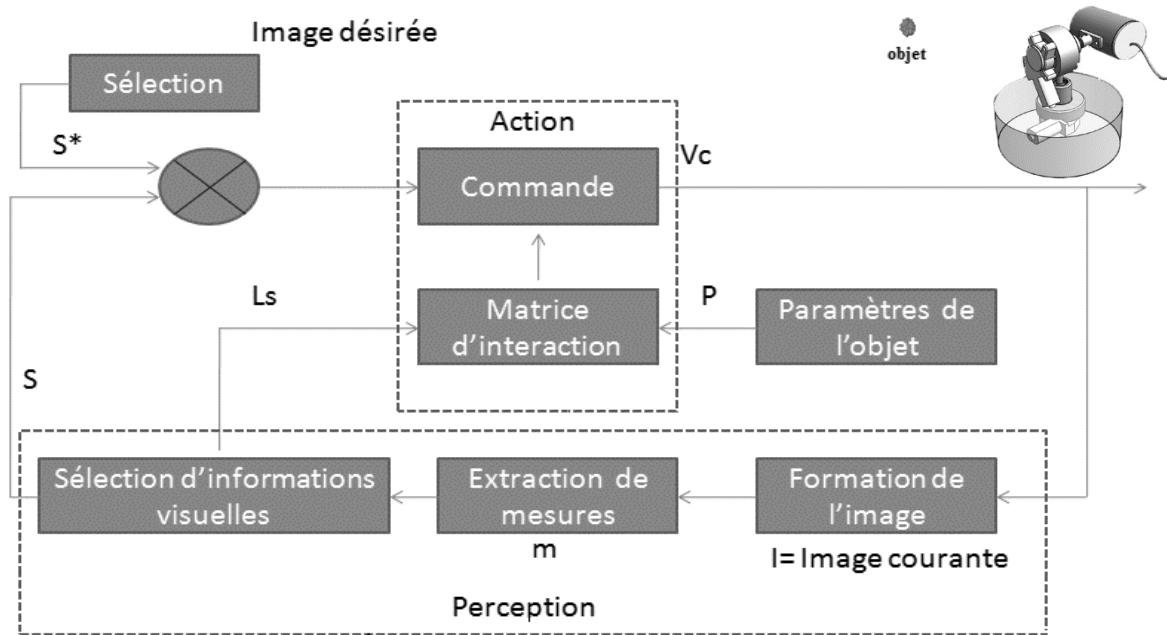


Fig.1.9 Schéma bloc de l'asservissement visuel

Comme il est indiqué sur la figure (Fig.1.9), la boucle d'asservissement visuel est composée essentiellement de deux parties à savoir la partie Perception (Look) et la partie action (Move) (Fig. 1.10).

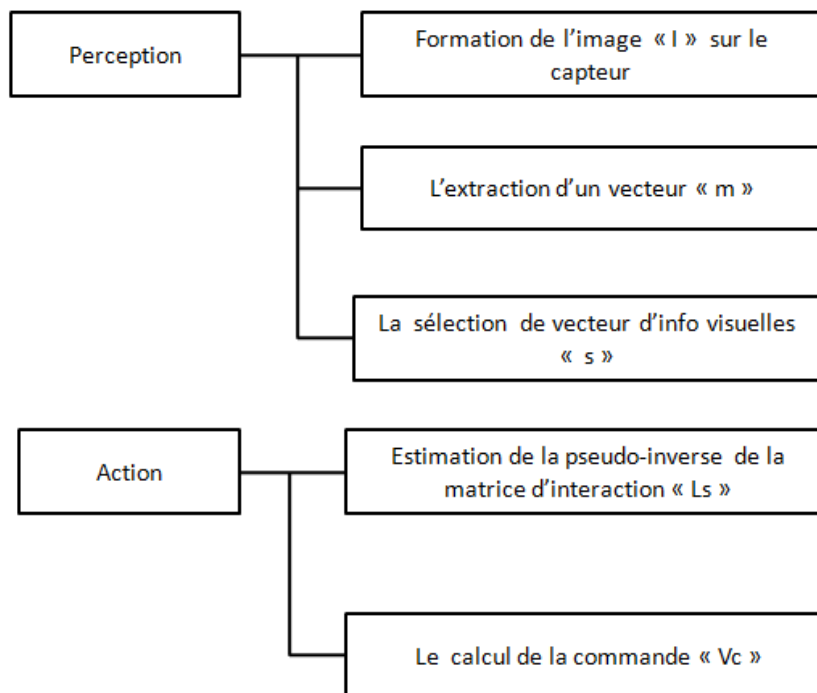


Fig.1.10 Diagramme explicatif de la partie Perception et Action

La matrice « l_s » est fondamentale en asservissement visuel car elle représente la relation vision-commande. Dans la suite, nous décrivons comment la matrice « l_s » intervient dans le calcul de la commande « v_c », ce qui nous permet de définir les propriétés souhaitables de la matrice « l_s » et par conséquent de définir des critères pour choisir le vecteur « s » d'informations visuelles idéal.

1.6 La loi de commande

Le but de la commande est de déplacer le robot suivant une consigne qui peut être fixée dans l'image. La tâche visuelle consiste à amener le robot d'une position initiale vers une position désirée. Il est possible d'exprimer cette tâche robotique comme la régulation de la fonction $e(r(t))$, r étant la position du robot. Cette fonction est donnée par la relation (1.1)

$$e(r(t)) = s(r(t)) - s^* \quad (1.1)$$

Où $s: SE(3) \rightarrow R^k$ et $r(t) \rightarrow e(r(t))$, une application différentiable définie sur un ensemble de « k » informations visuelles sélectionnées à la position $r(t)$.

Sachant que $SE(3) = R^k \times SO(3)$, groupe de Lie de déplacement et s^* , consigne sur les informations visuelles, nous considérons que s^* est constant.

Puisque « s » est différentiable alors « e » est aussi différentiable et nous obtenons la relation suivante (1.2) :

$$\dot{e} = \frac{\partial e}{\partial r} \dot{r} = \frac{\partial s}{\partial r} \dot{r} = l_s v_c \quad (1.2)$$

Où l_s est une matrice d'interaction.

Cette matrice décrit la variation temporelle des informations visuelles « s » dues à un mouvement de la caméra. Le vecteur « v_c » est le torseur cinématique formé par les vitesses de translations et de rotation de la

caméra. $v_c(\mathbf{v}, \boldsymbol{\omega})$ Avec $\mathbf{v}(v_x, v_y, v_z)$ le vecteur de vitesse de translation et $\boldsymbol{\omega}(\omega_x, \omega_y, \omega_z)$ le vecteur de vitesse de rotations

La sélection de « s » rend la conception du système du contrôle très simple et la conception d'un contrôleur de vitesse nécessite la connaissance de la relation entre la variation du temps de « s » et la vitesse de la caméra [6],.

Cette relation se traduit par l'équation (1.3) :

$$\dot{s} = l_s \cdot v_c \quad (1.3)$$

Pour assurer une régulation exponentielle découplée de la tâche « e », nous imposons que :

$$\dot{e} = -\lambda \cdot e \quad (1.4)$$

Où « λ » est un réel strictement positif, représente le gain de la tâche qui doit être réglé de manière adéquate pour avoir un temps rapide de convergence tout en préservant la stabilité du système.

En injectant la relation (1.2) dans (1.4) on déduit la commande idéale envoyée au robot.

$$v_c = -\lambda \cdot l_s^+ \cdot e \quad (1.5)$$

Où l_s^+ : est le pseudo-inverse de la matrice d'interaction « l_s ».

Ce pseudo-inverse est donné par la relation suivante :

$$l_s^+ = (l_s^* l_s)^{-1} l_s^* \quad (1.6)$$

Où l_s^* est la matrice adjointe de l_s .

Pour « k=6 », si $\det l_s \neq 0$, nous pouvons inverser cette matrice et la relation (1.5) peut s'écrire comme suit :

$$v_c = -\lambda \cdot l_s^{-1} \cdot e$$

Mais pour tout système d'asservissement visuel, il est impossible de connaître parfaitement que ce soit l_s ou l_s^+ .

La matrice d'interaction « l_s » dépend des paramètres de l'objet, ainsi que des paramètres intrinsèques « a » du système de vision et du vecteur « s » de l'information visuelle.

Comme, généralement, nous ne disposons que d'une estimation de « a », il en est de même pour les paramètres de l'objet et pour le vecteur « s ». C'est

pourquoi, en pratique, nous travaillons avec les estimations de ces trois paramètres.

De ce fait, la vitesse envoyée au robot et obtenue de (1.5) s'écrit (1.7) :

$$v_c = -\lambda. \hat{l}_s^+ . e \quad (1.7)$$

Où \hat{l}_s^+ est le pseudo-inverse de l'estimation de la matrice d'interaction.

La matrice d'interaction joue un rôle primordial pour la synthèse des lois de commande. Elle lie le mouvement de la caméra aux variations des primitives visuelles dans l'image. Il y a une proposition d'une méthode générale pour le calcul de la matrice pour toute information visuelle définissable à partir des primitives géométriques paramétrables pour une caméra réalisant une projection perspective (Fig.1.11).

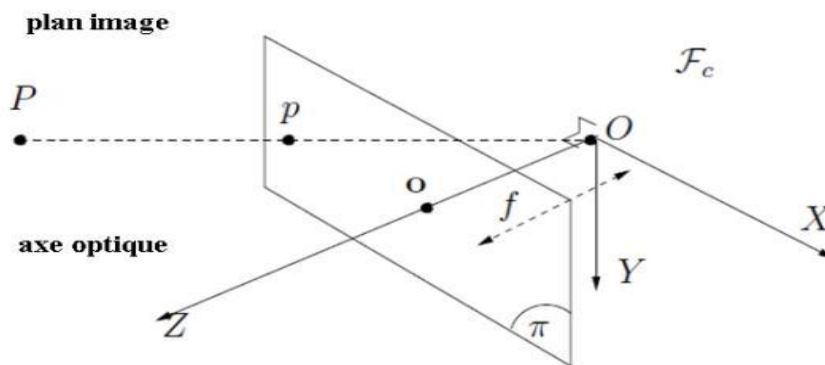


Fig.1.11 Modèle de la projection perspective

Où P est le point dans l'espace, p , le point représentatif du point P dans le plan 2D, O , le centre de la projection, o , le point d'intersection de l'axe optique issue de O avec le plan image, f , la distance focale, \mathcal{F}_c , le repère de référence de la caméra.

Pour le point « P » dans l'espace 3D avec des coordonnées (X, Y, Z) dans le cadre de la caméra, la projection en image comme un point « p » en deux dimensions avec des coordonnées $p = (x, y)$ est donnée par la transformation de la relation (1.8) :

$$\begin{cases} x = \frac{X}{Z} = (u - c_u)/f\alpha \\ y = \frac{Y}{Z} = (v - c_v)/f\alpha \end{cases} \quad (1.8)$$

Le couple (u, v) représente les coordonnées du point de l'image exprimé en pixel.

Soit le paramètre intrinsèque « a » de la caméra. Ce paramètre dépend des différents facteurs dont nous pouvons l'écrire sous la forme suivante (c_u, c_v, f, α) avec c_u et c_v , sont les coordonnées de notre point principal de la projection, f est la distance focale de la caméra et α est le rapport de dimension de pixel. En effectuant la dérivation par rapport au temps des équations de projection de la relation (1.8) nous obtenons le système d'équations (1.9) :

$$\begin{cases} \dot{x} = \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{cases} \quad (1.9)$$

Nous pouvons relier la vitesse du point 3D à la vitesse spatiale de la caméra en utilisant l'équation (1.9) :

$$\dot{X} = -v_c - \omega_c \cdot X \leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = -v_z - \omega_x Y + \omega_y X \end{cases} \quad (1.10)$$

En injectant (1.10) dans (1.9) on obtient le système d'équations (1.11) :

$$\begin{cases} \dot{x} = -\frac{v_x}{Z} + \frac{xv_z}{Z} + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} = -\frac{v_y}{Z} + \frac{yv_z}{Z} - xy\omega_y + (1+y^2)\omega_x - x\omega_z \end{cases} \quad (1.11)$$

Donc nous pouvons déduire que la dérivée du vecteur « X » est liée directement au torseur cinématique « v_c » par la relation suivante :

$$\dot{X} = l_s v_c$$

Comme résultat de la relation précédente, on peut définir la matrice d'interaction « l_s » comme suit :

$$l_s = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1-x^2) & y \\ 0 & -1/Z & x/Z & (1-y^2) & -xy & -x \end{bmatrix} \quad (1.12)$$

Dans la matrice « l_s », la valeur Z est la profondeur du point par rapport au cadre de la caméra. Par conséquent, tout système de contrôle qui utilise cette forme de la matrice d'interaction doit estimer la valeur de Z . Cette estimation doit être faite judicieusement afin de satisfaire nos besoins. De même, les paramètres intrinsèques de la caméra sont impliqués dans le calcul de x et y . D'une manière générale, pour contrôler le degré de liberté « 6 » au moins trois points sont nécessaires donc il suffit d'avoir « $k=6$ ». Pour ce faire trois points sont suffisants pour aborder notre problème. Si nous utilisons le vecteur de caractéristiques $x = (x_1, x_2, x_3)$, par un simple empilement des matrices d'interaction pour trois points, nous obtenons :

$$l_s = \begin{pmatrix} l_{s1} \\ l_{s2} \\ l_{s3} \end{pmatrix}$$

Comme nous avons dit précédemment, afin de faciliter l'étude de la matrice d'interaction nous faisons appel à la technique de l'estimation parce que la connaissance exacte de cette matrice est très complexe.

Le but de cette étude est d'améliorer la commande du système à travers le choix des informations visuelles « s » et également le choix du schéma de contrôle.

1.7 Applications de l'asservissement visuel

L'asservissement visuel est utilisé principalement pour réaliser des tâches de positionnement dont l'objectif est de ramener le robot à partir d'une position initiale à une position consigne mesurée par l'utilisation d'une caméra. Il peut être aussi utilisé pour le suivi d'une cible mobile, ce qui consiste à utiliser les images de la caméra pour ramener le robot à une situation d'équilibre par rapport à une cible mobile. Dans le cas de la navigation et de l'évitement d'obstacles par un robot mobile qui réalise un déplacement d'un point A à un point B tout en évitant tout obstacle rencontré sur son chemin, l'asservissement visuel est amplement utilisé.

1.8 Objectifs de notre projet

En ce qui nous concerne, l'application que nous proposons de réaliser dans le cadre de notre projet de fin d'études, elle s'inscrit dans les domaines de la commande numérique, de la robotique et de la vision artificielle. En effet, il consiste à étudier, sur le plan théorique et pratique, la commande d'un système composé d'une caméra numérique adapté en vue nocturne et diurne, un robot à deux axes dont un vertical et l'autre horizontal (2 degrés de libertés). Comme organe terminal de ce robot, un support portant la camera. Le fonctionnement de cet ensemble est piloté par un ordinateur à travers une carte de développement (acquisition et commande) qui est une ARDUINO UNO [1],. Ainsi, application consiste à faire détecter un objet en mouvement dans le champ de vision de la caméra sélectionnée et de suivre cet objet. Le système à commander est représenté en figure (Fig.1.12).

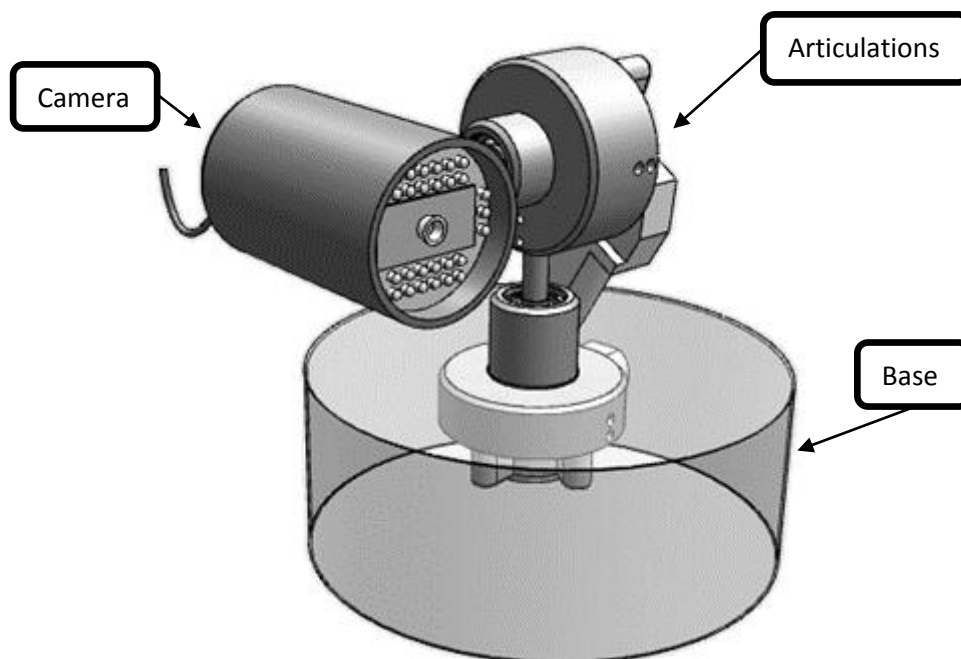


Fig.1.12 Schéma de principe du système à commander

1.9 Conclusion

Ce chapitre a été dédié à l'asservissement visuel. Il est utile de souligner que cette étude est récente et inspire nombreux chercheurs. Les applications relatives à cet axe sont nombreuses et utilisent parfois des concepts très poussés en matière de vision 3D, de navigation, d'évitement d'obstacles, poursuite et bien d'autres encore. Etant donné l'aspect théorique et réalisation pratique de notre travail, nous nous sommes fixés le but de développer un système qui gère le fonctionnement d'une poursuite d'un objet en mouvement pour que ce dernier reste dans le champ de vision de la caméra.

Deuxième chapitre

Détection de mouvement

2.1 Introduction

La plupart des algorithmes de suivi d'objets prenant en entrée les images fournies par une ou plusieurs cameras effectuent une première étape de détection de mouvement afin de déterminer les pixels de l'image courant appartenant à l'arrière-plan de la scène, et représentant des objets mobiles. Ce domaine de recherche est très actif depuis les débuts de l'analyse de séquences d'images à la fin des années 1970. Depuis, le nombre d'articles publiés chaque année sur ce sujet ne cesse de croître particulièrement depuis 1990 lorsque la puissance des ordinateurs a permis d'envisager des traitements en temps réel de données images.

2.2 Terminologie en détection de mouvements

Classer et hiérarchiser les différentes méthodes de détection de mouvement présentes dans la littérature n'est pas une tâche aisée. En effet, les auteurs utilisent souvent une terminologie différente pour désigner des méthodes ou algorithmes qui ont le même objectif. Par ailleurs, un même terme dans un article particulier peut désigner un petit module bien défini d'un algorithme alors que, dans un autre, il désignera la totalité d'un système informatique incluant ce module ainsi que beaucoup d'autres. Dans

ces conditions, il est nécessaire de définir avec précision les termes rencontres afin d'éviter toute confusion [16].

2.2.1 Détection de mouvement

Ce terme est le plus générique. Il indique uniquement que l'on parle d'une méthode qui a pour objet de trouver en quels points de l'image un mouvement a eu lieu. Un algorithme ayant cet objectif fournit en sortie une variable quantitative (quantité de mouvement) ou qualitative (booléenne) pour tout pixel de chaque image d'entrée. Toutes les méthodes présentées rentrent dans cette catégorie (Fig.2.1).

