RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE BATNA
FACULTE DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE

Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Électronique

OPTION : Robotique

Thème :

NAVIGATION VISUELLE D'UN ROBOT MOBILE

Etudié par :

Zoulikha BOUHAMATOU

(Ingénieur d'état en électronique)

Soutenu le : 22/04./2015

Devant le jury :

• Dr. Abdessemed Yassine	MCA à l'Université de Batna	Président
• Dr. Abdessemed Foudil	Prof à l'Université de Batna	Rapporteur
• Dr. Melaab Djamel	MCA à l'Université de Batna	Examinateur
• Dr. Ziet Lahcen	MCA à l'Université de Sétif	Examinateur

2014/2015

Table des matières

CHAPITRE 3 : La Modélisation Et La Calibration D'une Camera

CHAPITRE 4 : L'asservissement visuel

REMERCIEMENTS

En premier lieu, je remercie Dieu, le tout puissant, pour m'avoir donné, la patience, la volonté et la force nécessaires pour achever ce travail.

Je tiens à dire ma reconnaissance envers Monsieur le Professeur ***Abdessemed Foudil*** *qui a accepté sans réserve, de diriger ce mémoire. Il s'y est grandement impliqué par ses directives, ses remarques et suggestions, mais aussi par ses encouragements. Je tiens à le remercier aussi pour cette liberté qu'il m'a permise, sans laquelle le chercheur ne saurait affirmer sa manière de penser et de procéder, sa manière d'être, bref toute sa personnalité.*

Je remercie le président ***Dr.Abdessemed Yassine*** *pour avoir accepté de juger ce travail, et les membres du jury* ***Dr. Melaab Djamel*** *et* ***Dr. Ziet Lahcen*** *qui ont pris de leurs temps pour lire, juger ce travail.*

Enfin, je tiens à remercier tous ceux qui de près comme de loin m'ont aidé, soutenu et encourager aux moments opportuns. A tous, un grand MERCI !

Dédicace

Ma mère — Pour la disponibilité et l'aide

Mon père — **Tu as dépensé toute ta vie pour m'assurer une éducation exemplaire**

Ma sœur — **La source de mon succès**

Mes frères — *Pour les bons moments et la spontanéité à rendre service*

Fatima — Ma réussite est aussi la votre

Les deux Khadidja et djawhara — ***Pour leur soutien et pour m'avoir donné le goût d'apprendre***

Je dédie ce mémoire
Couronnement de mes études
Universitaires

Liste des figures

Liste des Tableaux

Liste des abréviations

f	La distance focal.
(I_x, I_y)	La longueur et la hauteur d'un pixel.
s	Le facteur d'échelle qui correspond à la profondeur du point 3D observé.
(k_u, k_v)	Le facteur d'échelle
(u_0, v_0)	Le point principal d'une l'image.
Rot (Δ, α)	Matrice de rotation : Une rotation α autour d'un axe Δ.
RCS	Les coordonnées du système robot (*Robot coordinate system*).
ICS	Les coordonnées du système image (*Image coordinate system*).
WCS	Les coordonnées du système monde (*Word coordinate system*).

INTRODUCTION GÉNÉRAL

La perception est un élément crucial lorsqu'il s'agit de faire naviguer un robot mobile dans un environnement inconnu et non structuré pour accomplir une mission d'exploration ou de portage. Pour atteindre cet objectif, le robot doit pouvoir être capable de percevoir et de réagir en cas de danger de collision ou de renversement. Jadis les robots mobiles qui réussissaient à naviguer dans des environnements inconnus ou partiellement connus, en utilisant des capteurs ultra-sons ou infra rouges ou encore des rayons laser pour se positionner et détecter les obstacles. Avec les caméras modernes et la possibilité de calcul offerte par les calculateurs, il est désormais possible de naviguer avec souplesse toute en ayant des performances meilleures, tout en fusionnant s'il y'a nécessite les autres capteurs.

On définit un robot mobile comme un véhicule intelligent équipé de capacités de perception, qui lui permettent de définir et identifier le chemin à suivre et les différents obstacles pour une navigation autonome dans un environnement inconnu. Les robots mobiles sont classés selon leurs types de locomotion. On peut citer :

- les robots mobiles qui se déplacent dans un espace terrestre comme les robots mobiles à roues, les robots mobiles à chenilles, les robots mobiles marcheurs, ou encore les robots mobiles rampants.
- les robots mobiles aériens comme les drones.
- les robots mobiles aquatiques.

Les robots mobiles les plus largement utilisés et développés sont les robots mobiles à roue. En effet, le contrôle des roues du robot mobile est plus facile et l'étude de leur navigation est plus facile. Les roues procurent une meilleure stabilité au véhicule et dissipent moins d'énergie. Cependant, l'inconvénient réside dans leur utilisation que sur des terrains durs et relativement plats. Par ailleurs, les robots mobiles à pattes sont caractérisés par de bonnes propriétés de locomotion en milieu difficile comme les milieux forestiers, agricoles et les montagnes.

Au cours des dernières années, le robot mobile est devenu de plus en plus important et largement utile dans la vie quotidienne des êtres humains. Parmi les types de robots mobiles à roues on trouve : les robots mobiles omnidirectionnels ou holonome qui ont attiré beaucoup d'attention dans le cadre des compétitions internationales de robotique « *Robocup* ». Contrairement au premier type, il existe des robots mobiles non holonomes comme les robots mobiles type voiture. Le robot mobile omnidirectionnel est un robot qui possède trois degrés

de liberté. Il permet de réaliser une translation longitude avancée ou reculée ; une translation latérales en allant vers la droite ou vers la gauche, et une rotation sur lui même en tournant vers la droite ou vers la gauche. Donc, les robots mobiles holonomes ont une capacité agile pour se déplacer dans n'importe quelle direction ainsi que de tourner facilement sur eux-mêmes. Par contre les robots mobiles non holonomes disposent seulement de deux degrés de liberté sur un plan, puisque les translations latérales sont impossibles. Il permet de réaliser une translation longitude avancée ou reculée et une rotation qui fait tourner le robot mobile vers la droite ou vers la gauche.

La navigation d'un robot mobile peut généralement être définie comme le processus de déterminer un chemin approprié entre un point de départ et un point d'arrivé et permettre au robot de le suivre. En particulier, dans les dernières décennies, la navigation visuelle des robots mobiles est devenue une source de contributions à d'innombrables recherches. Avec vision, le robot est capable de percevoir son environnement en recevant suffisamment d'informations lui permettant de se localiser et trouver des chemins efficaces et sécuritaires de manière autonome afin d'éviter des collisions et arriver à sa destination. L'asservissement visuel constitue le processus de contrôle basé sur les informations fournies par une ou plusieurs caméras.

L'objectif de cette recherche est de proposer une méthode de contrôle cinématique du robot omnidirectionnel pour parvenir à la stabilisation déplacement point à point et de poursuite de trajectoire. Notre objectif et de permettre au robot mobile de poursuivre une trajectoire tracée sur le sol grâce au seul capteur optique : Une caméra webcam.

Le mémoire est organisé en quatre chapitres répartis comme suit :

Chapitre 1 : Le premier chapitre donne l'état de l'art sur différents types de vision monoculaire et stéréovision et vision omnidirectionnelle et les dernières recherches réalisées dans ces domaines.

Chapitre 2 : Le deuxième chapitre décrit la modélisation cinématique et le contrôleur cinématique du robot mobile holonome "*Robotino*", et les résultats des simulations

Chapitre 3 : Le troisième chapitre est consacré à la modélisation et le calibrage de la caméra utilisée.

Chapitre 4 : Dans le quatrième chapitre, nous développons une méthode de poursuite de trajectoire par vision monoculaire. La méthode proposée est principalement composé de deux parties : La première, concerne l'acquisition puis le traitement d'image adéquat qui permet une segmentation de l'image et l'obtention de la courbe de la trajectoire à poursuivre. La seconde partie, offre au robot mobile les points de référence de la trajectoire désirée pour accomplir la tâche de poursuite. Tous les calculs permettant la représentation des points dans le repère particulier ont été développés en utilisant les transformations matricielles nécessaires à la boucle d'asservissement.

CHAPITRE I :
ÉTAT DE L'ART

CHAPITRE I

ÉTAT DE L'ART

LES SYSTÈMES DE VISION DANS LA ROBOTIQUE

I.1. Introduction :

La réalisation d'un système robotique mobile intelligent est liée directement à la puissance d'apprendre ou de s'adapter à son environnement; donc il est nécessaire d'utiliser des capteurs qui fournissent la perception requise de l'environnement pour une prise de décision intelligente. Le capteur utilisé, est une caméra webcam.

Dans le cadre de cette thèse, nous nous intéressons au problème de la navigation d'un robot mobile guidé par une caméra monoculaire placée au dessus du robot. Dans ce chapitre, nous présentons les différents systèmes de vision adaptés à la robotique.

I.2. Les systèmes de vision:

En robotique les systèmes de vision sont basés sur l'utilisation d'une caméra CCD (*Charge Coupled Device*). L'utilisation d'une seule camera ne peut fournir qu'une seule information 2D qu'on appelle vision monoculaire. Pour obtenir des informations 3D, il existe différentes techniques qui sont généralement liées à l'adjonction d'un autre capteur; par exemple: l'utilisation de deux caméra nous permet de reconstruire la structure 3D des objets avec deux images, on l'appelle: la stéréovision. On peut aussi ajouter un système de réflexion de type miroir donnant ainsi la vision omnidirectionnelle. Donc, on peut dire qu'on peut identifier trois techniques: La vision monoculaire, la stéréovision et La vision omnidirectionnelle.

I.2.1. La Vision Monoculaire :

Dans le cadre de cette thèse, nous nous intéressons à la vision monoculaire qui utilise une seule caméra qui observe directement la scène. L'inconvénient de ce système est: son champ visuel est limité et ne permet pas de restituer la profondeur de la scène. Il ne fournit qu'une seule représentation bidimensionnelle de l'image et de l'espace tridimensionnel, ce qui n'est pas suffisant. La profondeur est déterminée à partir d'une ou plusieurs images. En vision par

ordinateur, ces techniques de reconstruction 3D à partir d'images sont appelées techniques de « Shape-From-X », où X peut être « Stéréo », « Motion », « Shading », etc.

L'analyse de l'ombre "Shape-From-Shading":

La technique de « *Shape-From-Shading* » (SFS) a été développé par Horn [Horn75]; au début des années 70. Cette technique construit une surface 3D à partir d'une image 2D quand les propriétés de la réflexion de surface sont connues. Cette correspondance traite les surfaces avec des propriétés de réflexion de *Lambert*.

La modélisation du processus de formation des images permet en effet d'exprimer le niveau de gris d'un pixel de l'image en fonction de la direction de la source de lumière et de la normale à la surface de l'objet. Le but du SFS est donc, de retrouver la source de lumière et la normale à la surface pour chaque pixel de l'image, étant donné un niveau de gris de l'image (figure (1.1)) [Rémi10].

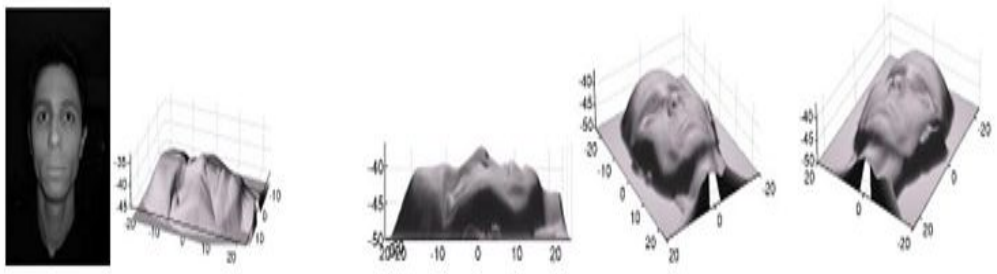


Figure (1.1) : Reconstruction 3D d'un visage par Shape-From-Shading. Source [Prados06].

L'analyse de texture "Shape- From- texture" :

Le premier qui a introduit la technique de *"Shape-From-texture"* est Gibson [Gibson50] en 1950. Il étudie la récupération des coordonnées 3D d'une surface dans une scène, par l'analyse de la déformation de sa texture projetée dans une image. Le problème de "Shape-From-texture" est généralement décomposée en deux étapes indépendantes; la première étape consiste à mesurer la distorsion de la texture dans l'image et la seconde est de récupérer les coordonnées de la surface de cette distorsion de texture (figure (1.2)).

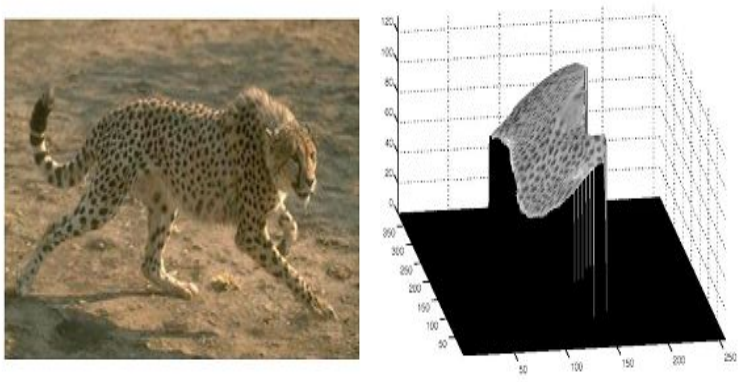


Figure (1.2): Exemple de reconstruction 3D d'un guépard par *Shape From Texture*.
Source [Forsyth02]

Shape-From-Contour:

Cette méthode donne une estimation de la forme 3D de tous les contours de l'objet. Le modèle 3D de l'objet peut être calculé à partir des contours de l'objet par les images prises de diverses directions d'observation; en utilisant comme méthode le "*volume-craving*" voir [Klette98]. Néanmoins, la méthode est incapable de récupérer certaines régions concaves sur la surface et la précision de récupérer le modèle 3D dépend de la résolution des directions d'observations. Néanmoins, cette méthode présente moins de restrictions sur les conditions d'éclairage, ce qui en fait une approche plus robuste pour obtenir le modèle 3D préliminaire d'un objet (figure (1.3)).

Figure (1.3): Application du Shape-From-Contour à la rectification d'images.
Source [Courteille07].

Shape-From-Photometric Stereo:

Woodham [Woodham80] a introduit la technique appelée en anglais " *Shape-From-Photometric Stereo* " en 1980. Cette méthode utilise deux ou plusieurs images d'intensité de l'objet en gardant le même point de vue sous différentes conditions d'illumination, en modifiant la direction de l'éclairage entre les clichés [Rémi10]. Le procédé de Photometric Stereo est capable d'obtenir rapidement les orientations d'une surface locales à partir de l'intensité des images de l'objet éclairé par des sources lumineuses. Les vecteurs d'orientation de la surface sont intégrés, soit globalement ou localement, pour fournir les valeurs de la profondeur de la surface. Cette technique permet d'obtenir de meilleurs résultats comme l'illustre la figure (1.4) :

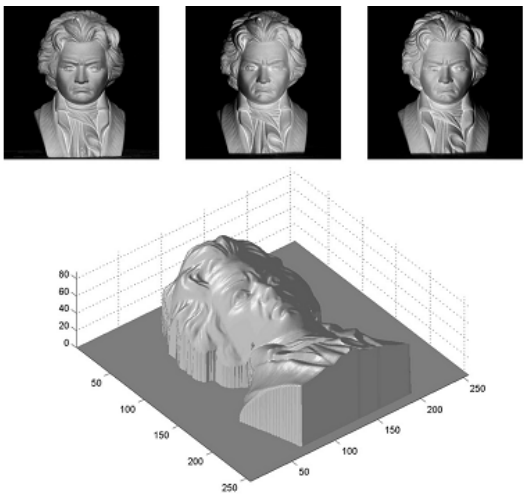


Figure (1.4): Reconstruction 3D d'un buste par stéréophotométrie.
Source [Durou07].

Shape-From-Motion:

Cette technique est connue sous le nom de "*Structure-From-Motion"* (SFM). Elle exploite le mouvement relatif entre la caméra et la scène. Similaire à la technique stéréo, le processus peut être divisé en sous-processus de la découverte de la correspondance de trames consécutives et la reconstruction de la scène. Les différences entre les trames consécutives sont, en moyenne**,** plus petites que celles de la paire stéréo typique, parce que les séquences d'images sont échantillonnées à un taux élevé**.** Contrairement à la stéréo**,** en mouvement le déplacement relatif 3D entre la visualisation de la caméra et la scène n'est pas nécessairement causé par une seule transformation 3D (figure (1.5)).