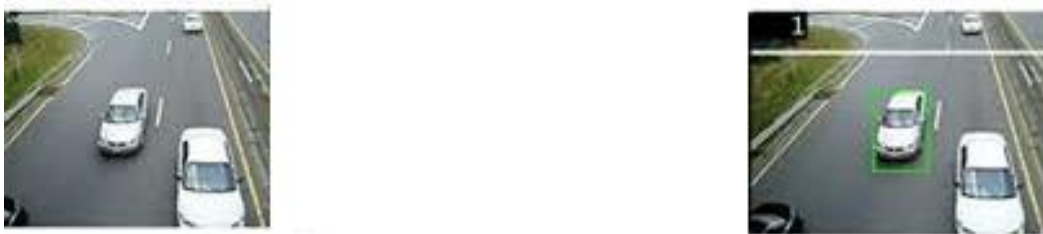


Fig.2.1 Détection de mouvement

2.2.2 Estimation de mouvement

La notion d'estimation de mouvements inclut la précédente en y ajoutant la contrainte que le résultat fourni doit être quantitatif. Ce terme est surtout utilisé pour les auteurs travaillant sur l'estimation du flot optique. Dans ce cas, le résultat de l'estimation est un champ de vecteurs à deux dimensions représentant la projection sur le plan de l'image du mouvement réel tridimensionnel ayant lieu dans la scène (Fig.2.2).



Fig.2.2 Estimation de mouvement par flot optique

2.2.3 Modélisation de l'arrière-plan

Cette catégorie regroupe toutes les méthodes de détection de mouvement qui consistent à créer un modèle de l'arrière-plan de la scène

filmée (sans aucun objet mobile). Ce modèle peut être une image créée à partir des pixels observés à différents instants de la séquence vidéo (Fig.2.3).

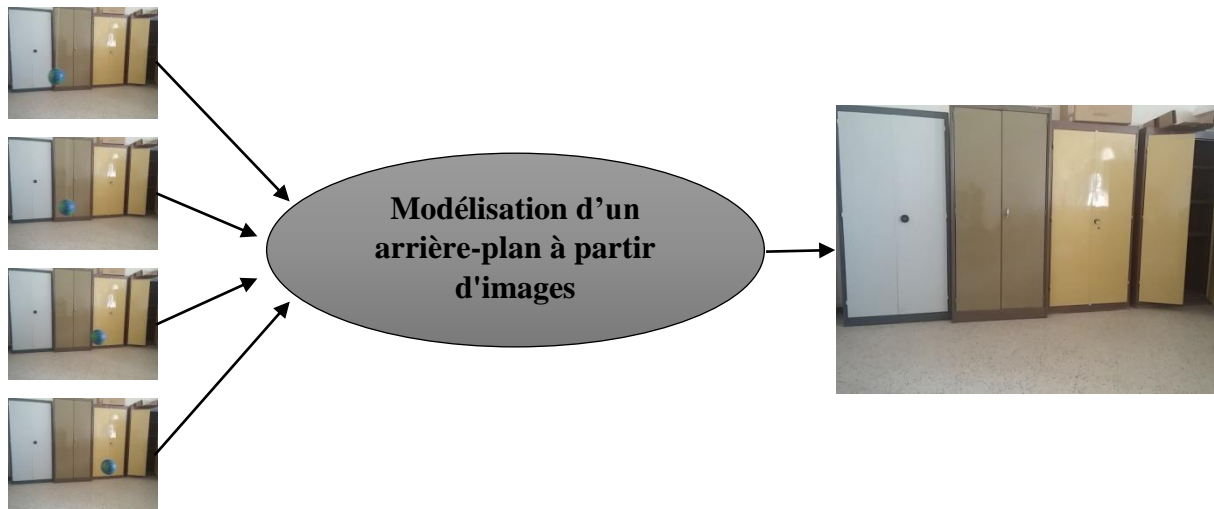


Fig.2.3 Modélisation d'arrière-plan

2.2.4 Soustraction de l'arrière-plan

La soustraction de l'arrière-plan est l'opération qui suit logiquement la modélisation de l'arrière-plan afin d'obtenir une différence en valeur absolue entre ce modèle et l'image courante est effectuée afin d'obtenir une détection de mouvement (Fig.2.4).

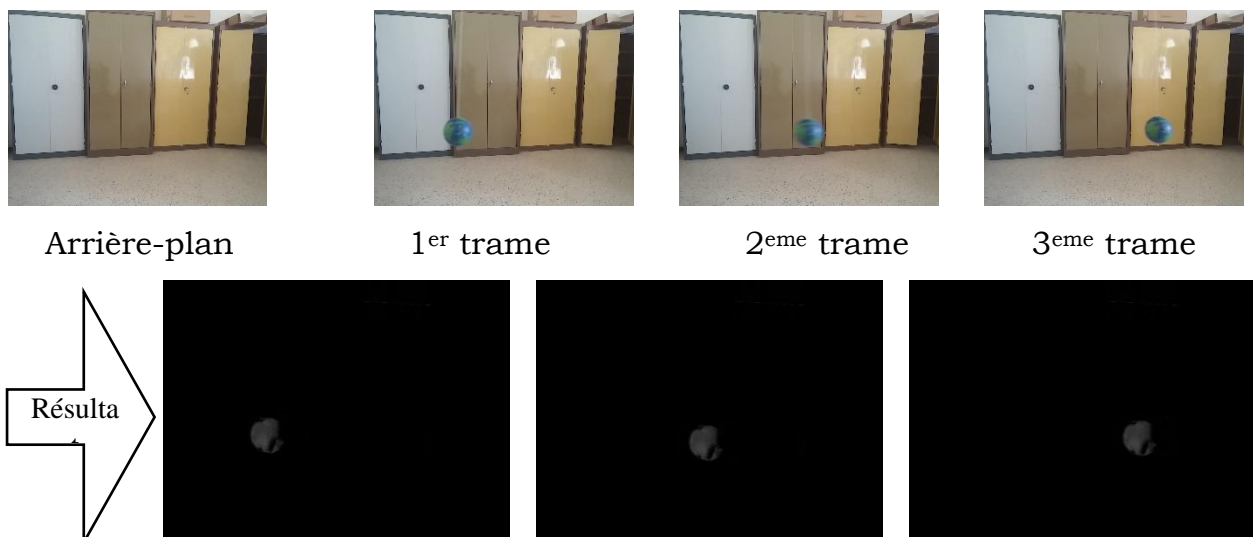


Fig.2.4 Détection d'un objet en mouvement par soustraction d'arrière-plan

2.2.5 Segmentation de mouvement

Cette tâche va au-delà de la détection de mouvement puisqu'il s'agit de segmenter chaque image en régions qui présentent une homogénéité du mouvement apparent. Cette opération est généralement réalisée à part d'une estimation du flot ou des dérivées spatiotemporelles de l'intensité lumineuse et surtout pour le suivi (tracking) (Fig.2.5).

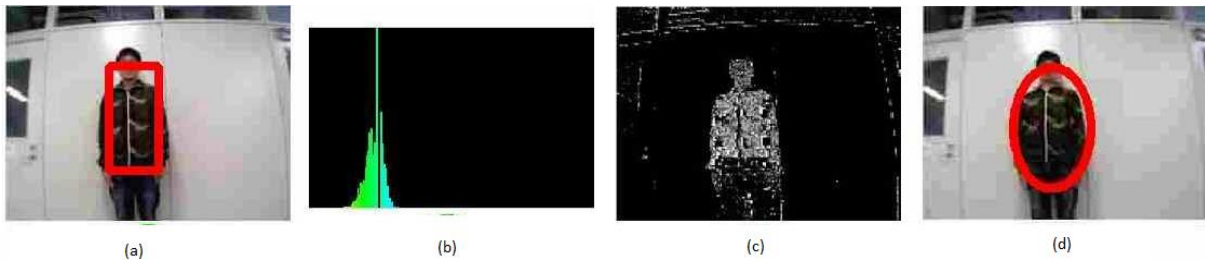


Fig.2.5 Segmentation par Camshift

2.3 Méthodes de détection de mouvements

L'ensemble des éléments du sous-titre vus précédemment nous permet d'établir une classification hiérarchisée des différentes méthodes de détection de mouvement. Nous distinguerons quatre grandes familles de méthodes en fonction de la modélisation de l'arrière-plan.

2.3.1 Détection sans modélisation de l'arrière-plan

Ces méthodes consistent à détecter le mouvement par le calcul en tout point de l'image d'une quantité mathématique qui est fonction de l'intensité ou de la couleur de l'ensemble des pixels et qui est censée refléter l'importance du mouvement visible dans la scène. La dérivée temporelle de l'intensité lumineuse, flot optique sont des exemples que nous présenterons dans la suite

2.3.2 Modélisation locale de l'arrière-plan

Ces méthodes consistent à associer à tout point de l'image une valeur ou une fonction permettant de modéliser l'apparence de l'arrière-plan en ce point. Le modèle d'apparence de l'arrière-plan en un point ne dépend que des observateurs qui ont eu lieu en ce point. Les autres pixels de l'image n'interviennent pas. La grande majorité des méthodes présentées dans la

littérature bâtissent une modèle statistique (ensemble de paramètres d'une loi, ou ensemble d'échantillons), mais il peut s'agir d'un processus stochastique, d'un filtre prédictif ou simplement d'une valeur d'intensité.

2.3.3 Modélisation semi-locale de l'arrière-plan

Ces méthodes sont très semblables à celles de la catégorie précédente, à la différence près que la modélisation de l'arrière-plan en un point dépend des observations qui ont eu lieu dans un certain voisinage de ce point ou dans la région de l'image à laquelle il appartient.

2.3.4 Modélisation globale de l'arrière-plan

Ces méthodes utilisent à chaque instant l'ensemble des observations pour construire un modèle de l'ensemble de l'arrière-plan.

Ses méthodes sont résumées dans la figure 2.6

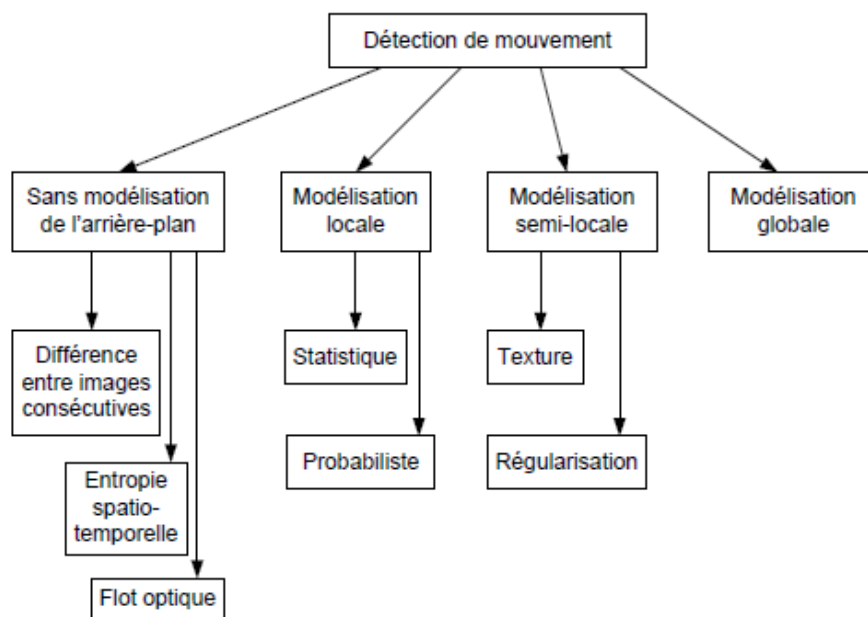


Fig.2.6 - Méthodes de détection de mouvement

2.4 Détection sans modélisation

Dans l'ensemble de cette section et dans le reste du chapitre, nous désignerons par $I(t)$ l'image en niveau de gris présente dans le flux d'entrée au temps t et par $I(x, y, t)$ le pixel de coordonnées $X(x, y)$.

Dans cette même image, nous appellerons E le domaine de définition des coordonnées des pixels. Les dérivées spatiales horizontale et verticale de l'image seront notées respectivement $I_x(x, y, t)$ et $I_y(x, y, t)$. La dérivée temporelle sera notée $I_t(x, y, t)$. Le gradient de l'image au point (x, y) et au temps t sera quant à lui noté

$$\nabla I(x, y, t) = [I_x(x, y, t) \ I_y(x, y, t)]^T \quad (2.1)$$

2.4.1 Dérivée temporelle

La manière la plus intuitive de détecter les zones en mouvement dans un champ de vision est de mesurer le changement d'apparence des pixels entre deux trames consécutives, soit la dérivée temporelle en tout point. La première utilisation de cette méthode dans l'analyse de séquence vidéo est généralement attribuée à Jain et Nagel [8]. L'intensité lumineuse d'un pixel étant un signal discret à une dimension, l'estimation de la dérivée temporelle instantanée du signal au temps t est donnée par :

$$\forall (x, y) \in E \ \forall t > 0 \ I_t(x, y, t) \approx |I(x, y, t) - I(x, y, t - \Delta t)| \quad (2.2)$$

L'équation (2.2) a été programmée et simulée par un exemple (Fig2.7)

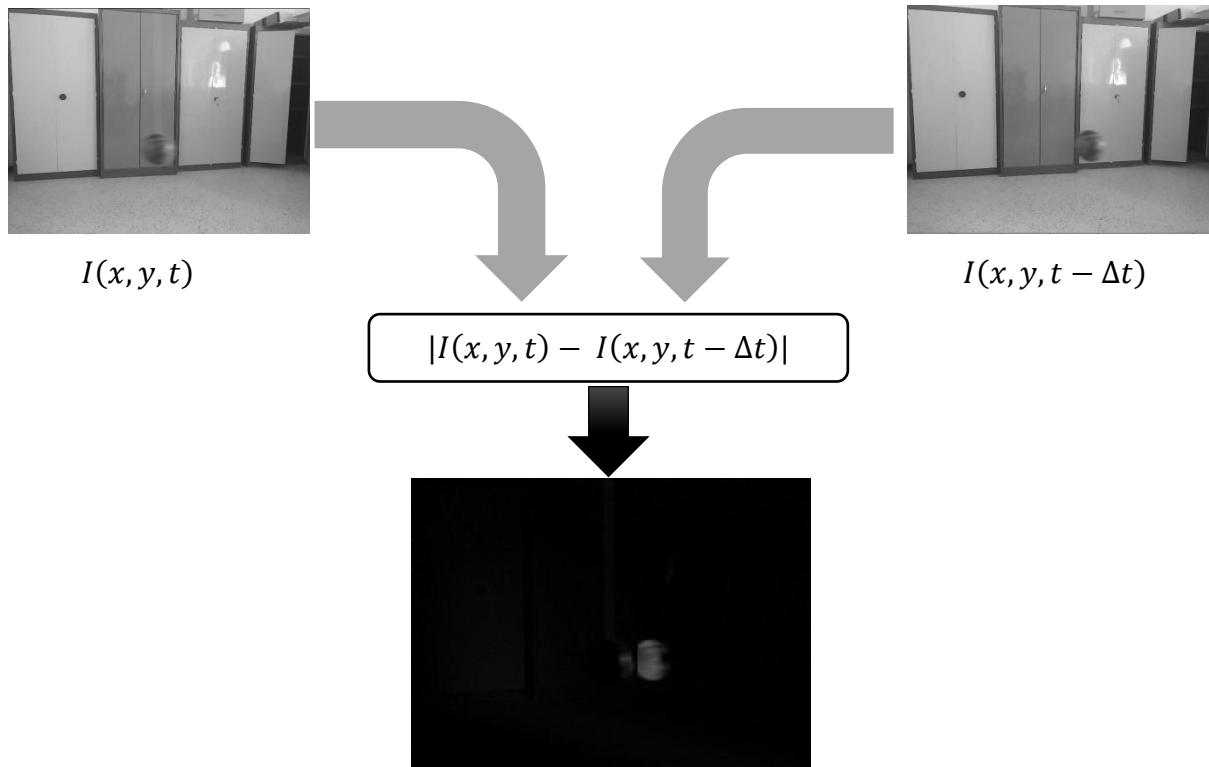


Fig.2.7 Détection de mouvement par dérivée temporelle

Mais cette méthode se montre peu robuste face à des phénomènes tels que les mouvements lents, les arrêts brefs d'un objet en mouvement ou encore la présence de trames redondant dans certaine séquence d'images. Il convient donc d'effectuer un lissage temporel de la séquence, c'est-à-dire d'appliquer un opérateur de moyenne mobile à la mesure obtenue. Ceci peut être fait à l'aide d'une matrice de même taille que les trames d'entrée, appelée accumulateur et que nous noterons A .

$$\forall (x, y) \in E \left\{ \begin{array}{l} A(x, y, 0) = 0 \\ \forall t > 0 \quad A(x, y, t) = \omega_A A(x, y, t - \Delta t) + (1 - \omega_A) I_t(x, y, t) \end{array} \right\} \quad (2.3)$$

Avec $0 \leq \omega_A \leq 1$. Dans l'équation (2.3), le terme de ω_A pondère la contribution des mesures passées par rapport à la dernière mesure en date. Avec un ω_A faible, les effets du lissage temporel sont peu visibles, et les problèmes qui avaient motivé l'utilisation de cette méthode risquent d'apparaître tout de même. Avec un ω_A élevé, le lissage est important et un effet de persistance des entités détectées peut se produire (effet fantôme).

Ce cas a été illustré dans la figure ci-dessous (Fig.2.8)

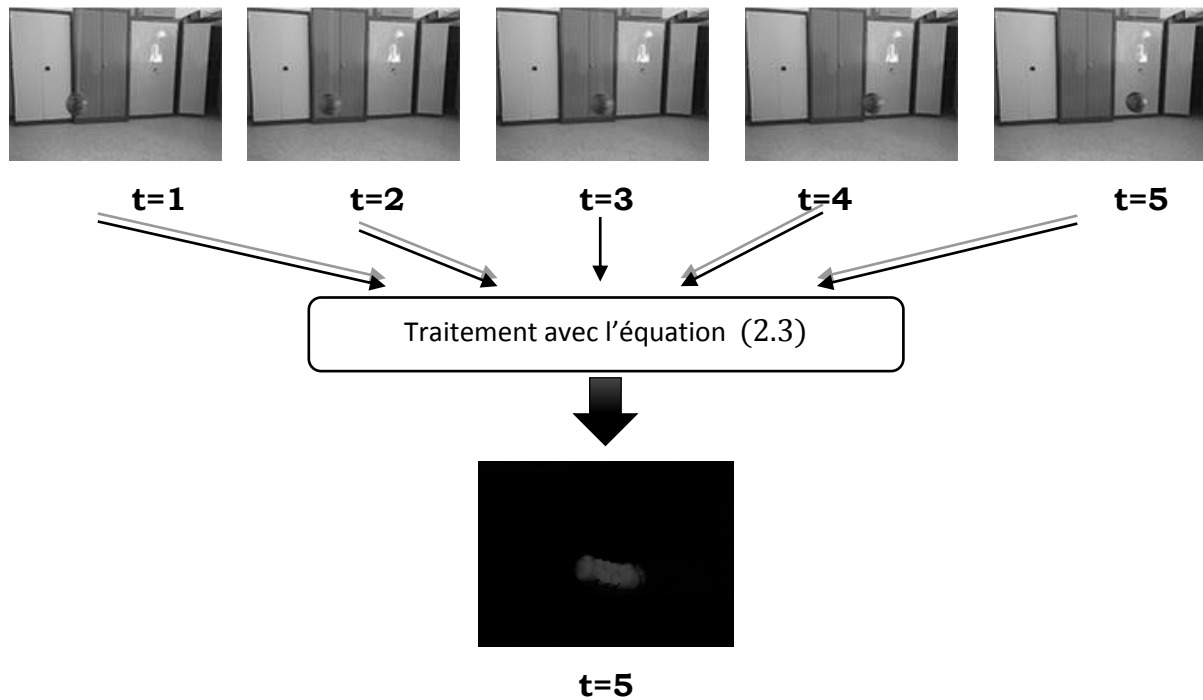


Fig.2.8 Détection de mouvement par lissage temporel de la séquence pour $\omega_A = 0.8$

2.4.2 Entropie spatio-temporelle

L'entropie est une mesure issue de la thermodynamique, associée au degré de désordre d'un système. Dans notre cas, il s'agira de mesurer en chaque point, la variabilité de la grandeur mesurée. Concrètement; plus l'intensité lumineuse aura pris des valeurs dissemblables en un point pendant un certains intervalles de temps, plus l'entropie sera élevée sur ce point. Dans notre cas, nous nous intéressons à une méthode à base d'histogrammes. Il s'agit, en chaque pixel désigné par (x, y, t) , de créer un histogramme à partir des points (v, v, τ) d'un voisinage spatio-temporel de diamètre W dans le domaine spatial et de longueur L dans le domaine temporel. Ainsi pour une image à Q niveaux de gris, on note $H_{x, y, t}(q)$ la fréquence de la q -ème classe de l'histogramme ($0 \leq q \leq Q - 1$) associée au point (x, y, t) . Si on normalise cet histogramme, on obtient une fonction de densité de probabilité.

$$\forall (x, y) \in E \quad \forall t \geq L \quad P_{x,y,t}(q) = \frac{H_{x,y,t}(q)}{W^2L} \quad \text{avec} \quad \sum_{q=0}^{Q-1} P_{x,y,t}(q) = 1 \quad (2.4)$$

Cette expression nous permet de déterminer en tout point l'entropie associée à la répartition des niveaux de gris au voisinage dudit point.

$$E(x, y, t) = - \sum_{q=0}^{Q-1} P_{x,y,t}(q) \log(P_{x,y,t}(q)) \quad (2.5)$$

L'entropie aussi calculée peut être quantifiée en 256 niveaux, et être représentée sous la forme d'une image appelée STEI (*Spatial Temporal Entropy Image*) (Fig.2.9). Dans cette image, sont mis en évidence les points où les variations d'intensité lumineuse sont importantes au cours de temps, et dans une moindre mesure, au sein d'un voisinage spatial. Les zones de mouvement sont donc bien mises en évidence, mais les contours (zones à fort gradient) le sont également. On peut atténuer cet effet en donnant plus d'importance aux variations observées dans le domaine temporel que dans le domaine spatial, mais ce phénomène persistera toujours.

Tandis que Ma et Zhang [12] proposent de filtrer le STEI à l'aide de filtres morphologique afin d'éliminer le faux positifs dus aux contours, pour notre part, nous utilisons, comme données, la différence entre trames consécutives telle que calculée dans la section 2.4.1. L'image aussi obtenue est appelée DSTEI (*Difference-based Spatial Temporal Entropy Image*) [16]



Fig.2.9 Détection de mouvement par Entropie spatio-temporelle

2.4.3 Flot optique

Alors que la dérivée temporelle quantifie la variation de l'aspect de chaque pixel considéré individuelle, le flot optique est un champ de vecteurs à deux dimensions représentant la projection sur le plan image du mouvement réel observé (tridimensionnel). De nombreuses méthodes ont été proposées depuis l'article précurseur de Horn et Schunck [7]. Ces méthodes sont détaillées dans plusieurs articles comme celui Barron et al. [2], où neuf algorithmes sont étudiés et comparés selon des critères de précision et de la

densité du champ obtenu. Cependant, aucune mention n'est faite de la complexité algorithmique. Quelle que soit la méthode choisie, le calcul du flot optique reste une opération très coûteuse en temps de calcul. Le temps-réel peut néanmoins être atteint en sous-échantillonnant les trames et en choisissant un algorithme rapide. Parmi les algorithmes les plus rapides et les plus populaires, nous pouvons citer celui de Lucas et Kanade [11]. Il s'agit de calculer en temps t , le déplacement $d = (d_x, d_y)^T$ du point $x = (x, y)^T$. L'hypothèse sur laquelle se base la méthode est celle de la conservation de l'intensité lumineuse (2.6).

$$\forall (x, y) \in E \quad \forall t > 0 \quad I(x + d, t + 1) - I(x, t) = 0 \quad (2.6)$$

Ainsi, l'estimation du flot optique au point est le vecteur d^* qui minimise la fonction d'erreur quadratique ε calculée sur un voisinage $\mathfrak{N}(x)$ et définie par

$$\varepsilon(d) = \sum_{x' \in \mathfrak{N}(x)} [I(x' + d, t + 1) - I(x', t)]^2 = 0 \quad (2.7)$$

La minimisation peut être obtenue de manière récursive par l'expression (2.8)

$$\left\{ \begin{array}{l} d_0 = 0 \\ d_{n+1} = d_n + \left(\sum_{x' \in \mathfrak{N}(x)} \nabla I(x' + d_n, t + 1)^T I_t(x', t + 1) \right) \\ \left(\sum_{x' \in \mathfrak{N}(x)} \nabla I(x' + d_n, t + 1) \nabla I(x' + d_n, t + 1)^T \right) \end{array} \right. \quad (2.8)$$

2.5 Détection par modélisation locale

Parmi les méthodes de modélisation locale, nous pouvons en citer les trois suivantes.

2.5.1 Modélisation par image

Le modèle d'arrière-plan le plus simple serait une image représentant la scène dépourvue d'objet. Cette méthode présente l'avantage de détecter aussi bien les mouvements lents que les mouvements rapides. Par ailleurs même les objets momentanément immobiles sont détectés (Fig.2.10). Cependant, en environnement extérieur, les variations d'intensité lumineuse

rendent rapidement obsolète un tel modèle, et n'est pas toujours possible d'obtenir une image de la scène totalement vide. Dans ces conditions il est nécessaire de mettre à jours l'image de l'arrière-plan.

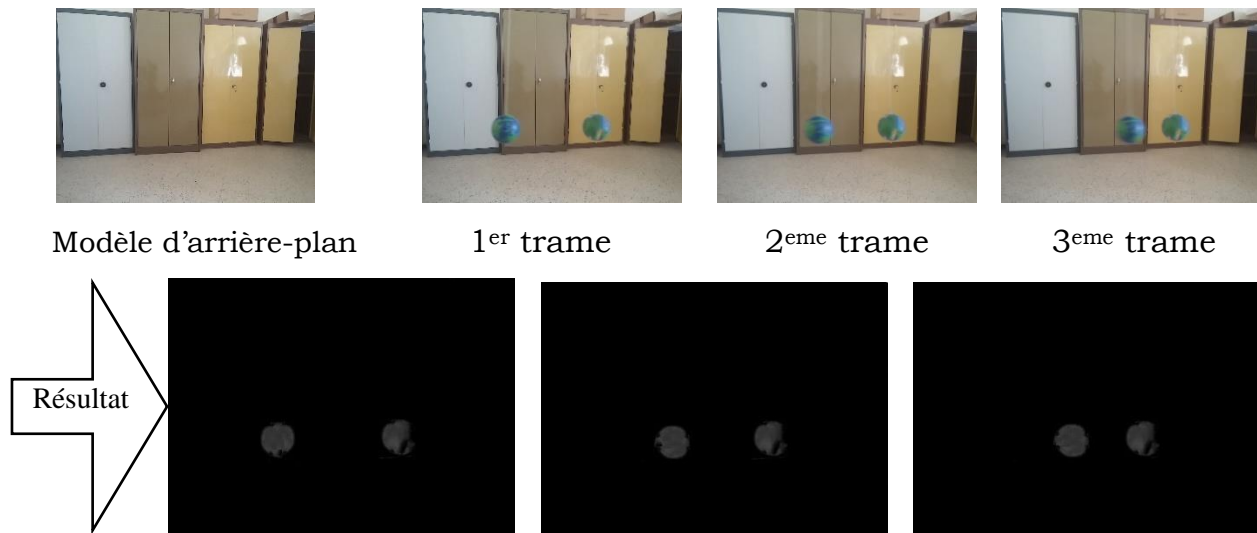


Fig.2.10 Modélisation d'arrière-plan par une image dépourvue d'objet

Dans le travail de Yang et al. [17], les auteures proposent d'utiliser la différence entre images consécutives pour y parvenir. Ils considèrent la première image de la séquence comme une première approximation du modèle de l'arrière-plan. Ensuite, à chaque nouvelle trame, la différence par rapport à l'image précédente est calculée, et les pixels où aucun mouvement n'est détecté sont utilisés pour mettre à jours le modèle du fond. Pour plus de robustesse, les auteurs préconisent de ne considérer que les points auxquels la dérivée temporelle a été négligeable pendant un certain intervalle de temps.

2.5.2 Modélisation statistique

Le problème de la modélisation de l'arrière-plan peut être exprimé d'un point de vue statistique. Il s'agit, pour chaque pixel, d'estimer la probabilité d'y observer telle ou telle couleur (ou niveau de gris) en se basant sur un modèle appris, censé représenter l'arrière-plan de la scène. Le modèle consiste en un ensemble de fonction de densité de probabilité, une par pixel de l'image les mesures dont la probabilité d'être observées est élevée

correspondent à des pixels qui seront étiquetés comme arrière-plan, tandis que celles dont la probabilité d'être observées est faible correspondent à des pixels qui seront étiquetés comme avant-plan.

2.5.3 Modélisation prédictive

Une autre approche, assez semblable à l'approche statistique, consiste à utiliser un filtre de Wiener ou de Kalman pour prédire la prochaine valeur que l'on devrait observer en chaque point. C'est l'écart entre la prédiction et l'observation qui sera utilisé pour estimer l'amplitude du mouvement. Ces méthodes permettent de gérer les problèmes d'arrière-plans non statiques, et les perturbations intervenant à intervalles réguliers (par exemple, la bande horizontale qui apparaît lorsque l'on filme un écran à tube cathodique) [16].

2.6 Conclusion

L'étude des différentes méthodes de détection de mouvement proposées dans ce chapitre nous a permis de constater que la plupart des solutions proposées consistent à considérer les séquences vidéo comme des successions d'images, classées généralement en deux catégories : l'image courante, et l'image passée. L'approche la plus fréquente consiste à construire un modèle plus ou moins compact censé représenter tout le passé, et à confronter l'image courante à ce modèle afin de décider en tout point, si celui-ci représente l'arrière-plan ou un objet mobile. La méthode choisie pour notre réalisation est une détection sans modélisation d'arrière-plan qui est la méthode de la dérivée temporelle.

Troisième chapitre

Etude et réalisation d'un système de détection et de suivi d'objets en mouvements

3.1 Introduction

Dans ce chapitre, nous allons présenter le système que nous avons réalisé ayant pour objectif la détection et le suivi d'un objet en mouvement. Il est composé d'un système mécanique articulé et d'un système de vision, le tout, piloté par un microordinateur via une carte d'acquisition et de restitution de signaux (Annexe 3). Ainsi, on garantit l'orientation et le pilotage de la caméra dans l'espace 3D en utilisant l'asservissement visuel. Cette caméra reste tout le temps active pour détecter tout objet qui entre dans son champ de vision. Le reste du système reste inactif tant que qu'aucun mouvement n'est détecté. Dès qu'un objet est détecté, le système positionne la camera sur cet objet qu'elle suit tout en ramenant et en gardant son image au centre du champ de vision.

Dans la prochaine section, nous décrivons la procédure de la détection de mouvement. La troisième section est consacrée à l'étude et réalisation électrique de la commande du système. La conception mécanique est présentée dans la quatrième section. Dans la cinquième section, nous présentons la partie commande des moteurs utilisés. Le côté software et

hardware de notre application fait l'objet de la section 6. Nous terminons ce chapitre par une conclusion.

3.2. Détection d'un objet en mouvement

L'acquisition d'images et leurs traitements est la première étape dans un système de vision artificielle. Dans notre cas, il s'agit de la détection d'un objet en mouvement dans une séquence d'images prises et traitées en temps réel. Pour ce faire, nous allons utiliser une des méthodes qui ont été exposées dans le chapitre précédant à savoir, la dérivée temporelle. Cette méthode est basée la mesure du changement d'apparence des pixels entre deux trames consécutives en procédant à une soustraction pixel par pixel de deux images. Dans ces conditions il y aurait des valeurs négatives. Ce sont de fausses valeurs dans le sens où si un pixel dont l'amplitude décroît entre deux trames successives, le résultat de la soustraction de ces deux pixel est négatif puisque, en terme de niveau de gris, toute valeur d'amplitude négative est représentée par le noir, d'où sa confusion avec les pixels d'amplitude 0 qui symbolise dans notre cas aucune variation. Une solution à ce problème est de calculer la valeur absolue de tous les éléments de la matrice qui symbolise l'image résultat. Ainsi, on calcule, non seulement la différence entre deux images, mais aussi calcule la valeur absolue de tous les éléments de cette matrice résultat (Fig.3.1).

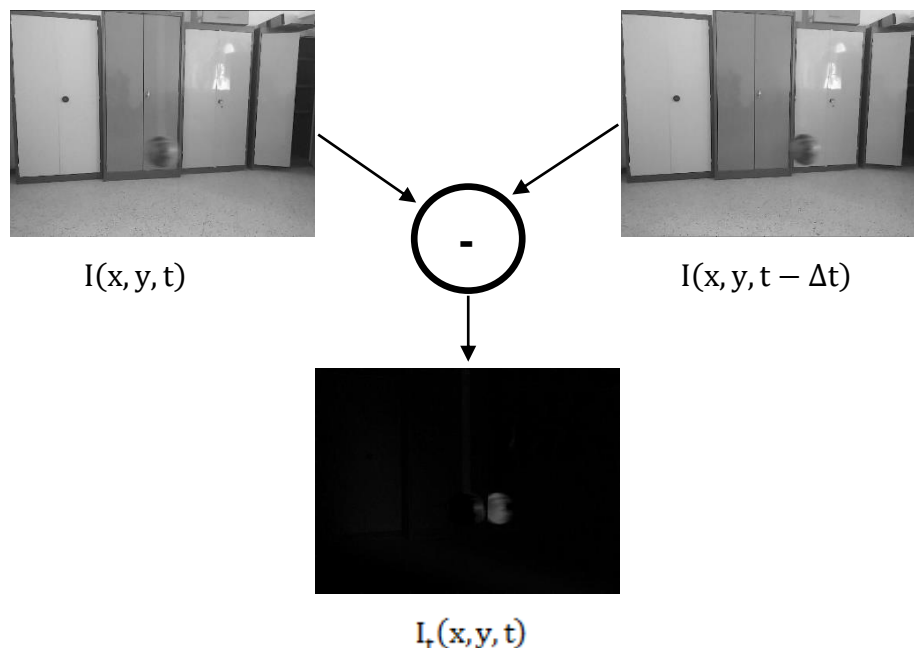


Fig.3.1 Etape de soustraction

Pour modéliser ce mouvement, une étape de segmentation été nécessaire. Pour ce fait, le type de segmentation utilisée est une segmentation par seuillage qui nous donne une image binarisée (Fig.3.2).

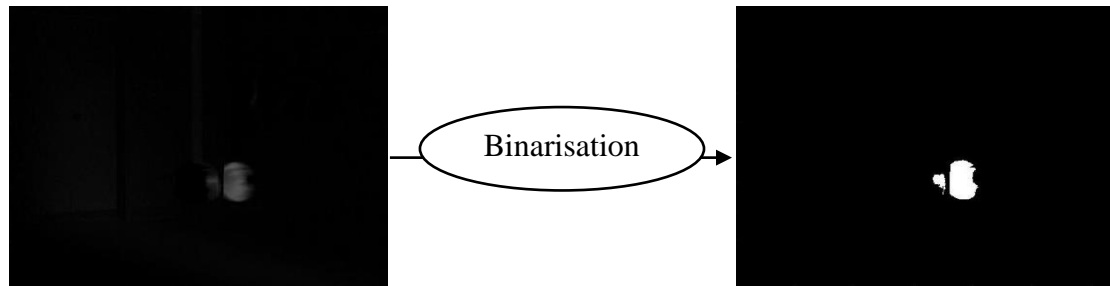


Fig.3.2 Etape de binarisation

L'image segmentée de l'objet en mouvement peut être fragmentée, ce qui peut être interprété comme étant constituée de plusieurs objets en mouvement. Pour remédier à ce problème, on a fait appel à des opérateurs morphologie mathématique telle que la dilatation, la fermeture et l'érosion pour défragmenter l'image de l'objet en mouvement (Fig.3.3).

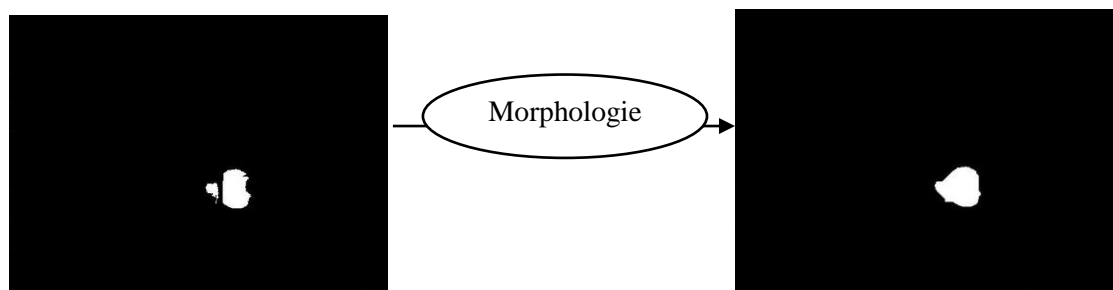


Fig.3.3 Etape de traitement morphologique

Finalement, l'image est représentée avec son centre de gravité (Fig.3.4).