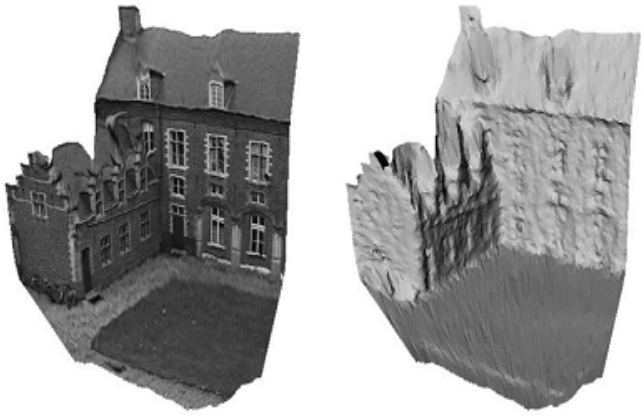
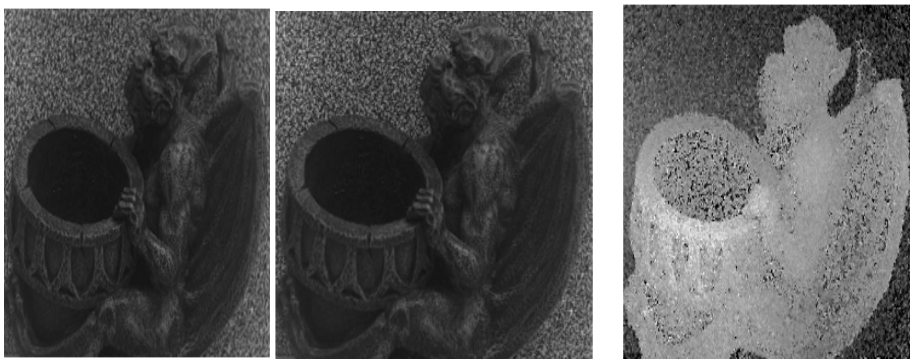


Figure (1.5) : Exemple de modèles 3D obtenus par *Shape From motion*.
Source [Pollefeys04]

Shape-From-Focus/Defocus:

Le problème de *" **Shape-From-Focus/Defocus*** " **est** estimé la surface d'une scène 3D à partir d'un ensemble de deux ou plusieurs images de la scène**.** Les images sont obtenues en modifiant les paramètres de la caméra (**généralement le** paramètre de focale ou la position axiale du plan d'image**),** et prendre **le** même point de vue.

La différence entre " ***Shape-From-Focus"*** et ***"Shape-From-Defocus"*** **est :** dans le premier cas, il est possible de modifier dynamiquement les paramètres de la caméra pendant le processus d'estimation de surface**,** tandis que dans le second cas ce n'est pas autorisé.



(a) (b) (c)

Figure (1.6) : Résultat obtenu par Shape-From-Defocus.
(a) et (b) : premières ensembles d'image originale acquises (de taille 640 × 480 pixels) L'image (a) est près de focus (à 0,70 m), l'image (b) est de focus (à 0,77m) (c) Carte de profondeur obtenue. Source [Favaro05]

De nombreux travaux de ce type de vision (vision monoculaire) sont utilisés dans différents domaine en robotique. Dans l'approche proposée par [Michel04]: la méthode dédiée à l'application se base sur l'utilisation des capteurs embarqués sur le robot afin de fournir une bonne estimation de la localisation du robot. Cette méthode, devra être réalisée en cours de mouvement, elle devra être la moins couteuse possible en temps de calcul. Elle se base sur une information odométrique corrigée en position et en orientation à l'aide des distances mesurées par les capteurs à ultrasons. L'utilisation d'un système de vision monoculaire vient pour préciser l'orientation. [Eric *et al*05] déterminent la localisation d'un robot mobile par rapport à une séquence vidéo d'apprentissage. Dans un premier temps, le robot est conduit manuellement sur une trajectoire et une séquence vidéo de référence est enregistrée par une seule caméra. Puis un calcul hors ligne donne une reconstruction 3D du chemin suivi et de l'environnement. [Eric06] présente la réalisation d'un système de localisation pour un robot mobile autonome fondé sur la vision monoculaire. Les travaux de Renaud [Renaud08] concernant l'évitement d'obstacles sont basés sur la vision monoculaire et se divisent en deux grandes classes. Les premières, sont les méthodes basées sur le mouvement, et utilisent en général le flux optique. Les secondes sont les méthodes basées sur l'apparence. [Baptiste12] présente un système de localisation par vision monoculaire pour un robot mobile circulant dans un milieu urbain; une première phase d'apprentissage où le robot est conduit manuellement est réalisée pour enregistrer une séquence vidéo. Les images acquises sont ensuite utilisées dans une phase hors ligne pour construire une carte 3D de l'environnement Par la suite, le véhicule peut se déplacer dans la zone de manière autonome ou non, et l'image reçue par la caméra permet de le positionner dans la carte. Le travail de Vivianne, [Viviane11] est basé sur la navigation d'un robot mobile par commande référencée multi-capteur. **Il** considère des véhicules non holonomes équipés de capteurs et d'une caméra commandable en lacet. [Durand12] se base sur la navigation par asservissement visuel d'un robot mobile dans un environnement libre. La navigation de robots composés d'une base mobile non holonome équipée d'une caméra montée sur une platine commandable en lacet; la caméra de type Axis 214 PTZ numérique couleur, elle possède des capteurs CCD ExView HAD de 1/4 pouce, La caméra est modélisée à l'aide du modèle sténopé (*Pinhole Camera Model*). Ce modèle est couramment utilisé pour les capteurs projectifs.

I.2.2. La Stéréovision:

La stéréovision consiste à utiliser deux caméra séparées l'une de l'autre d'une certaine distance connue et observant la même scène figure (1.7). Lorsqu'on utilise deux caméras, c'est le système binoculaire, il est suffisant pour reconstruire en 3D la scène observée ou retrouver l'information de la profondeur par triangulation qui peut être utilisée pour l'évitement d'obstacles ou pour la cartographie.

La principale difficulté de la stéréovision réside dans la mise en correspondance des pixels, c'est-à-dire la détermination dans les images des projections $\mathbf{m_1}$ et $\mathbf{m_2}$ d'un même point **M**. Une fois les pixels sont mis en correspondance, les coordonnées du point **M** sont obtenues par triangulation des points $\mathbf{m_1}$ et $\mathbf{m_2}$ à partir des centres optiques $\mathbf{C_1}$ et $\mathbf{C_2}$ des caméras [Rémi10], comme l'illustre la figure (1.8)

Figure (1.7) : capteur stéréovision

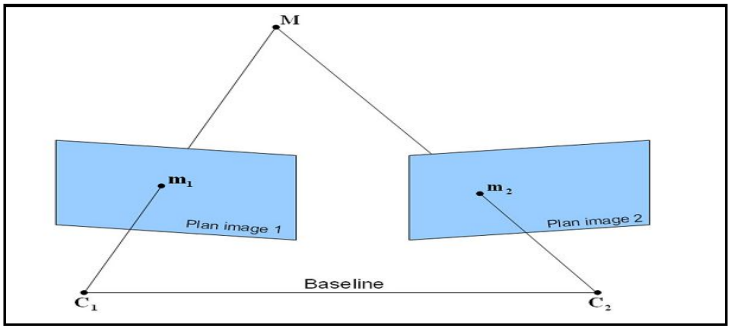


Figure (1.8): Principe de la stéréovision

Figure (1.9) : Reconstruction 3D d'un visage par stéréovision.
Source [Devernay97]

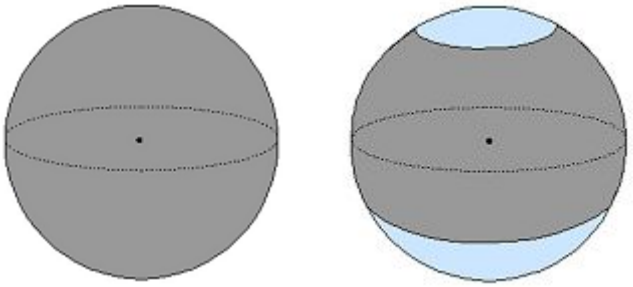
Dans cette recherche, Miguel [Miguel *et al*04] a proposé un nouvel algorithme pour déterminer la trajectoire d'un robot mobile et, en même temps, la création d'un modèle 3D volumétrique détaillée de son espace de travail. L'algorithme utilise exclusivement des informations fournies par un seul système de vision stéréo. [Juan05] se base sur l'étude du déplacement du robot dans les milieux agricoles en percevant visuellement les principaux objets qui l'entoure, le système de vision est constitué par une tête stéréo constituée de caméras Micropix C-1024. Ce modèle de caméras a la particularité d'avoir un seul capteur CCD équipé d'un filtre ou mosaïque Bayer; qui permet d'obtenir des images couleur de haute qualité (résolution maximale 1024 * 768). [Vincent05] se base sur l'analyse des applications possibles de la stéréovision embarquée sur un véhicule qui utilise un calibrage automatique du capteur stéréoscopique et traite deux applications majeures d'aide à la conduite. La première application réalise une fonction de détection d'obstacles en milieu routier et la deuxième consiste en la modélisation de l'environnement en milieu urbain pour une fonction d'aide au parking. Le travail de Mathias [Mathias08] est basé sur la détection d'obstacles en utilisant des multi-capteurs supervisée par stéréovision. Les travaux d'Oualid [Oualid10] se basent sur l'étude de la localisation et le guidage du robot mobile "Atrv2" dans un environnement naturel, en utilisant un système de vision composé de deux caméras CCD couleurs montées sur une tourelle Pan Tilt. [Saurav *et al*10] a utilisé la fusion de capteur Laser & Stéréo Vision Camera pour l'estimation de la profondeur et l'évitement des obstacles.

I.2.3. La Vision Omnidirectionnelle:

Le terme de la vision omnidirectionnelle se réfère au capteur de vision avec un très large champ de vue, c'est-à-dire un capteur de champ de vue de 360°. Comme il existe un autre terme pour un champ de vue large c'est la vision panoramique. Ces deux termes sont généralement employés indifféremment alors qu'il existe une différence entre eux.

- La vision omnidirectionnelle est le procédé de vision avec un champ de vue de 360° horizontale avec un champ de vision vertical qui peut être limité ou non, donc son champ de vue doit être sphérique, et permet d'observer le monde à partir de son centre, comme l'illustre la figure (1.10 (a)).
- La vision panoramique est une réduction du premier procédé, son champ de vue est limité. Il est de 360° horizontale avec un champ de vue vertical limité. Donc elle ne

couvre pas tout l'environnement dans un autre sens comme l'illustre la figure (1.10 (b)).



(a) (b)

Figure (1.10): Champ de vue des systèmes de vision (représenté en gris).
(a) Vision panoramique - (b) Vision omnidirectionnelle

Plusieurs techniques ont été développées pour augmenter le champ de vue panoramique et omnidirectionnel que l'on peut classer en trois catégories [Benosman01], [Yagi99], [Nayar97] [Barbosa04]:

- **l'utilisation de plusieurs images pour former un panorama.**
- **l'utilisation d'objectifs grands angles.**
- **l'utilisation d'un miroir.**

I.2.3.1. Fusion de plusieurs images :

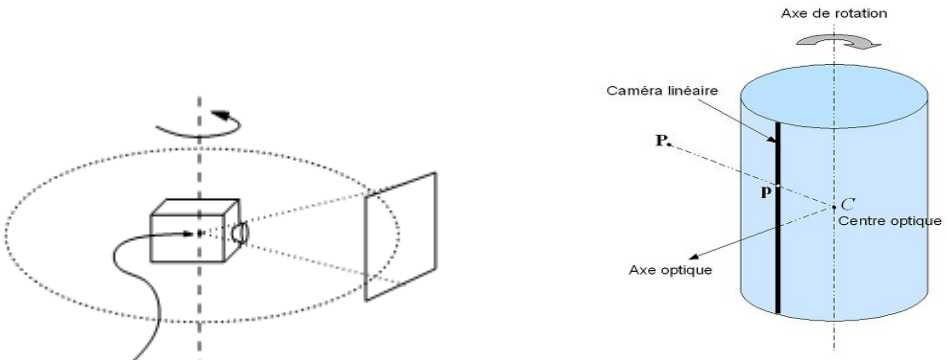
La première catégorie est basée sur l'utilisation de plusieurs images pour former une image panoramique. Cependant ce système pose plusieurs problèmes en pratique. Deux techniques peuvent être mises en œuvre pour l'acquisition des images:

- La rotation d'une caméra autour d'un axe;
- Utilisation de plusieurs caméras.

Rotation d'une caméra:

La première technique consiste à effectuer une acquisition d'une image par rotation d'une caméra autour d'un axe vertical pour acquérir l'image panoramique la figure (1.11(a)). Aussi, il est possible d'utiliser une caméra matricielle. C'est notamment ce que fait Ishigo [IYT90]. La caméra utilisée dans un tel système est en général linéaire puisqu'il suffit alors de concaténer les lignes verticales pour former l'image panoramique.

L'autre type de capteur nommé "capteur cylindrique" est modélisé par un cylindre comme l'illustre la figure (1.11(b)) [Benosman96]. Ce capteur permet de donner à l'image panoramique une haute résolution. La figure (1.12) l'illustre une image acquise par un capteur cylindrique et le zoom effectué sur une petite région pour mettre en évidence la haute résolution.



(a) centre de projection (b) capteur cylindrique

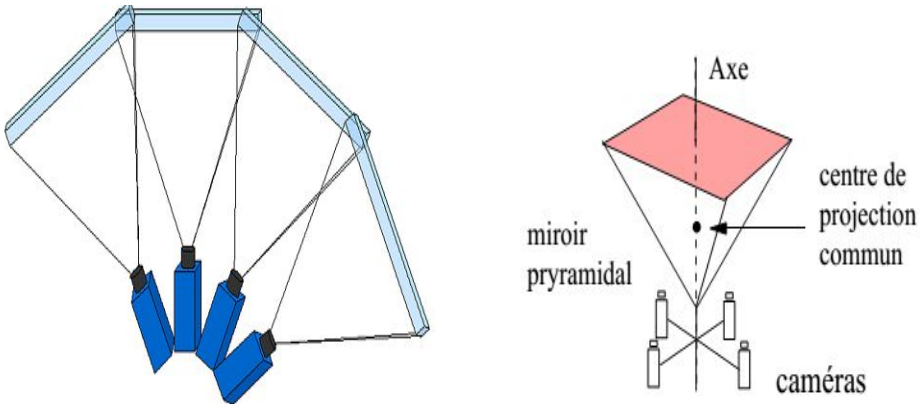
Figure (1.11) : Image omnidirectionnelle générée par rotation d'une caméra

Figure (1.12) : Image acquise par un capteur cylindrique et zoom sur une région. La résolution de cette image est de 9 000 x 59 000 pixels. Source [Panoscan]

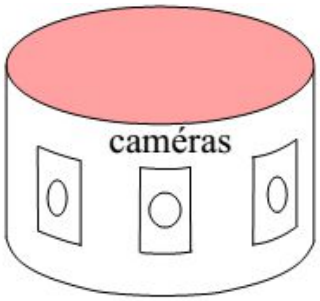
Systèmes multi-caméras:

La deuxième technique est basée sur l'utilisation de plusieurs caméras de manière à obtenir une acquisition du panorama instantanément, contrairement à la première technique.

La figure (1.13) illustre des exemples sur quelques caméras qui permettent le recouvrement des champs de vision panoramiques de l'environnement :



(a) Système «fullview» des laboratoires Bell [Nalwa96]



(b) Système «RingCam» de Microsoft

Figure (1.13): Principe du système multi-caméras.

Quelques systèmes réalisés sur ce principe sont : le capteur DodecaTM 1000, développé par Immersive Media et la série de capteurs Astro mis au point par la société ViewPlus.

(a) (b)

Figure (1.14) : Systèmes multi-caméras.
(a) DodecaTM 1000 - (b) Astro Sensor

L'avantage de ces systèmes "multi-caméras" est l'obtention d'une haute résolution, par contre son inconvénient réside dans:

- Son prix qui reste très cher.
- La difficulté de synchroniser toutes les caméras.
- L'importance du flux de données à traiter.

I.2.3.2. Utilisation d'un objectif grand angle:

Le grand angle, appelé aussi grand angulaire, est un objectif de courte focale. Certains objectifs permettant d'obtenir un grand angle sont les objectifs type "*Fish-eye*" (œil de poisson). Une caméra "*fish-eye*" est une caméra catadioptrique possédant une focale très courte et une association d'un miroir de révolution placé devant la caméra pour agrandir le champ de vue. Cette caméra fournit en effet un grand champ de vue remarquable (près de 180 degré) par rapport à une lentille classique, ce qui donne le possible de traiter un espace semi-sphérique. Vu que ces caméras sont relativement petites, elles peuvent être montées facilement sur un véhicule ou un robot. Lorsqu'elle est installée sur un robot mobile, elle est ajustée de manière à ce que sa direction pointe vers le bas, comme un oiseau volant qui oriente sa tête vers la terre. L'inconvénient d'un tel système réside dans sa très faible résolution aux alentours de la périphérie de l'image, bien qu'au centre elle est bonne. La figure (1.15(b)) illustre ce phénomène où on constate la déformation au niveau de la périphérie. Dans ce cas, ce type de caméra n'est pas souhaitable pour des utilisations robotiques à cause du long traitement de l'image.

(a) (b)

Figure (1.15): (a)Objectif fish-eye, (b) Image acquise par une caméra fish-eye

I.2.3.3. Utilisation d'un capteur catadioptrique:

Les caméras qui utilisent uniquement l'effet de réfraction de la lentille de refléter la lumière sont appelés caméras dioptriques (la dioptrie est la science de éléments de réfraction**).** Les caméras qui utilisent les effets combinés de la réflexion sur un miroir et de réfraction d'une lentille sont appelés les caméras catadioptriques (à partir de catoptrique, c'est à dire la science des surfaces de réflexion et dioptrique). Un capteur catadioptrique figure (1.16) est composé d'une caméra et d'un miroir ayant en général un axe de révolution.