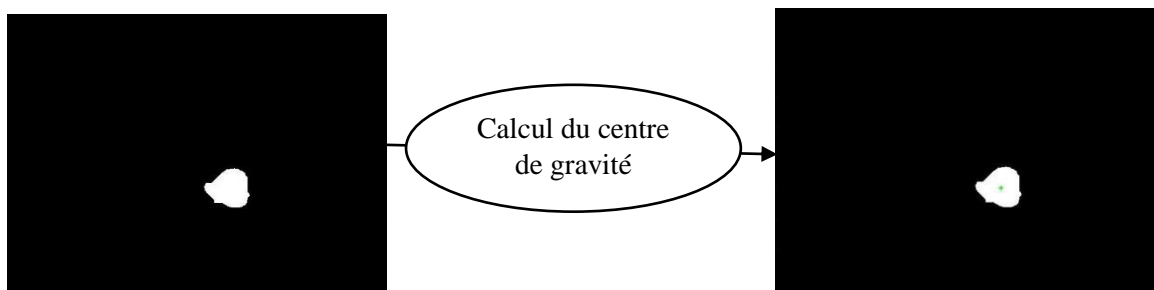


Fig.3.4 Calcule du centre de gravité

L'ensemble de ces opérations est réalisé selon l'algorithme de la figure 3.5

Algorithme : Détection de mouvement

Début

Configuration de la camera

Configurer les éléments structurants

Pour tout

X ← capturer d'image à l'instant t;

X ← convertir l'image X en niveau de gris ;

Y ← capturer d'image à l'instant t + Δt ; // Δt est le temps d'acquisition de deux frames
successives

Y ← convertir l'image Y en niveau de gris ;

Z ← La soustraction entre X et Y et calculer sa valeur absolue ;

Filtrer Z par un filtre médian ;

Z ← Binariser Z ;

Z ← Dilater Z ; niveau de gris

Z ← Fermer Z ;

Z ← Eroser Z ;

Enlever les petites surfaces ;

Calculer le centre de gravité

Fin pour tout

Fin

Fig.3.5 Algorithme pour la détection de mouvement

Cependant, cette technique est très sensible à la moindre variation ou aux bruits et aux variations de luminosité. La technique est applicable que dans le cas où la caméra est statique ou fixe. Par contre, dans notre application l'environnement est inconnu et l'organe visuel est dynamique (caméra mobile). Dans ce cas, nous proposons de ne pas prendre en considération les petits objets en mouvement, en ignorant les images des objets à petite surface et aussi les images fournies par la caméra durant ce déplacement.

3.3 Réalisation électrique

Les éléments qui composent cette partie sont deux moteurs à courant continu avec des réducteurs. Leur rôle est de déplacer la caméra dans l'espace 3D. La caméra a été adaptée à fonction en diurne et en nocturne. Pour ce pilotage, nous avons utilisé un circuit de puissance le L293D [9] (annexe 1) et une carte de commande et d'acquisition de type carte ARDUINO UNO qui est branchée à une entrée USB du microordinateur.

3.3.1 Moteur à courant continu

Le choix du moteur à courant continu présente quelques difficultés liées à la disponibilité, au prix et caractéristiques techniques comme le couple, la vitesse, le réducteur etc. Toutes ces caractéristiques sont à satisfaire selon le cahier des charges. Dans notre application, nous avons utilisé deux moteurs lève-glaces de voiture. L'avantage de ces types de moteurs à courant continu est qu'ils sont équipés d'un réducteur de type vice-sans-fin, ce qui délivre un bon couple mécanique en sortie (Fig.3.6).

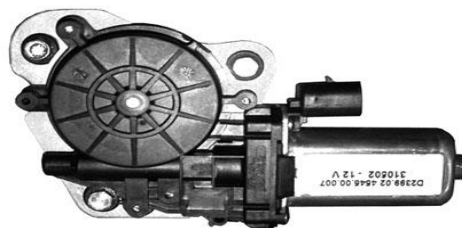


Fig.3.6 Moteur à courant continu avec réducteur

Les moteurs à courant continu utilisés sont à aimant permanent dont la tension et la vitesse nominales sont respectivement de 12 volts et de 5000 tr/min. Pour déterminer la fonction de transfert du moteur, un capteur de vitesse (génératrice tachymétrique) couplée avec le moteur a été utilisé. Ce capteur fournit en sortie une tension continue u_T dont la valeur est proportionnelle à la vitesse du moteur Ω_m , et, sa polarité indique le sens de rotation du moteur.

La tension continue u_T est donnée par l'expression (3.1).

$$u_T = K_T \cdot \Omega_m \quad (3.1)$$

Où K_T est le gain ou sensibilité de la génératrice tachymétrique et

$$K_T = 0.09 \text{ V} \cdot \text{sec}/\text{tr}.$$

On considère le comportement du capteur proche proportionnel et sans retard quoique des ondulations de hautes fréquences se superposent à la tension continue de sortie. L'utilisation d'un filtre passif (de lissage) en utilisant le circuit RC représenté en figure 3.7. Son rôle est d'éliminer le bruit de mesure de vitesse par un lissage de la tension de sortie du capteur de vitesse (génératrice tachymétrique).

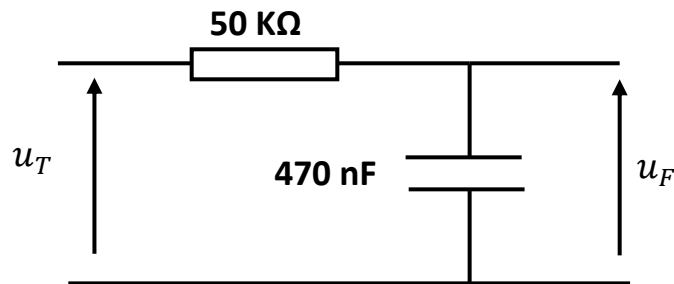


Fig.3.7 Schéma du circuit RC

Le diagramme de Bode du circuit de la figure 3.7 dont la fonction de transfert est donnée par l'expression 3.2 est montré en figure 3.8.

$$F(p) = \frac{1}{1 + RCP} = \frac{1}{1 + 0.0235P} \quad (3.2)$$

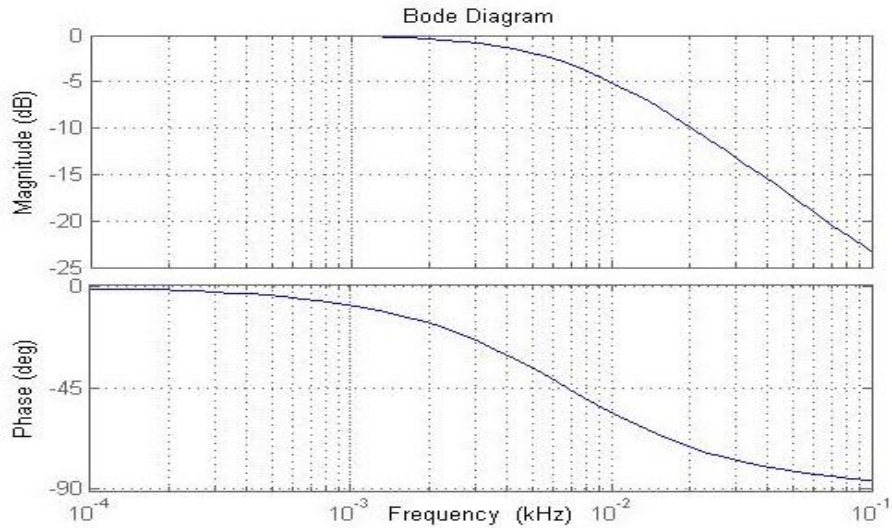


Fig.3.8 Diagramme de Bode du filtre passe-bas.

La fréquence de coupure est calculée par la relation suivante (3.3) :

$$f_c = \frac{\omega_c}{2\pi} \quad (3.3)$$

$$= \frac{1}{2\pi\tau} = \frac{1}{2\pi RC} = 6.773Hz$$

où ω_c est la pulsation de coupure et tel que : $\omega_c = \frac{1}{\tau}$, τ étant la constante de temps.

En effet, ce filtre atténue fortement la composante de hautes fréquences représentant le bruit de mesure et il laisse passer la composante de basses fréquences, correspondante à la vitesse du moteur, avec une atténuation et un déphasage (retard) de 0.005sec que nous pouvons considérer négligeable ($u_F \approx u_T$). Pour illustrer l'action du filtre, un exemple d'une tension générée par le capteur de vitesse avant et après le filtrage est représenté sur la figure 3.9.

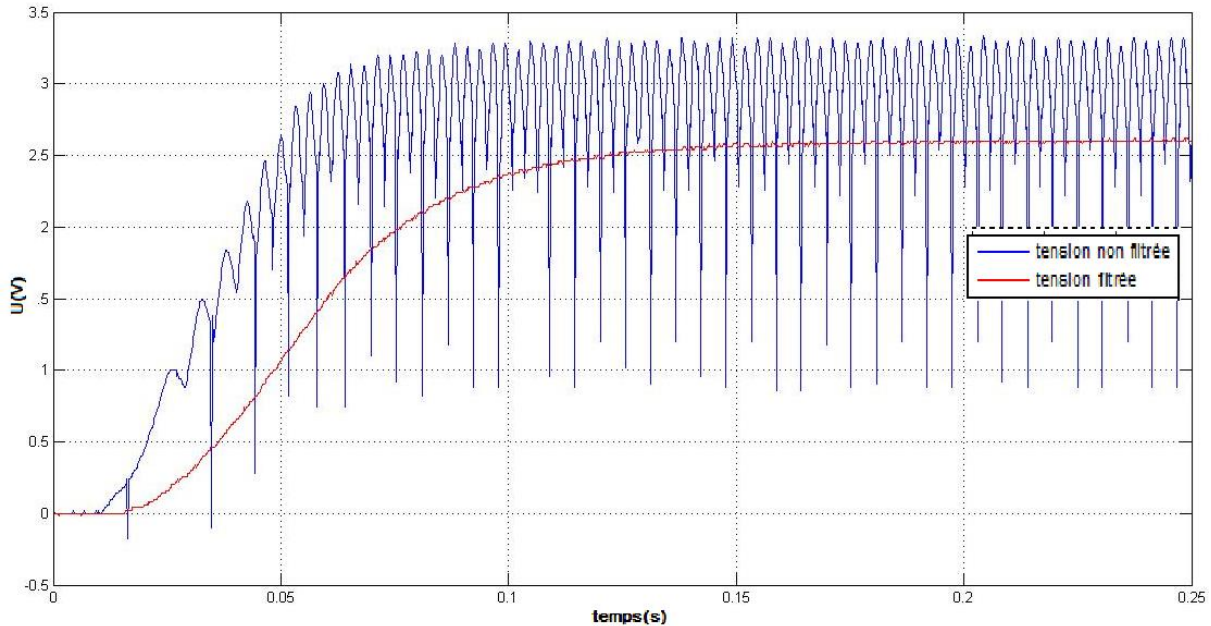


Fig.3.9 Tension générée par le capteur de vitesse avant et après filtrage

Le réducteur de vitesse utilisé est réalisé avec un système roue-vis sans fin. Il est caractérisé par un rapport de réduction $K_R = 1/64$ et un rendement n . Ainsi, pour déplacer en rotation à la camera à une vitesse Ω , le moteur doit tourner à une vitesse Ω_m égale à 64 fois Ω , en générant une puissance mécanique égale à $69n$ fois moins la puissance nécessaire pour faire tourner la camera sur un axe par un moteur sans réducteur [15].

- *Calcul de la fonction de transfert des moteurs.*

Il s'agit de trouver la relation mathématique liant la tension de commande du circuit de puissance, d'abord, à la vitesse de rotation du moteur Ω_m puis à la vitesse rotation après la réduction Ω . Pour ce faire, la structure de la fonction de transfert est déterminée à partir des équations régissant le fonctionnement des moteurs. Les paramètres de cette fonction de transfert sont, ensuite, identifiés par un ensemble de tests pratiques.

- *Détermination de la structure de la fonction de transfert*

En tenant compte de la description des moteurs présentées précédemment et des équations de base d'un moteur à courant continu à aimant permanent [4], les équations régissant le fonctionnement des articulations sont :

$$u_m(t) = E + R i(t) + L \frac{di(t)}{dt} \quad (3.4)$$

$$E(t) = k \Omega_m(t) \quad (3.5)$$

$$J \frac{d\Omega_m(t)}{dt} = C_m(t) - C_r(t) \quad (3.6)$$

$$C_m(t) = k i(t) \quad (3.7)$$

$$\Omega(t) = K_R \Omega_m(t) \quad (3.8)$$

$$u_m(t) = K_P u(t) \quad (3.9)$$

$$u_F(t) = K_T \Omega_m(t) \quad (3.10)$$

où C_m est le couple électromagnétique du moteur, J , le moment d'inertie de l'ensemble des pièces tournantes ramenées à l'arbre du moteur, C_r , le couple résistant dû à toutes les charges tournantes et appliqué au moteur par l'intermédiaire du réducteur, k , la constante du moteur, K_P , le gain du circuit de puissance, I , le courant traversant l'induit du moteur, E , la *FEM.* du moteur, R , la résistance de l'induit du moteur, L , l'inductance de l'induit du moteur et u_F , la tension après filtrage.

La transformée de Laplace appliquée aux équations (3.4) à (3.10) donne les expressions (3.11) au (3.17).

$$U_m(p) = E(p) + R I(p) + Lp I(p) \quad (3.11)$$

$$E(p) = k \Omega_m(p) \quad (3.12)$$

$$Jp \Omega_m(p) = C_m(p) - C_r(p) \quad (3.13)$$

$$C_m(p) = k I(p) \quad (3.14)$$

$$\Omega(p) = K_R \Omega_m(p) \quad (3.15)$$

$$U_m(p) = K_P U(p) \quad (3.16)$$

$$U_F(p) = K_T \Omega_m(p) \quad (3.17)$$

Ceci nous conduit au schéma fonctionnel du système (Fig.3.10).

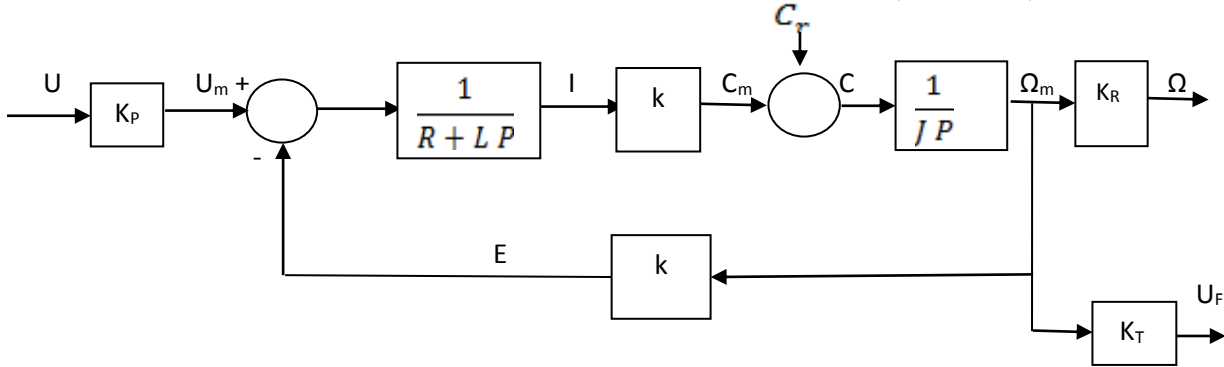


Fig.3.10 Schéma fonctionnel du de la table tournante.

Le couple résistant C_r est composé d'un couple, proportionnel à la vitesse, correspondant au couple de frottement dû principalement au réducteur de vitesse et d'un couple dépendant de la charge C_c (3.18)

$$C_r(t) = C_c + \lambda \Omega_m(t) \quad (3.18)$$

Où λ est le coefficient des frottements.

Du fait de l'importance du rapport de réduction (1/64) du réducteur de vitesse, la charge influe peu sur l'arbre du moteur. Ainsi, le couple de la charge est négligeable devant le couple de frottement. Par conséquent, l'expression (3.18) s'écrit sous la forme de l'expression (3.19) suivante:

$$C_r(t) = \lambda \Omega_m(t) \quad (3.19)$$

Ainsi, l'expression (3.6) devient

$$J \frac{d\Omega_m(t)}{dt} = C_m(t) - \lambda \Omega_m(t) \quad (3.20)$$

La transformée de Laplace de l'expression (3.20) est :

$$Jp \Omega_m(p) = C_m(p) - \lambda \Omega_m(p) \quad (3.21)$$

En combinant ces équations, on obtient la fonction de transfert suivante (3.22)

$$G(P) = \frac{\Omega(p)}{U(P)} = \frac{k K_P K_R}{k^2 + R\lambda + (RJ + L\lambda)p + LJ p^2} \quad (3.22)$$

Sous sa forme canonique cette fonction de transfert devient:

$$G(p) = \frac{\Omega(P)}{U(P)} = \frac{G}{1 + \frac{2h}{w_n} p + \frac{1}{w_n^2} p^2} \quad (3.23)$$

où $G = \frac{k K_P K_R}{k^2 + R\lambda}$, $\frac{2h}{w_n} = \frac{RJ + L\lambda}{k^2 + R\lambda}$ et $\frac{1}{w_n^2} = \frac{LJ}{k^2 + R\lambda}$

Ainsi, pour déterminer un modèle pour le moteur, il suffit d'identifier les paramètres G, h et w_n en utilisant la méthode des moindres carrés.

Pour ce faire, en premier lieu, nous avons réalisé sur le moteur, sans le circuit de puissance et sans réducteur, un essai indiciel en appliquant un échelon de 7 volts. La réponse indicielle relative à la vitesse est enregistrée à l'aide d'un oscilloscope numérique de fréquence de 200MHz en prenant en compte 1750 points (Fig.3.11). A partir de l'allure de cette courbe, on conclut que la réponse indicielle en vitesse présente un comportement apériodique [4]. Par conséquent elle est décrite par l'équation (3.24),

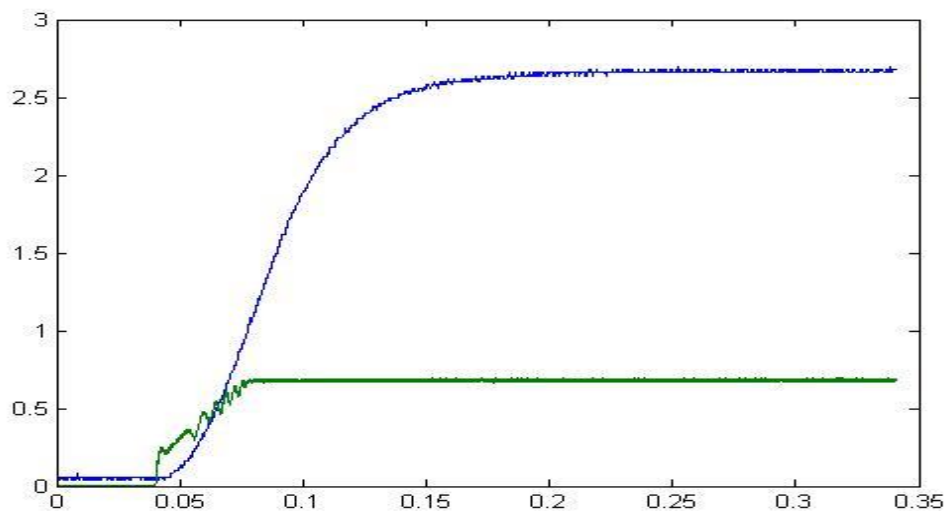


Fig.3.11 Tension d'alimentation et vitesse mesurée de la table.

$$\Omega(t) = \Omega_{\infty} \left(1 + \frac{\tau_1}{\tau_2 - \tau_1} e^{-\frac{t}{\tau_1}} + \frac{\tau_2}{\tau_1 - \tau_2} e^{-\frac{t}{\tau_2}} \right) \quad (3.24)$$

$$\text{Avec : } \tau_1 = \frac{1}{w_n} (h + \sqrt{h^2 - 1})$$

$$\tau_2 = \frac{1}{w_n} (h - \sqrt{h^2 - 1})$$

$$\Omega_\infty = U_0 G_m \quad \text{avec} \quad G_m = \frac{G}{k_p}$$

En second lieu, la méthode des moindres carrés [15] a été appliquée pour déterminer les paramètres G, h et w_n par la minimisation du critère quadratique suivant (3.25) :

$$J = \sum_{k=0}^N [\Omega(kT) - \Omega_{mes}(kT)]^2 \quad \zeta \quad (3.25)$$

Où $\Omega_{mes}(kT)$ est la mesure de la réponse indicielle prise à l'instant kT et $\Omega(kT)$, la valeur de la réponse indicielle à estimer à l'instant kT

Les résultats de cette minimisation sont :

$$G_m = \frac{61.1}{7}, \quad G = 8.7, \quad h = 0.89 \quad \text{et} \quad w_n = 44.72$$

Ainsi, la fonction de transfert suivante est obtenue (3.26) :

$$G(P) = \frac{8.7}{1 + 0.039p + 0.0005p^2} \quad (3.26)$$

La figure 3.12 représente la réponse indicielle mesurée. La réponse indicielle (estimée) courant/vitesse du moteur sans réducteur est obtenue par identification. Les deux réponses indicelles sont mesurées en tr/sec.

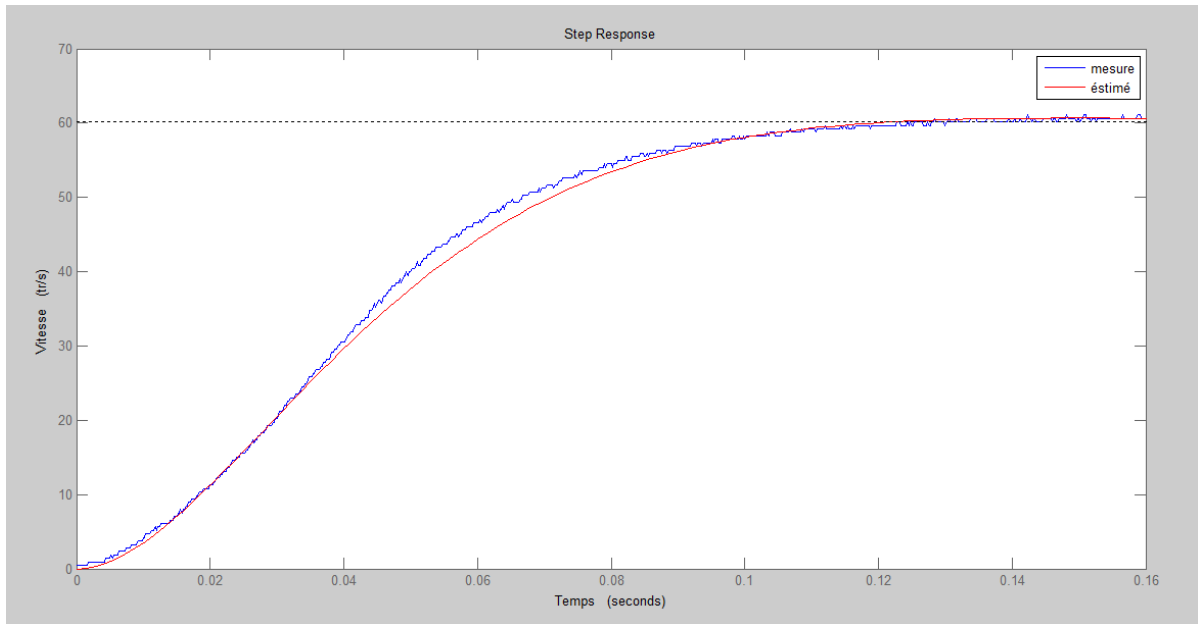


Fig.3.12 Réponse indicielle mesurée (en bleu),
réponse indicielle estimée (en rouge) de la sortie du moteur.

Nous constatons que le temps de réponse t_r de la table est de 0.09 sec et que ce système présente un petit dépassement difficilement distinguable vu que le coefficient d'amortissement est inférieur et presque égale à 1.

En prenant comme nouvelle sortie la vitesse de l'arbre après réduction. De la même manière avec la première estimation sans réducteur nous obtiendrons ce qui suit :

$$G_m = \frac{0.87}{7}, \quad G = 0.125, \quad h = 0.72 \text{ et } w_n = 40.8$$

Ceci nous conduit à la fonction de transfert suivante (3.27) :

$$G(P) = \frac{0.125}{1 + 0.038p + 0.0006p^2} \quad (3.27)$$

La figure (Fig.3.13) représente la réponse indicielle mesurée et la réponse indicielle (estimée) courant/vitesse du moteur avec réducteur et obtenue par identification. Les deux réponses indicielles sont mesurées en tr/sec.

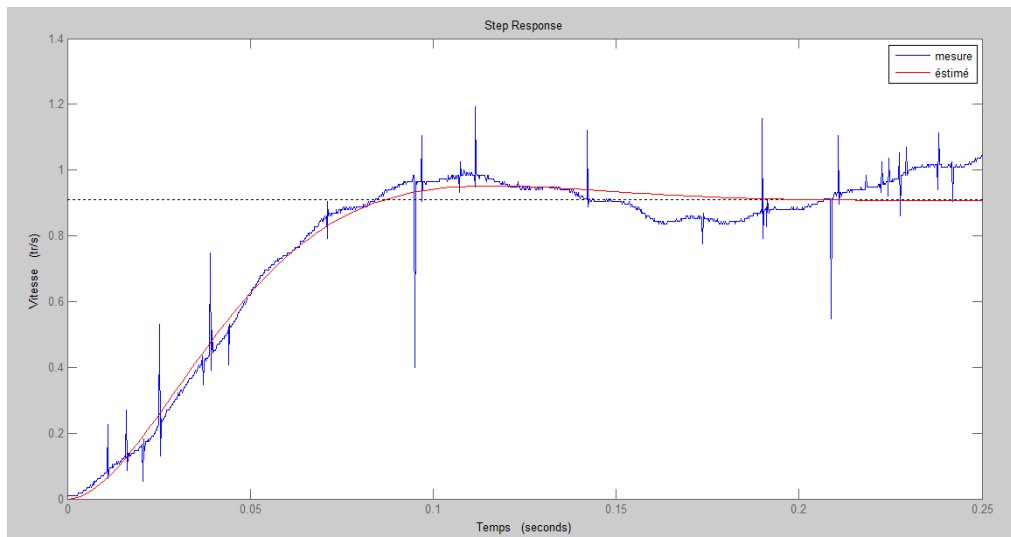


Fig.3.13 Réponse indicielle mesurée (en bleu), réponse indicielle estimée (en rouge) de la sortie du réducteur.

Les résultats trouvés précédemment montrent que après réduction, le dépassement est plus important qui est dû au moment d'inertie qui s'accroît en présence du réducteur.

3.3.2 Le circuit de puissance

Le L293D est un circuit qui comprend deux pont-H (Fig.3.14) et a comme rôle d'alimenter le moteur par une tension de 10.5 volts lorsqu'il reçoit une tension de 5 volts à son entrée de commande. Sachant que ce dernier peut commander indépendamment deux moteurs à courant continu en sens, un seul circuit L293D est suffisant dans notre travail. Notons que le L293D peut délivrer au maximum 600mA.



Fig.3.14 Circuit intégré L293D

Les caractéristiques techniques d ce composant L293D sont les suivantes:

- Nombre de pont-H: 2
- Courant maximal régime continu: 600mA (x2)
- Courant de pointe Max : 1200mA
- Tension maximal alimentation du moteur (Vs): 36v
- Alimentation maximal logique (Vss): 7v
- Nombre de broches: 16 Pins
- Perte de tension: 1.3v
- Temps de repense : 2 ms

Le schéma suivant détaille les différentes broches du composant L293D (Fig.3.15) :

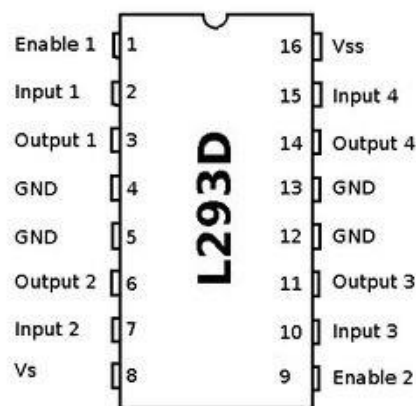


Fig.3.15 L293D pins

1. Enable1 permet d'envoyer ou non la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2 et commande l'activation/désactivation du premier Pont-H. Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas. Si ENABLE1 = Vss, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 et INPUT2.
2. Input1 et Input 2 sont les broches de commande du Pont-H Output1/Output2 commandent le sens du courant entre Output 1 et Output 2.
3. Ouput1 et Output 2 sont les sorties du pont-H branché directement sur le moteur 1.

4. GND est raccordé à la masse de la source d'alimentation de puissance Vs et à la masse de la source d'alimentation de la logique Vss (et GND Arduino).
5. GND
6. Output2 et Output 1.
7. Input2 et Input 1.
8. Vs: Alimentation de puissance des moteurs.
9. Enable2: commande l'activation du second pont-H constitué de Output3/Output4
10. Input3 avec Input 4, sont les broches de commande du Pont-H Output3/Output4 commandent le sens du courant entre Output 3 et Output 4.
11. Output3 et Output 4 sont les sorties du pont-H branché directement sur le moteur 2.
12. GND
13. GND
14. Output4 et Output 3.
15. Vss est l'alimentation de la logique de commande (5V). (A raccorder à la borne +5V de votre Arduino).

Notant que les pins Enable1 et Enable2 permettent de moduler la vitesse du moteur en utilisant un signal de type PWM (Pulse Width Modulation) comme entrée de ces broches.

Les différentes configurations pour contrôler la marche et l'arrêt et le contrôle du sens de rotation du moteur à courant continu en utilisant le composant L293D sont résumées dans le tableau 3.1:

Tab.3.1 Configurations de fonctionnement du L293D

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

3.3.3 Carte ARDUINO UNO

Le modèle UNO (Fig3.16) de la société ARDUINO est une carte électronique dont le noyau est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des sorties PWM (Pulse Width Modulation)), 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits de résolution, un résonateur céramique (quartz) de 16 MHz, un connecteur ICSP (In Circuit Serial Programming) qui permet d'injecter le bootloader à l'intérieur du microcontrôleur, un connecteur jack pour une alimentation extérieure, un bouton de reset pour mettre le processus à zéro.

L'avantage de cette carte est qu'elle n'a pas besoin de pilote pour faire la conversion USB/série. Elle a juste un petit microcontrôleur ATMega8 (pour la version 2) programmé comme convertisseur USB/série Elle contient tout ce que le microcontrôleur a besoin pour fonctionner. Il suffit simplement de la connecter avec un câble USB à un microordinateur ou avec une alimentation externe.

L'intérêt principal des cartes ARDUINO est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open-source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions sont également fournies pour l'exploitation d'entrées-sorties, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C. La carte ARDUINO peut être alimentée soit via la connexion USB qui fournit 5V 500mA ou à l'aide d'une alimentation externe ; dans notre application la carte ARDUINO est alimenté via le port USB.

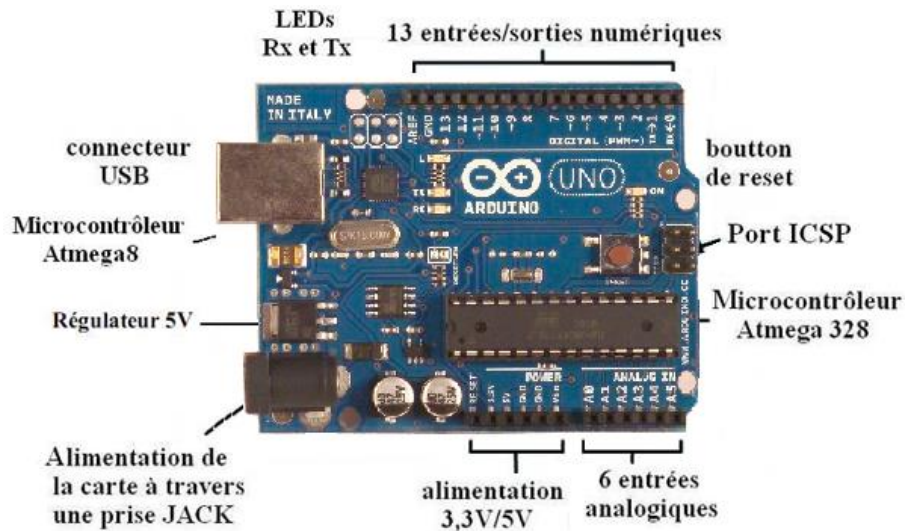


Fig.3.16 : Description de la carte ARDUINO UNO

3.3.4 Adaptation de la camera en mode nocturne

Bien que la disponibilité des caméras dotée d'une vision nocturne sur le marché, leur prix reste exorbitants ce qui nous a poussés à adapter une caméra numérique de type webcam en mode nocturne sachant qu'une webcam n'est pas sensible à la lumière infrarouge, elle-même est équipée d'un filtre infrarouge. Ainsi, nous avons enlevé le filtre et mis en place un projecteur infrarouge composé à l'aide de 50 LEDs LTE5128A [10] (Fig.3.17).

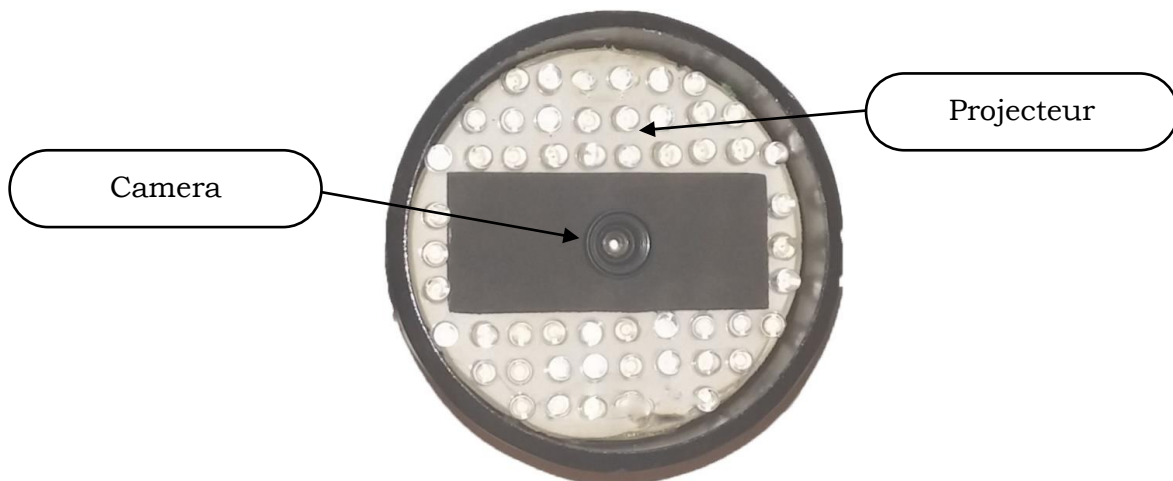


Fig.3.17 Camera adapté en mode nocturne

3.3.5 Alimentation

Pour l'alimentation en courant et en tension qui conviendrait à toutes les parties qui demandent une alimentation électrique à savoir, la carte ARDUINO, les moteurs, le projecteur infrarouge, le L293D et la camera. Les

Ports USB de tous les microordinateurs sont dotés d'une alimentation de (5V, 1A) pour alimenter les dispositifs qui seront branchés à ses Ports. Ainsi, dans notre cas, la camera et la carte ARDUINO sont alimentées via le Port USB. Le L293D qui commande les deux moteurs à courant continu nécessite deux alimentations. L'alimentation $V_{ss}=5V$ est branchée à la carte l'ARDUINO sachant que cette dernière a deux alimentations une de 5V et l'autre de 3.3V. Pour la seconde alimentation V_s qui alimente les moteurs est une alimentation externe de 12V, 2A. Enfin pour l'alimentation du projecteur infrarouge, nous avons utilisé une deuxième alimentation externe de (4V, 1A) (Fig.3.18)

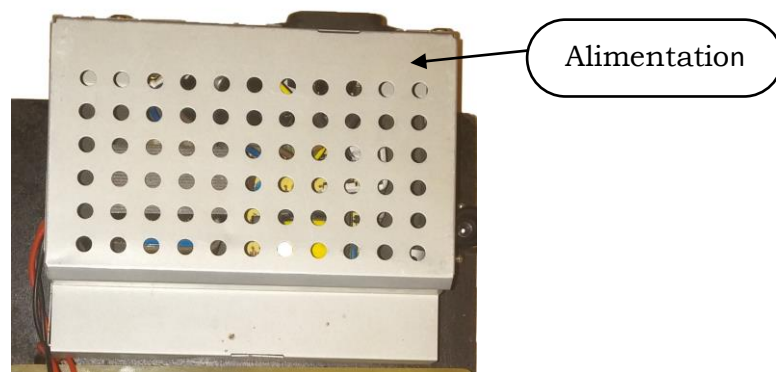


Fig.3.18 Alimentation exetne

3.3.6 Branchement ARDUINO

Le branchement entre la carte ARDUINO (circuit de commande), le circuit intégrer L293D (circuit de puissance) et les moteurs est représenté dans la figure (Fig.3.19)

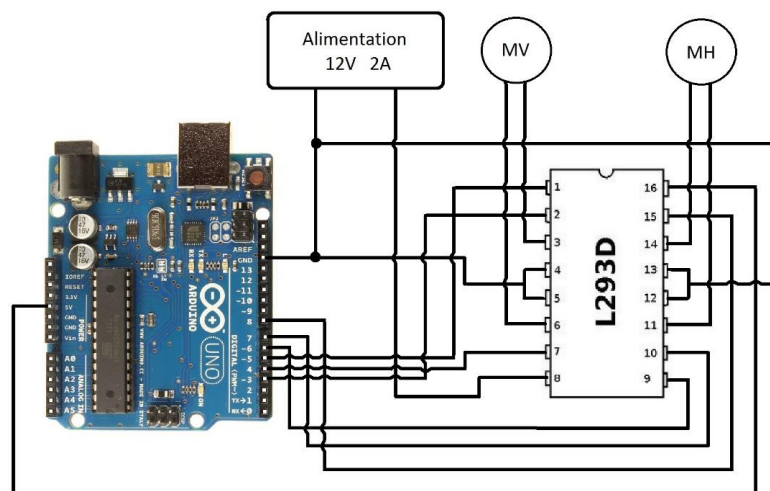


Fig.3.19 Branchement ARDUINO avec le L293D

Pour une utilisation manuelle, le branchement du joystick à la carte ARDUINO est représenté à la figure (Fig.3.20)

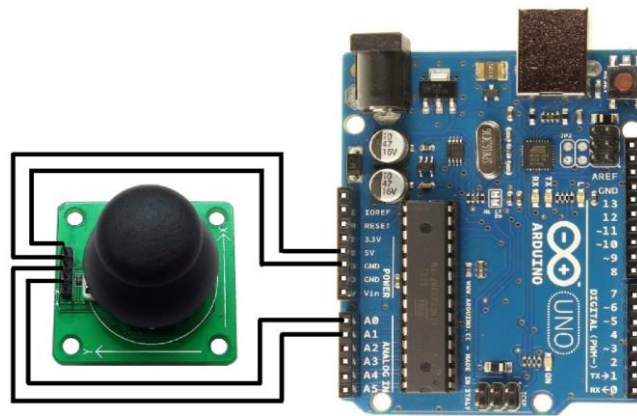


Fig.3.20 Branchement ARDUINI avec le joystick

Le circuit global est présenté en annexe 2.

3.4 Conception de la partie mécanique

Le cahier des charges communiqué à un bureau d'étude en génie mécanique pour cette conception est le suivant:

- Le système articulé est à deux degrés de liberté, l'un vertical et l'autre horizontal. L'articulation horizontale porte l'articulation verticale et cette dernière porte l'organe visuel (Fig.3.21).