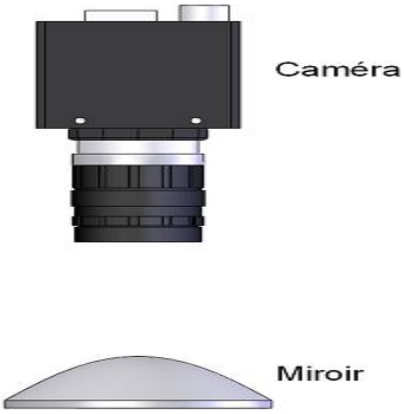


Figure (1.16) : Exemple de capteur catadioptrique.

Nous allons donner un bref aperçu sur les types de miroir omnidirectionnel les plus utilisés. Seules certaines formes de miroir de révolution permettent de conserver la contrainte de point de vue unique, à savoir les miroirs sphériques, ellipsoïdes et hyperboloïdes avec une caméra perspective et le miroir paraboloïde avec une caméra orthographique [Baker99]. Ces formes ont été largement utilisées dans les robots mobiles qu'elles peuvent être appelées **les** formes classiques de miroir**. C**hacune de ces formes de miroir réalise une cartographie différente entre les points du monde et les points d'image.

Formation d'image:

1) La projection induite par une caméra est une fonction de l'espace 3D vers le plan image 2D f:$\boldsymbol{R^3 \rightarrow R^2}$.
2) L'image inverse d'un point de l'image dans l'espace est une droite (figure (1.17)).

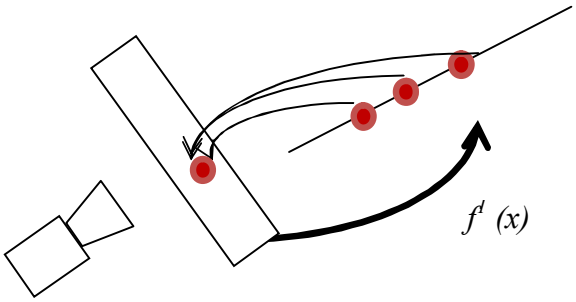


Figure (1.17) : Inverse d'un point de l'image dans l'espace.

3) Pour beaucoup de caméras, toutes ces droites ne possèdent pas nécessairement un seul point d'intersection figure (1.18). Leur enveloppe est appelée caustique et représente le lieu des points de vue. Projection non centrale.

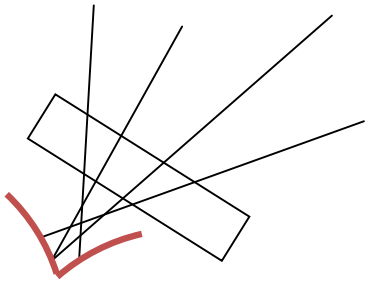


Figure (1.18) : Projection non centrale.

4) Si toutes les droites se coupent en un seul point alors le système a un point de vue effectif unique et c'est une projection centrale.

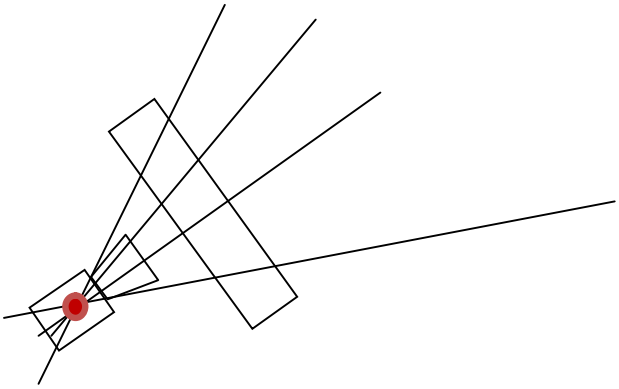


Figure (1.19): Projection centrale.

Le miroir plan:

Le point de vue unique du miroir plan situé sur l'axe optique à une distance égale à celle séparant le miroir du centre optique de la caméra figure (1.20).

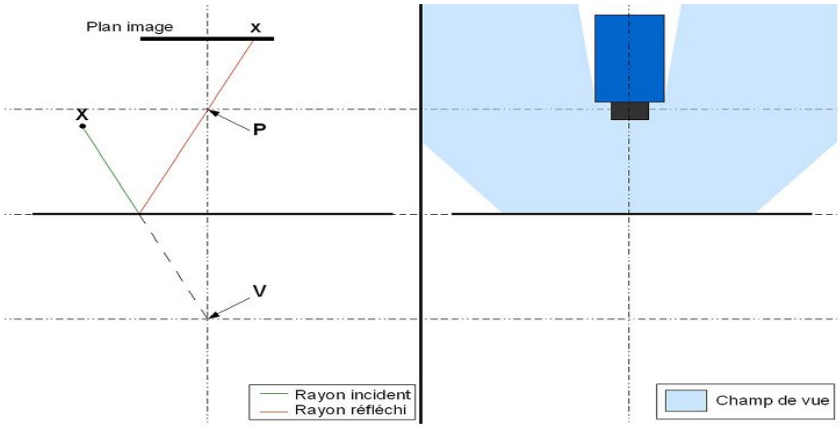


Figure (1.20) : miroir plan

L'utilisation d'un seul miroir plan et une caméra ne permet pas d'augmenter l'angle de champ de vue puisque celui-ci sera identique à celui de la caméra. Pour augmenter l'angle de champ de vue avec la conservation d'un point de vue unique il faut multiplier le nombre de miroir et le nombre des caméras à condition d'associer une caméra à chaque miroir.

[Nalwa96] a réalisé un capteur panoramique à l'aide de quatre miroirs plans disposés en pyramide et de quatre caméras CDD. Il a réussi à obtenir un point de vue unique et un champ de 360°x50°.

Le miroir conique :

Le miroir conique possède un point de vue unique confondus avec le centre optique de la caméra **P** au sommet du cône figure (1.21).

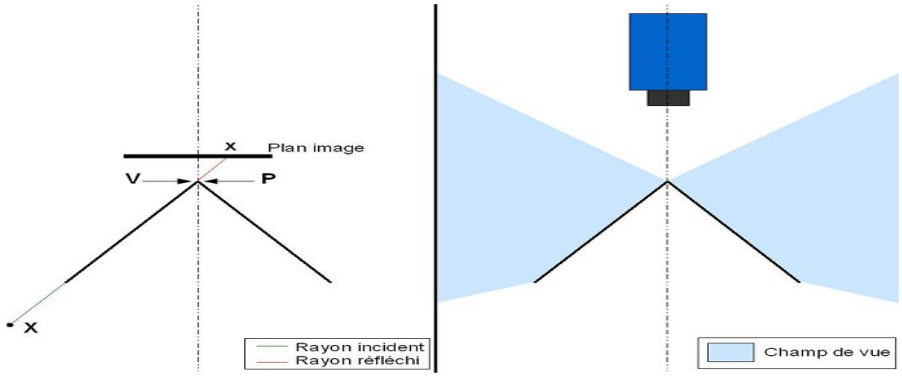


Figure (1.21) : Miroir conique.

Le miroir ellipsoïde:

Le miroir ellipsoïde respecte la contrainte du point de vue unique lorsque le centre optique **P** de la caméra est situé sur le second foyer de l'ellipsoïde, le premier étant le point de vue **V** figure (1.22)

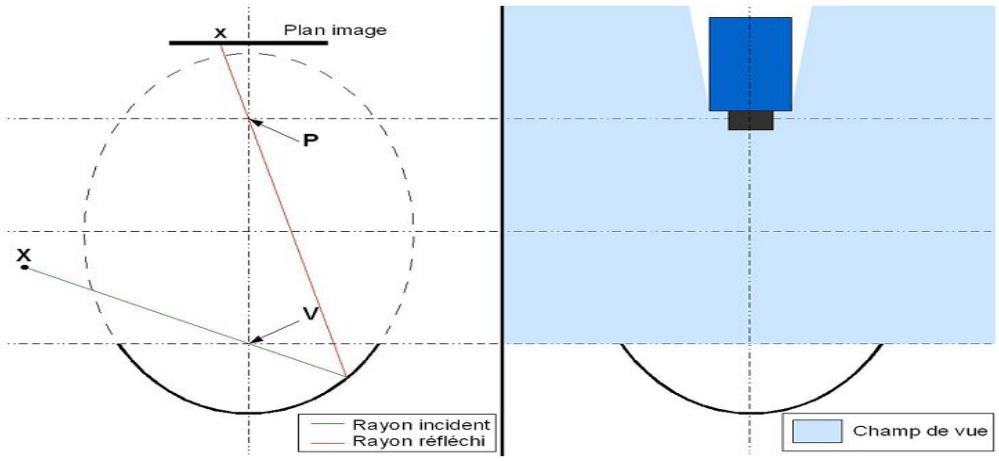


Figure (1.22) : Miroir ellipsoïde.

Le miroir sphérique:

Ce type de vision présente la particularité d'inclure les deux points dans le centre de la sphère. Donc, le point de vue unique **V** et le centre optique **P** sont confondus (figure (1.23)).

L'image obtenue par ce système a une bonne résolution dans la région centrale et une faible résolution dans la région périphérique.

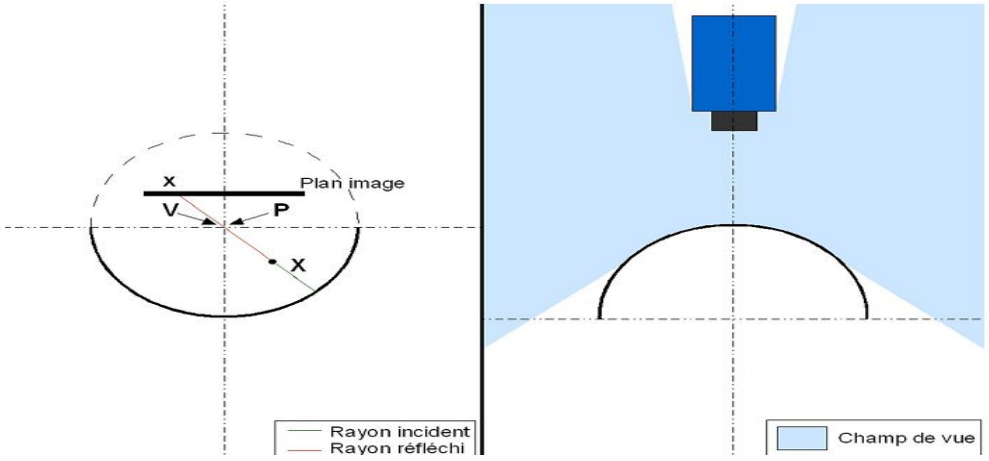


Figure (1.23) : Miroir sphérique.

Le miroir hyperboloïde :

L'avantage de ce miroir " miroir hyperboloïde" est le point de vue unique **V** coïncide avec le centre optique **P** (figure (1.24)). **C**et avantage permet de reconstruire une image libre de distorsion et elle donne une bonne résolution dans la zone centrale.

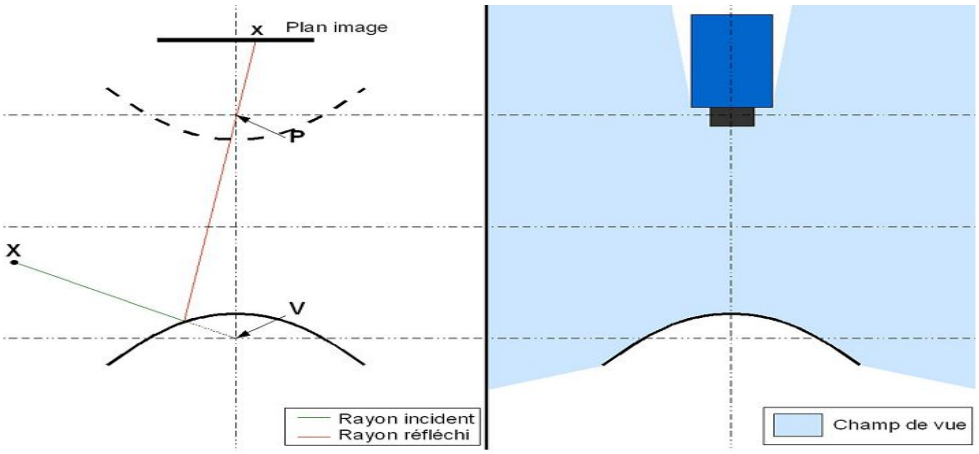


Figure (1.24) : Miroir hyperboloïde

A partir de l'image d'un miroir hyperboloïde, il est possible de reconstituer un panoramique cylindre ou des images perspective à des angles désirés comme l'illustre la figure (1.25)

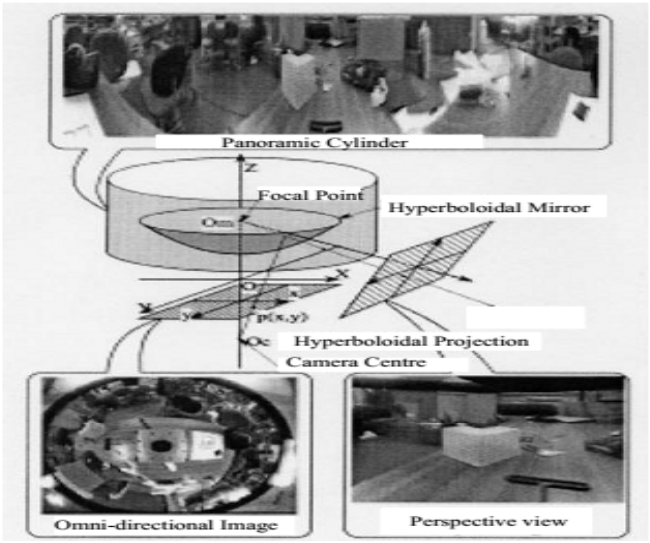


Figure (1.25): Projection hyperboloïde et des exemples d'images transformées.
(Courtesy of Prof. Y. Yagi at Osaka University)

Le miroir paraboloïde:

Le miroir paraboloïde présente un seul point de vue lorsqu'il est utilisé avec un appareil photo de projection orthographique**,** c'est-à-dire les rayons qui passent par le point de vue **V** sont réfléchis parallèlement à l'axe optique (figure (1.26)). Il est facile de réaliser le calibrage et le calcul des images perspectives est simple.

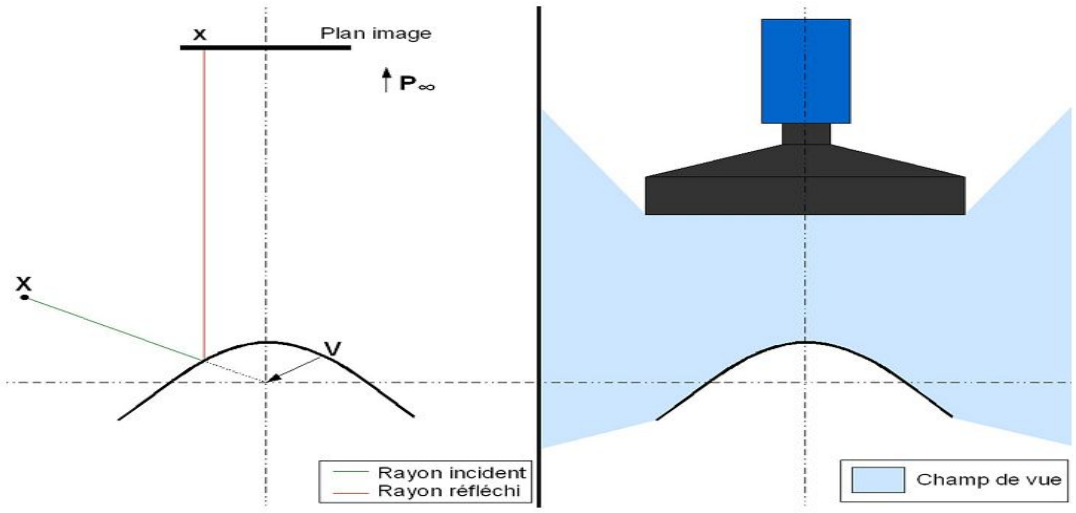


Figure (1.26): Miroir paraboloïde.

Le miroir parabolique possède la même résolution que le miroir hyperboloïdal**,** même avec un angle de vision plus petit. Seules deux configurations sont possibles à réaliser :

1. Miroir hyperboloïde avec une caméra perspective dont le centre optique coïncide avec le second foyer du miroir,
2. Miroir paraboloïde avec caméra orthographique.