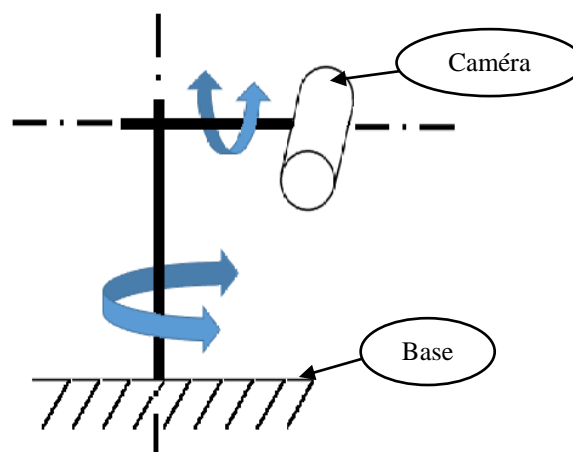


Fig.3.21 Schéma basique du système mécanique

- Le poids et le volume doivent être réduits.

- La base du système doit être cylindrique avec un grand diamètre pour une stabilité maximale.
- Les moteurs sont à courant continu.
- Le guidage en rotation doit être précis en vertical et en horizontal.
- Tous les mouvements doivent être stables.
- Des emplacements de butée de fin de course pour les deux mouvements vertical, horizontal doivent être prévus.

On prenant en considération les problématiques imposé dans le cahier des charges, plusieurs simulations 3D ont été faites pour aboutir à un résultat final représenter dans la figure 3.22.

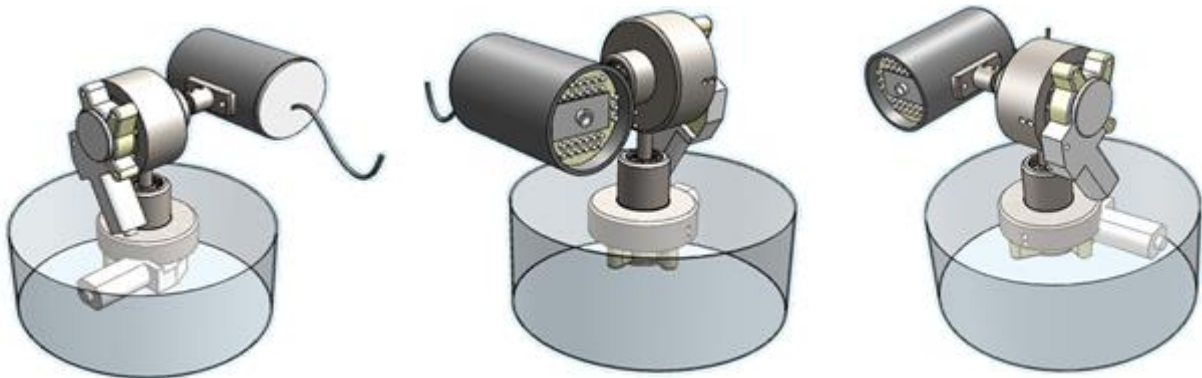


Fig.3.22 Simulation 3D de la solution proposé

Cette conception est composée de 7 pièces (annexes du 4 au 10)

3.5 Commande des moteurs

Dans notre cas la commande adéquate est celle de l'asservissement visuel. La commande est basée sur la technique de « look end move ». La partie « move » de l'asservissement visuel est décrit dans la section 3.2. L'objectif est de trouver une loi de commande qui nous permet de faire un asservissement en position de la camera de façon à ce que le point représentant l'objet en mouvement soit ramené de sa position à une position souhaitée (consigne) qui représente le centre du champ de vision (Fig.3.23)

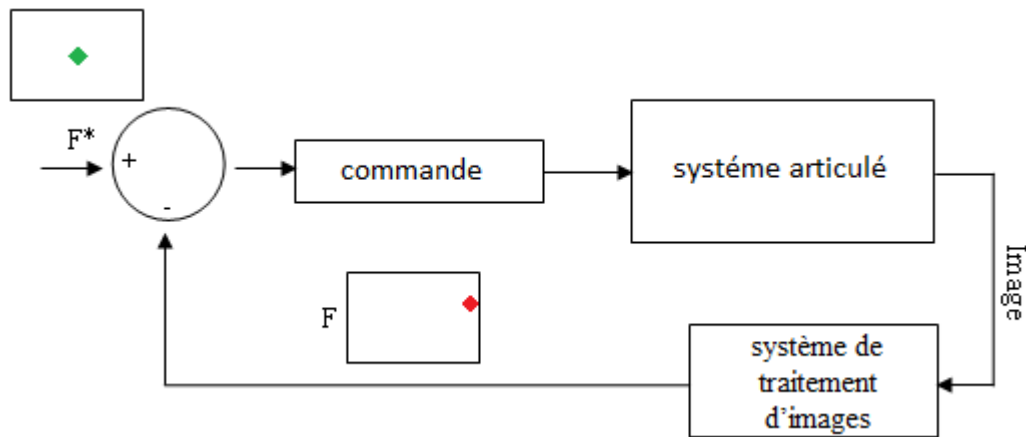


Fig.3.23 Boucle d'asservissement visuel

3.5.1 Calibration du pixel fournie par la camera

Pour connaître l'angle du champ de vision, des tests ont été effectués. Nous plaçons la caméra à une distance donnée face à plan verticale et nous calculons les limites de son champ de vision comme le montre la figure ci-dessous (Fig.3.24)

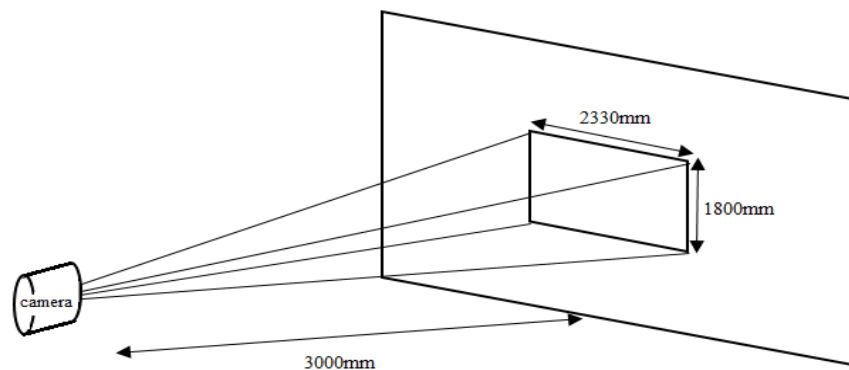


Fig.3.24 Méthode de détermination des paramètres du champ de vision

Les dimensions du rectangle du champ de vision de la caméra sur ce plan qui est d'une distance de 3000mm de la camera sont :

-Longueur : 2330mm

-Largeur : 1800mm

Les deux angles de vision sont calculés tel que (Fig.3.25) :

$$\tan^{-1} \frac{1165}{3000} = 21.21^\circ$$

$$\tan^{-1} \frac{900}{3000} = 16.07^\circ$$

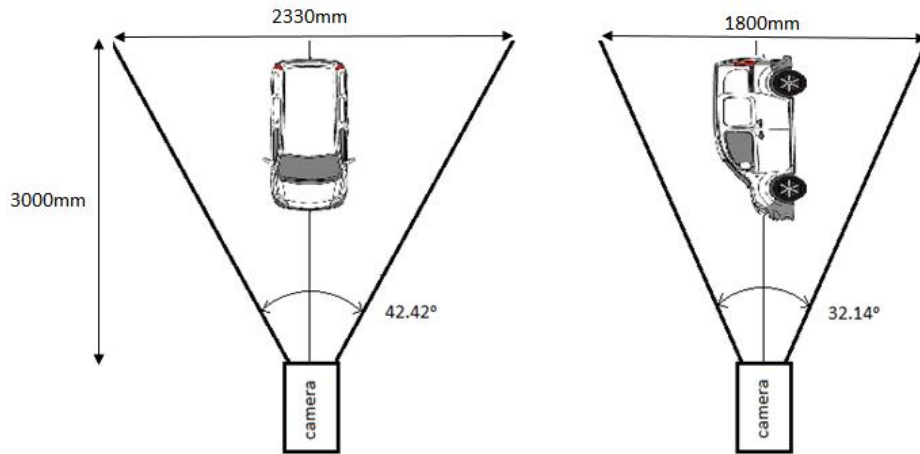


Fig.3.25 Angles du champ de vision

La caméra utilisée a une résolution de 320x240 qui fait que chaque pixel couvre un angle, vu que le pixel a une forme carrée, en calculant cet angle on obtient 0.132°.

3.5.2 Loi de commande

Pour la commande en position des moteurs, nous proposons une commande impulsionnelle, en utilisant comme signal d'entrée une impulsion de largeur T et l'amplitude $A=10.5$ V (Fig.3.26).

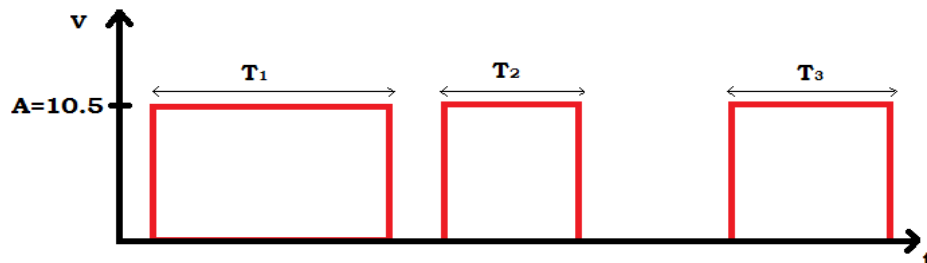


Fig.3.26 Signal d'entre

Pour connaître la nature de la relation liant le signal d'entrée qui est une impulsion de largeur d'impulsion T et la sortie qui est l'angle $\nabla\alpha = \alpha_i - \alpha_f$ telle que α_i est l'angle initial, α_f est l'angle final. Des essais ont été faits. Les résultats sont portés sur le tableau ci-dessous (Tab.3.2)

Tab.3.2 tableau de mesure entre la longueur de l'imputation et l'angle

T(s)	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$\nabla\alpha(^{\circ})$	00	03	07	14	18	23	30	32	35

En traçant les données du tableau (Tab.3.1), nous obtenons la courbe de la figure 3.27.

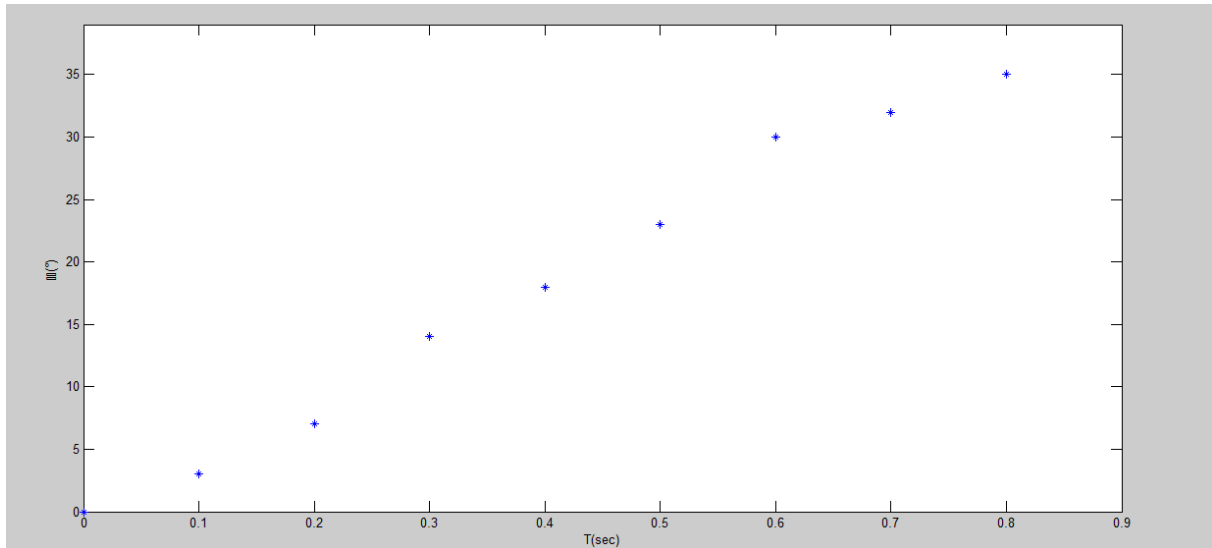


Fig.3.27 Tracé des points du Tab3.2

Nous remarquons que la relation entre l'entrée et la sortie est de nature linéaire et proportionnelle (3.28).

$$\nabla \alpha = kT \quad (3.28)$$

Où k est le rapport entre la sortie et l'entrée.

Pour trouver k de façon optimale, nous utilisons la méthode des moindres carrés » [14]. Il s'agit donc de trouver la fonction f du modèle retenu qui minimise la somme des carrés des écarts entre les valeurs y_i observées et les valeurs $f(x_i)$ données par le modèle. La fonction f doit donc minimiser l'expression (2.29) :

$$\sum_{i=1}^n (y_i - f(x_i))^2 \quad (3.29)$$

En premier lieu, nous calculons le point moyen G de coordonnées $(\bar{x}; \bar{y})$ où \bar{x} représente la moyenne des x_i et \bar{y} représente la moyenne des y_i où :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad , \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Lors d'un ajustement affine par la méthode des moindres carrés, la droite « d » servant à l'ajustement de y en x, la pente k de la droite estimée à « d » est :

$$k = \frac{cov(x, y)}{V(x)}$$

Où $cov(x, y)$ est la covariance de x et de y

$$cov(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Et $V(x)$ est la variance du caractère x

$$V(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Dans notre cas x est T et y est $\nabla\alpha$

Le point moyen est G (0.4 ; 18)

La covariance est $cov(T, \nabla\alpha) = 0.8297$

La variance est $V(T)=0.0178$

La pente de la droite $k = 46.67$

$\nabla\alpha = 46.67 T$

En représentant cette droite passant par l'origine (Fig.3.28)

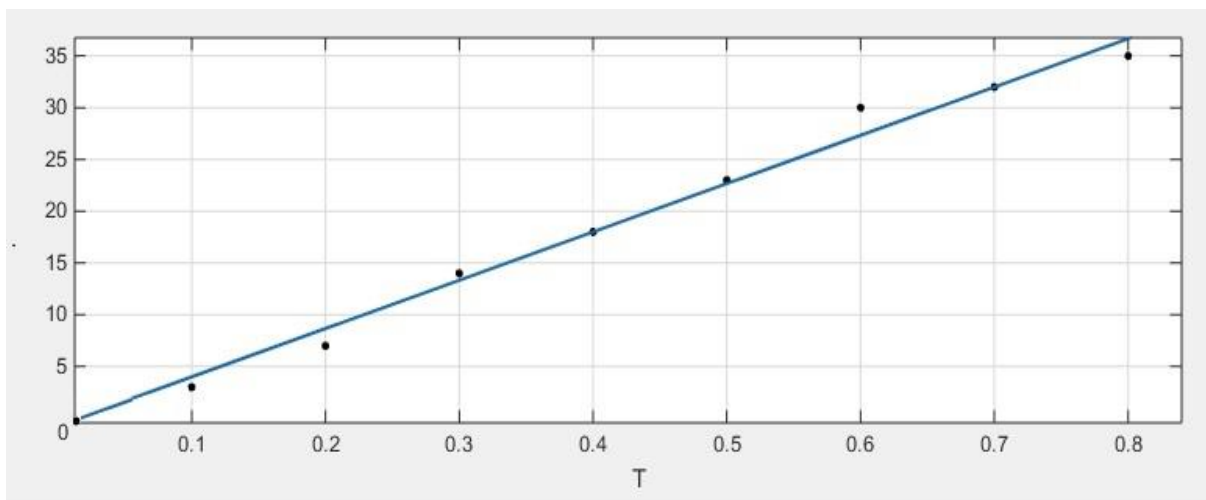


Fig.3.28 Tracé de la droite $\nabla\alpha(T)$

Par conséquent pour un signal de commande de type impulsion d'amplitude $A=10.5V$ et de largeur $T=1$ seconde, le moteur fait une rotation de 46.67° .

En utilisant les résultats trouvés dans la sous-section 3.5.1 pour déplacer le point représentant l'objet en mouvement d'un pixel à pixel voisin, le signal

d'amplitude $A = \mp 10.5V$ selon le sens de rotation adéquat est de largeur $T = 2.83ms$.

3.5.3 Exemple illustratif

Soient C le point consigne de coordonnées (160; 120) et P est le point représentant l'objet en mouvement de coordonnées (100 ; 100). L'erreur entre la consigne et la mesure est représenté dans le vecteur d'erreur e. dans notre cas $e = [60, 20]$. En convertissant cette erreur à deux signaux de commande d'amplitude $A = 10.5V$, le premier de largeur $60 * 2.83 = 168.6ms$ pour le moteur « H » qui contrôle la rotation horizontale et le seconde de largeur $20 * 2.83 = 56.2ms$ pour le moteur « V » qui contrôle la rotation verticale.

Si cette erreur est négative, nous calculons la valeur absolue pour avoir une largeur T positive mais l'amplitude de ce signal sera négative ($A = -10.5V$) pour contrôler le sens du moteur (Fig.3.29).

Tout ça est résumé dans un algorithme (Fig.3.30).

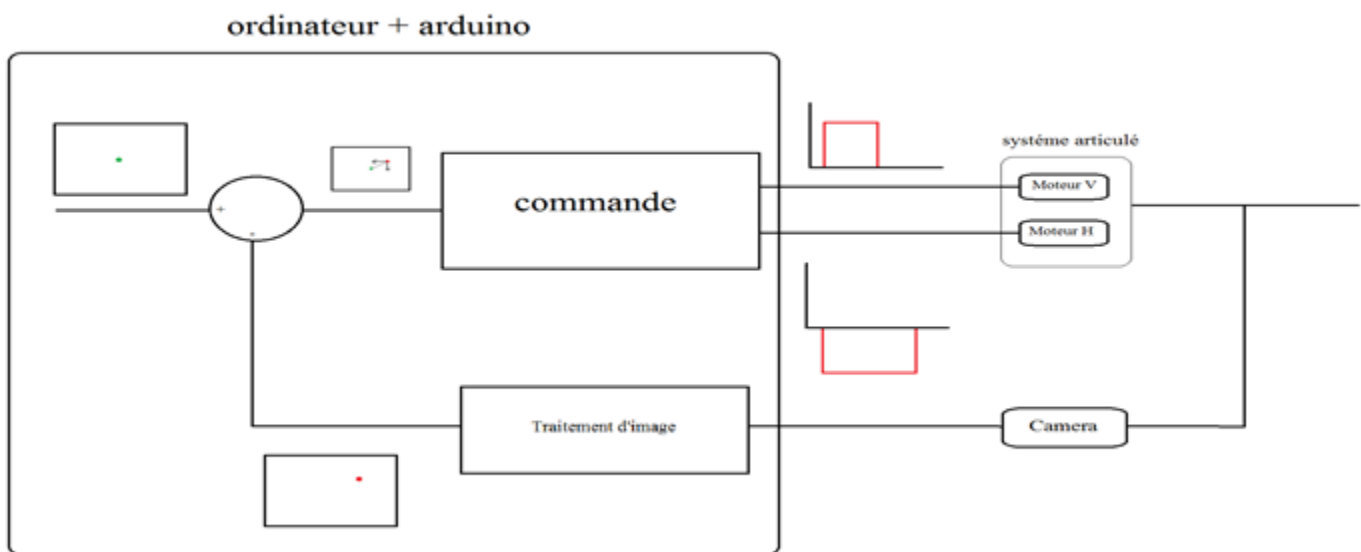


Fig.3.29 Exemple de commande d'un traitement

Algorithme : générer un signal de commande

// $(x_o; y_o)$ Coordonnées du pixel qui modélise l'objet en mouvement.