Les travaux de Cyril [Cyril02], abordent le problème de la localisation et la modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels. Il s'agit du capteur omnidirectionnel "SYCLOP" composé d'un miroir conique et d'une caméra CCD. L'objectif de son travail est de construire un modèle sensoriel basé sur l'utilisation conjointe de deux capteurs omnidirectionnels. Dans une deuxième phase, il utilise ces informations pour mettre en œuvre une méthode de modélisation incrémentale. Le système de perception proposé par Sylvain [Sylvain04], pour les robots mobiles autonomes, conçus pour participer à la compétition internationale de soccer robotisé," *Robocup1*", est supposé être en mesure de fonctionner parfaitement dans l'environnement prévu pour cet effet. Le système de

vision est basé principalement sur 2 types de capteurs: une caméra omnidirectionnelle et des encodeurs optiques mesurant le déplacement des roues motrices. Il s'agit d'une caméra orientée vers un miroir convexe situé dans le haut du montage. Courbon et ses coéquipiers [Courbon *et al*, 09] proposent une méthode basée sur un graphe d'images générique (image fish-eye, omnidirectionnelle) définissant un chemin visuel à suivre. Le graphe est utilisé pour un suivi de trajectoire. Dans [Jogan *et al,* 00]: la méthode de localisation proposée est basée sur des panoramas cylindriques. Les travaux de Remi [Rémi 10], ont permis de développer un capteur de stéréovision omnidirectionnelle, composé de deux capteurs catadioptriques, pour tirer partie des avantages de la vision omnidirectionnelle et de la stéréovision. Guillaume [Guillaume 10], aborde les questions d'estimation de position et d'orientation de caméras omnidirectionnelles conjointement au positionnement de robot guidé par de telles caméras.

I.4. Conclusion :

Dans ce chapitre on a essayé d'étaler un ensemble de systèmes des visions dont beaucoup sont utilisées en robotique: Il s'agit essentiellement de la vision monoculaire, la stéréovision et la vision omnidirectionnelle. Tous ces systèmes sont basés sur l'utilisation d'une ou plusieurs cameras. D'autre part, on a présenté d'une manière générale les différents avantages et inconvénients de chaque système.

Comme le domaine de la vision est très riche dans ses applications dans plusieurs domaines, on ne recense dans notre cas que les travaux appliqués à la robotique. Ainsi un état de l'art a été donné en survolant quelques travaux que nous avons jugés utile.

CHAPITRE II :

La Modélisation Et La Commande D'un Robot Omnidirectionnel

CHAPITRE II

LA MODÉLISATION ET LA COMMANDE D'UN ROBOT OMNIDIRECTIONNEL

II.1. Introduction :

La recherche d'algorithmes de planification et de stratégies de commande des véhicules holonomes constitue aujourd'hui l'un des principaux axes de la recherche dans la robotique moderne. Dans ce chapitre nous allons présenter le modèle cinématique et un contrôle non linéaire d'un robot mobile omnidirectionnel à roues. Le contrôleur cinématique est proposé en fonction d'une linéarisation par bouclage en anglais "*feedback linéarization*" pour parvenir à la stabilisation asymptotique en un point d'équilibre à poursuite d'une trajectoire. Les simulations et les résultats expérimentaux sont effectués pour démontrer la faisabilité et l'efficacité de la méthode.

L'étude prend comme modèle " *Robotino* ", un robot mobile omnidirectionnel de *Festo didactic* [festo], composé d'une base circulaire reposant sur des roues suédoises. Les roues possèdent une bande de roulement formée d'une paire de 3 rouleaux. Il est conduit par 3 unités motrices indépendantes montées avec un angle de 120° entre elles, un moteur à courant continu avec un encodeur, un réducteur à engrenages, une réductrice poulie / courroie synchrone et une roue suédoise.

Trois roues omnidirectionnelles lui permettent de se déplacer aisément dans des lieux très encombrés. Le robot est équipé d'une caméra montée sur une platine commandable en lacet. Les figures (2.1), (2.2) et (2.3) illustrent clairement ce que nous avons décrit.

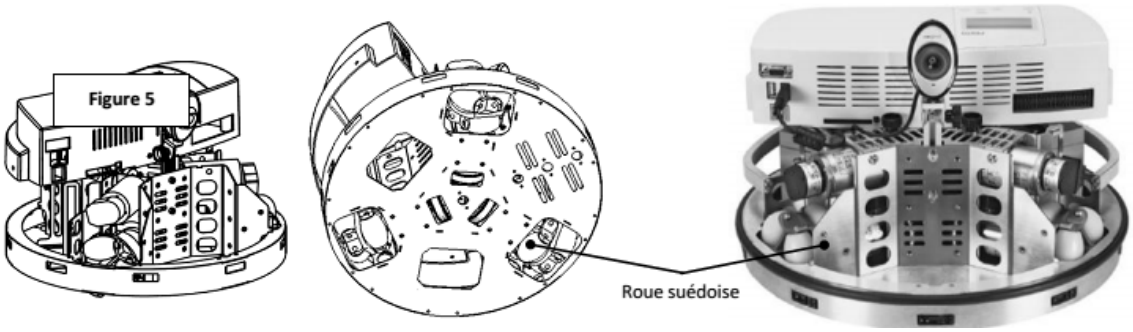


Figure (2.1) : Photographie de la base mobile" Robotino"

Figure (2.2) : Roue suédoise

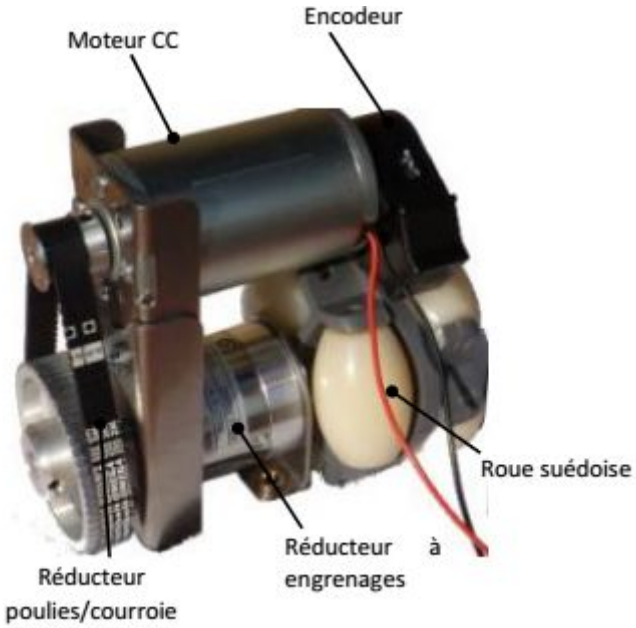


Figure (2.3) : Unité motrice

II.2. Model Cinématique :

Le but de cette section est de développer une relation entre les vitesses de chaque roue et les vitesses du robot. La vitesse en X est appelée "vitesse **longitudinale"** et la vitesse en Y est appelée " vitesse **latérale "**, la troisième vitesse représente la vitesse de rotation du robot autour de l'axe verticale et est appelée " vitesse de **lacet ".** Le modèle cinématique utilise trois équations pour déterminer ces vitesses qui sont obtenues en projetant les vitesses sur les deux axes pour $\dot{X}r$ (V_x) et $\dot{Y}r$ (V_y). Ainsi à partir des trois vitesses angulaires $\dot{\theta}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$, des trois roues, on peut obtenir les vitesses du robot à un instant précis. Les équations du modèle cinématique sont reproduites ci-après :

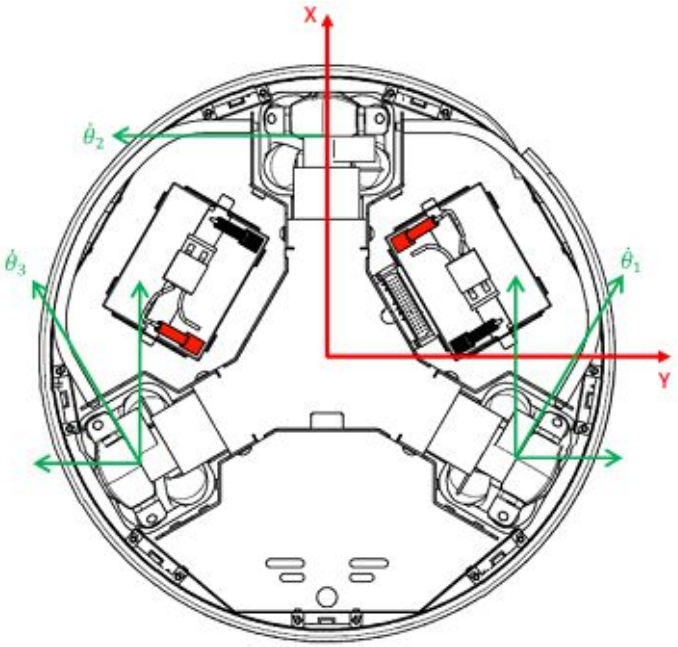


Figure (2.4) : Géométrie du robot omnidirectionnel

$$\begin{cases} \dot{X}r = \frac{R}{3}[(\dot{\theta}_1 + \dot{\theta}_3).\cos(\frac{\pi}{6})] & (2.1) \\ \dot{Y}r = \frac{R}{3}[(\dot{\theta}_1 - \dot{\theta}_3).\sin(\frac{\pi}{6}) - \dot{\theta}_2] & (2.2) \\ \vartheta_r = \frac{R}{L}[(\dot{\theta}_1 + \dot{\theta}_3).\cos(\frac{\pi}{6})] + \frac{R}{L}[\dot{\theta}_2 - (\dot{\theta}_1 - \dot{\theta}_3).\sin(\frac{\pi}{6})] & (2.3) \end{cases}$$

Donc :

$$\begin{bmatrix}\dot{X}\mathrm{r}\\ \dot{Y}\mathrm{r}\\ \vartheta_r\end{bmatrix}= R\begin{bmatrix}\frac{1}{3}.\cos\frac{\pi}{6} & 0 & \frac{1}{3}.\cos\frac{\pi}{6}\\ \frac{1}{3}.\sin\frac{\pi}{6} & -1 & -\frac{1}{3}.\sin\frac{\pi}{6}\\ \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{\mathrm{L}} & \frac{1}{\mathrm{L}} & \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{\mathrm{L}}\end{bmatrix}*\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}$$

$$\begin{bmatrix}\dot{X}\mathrm{r}\\ \dot{Y}\mathrm{r}\\ \vartheta_r\end{bmatrix}=R * p^{*}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}$$

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}=\frac{1}{R} * p^{-1}*\begin{bmatrix}\dot{X}\mathrm{r}\\ \dot{Y}\mathrm{r}\\ \vartheta_r\end{bmatrix}$$

Où : $p=\begin{bmatrix}\frac{1}{3}.\cos\frac{\pi}{6} & 0 & \frac{1}{3}.\cos\frac{\pi}{6}\\ \frac{1}{3}.\sin\frac{\pi}{6} & -1 & -\frac{1}{3}.\sin\frac{\pi}{6}\\ \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{\mathrm{L}} & \frac{1}{\mathrm{L}} & \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{\mathrm{L}}\end{bmatrix}$

Où : $\mathrm{V_i} = R * \dot{\theta}_\mathrm{i}$, i= 1,2,3

$$\begin{bmatrix}\mathrm{V}_1\\ \mathrm{V}_2\\ \mathrm{V}_3\end{bmatrix}=p^{-1}*\begin{bmatrix}\dot{X}\mathrm{r}\\ \dot{Y}\mathrm{r}\\ \vartheta_r\end{bmatrix} \qquad (2.4)$$

Avec :

$\dot{X}\mathrm{r}$: vitesse longitudinale

$\dot{Y}\mathrm{r}$: vitesse latérale

ϑ_r : vitesse de lacet

Le passage d'un référentiel à l'autre ce fait par simple projection vectorielle, la matrice de passage homogène entre repère robot à repère monde est défini comme suit :

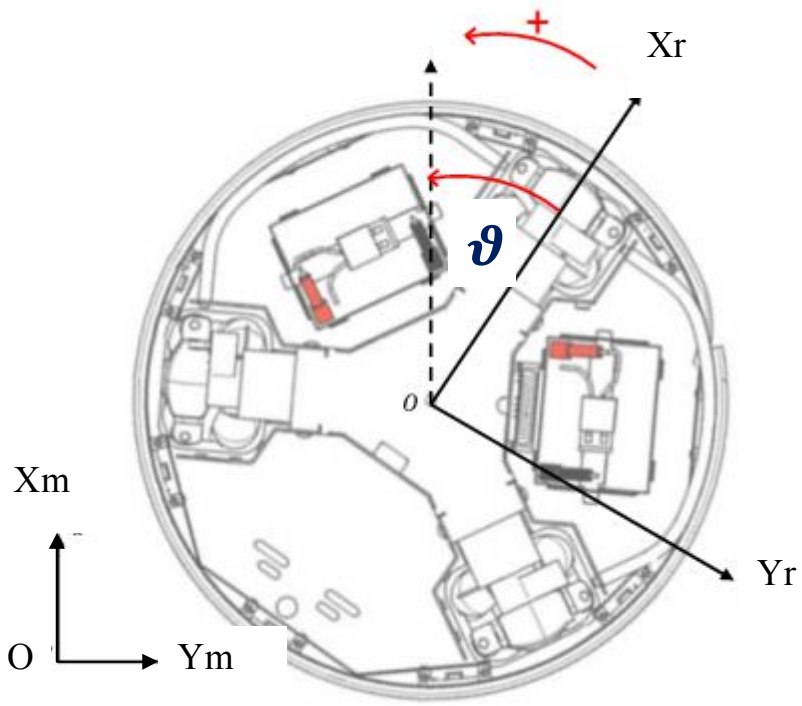


Figure (2.5) : Changement des repères

$$\begin{bmatrix} X_m \\ Y_m \\ \vartheta \end{bmatrix} = \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 \\ \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_r \\ Y_r \\ \vartheta_r \end{bmatrix}$$

Avec :

Xr, Yr : repère du robot

Xm, Ym : repère monde

ϑ : Angle de rotation

L'équation (2.4) devient :

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = p^{-1} * \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 \\ \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} * \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\vartheta} \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = P(\vartheta)^{-1} * \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\vartheta} \end{bmatrix} \qquad (2.5)$$

Avec : **P(ϑ)** est toujours inversible pour tout **ϑ , et :**

$$\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\vartheta}_m \end{bmatrix} = P(\vartheta) * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad (2.6)$$

Où :

$$P(\vartheta)^{-1} = \begin{bmatrix} \frac{1}{3}.\cos\frac{\pi}{6} & 0 & \frac{1}{3}.\cos\frac{\pi}{6} \\ \frac{1}{3}.\sin\frac{\pi}{6} & -1 & -\frac{1}{3}.\sin\frac{\pi}{6} \\ \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{L} & \frac{1}{L} & \frac{\cos\frac{\pi}{6}-\sin\frac{\pi}{6}}{L} \end{bmatrix}^{-1} * \begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 \\ \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

Tels que :

R : rayon de chaque roue

v_i : Vitesse de chaque roue avec i= 1, 2,3

$\dot{\theta}_1$: Vitesse angulaire de chaque roue

L : Distance entre le centre du robot et le centre d'une roue

II.3.Robot mobile omnidirectionnel pour une poursuite de trajectoire :

II.3.1. La commande en boucle fermé :

Un système en boucle fermée est un système dont la sortie du procédé est prise en compte pour calculer l'erreur. En général, le contrôleur effectue une action en fonction de l'erreur entre la mesure et la consigne**.**

Le problème de poursuite de trajectoire :

- ✓ Poursuite asymptotique d'une trajectoire cartésienne désirée $[x_d(t)\ y_d(t)\ ,\vartheta_d(t)]$] à partir d'une configuration initiale $q_0 = [x_0, y_0, \vartheta_0]^T$
- ✓ la détermination d'une commande permet de stabiliser asymptotiquement l'erreur de poursuite $\boldsymbol{q_e} = \boldsymbol{q_d} - \boldsymbol{q_m}$,

  Où: $\boldsymbol{q_m}$: La position mesurée avec $\boldsymbol{q_m} = [\ x_m(t), y_m(t), \vartheta_m(t)]$

  $\boldsymbol{q_d}$: La position désirée avec $\boldsymbol{q_d} = [\ x_d(t)\ , y_d(t)\ , \vartheta_d(t)]$

Prenant l'exemple du robot « *Robotino* ». Le problème est de stabiliser à l'origine l'erreur :

$$\begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ \vartheta_d \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \\ \vartheta_m \end{bmatrix} \qquad (2.7)$$

Avec : q_d la position désirée de la trajectoire et q_m la position en cours du robot mobile.

Dans ce travail, nous avons adopté la méthode qui a été proposé par [ching *et al,*05] pour permettre au robot mobile de se déplacer d'un point initial vers un point final spécifié ou bien poursuivre une trajectoire désirée

II.3.2. Le correcteur PI :

Le régulateur type PID est le plus courant dans les boucles de régulation. Il est devenu l'outil standard depuis que la commande des processus a émergé dans les années 1940. Malgré l'apparition de plusieurs variantes de régulateurs, il demeure toujours le plus désiré dans le contrôle des processus industriels. On a constaté que, plus de 95% des boucles de régulation sont du type PID « Proportionnel Intégral Dérivé ». Dans ce travail, on utilise les actions proportionnelle et intégrale dans une commande cinématique.

II.3.3. Stabilisation de point à point :

L'objectif de la commande de stabilisation point à point est de trouver le vecteur des vitesses linéaires $[\boldsymbol{V_1\ V_2\ V_3}]^T$ ou le vecteur des vitesses angulaire $[\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\ ]^T$,afin de ramener le robot mobile d'une configuration initiale $[\boldsymbol{x_0\ y_0\ \vartheta_0}]^T$ vers une configuration finale désiré $[\boldsymbol{x_d\ y_d\ \vartheta_d}]^T$, sachant que la position actuelle du robot est $[\boldsymbol{x\ \ y\ \ \vartheta}]^{\boldsymbol{T}}$.