// $(x_c; y_c)$ Coordonnées du pixel consigne.

Début $x_c \leftarrow 160;$

$y_c \leftarrow 120;$

$LIPP \leftarrow 2.83$ // Largeur d'impulsion par pixel

Pour tout

Calcule des coordonnées $(x_o; y_o)$;

$x = x_c - x_o$; // Erreur sur l'axe des l'ordonnées

$T_x = x * LIPP;$

$y = y_c - y_o$; // Erreur sur l'axe des abscisses

$T_y = y * LIPP;$

Signal de commande du moteur MV

$T1 = \text{abs}(T_x)$; //largeur d'impulsion du signal de commande du moteur MV

Si $(T_x \geq 0)$ **faire**

$A1 = 10.5V$; // amplitude de impulsion du signal de commande du moteur MV

Si non si $(T_x > 0)$ **faire**

$A1 = -10.5V$; // amplitude de impulsion du signal de commande du moteur MV

Fin si

Signal de commande du moteur MH

$T2 = \text{abs}(T_y)$; //largeur d'impulsion du signal de commande du moteur MH

Si $(T_y \geq 0)$ **faire**

$A2 = 10.5V$; // amplitude de impulsion du signal de commande du moteur MH

Si non si $(T_x > 0)$ **faire**

$A2 = -10.5V$; // amplitude de impulsion du signal de commande du moteur MH

Fin si

Fin pour tout

Fin

Fig.3.30 Algorithme pour générer un signal de commande

3.6 Partie software du projet

Dans notre travail, le langage de programmation Matlab, version R2014b, a été utilisé. Pour rendre notre application plus conviviale, une interface graphique est prévue grâce à l'outil GUIDE (Graphical User Interface Design Environment) de Matlab qui permet de créer facilement des interfaces graphiques.

Avant de créer une interface, nous devons d'abord la définir. Dans notre cas, il s'agit de caméra de surveillance dotée d'une poursuite d'un objet en mouvement en mode automatique (poursuite en boucle fermé) ou en mode manuel (poursuite en boucle ouverte ou l'opérateur fait une poursuite manuelle) à l'aide de boutons visualisés sur l'interface et l'autre par joystick. Ainsi, sont prévues :

- deux fenêtres pour visualiser, la première pour l'aperçu de la caméra et le deuxième pour les résultats après le traitement de l'image pour la détection de mouvement.
- l'activation et la désactivation de la commande en boucle ouverte en joystick.
- l'activation du mode manuel pour activer toute commande en boucle ouverte.
- quatre boutons pour contrôler les quatre sens.
- des boutons pour garder l'aperçu de la caméra comme une image et d'autre pour gérer ces images (enregistrer, effacer...).
- l'affichage des images capturées.
- des boutons pour choisir le mode de vue vu que la camera peut être configuré en mode nocturne et diurne.
- un bouton d'arrêt du système.
- un bouton pour activer le mode automatique de la poursuite (boucle fermée).
- L'affichage de la vitesse de traitement (nombre de traitement par seconde).

En prenant en considération tous ces points, nous proposons l'interface de la figure (Fig.3.31)

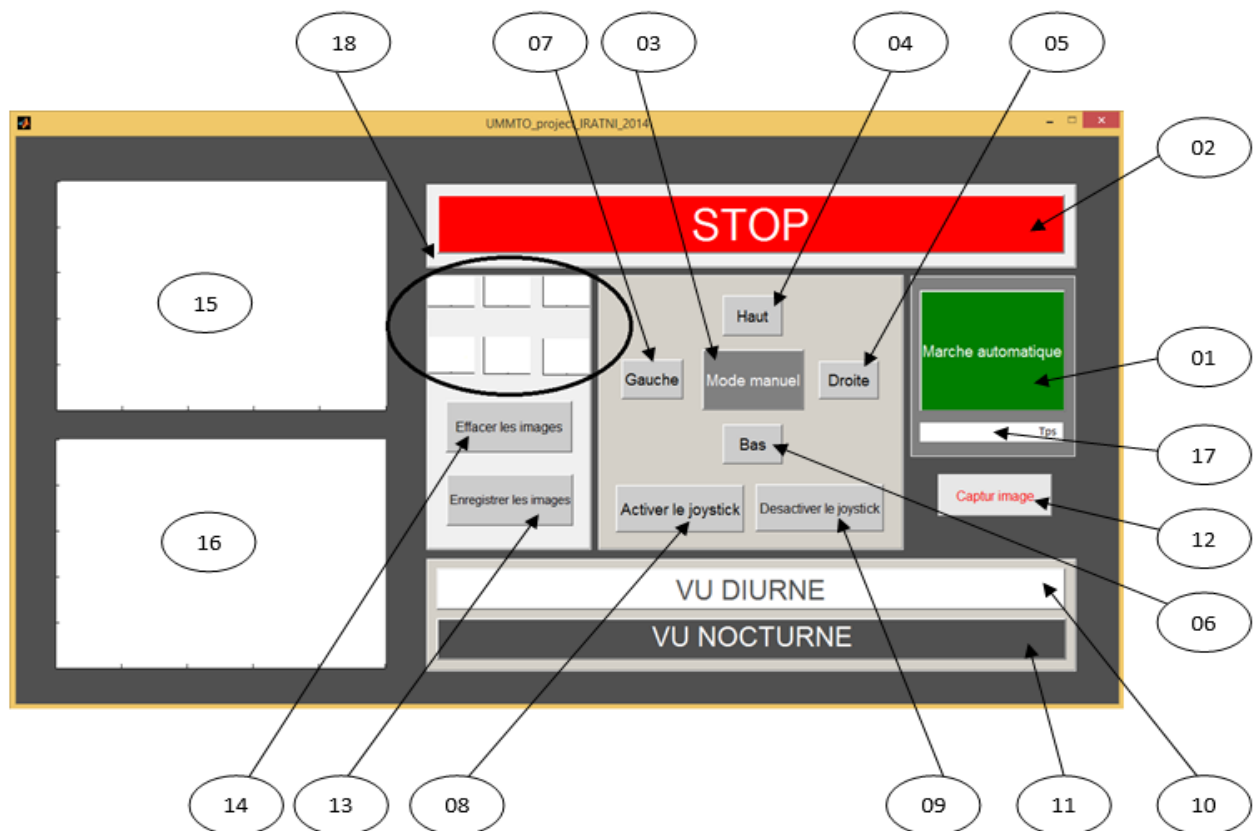


Fig.3.31 Interface du projet

- **Description des fonctions**

1. Le bouton « Marche automatique » permet de passer à une détection et une poursuite automatique d'un objet en mouvement.
2. Le bouton « STOP » permet de stopper toutes les fonctions de l'application.
3. Le bouton « Mode manuel » active le contrôle de la camera d'une façon manuelle et désactive la marche automatique.
4. Le bouton « Haut » guide la camera vers le haut si on est en mode manuel.
5. Le bouton « Droite » guide la camera vers la droite si on est en mode manuel.
6. Le bouton « Bas » guide la caméra vers le bas si on est en mode manuel.

- 7.** Le bouton « Gauche » guide la camera vers le gauche si on est en mode manuel.
- 8.** Le bouton « activer le joystick » permet de passe en commande de la camera par joystick et bloque la commande par bouton.
- 9.** Le bouton « désactiver le joystick » permet de passe en commande par bouton toute en désactivant la commande par joystick.
- 10.** Le bouton « VU DIURNE » permet de choisir comme mode de vue le mode diurne.
- 11.** Le bouton « VU NOCTURNE » permet de choisir comme mode de vue le mode nocturne.
- 12.** Le bouton « capture image » permet de capturer une frame.
- 13.** Le bouton « enregistrer les images » permet enregistrer les frames capturé.
- 14.** Le bouton « effacer les images » permet effacer les frames capturé.
- 15.** Une fenêtre pour afficher la séquence en temps réel fournie par la camera.
- 16.** Une fenêtre pour afficher la séquence en temps réel traiter pour détection de mouvement.
- 17.** L'affichage de la vitesse de traitement en Tps (Traitement par seconde).
- 18.** L'affichage des images capturées (maximum 6 images).

La figure 3.32 montre l'ensemble du projet et de ses composantes.

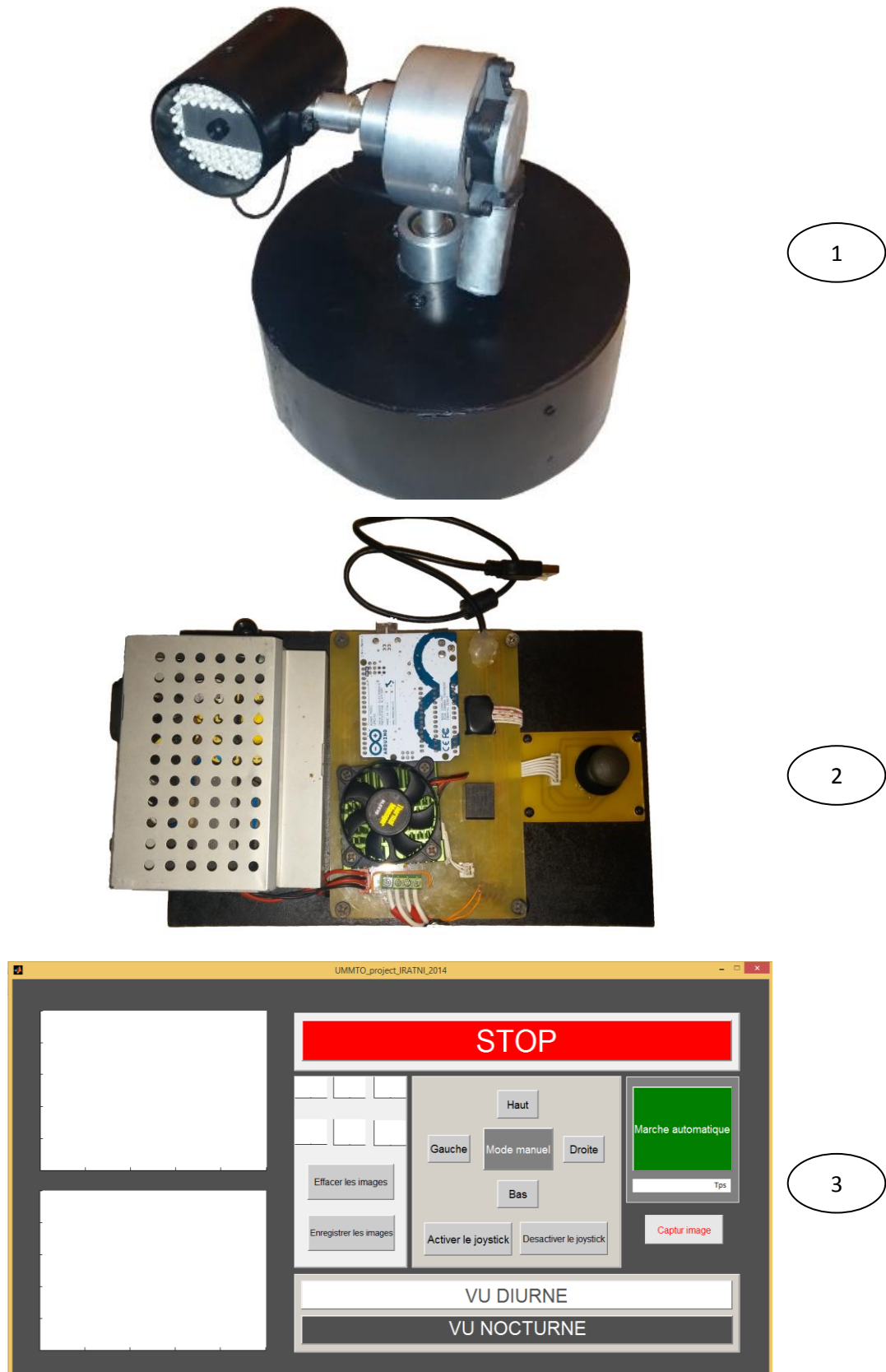


Fig.3.32 Parties conçu dans ce projet

1-système articulé, 2- circuit de puissance et de commande, 3-interface graphique

3.8 Conclusion

Après une description des objectifs de notre réalisation et de ses différentes composantes, les étapes permettant la commande à savoir l'étape de la détection de mouvement, l'étape de la modélisation des moteurs, le calcul de la loi de commande et l'étape de la réalisation pratique de la commande et de l'interface. Dans la première étape, nous avons présenté une synthèse d'un algorithme permettant de détecter le mouvement d'un objet et de calculer ses coordonnées. En seconde partie, un modèle est calculé sous forme d'une fonction de transfert du second ordre à partir des équations physiques régissant le fonctionnement du moteur à courant continu. Ceci est effectué en utilisant la méthode des moindres carrés. Dans la troisième partie, nous réalisons la commande de l'asservissement visuel en utilisant les paramètres géométriques de la caméra et les pixels. La dernière partie a été consacrée à la création de l'interface et présentation du projet.

Conclusion générale

Notre travail consistait à étudier, sur le plan théorique et pratique, la commande d'un système composé d'un système articulé à deux axes et d'une caméra. Le fonctionnement de cet ensemble piloté par une carte de développement ARDUINO consiste à suivre, en temps réel, un objet en mouvement en commandant le système articulé de telle manière à ce que cet objet soit toujours perçu dans le champ de vision de la camera.

Pour atteindre cet objectif, nous avons subdivisé notre travail en deux étapes. La première est consacrée à la réalisation d'une boucle de commandes numérique en position du moteur à courant continu assurant le mouvement de rotation du système articulé. La détection, la localisation de l'objet en mouvement a fait l'objet la seconde partie relative à la vision.

Pour la mise en œuvre de la commande du système articulé en première lieu nous avons étudié les constituants du système articulé puis nous avons réalisé les étapes permettant la commande de ce dernier à savoir l'étape de modélisation, l'étape de calcul de la loi de commande, l'étape de l'essai et validation de la loi de commande calculée. L'étape de modélisation comprend deux étapes. Dans la première, à partir des équations physiques régissant le fonctionnement des moteurs nous avons déterminé la structure du modèle sous forme de fonction de transfert du second ordre. Dans la seconde, nous avons relevé pratiquement la réponse indicielle des moteurs puis nous avons utilisé le principe des moindres carrés pour déterminer les paramètres optimaux impliquant la réponse indicielle la plus proche possible de celle relevé pratiquement. Pour rendre le fonctionnement de ses moteurs performant entre autre réduire le temps de réponse, il est nécessaire d'introduire un régulateur.

Dans la partie vision, pour détecter les objets en mouvement la stratégie utilisée est simple. Elle consiste à faire une soustraction de deux frames

successive puis segmenter ce résultat en le binarisant. Pour améliorer les résultats, nous avons introduit opérateurs morphologiques. La localisation d'objet passe par la détermination du modèle de la caméra à partir duquel les coordonnées du centre de gravité de l'objet en mouvement peuvent être déterminés par la connaissance de ses coordonnées dans l'image.

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

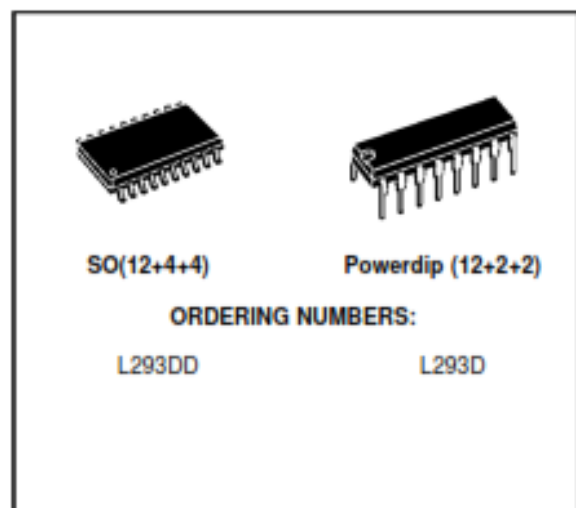
The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

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.

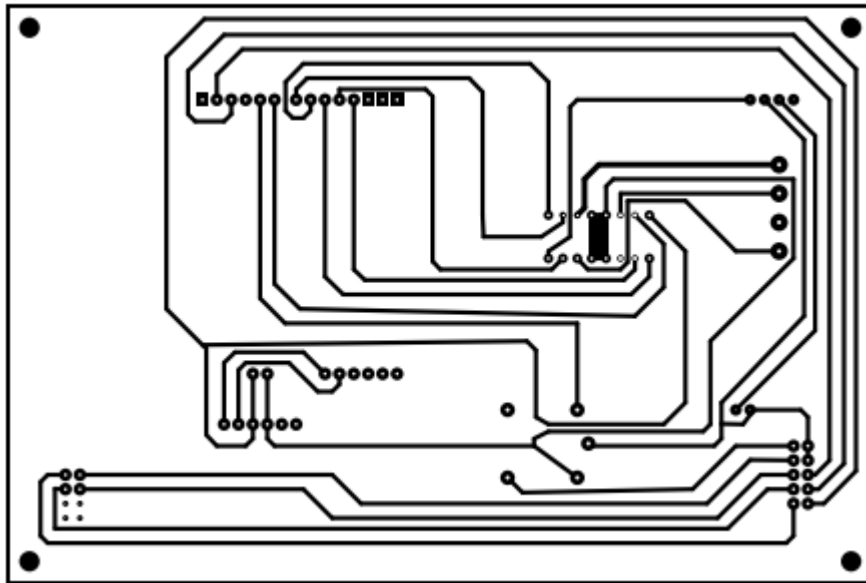
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{cc}	Supply Voltage	36	V
V_{ccs}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 μ s non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at $T_{pma} = 90$ °C	4	W
T_{stg}, T_j	Storage and Junction Temperature	- 40 to 150	°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.