Pour concevoir le dispositif de commande, le vecteur des erreurs de configuration est défini dans l'équation (2.7) comme suit :

$$\begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ \vartheta_d \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \\ \vartheta_m \end{bmatrix}$$

Dans le cas de l'asservissement point à point, seulement la configuration initiale et finale sont données. Donc les positions désirées sont des points ; sont présenté par des valeurs constantes et le dérivé d'un constant est zéro. Donc il est facile de vérifier que la dérivée de vecteur des erreurs est donné par :

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\vartheta}_e \end{bmatrix} = -\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix} = -P(\vartheta) * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad (2.8)$$

Pour une stabilité asymptotique la loi de commande PI est proposée, telle que :

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = -P(\vartheta)^{-1} * \left[-K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} \right] \quad (2.9)$$

Où les matrices de gain K_p et K_I sont symétriques et positive définies,

Avec $K_p = K_p{}^T > 0$, $K_I = K_I{}^T > 0$.

Dans ce cas, l'erreur du système en boucle fermée devient :

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\vartheta}_e \end{bmatrix} = -P(\vartheta) * -P(\vartheta)^{-1} * \left[-K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} \right] = \left[-K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} \right]$$

II.3.4. La commande en poursuite de trajectoire :

Dans cette partie, nous considérons le problème de suivi de trajectoire variant dans le temps tel qu'il a été formulé par les auteurs [*Ching et al*]. Contrairement aux robots non-holonomes, les trajectoires des robots mobiles holonomes ne peuvent pas être générées en utilisant leurs modèles cinématiques. Autrement dit ; les trajectoires des robots omnidirectionnels peuvent être arbitrairement planifiées. Étant donnée la trajectoire $[\boldsymbol{x_d(t)}\ \boldsymbol{y_d(t)}\ \boldsymbol{\vartheta_d(t)}]^T$ .Le vecteur d'erreur de poursuite de trajectoire est définie comme :

$$\begin{bmatrix} x(t)_e \\ y(t)_e \\ \vartheta(t)_e \end{bmatrix} = \begin{bmatrix} x_d(t) \\ y_d(t) \\ \vartheta_d(t) \end{bmatrix} - \begin{bmatrix} x_m(t) \\ y_m(t) \\ \vartheta_m(t) \end{bmatrix} \quad (2.10)$$

Dans le cas de l'asservissement de poursuite d'une trajectoire la position désirée est en fonction du temps, dans ce cas la dérivée du vecteur des erreurs s'écrit

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\vartheta}_e \end{bmatrix} = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix} - \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\vartheta}_m \end{bmatrix} = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix} - P(\vartheta) * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (2.11)$$

L'objectif de la commande est de contrôler le vecteur de vitesse $[\boldsymbol{V_1}\ \boldsymbol{V_2}\ \boldsymbol{V_3}]^T$ de telle sorte que le système d'erreur en boucle fermée soit asymptotiquement stable. Dans ce cas, le contrôleur PI donné par l'équation (2.5) a été proposé.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = -P(\vartheta)^{-1} * \left[-K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} - \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix} \right] \quad (2.12)$$

Obtenant ainsi l'erreur du système en boucle fermée telle que :

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\vartheta}_e \end{bmatrix} = -P(\vartheta) * -P(\vartheta)^{-1} * \left[-K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} - \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix} \right] + \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix}$$

$$= -K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} - \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix} + \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\vartheta}_d \end{bmatrix}$$

$$= -K_p \begin{bmatrix} x_e \\ y_e \\ \vartheta_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \vartheta_e \end{bmatrix} \quad (2.13)$$

Le schéma synoptique donné par la figure (2.6) permet de visualiser clairement le schéma de La commande appliqué :

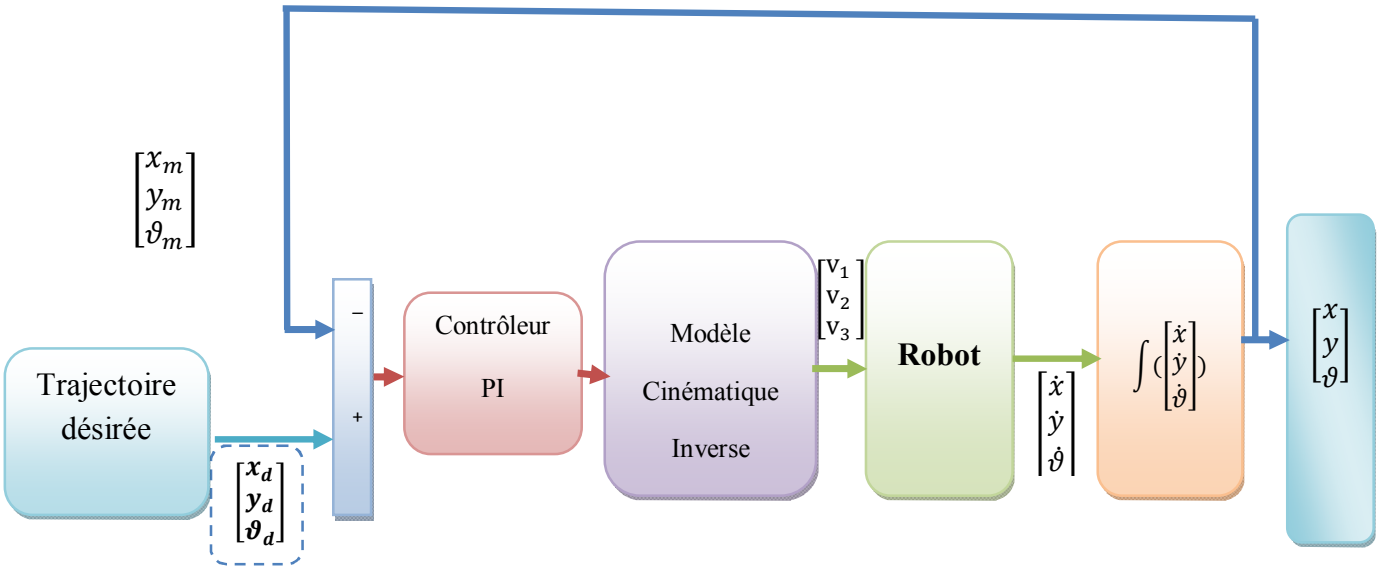


Figure (2.6) : Architecture de la commande

II.3.5. Stabilisation du système :

Première méthode :

Pour la stabilité asymptotique de système d'erreur en boucle fermée, on utilise la fonction de Lyapunov comme suit :

$$V=\frac{1}{2}[\,x_e\ y_e\ \vartheta_e]\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}+\frac{1}{2}[\int x_e\ \int y_e\ \int \vartheta_e]K_I\begin{bmatrix}\int x_e\\\int y_e\\\int \vartheta_e\end{bmatrix}$$

En prenant la dérivée temporelle de V le long de la trajectoire de l'équation (2.11) obtient :

$$\dot{V}=[\,x_e\ y_e\ \vartheta_e]\begin{bmatrix}\dot{x}_e\\\dot{y}_e\\\dot{\vartheta}_e\end{bmatrix}+[\int x_e\ \int y_e\ \int \vartheta_e]K_I\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}$$

$$=[\,x_e\ y_e\ \vartheta_e]\left[-K_p\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}-K_I\begin{bmatrix}\int x_e\\\int y_e\\\int \vartheta_e\end{bmatrix}\right]+[\int x_e\ \int y_e\ \int \vartheta_e]K_I\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}$$

$$=-[\,x_e\ y_e\ \vartheta_e]K_p\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}<0$$

Étant donné que : $\dot{\boldsymbol{V}}$ est défini négative, donc le système est asymptotiquement stable.

Deuxième méthode :

Détermination des valeurs de K_p et K_I :

Pour déterminer les valeurs K_p et K_I on utilise les équations de vecteur des erreurs du système en boucle fermé (2.13) :

$$\begin{bmatrix}\dot{x}_e\\\dot{y}_e\\\dot{\vartheta}_e\end{bmatrix}=-K_p\begin{bmatrix}x_e\\y_e\\\vartheta_e\end{bmatrix}-K_I\begin{bmatrix}\int x_e\\\int y_e\\\int \vartheta_e\end{bmatrix}$$

La fonction de vecteur des erreurs est linéaire

$$\dot{e} = -K_p * e - K_I * \int e \quad (2.14)$$

Après La transformation de la place on obtient :

$$s*E(s) = -K_p * E(s) - K_I * \frac{E(s)}{s} \quad (2.15)$$

$$s^2 + K_p s + K_I = 0 \quad (2.16)$$

Le déterminant de l'équation de deuxième degré (2.16) devient :

$$\Delta = K_p{}^2 - 4 * K_I \quad (2.17)$$

- ***Pour un système apériodique*** $\Delta > 0$

$$\Delta > 0 \rightarrow K_p{}^2 - 4 * K_I > 0 \rightarrow K_p{}^2 > 4 * K_I \rightarrow K_p > 2 * \sqrt{K_I}$$

Pour $\Delta > 0$, on a deux pôles real :

$$s_1 = \frac{-K_p + \sqrt{K_p{}^2 - 4*K_I}}{2} \quad (2.18)$$

$$s_2 = \frac{-K_p - \sqrt{K_p{}^2 - 4*K_I}}{2} \quad (2.19)$$

Pour un système stable il faut vérifier que s_1 *et* s_2 sont négative.

- La valeur de la racine de déterminant est toujours positive $\sqrt{K_p{}^2 - 4 * K_I} > 0$
- et $K_p > \sqrt{K_p{}^2 - 4 * K_I}$

Comme on peut le constater, les pôles s_1 et s_2 appartiennent toujours au demi-plan gauche, induisant ainsi un système stable. Dans le Tableau (2.1), on a reporté quelques valeurs obtenues suite à des tests effectués sur le système chaque fois qu'on varie les valeurs des paramètres du contrôleur PI. Suivant les valeurs choisies, on peut avoir les trois régimes de fonctionnement du système, à savoir : Apériodique, Critique et sous amorti.

$\Delta = K_p{}^2 - 4 * K_I$	***Régime Apériodique*** $\Delta > 0$		
K_p	4	5	10
K_I	1	0.2	0.2
$K_p > 2 * \sqrt{K_I}$	Vérifier	Vérifier	Vérifier
$\sqrt{K_p{}^2 - 4 * K_I}$	3.46	4.9193	9.95
$K_p > \sqrt{K_p{}^2 - 4 * K_I}$	Vérifier	Vérifier	Vérifier
$S_{1,2}$	< 0 (système stable)		

Tableau (2.1) : Les paramètres du contrôleur PI pour "Régime apériodique"

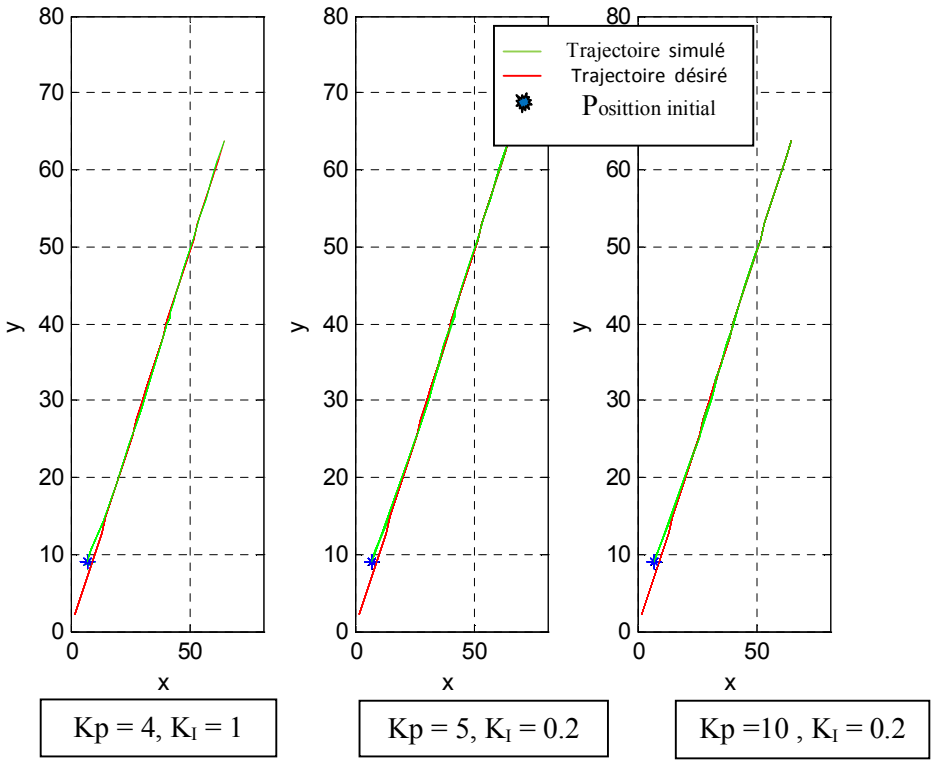


Figure (2.7) : Poursuite d'une trajectoire linéaire.

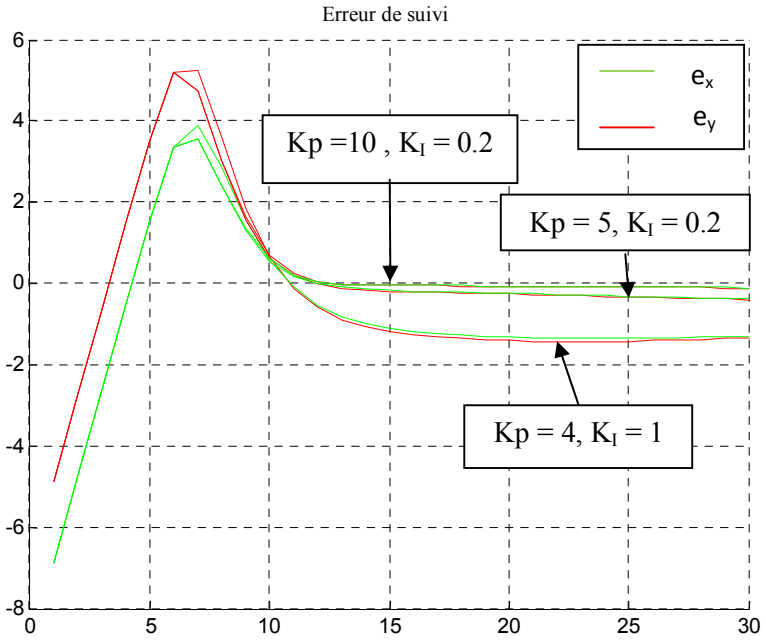


Figure (2.8) : Erreur de suivi.

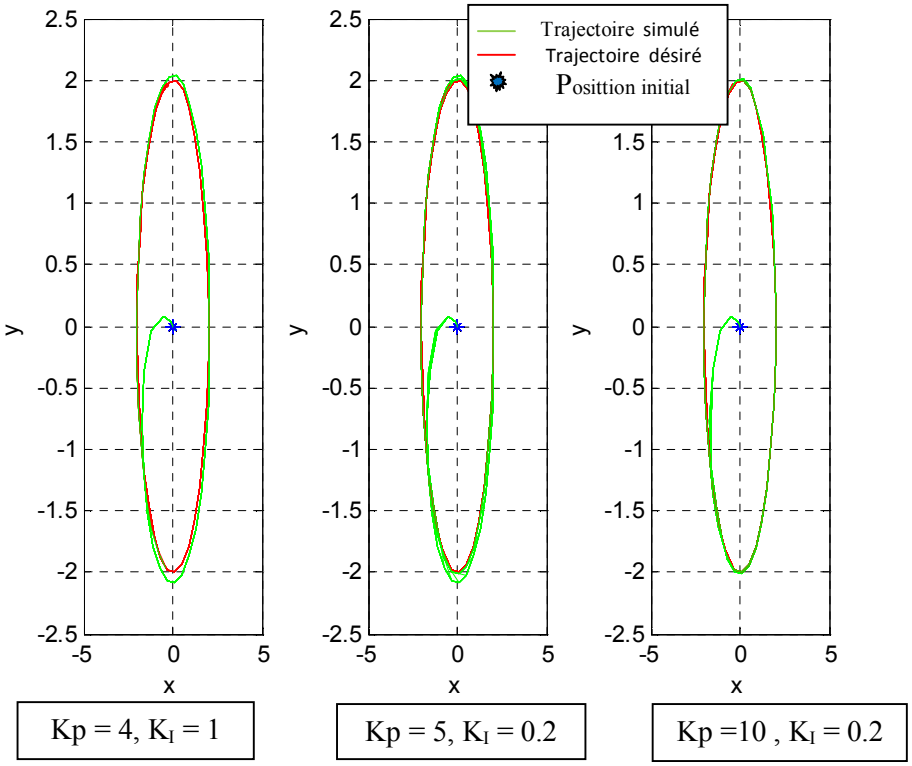


Figure (2.9) : Poursuite d'une trajectoire circulaire.