Annexe 2 Circuit électronique globale



Product Overview

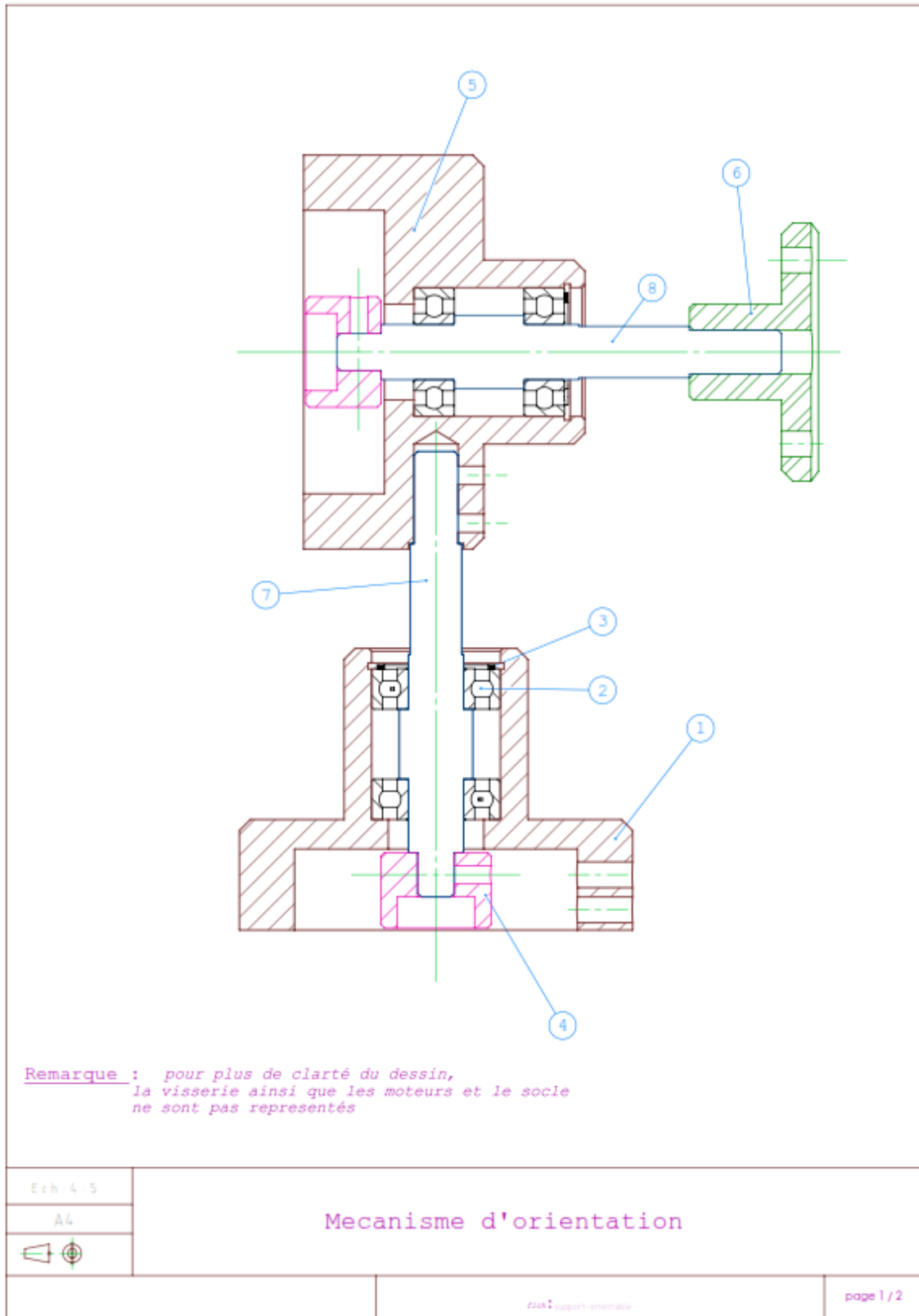
The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed 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 Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

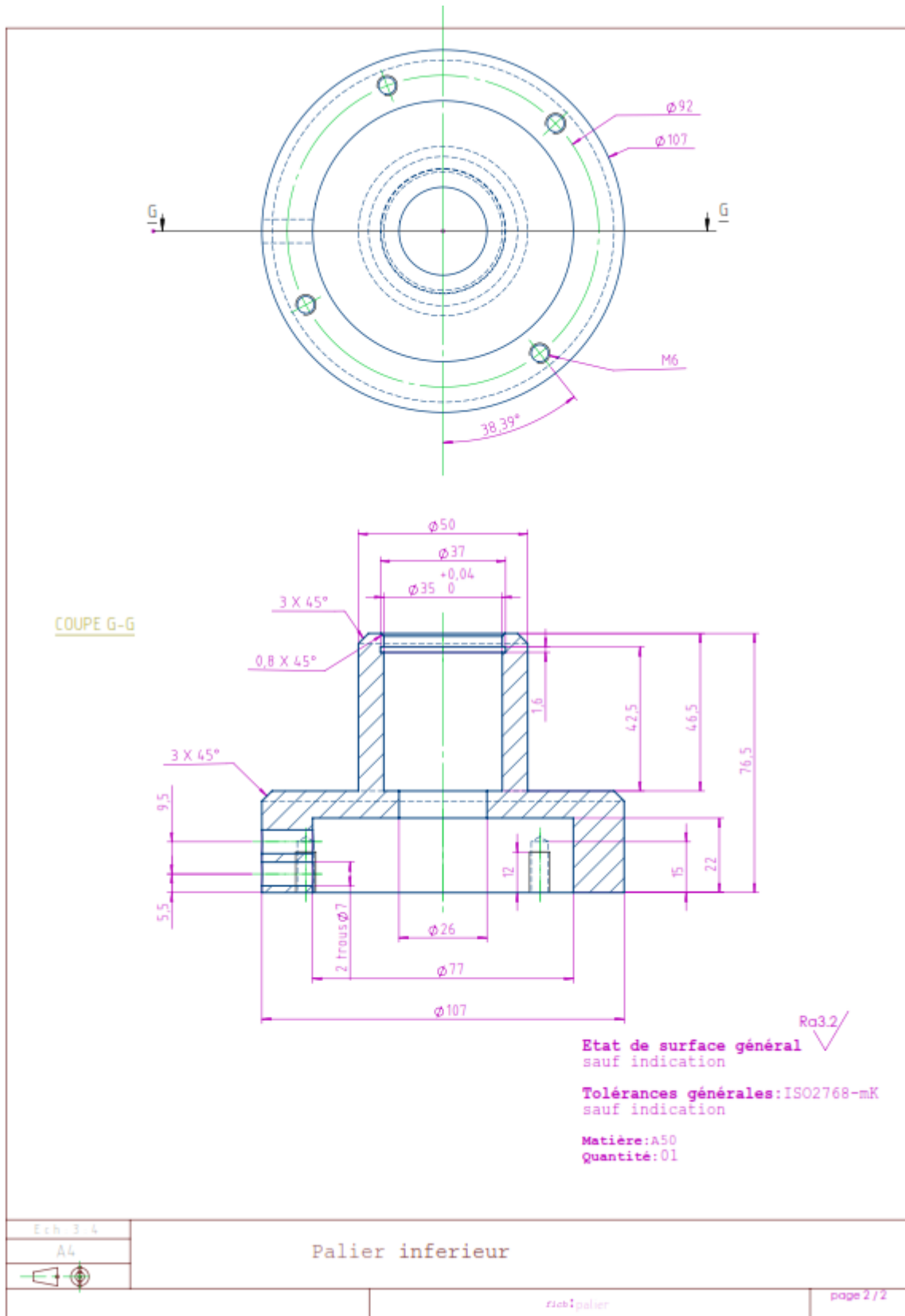
Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

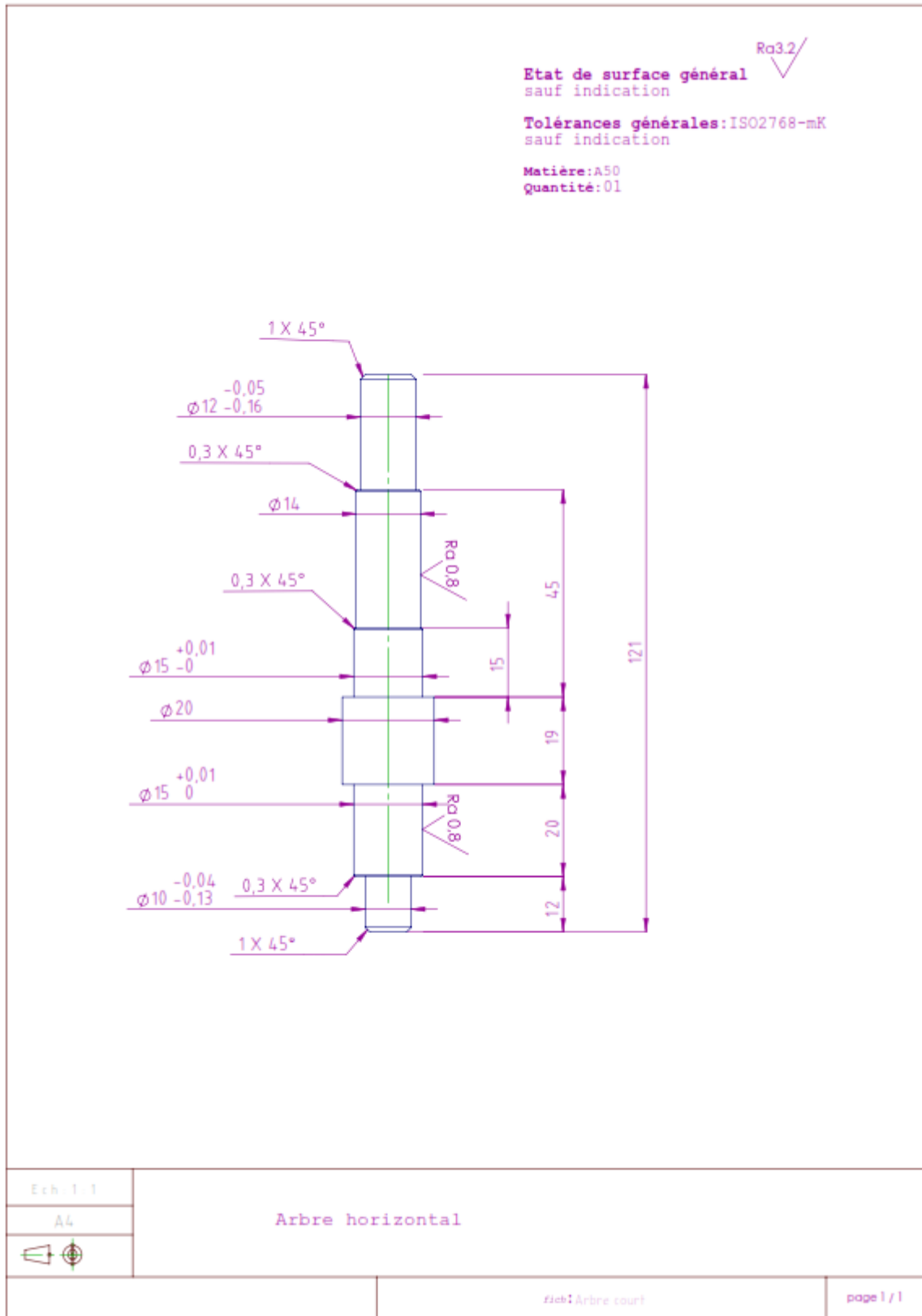
Annexe 3 ARDUINO UNO



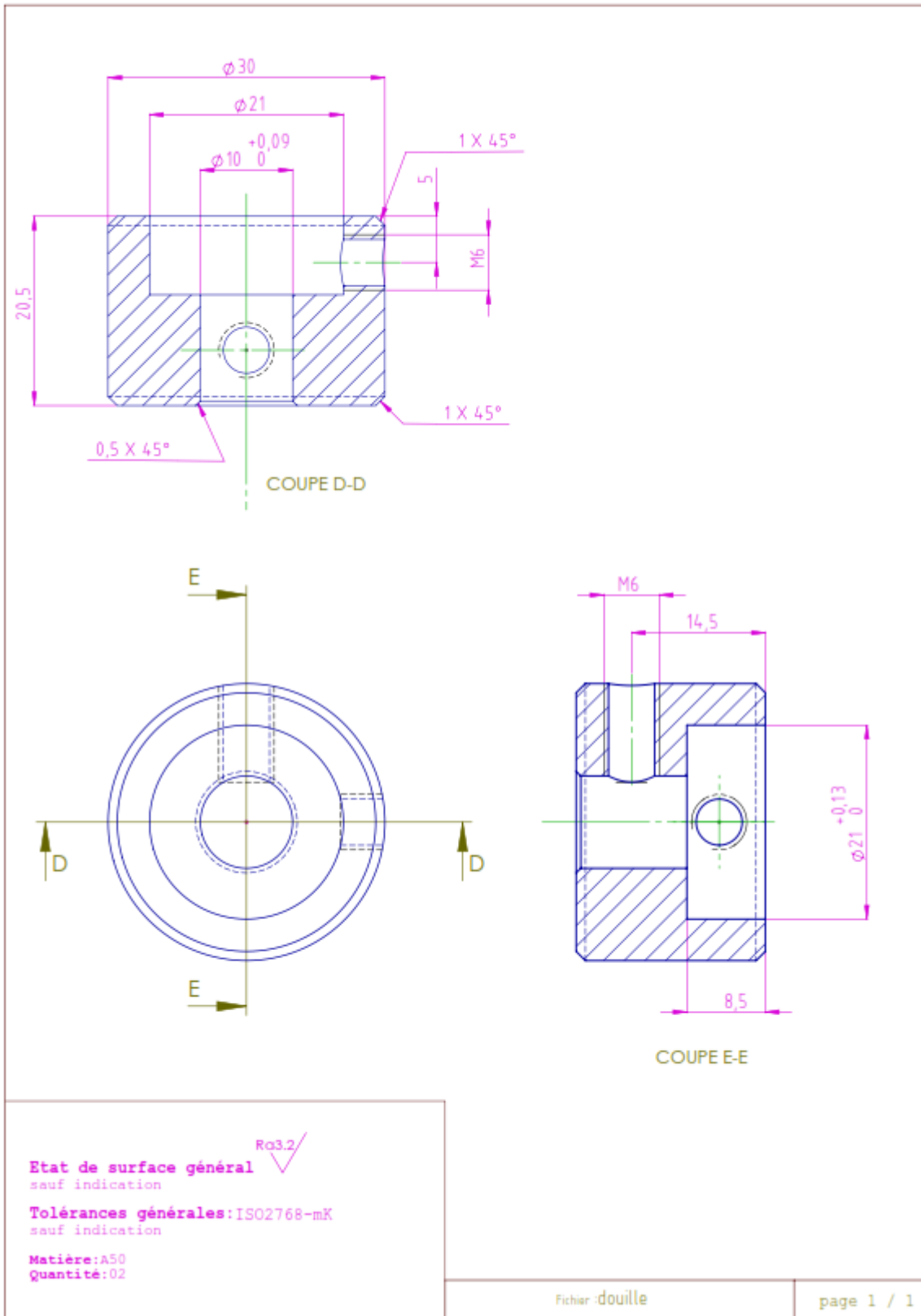
Annexe 4 Mécanisme d'orientation



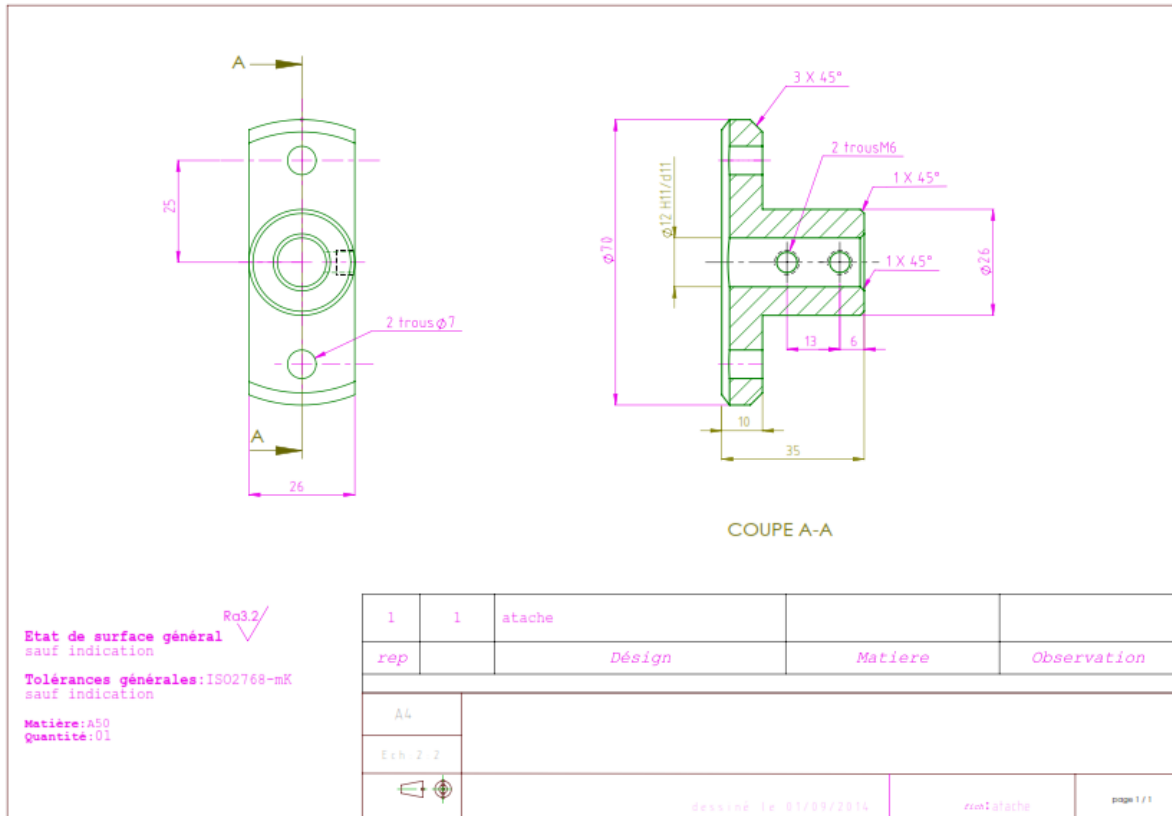
Annexe 7 Palier inférieur



Annexe 8 Arbre horizontal



Annexe 9 Douille



Annexe 10 Attache

Bibliographie

- [1] ARDUINO UNO. Datasheet, www.arduino.cc.
- [2] Barron, J. L., Fleet, D. J. et Beauchemin, S. S. (1994). Performance of optical flow techniques. *Journal of Computer Vision*, 12(1):43 77.
- [3] Bellier, M. (1984). *Machines électrique*. Édition delagrave.
- [4] Boukas, M (2001). *Systèmes asservis*. L'école de polytechnique de montréal.
- [5] Burel, G. (2001). *Introduction au traitement d'images, simulation sous Matlab*, Lavoisier.
- [6] Chaumette, F et Hutchinson, S. (2006) Visual servo control, Part I: Basic approaches. *Robotics and Automation Magazine*, pages 82 90.
- [7] Horn, B. K. P. et Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, pages 185 203.
- [8] Jain, R. et Nagel, H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. *Pattern Analysis and Machine Intelligence*, pages 206 214.
- [9] L293D, datasheet, www.st.com.
- [10] LTE5128A, datasheet, www.alldatasheet.com.
- [11] Lucas, B. D. et Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Artificial Intelligence*, pages 674 679, Vancouver, Canada.
- [12] Ma, Y. F. et Zhang, H. J. (2001). Detecting motion object by spatio-temporal entropy. *Multimedia and Expo* pages 265 268, Tokyo, Japon.
- [13] Malis, E. (1998) Contributions à la modélisation et à la commande en asservissement visuel. *Thèse de doctorat, Université de Rennes1*, France.
- [14] Rossignol, C. (2010). *Statistiques à deux variables, Ajustements affines*.
- [15] Triki, A. (2010) contribution à l'étude et réalisation d'un asservissement comprenant une table tournante, un robot et un système de vision artificielle. *Thèse de magister Université de Mouloud MAMMERI de tizi ouzou*, Algérie.
- [16] Verbeke, N. (2007). Suivi d'objet en mouvement dans une séquence vidéo. *Thèse de doctorat Université de paris DESCARTES*, France.
- [17] Yang, T., Li, S. Z., Pan, Q. et Li, J. (2004). Real-time and accurate segmentation of moving objects in dynamic scene. *Workshop on Video Surveillance & Sensor Networks*, pages 136 143, New York, usa.