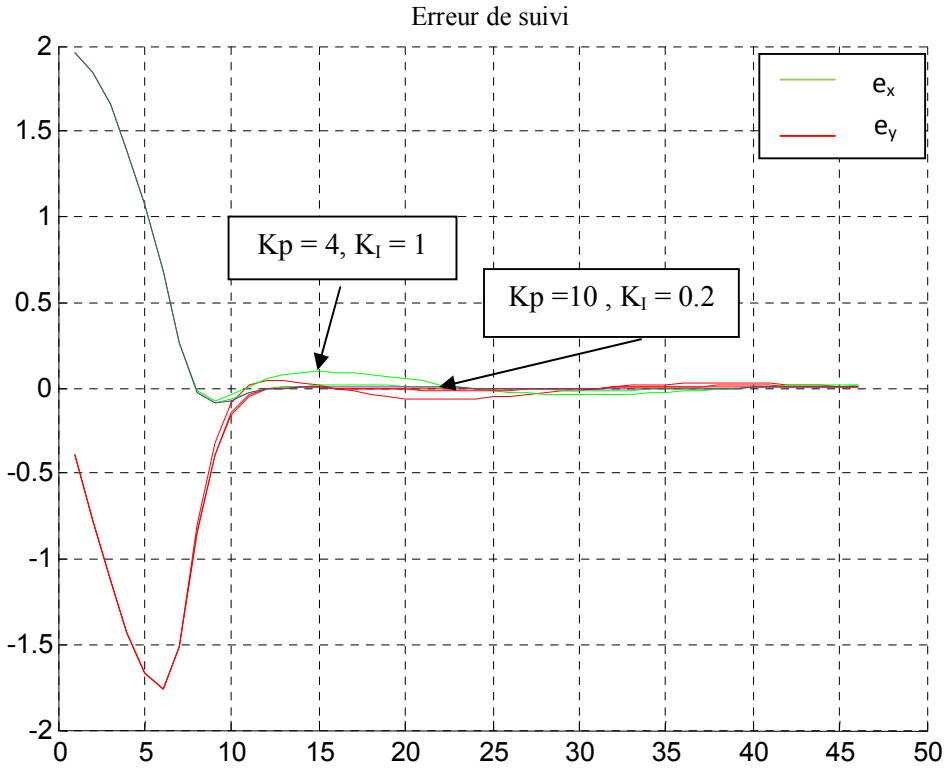


Figure (2.10) : Erreur de suivi

Les résultats de simulation montrent que le régime apériodique donne un bon résultat vis à vis du temps de réponse et de l'erreur de suivi. Dans ce cas la valeur de Kp est relativement grande par rapport à K_I. Avec les valeurs des paramètres Kp = 10 et K_I =0.2 est le meilleur choix obtenu dans ce cas de figure.

- ***Pour un système critique*** $\Delta = 0$

$$\Delta = 0 \rightarrow {K_p}^2 - 4 * K_I = 0 \rightarrow {K_p}^2 = 4 * K_I \rightarrow K_p = 2 * \sqrt{K_I}$$

Pour $\Delta = 0$ donc j'ai un pôle double real :

$$s = \frac{-K_p}{2} \qquad (2.20)$$

$\Delta = K_p{}^2 - 4 * K_I$	***Régime critique*** $\Delta = 0$		
K_p	4	8	20
K_I	4	16	100
$K_p = 2 * \sqrt{K_I}$	Vérifier	Vérifier	Vérifier
$s = \frac{-K_p}{2}$	< 0 (système stable)		

Figure (2.2): Les paramètres du contrôleur PI pour "Régime Critique"

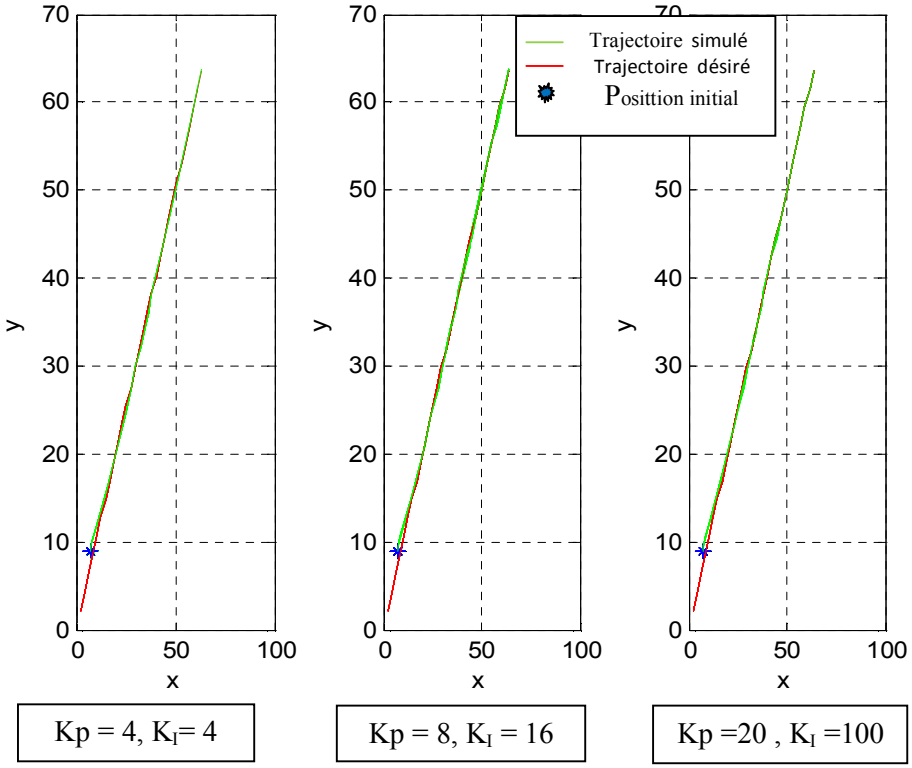


Figure (2.11) : Poursuite d'une trajectoire linéaire

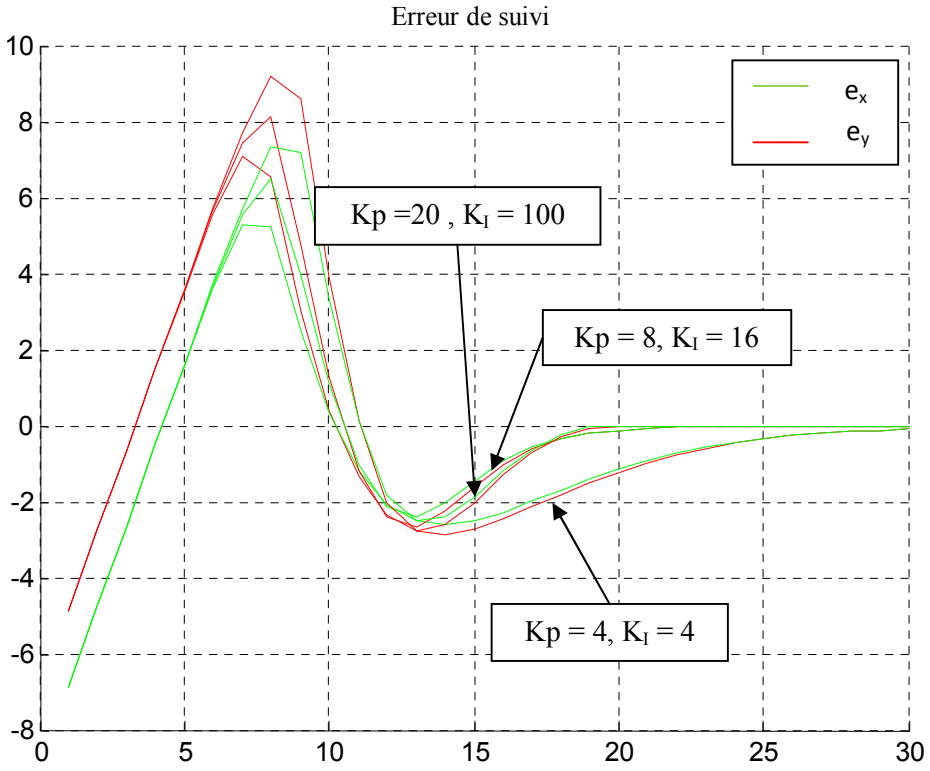


Figure (2.12) : Erreur de suivi

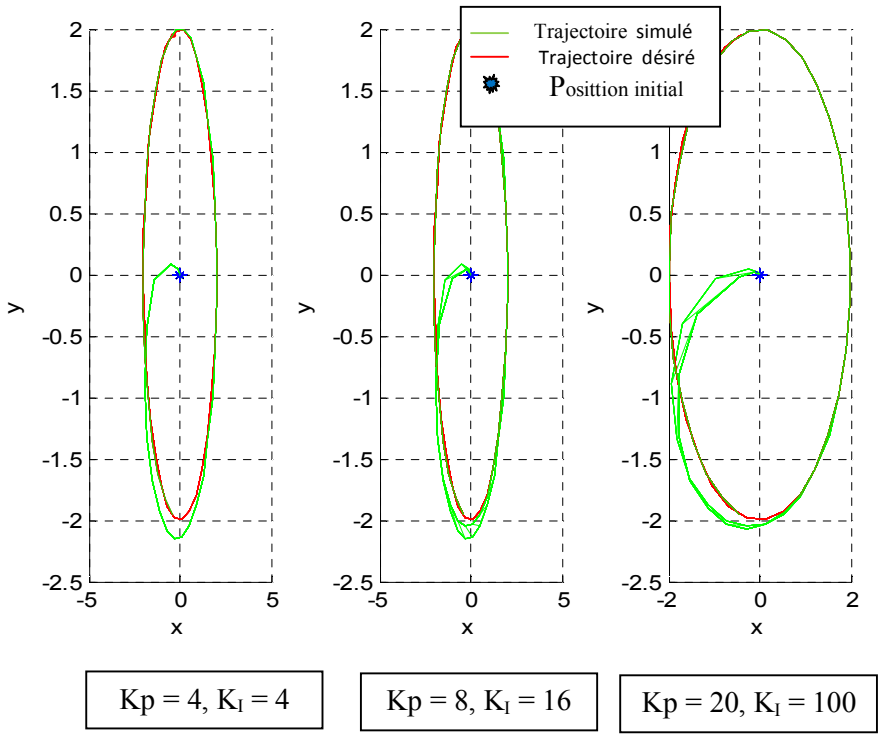


Figure (2.13) : Poursuite d'une trajectoire circulaire.

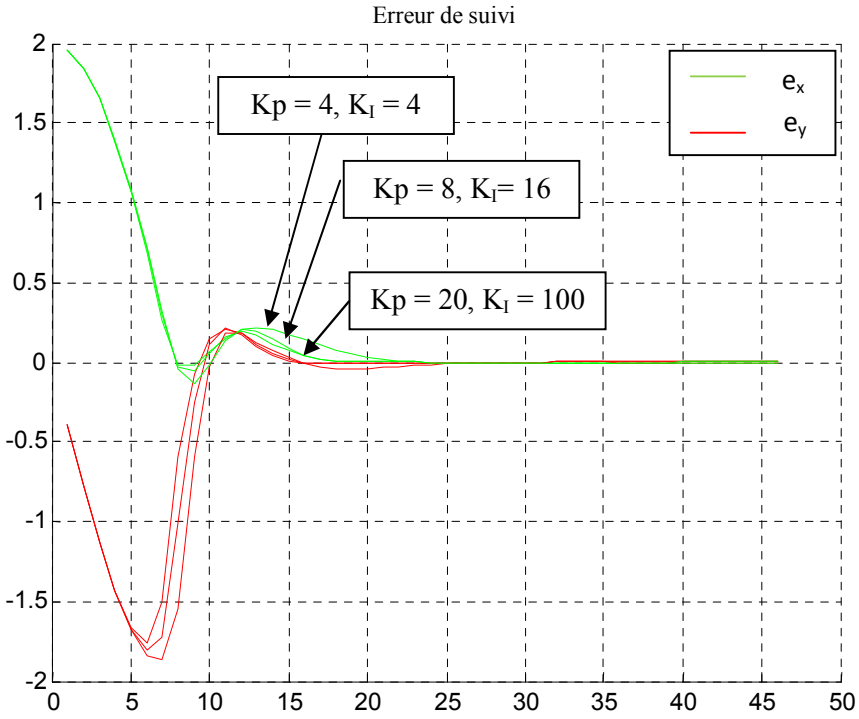


Figure (2.14) : Erreur de suivi.

Les figures de simulation (2.11) à (2.14) montent que le système critique donne un bon résultat " le temps de réponse, l'erreur de suivi "dans le cas ou Kp est relativement petit que K_I

- ***Pour un système Sous amorti*** $\Delta < 0$

$$\Delta < 0 \rightarrow K_p{}^2 - 4 * K_I < 0 \rightarrow K_p{}^2 < 4 * K_I \rightarrow K_p < 2 * \sqrt{K_I}$$

Pour $\Delta < 0$ donc j'ai deux pôles :

$$s_1 = \frac{-K_p + i\sqrt{-(K_p{}^2 - 4*K_I)}}{2} \qquad (2.17)$$

$$s_2 = \frac{-K_p - i\sqrt{-(K_p{}^2 - 4*K_I)}}{2} \qquad (2.17)$$

Pour un régime sous amorti stable il faut vérifier que la partie real de s_1 et s_2 négative et la valeur du partie imaginaire est très proche à l'axe des ordonnées.

$\Delta = K_p{}^2 - 4 * K_I$	***Régime sous amorti*** $\Delta < 0$		
K_p	1	4	7
K_I	4	16	16
$K_p < 2 * \sqrt{K_I}$	Vérifier	Vérifier	Vérifier
$S_{1,2}$	< 0 (système stable)		

Figure (2.3): Les paramètres du correcteur PI pour "sous amorti"

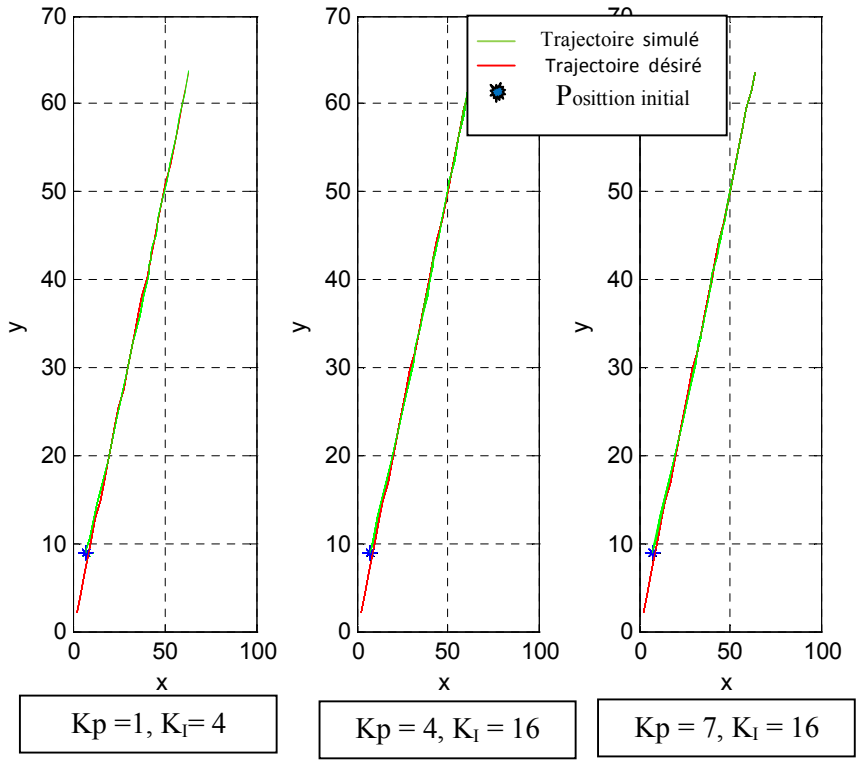


Figure (2.15) :poursuite d'une trajectoire linéaire.

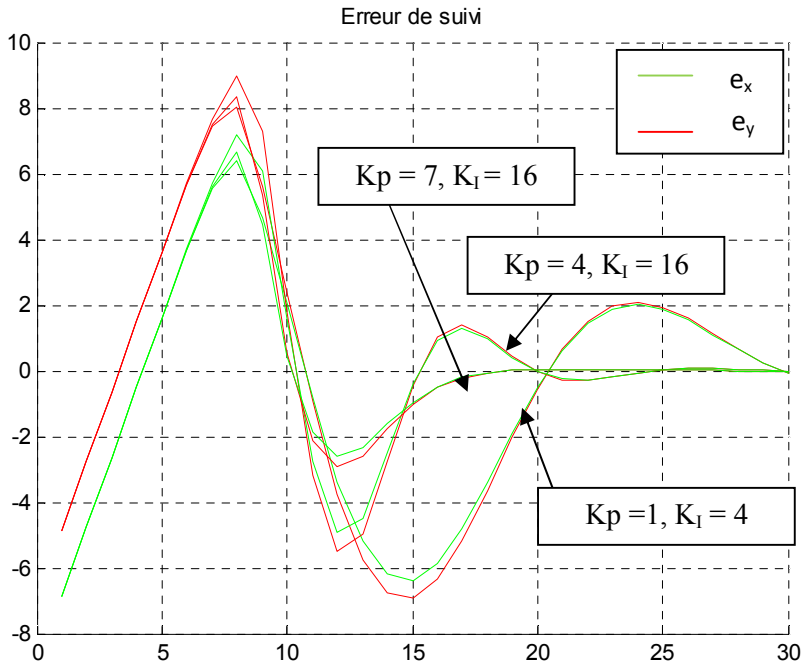


Figure (2.16) :Erreur de suivi.

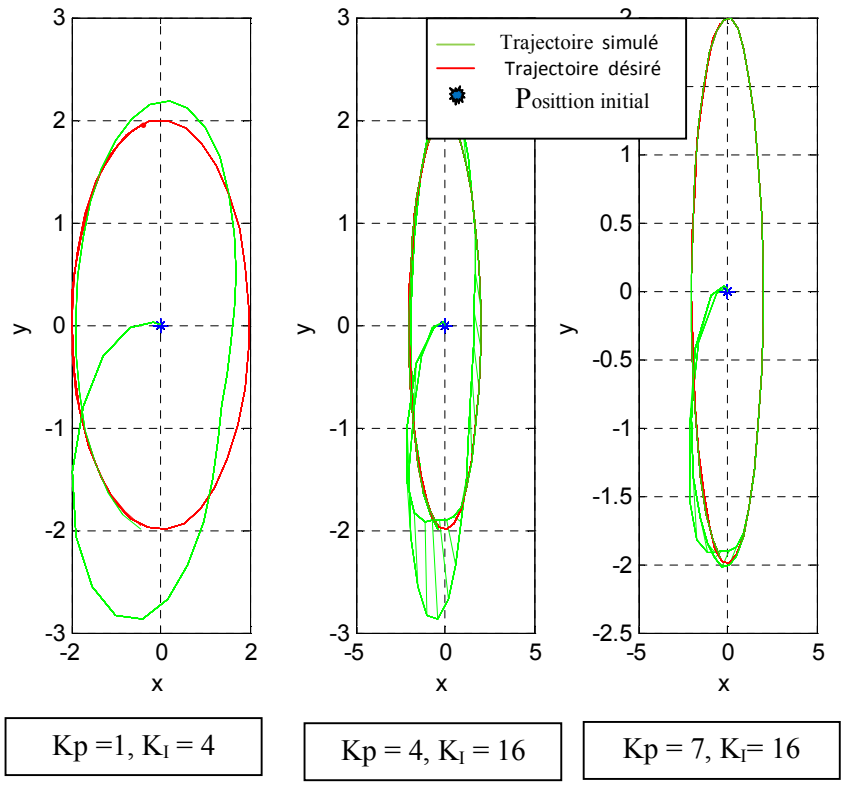


Figure (2.17) :poursuite d'une trajectoire circulaire.

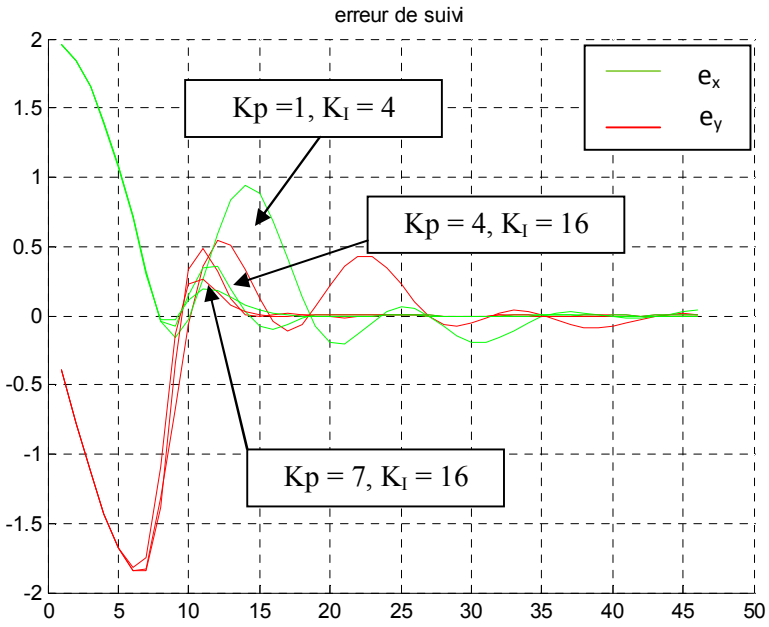


Figure (2.18) : Erreur de suivi.

Les figures de simulation (2.15) à (2.18) montent que le système sous amorti donne un bon résultat dans le cas ou les pôles présentent des parties imaginaires proches à l'axe des ordonnées mais le temps de réponse reste toujours grand et on observe des dépassements.

En comparant, on remarque que le régime apériodique donne le meilleur résultat.

II.4. Résultats de Simulations et discussion :

Le but de ces simulations est d'examiner l'efficacité et les performances de la méthodologie utilisée. En optant pour les valeurs des paramètres du contrôleur PI K_p et K_I ; tels que $\mathbf{K_p = diag\{10, 10, 10\}}$ **et** $\mathbf{K_I = diag\{0.2, 0.2, 0.2\}}$, on obtient les résultats suivants.

II.4.1. La stabilisation point-à-point :

La première simulation a été réalisée pour étudier les performances de la régulation point-à-point. On prend comme configuration initiale de la base mobile omnidirectionnelle l'origine, tel que : $[\boldsymbol{x_0\ y_0\ \vartheta_0}]^T = [0\ 0\ 0]^T$. Les configurations finales désirées sont 8 et 16 point situés sur un cercle comme l'indiquent dans les figures suivantes :

La configuration initiale est $[x_0\ y_0\ \vartheta_0]^T = [0\ 0\ 0]^T$

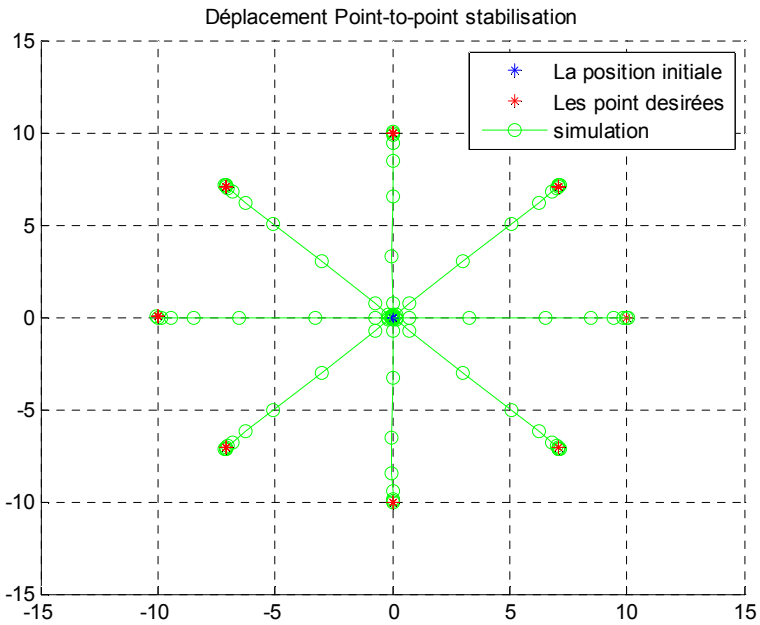


Figure (2.19) : Déplacement du point -à- point (1 vers 8)

La configuration initiale est $[x_0\ y_0\ \vartheta_0]^T = [0\ 0\ 0]^T$

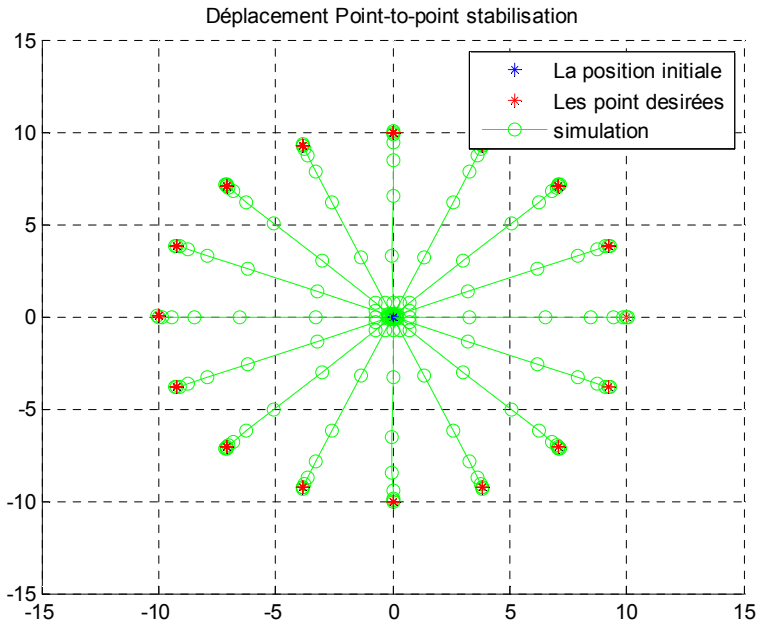


Figure (2.20) : Déplacement point -à- point (1vers16)

Les figures (2.19) à (2.20**)** montrent toutes les trajectoires simulées du robot mobile sous forme de segments de droites connectant le point initial, aux points finaux. Ce qui prouve que les trajectoires obtenues sont de longueurs minimales. En outre, il est important de remarquer que le robot mobile atteint sa destination finale avec la configuration désirée, C'est-à-dire tout en satisfaisant l'orientation souhaitée, montrant ainsi la propriété unique du robot mobile omnidirectionnel à roues.

II.4.2. Poursuite d'une trajectoire linéaire :

La deuxième simulation a été utilisée pour étudier les performances de la commande cinématique pour une poursuite de trajectoire linéaire.

La configuration initiale est $[x_0\ y_0\ \vartheta_0]^T = [0\ 0\ 0]^T$

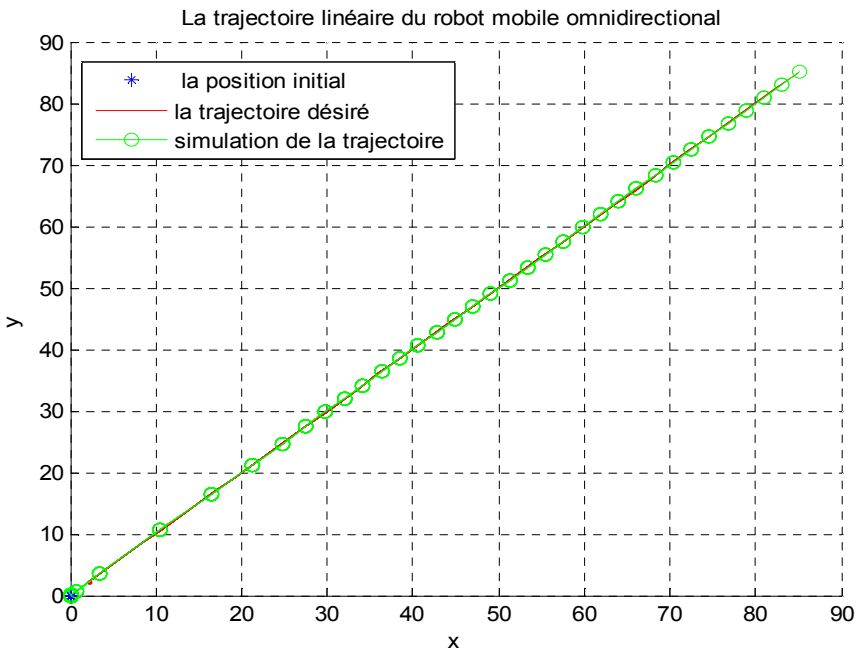


Figure (2.21) : Poursuite d'une trajectoire linéaire

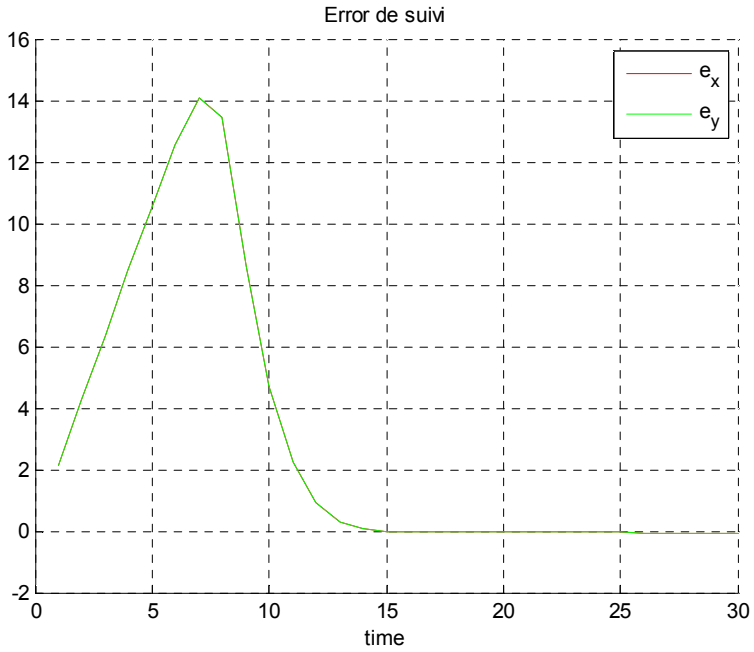


Figure (2.22) : erreur de poursuit

La configuration initiale est $[x_0\ y_0\ \vartheta_0]^T = [2\ 7\ 0]^T$

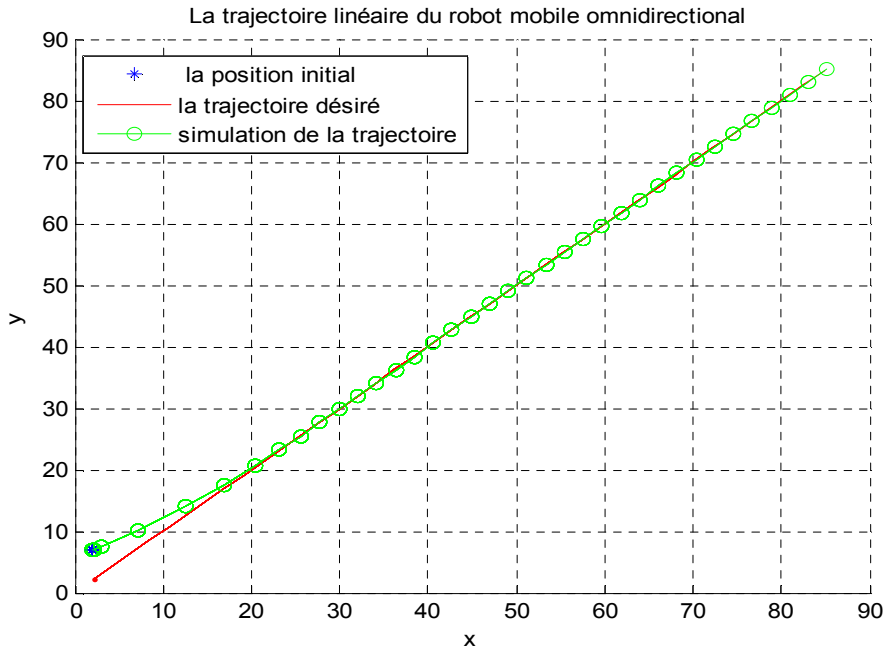


Figure (2.23) : Poursuite une trajectoire linéaire

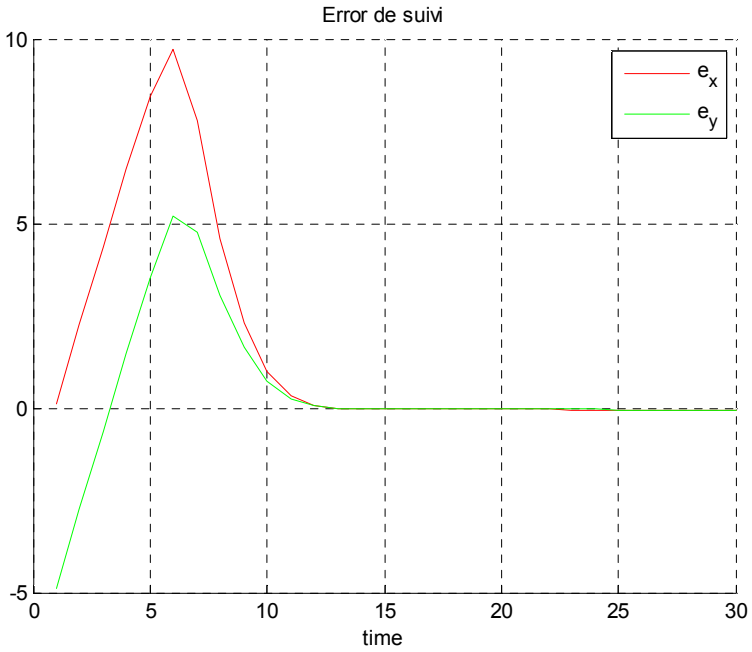


Figure (2.24) : erreur de poursuite

Les résultats indiqués par les figures (2.21) à (2.24) montrent que la loi de commande proposée est capable de commander le robot mobile pour une poursuite adéquate avec une erreur qui tend exponentiellement vers zéro

II.4.3. Poursuite d'une trajectoire circulaire :

Ce paragraphe vise à observer le comportement du contrôleur PI lorsque le robot mobile se déplace sur une trajectoire circulaire. La simulation est réalisée avec une configuration initiale à l'origine telle que :

La configuration initiale est : $[x_0\ y_0\ \vartheta_0]^T = [0\ 0\ 0]^T$

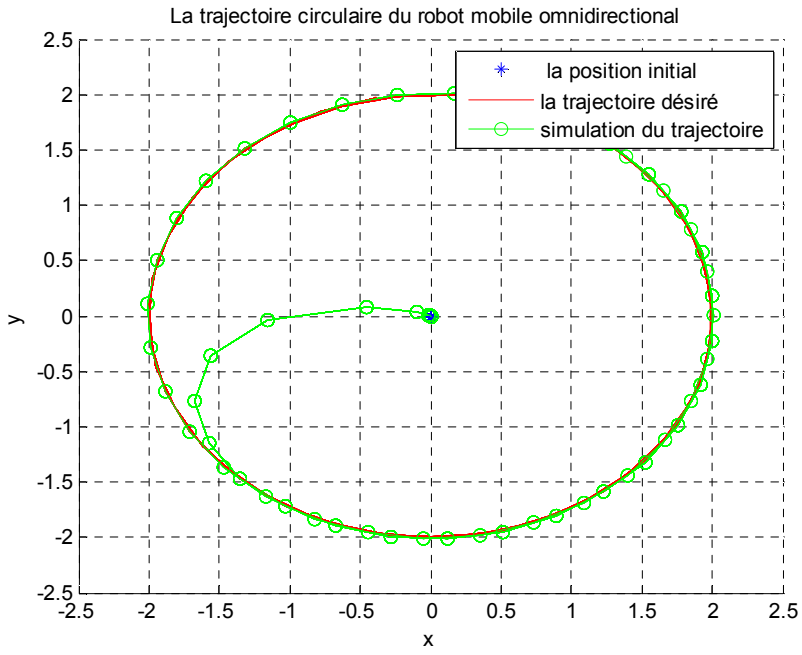


Figure (2.25) : Poursuite d'une trajectoire circulaire

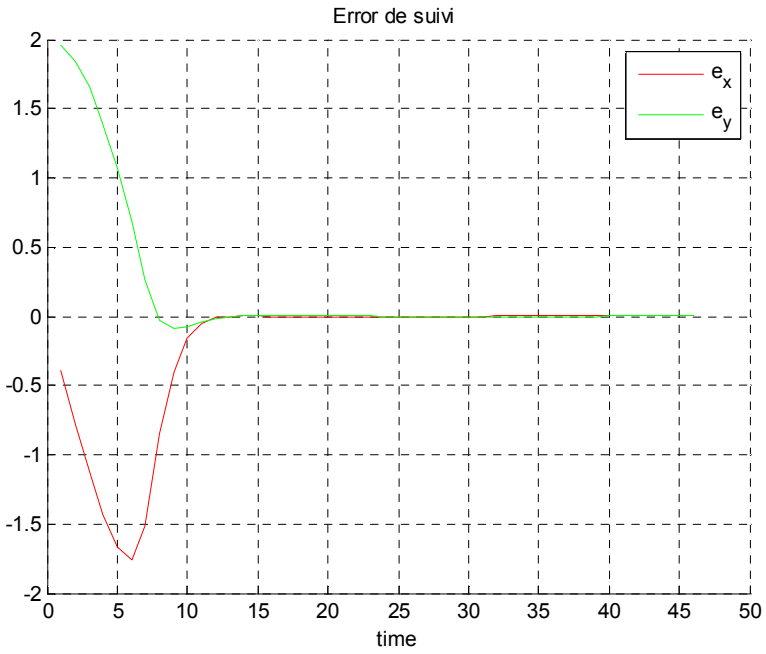


Figure (2.26) : erreur de poursuite

La configuration initiale est : $[x_0\ y_0\ \vartheta_0]^T = [3\ 3\ 0]^T$

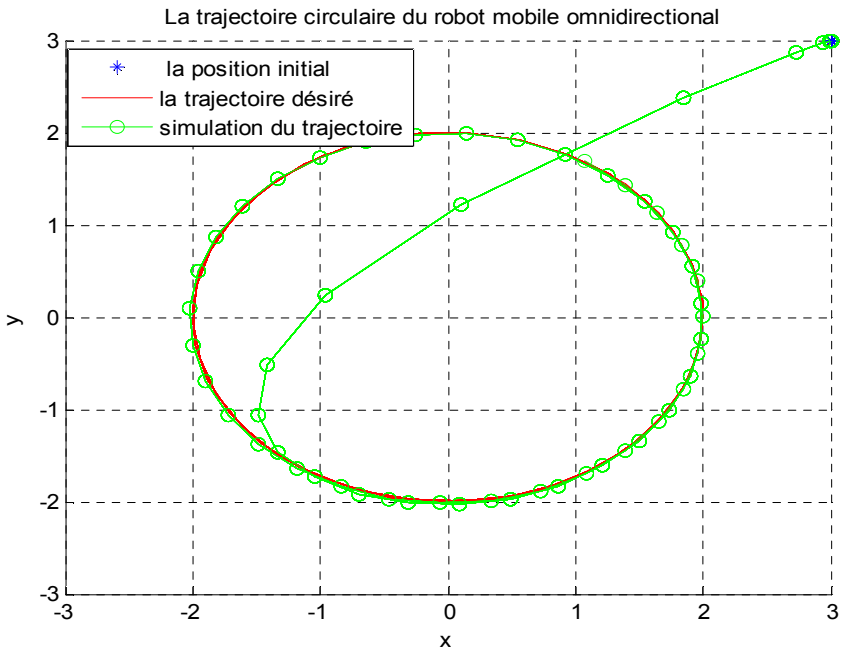


Figure (2.27) : Poursuite d'une trajectoire circulaire

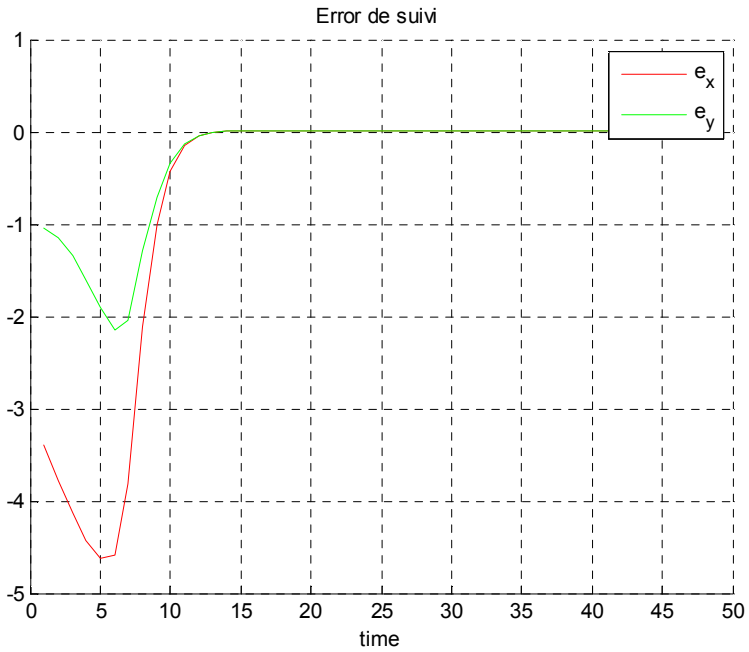


Figure (2.28) : erreur de poursuite

Les figures (2.25) à (2.28) montent les résultats de simulation d'une poursuite de trajectoire circulaire par un robot mobile. Ces résultats prouvent que le contrôleur cinématique proposé est efficace pour les applications de déplacement point à point et de poursuite.

II.4.4. Poursuite d'une trajectoire elliptique :

La dernière simulation montre les capacités du contrôleur PI à orienter le robot de manière à le laisser suivre une trajectoire elliptique.

La configuration initial est : $[x_0\ y_0\ \vartheta_0]^T = [0\ 0\ 0]^T$

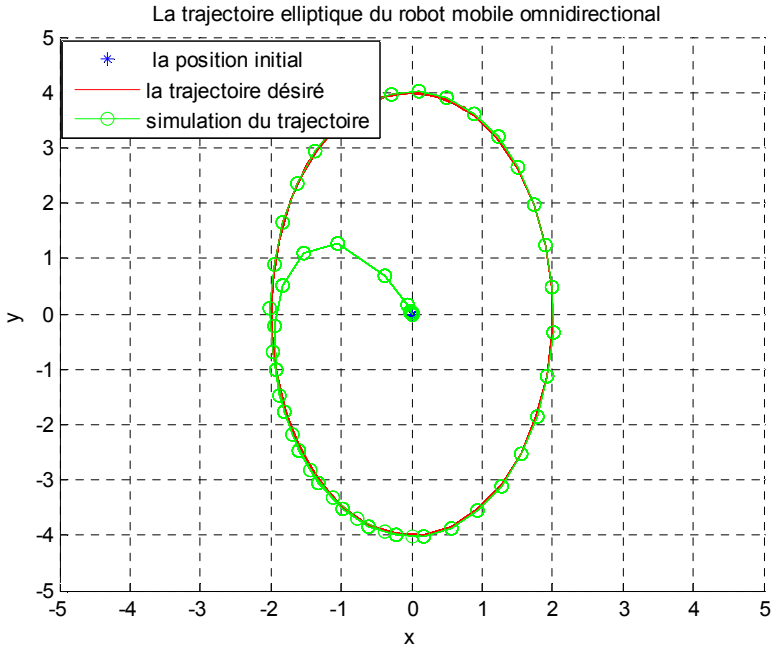


Figure (2.29) : Poursuite une trajectoire elliptique

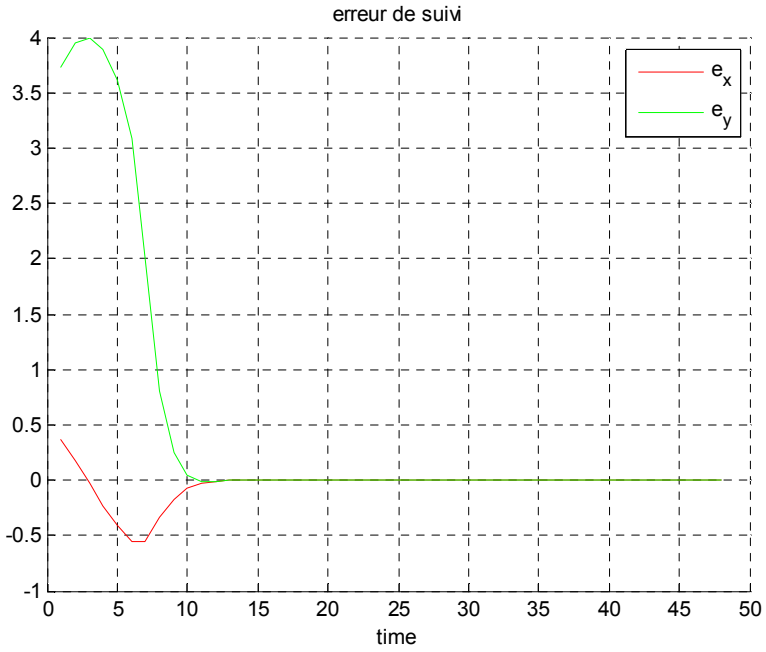


Figure (2.30) : erreur de poursuite.

La configuration initial est : $[x_0\ y_0\ \vartheta_0]^T = [3\ 3\ 0]^T$

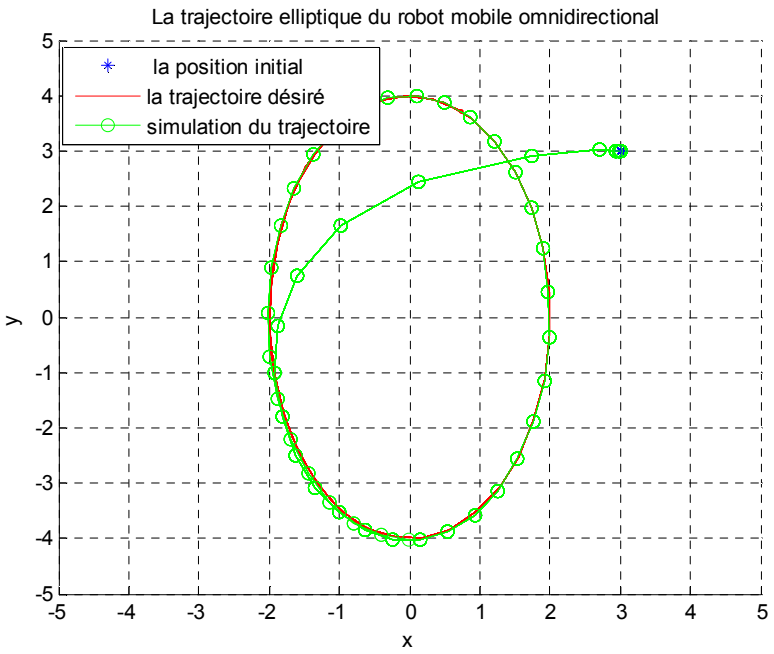


Figure (2.31) : Poursuite une trajectoire elliptique

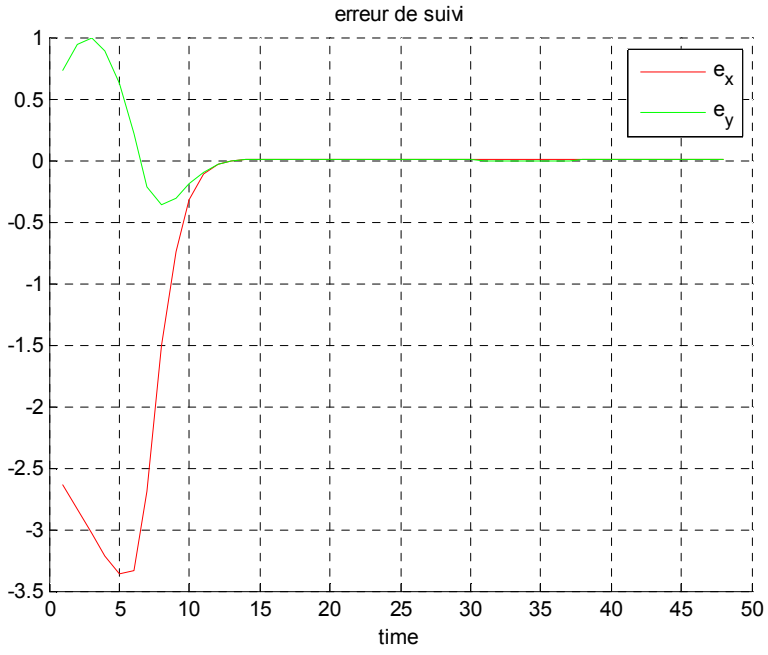


Figure (2.32) : erreur de poursuite

Les figures (2.29) à (2.32) montrent la poursuite de la trajectoire elliptique pour une configuration initiale différente de celle de l'origine. Ces résultats montrent que le dispositif

de la commande cinématique est capable de diriger la base mobile omnidirectionnelle en la laissant se déplacer exactement sur la trajectoire elliptique.

II.5. Conclusion :

Dans ce chapitre nous avons développé des méthodologies pour la modélisation et le contrôle d'un robot mobile omnidirectionnel "*Robotino*". On détermine le modèle cinématique et on applique une commande non linéaire " feedback linéarisation" qui basé sur l'utilisation du contrôleur PI " proportionnable intégrateur". Il a été choisi les valeurs du contrôleur PI pour un régime apériodique qui donne un millier résulta vis à vis du temps de réponse et de l'erreur de suivi. A la fin nous avons présentes les résultats des simulations pour déplacement point-à-point et poursuite d'une trajectoire " linéaire, circulaire et elliptique".

CHAPITRE III :

La Modélisation Et La Calibration D'une Camera

CHAPITRE III

LA MODÉLISATION ET LA CALIBRATION DE LA CAMERA

III.1. Introduction :

Ce chapitre est consacré à la description du modèle qui correspond au processus de formation des images prises par une caméra. C'est pourquoi nous allons présenter le modèle **sténopé** qui peut nous permettre l'accès aux paramètres extrinsèques et intrinsèques de la caméra ainsi que les techniques utilisées pour son calibrage. Nous présenterons les relations ou les transformations homogènes entre les différents repères à savoir ; le repère caméra, le repère robot et le repère monde.

III.2. Modélisation de la Caméra :

III.2.1-Conception :

Une caméra doit réaliser une transformation ponctuelle qui fait passer d'un point physique de l'espace réel 3D à un point 2D sur le plan image. Ce qui revient à une transformation mathématique de R^3 vers R^2.

Il existe plusieurs modèles dans la modélisation de la formation des images numériques. Notre étude prend comme modèle celui du **sténopé** ; appelé également le modèle perspectif (*pin-hole* en anglais) ; c'est un dispositif optique et le plus utilisé dans la vision par ordinateur. Ce modèle permet d'établir une relation entre un point de coordonnées 3D de la scène observée et sa projection dans l'image en 2D.

III.2.2- Composants d'une caméra :

Une caméra se compose d'une boîte dont l'une de ses faces est percée d'un trou minuscule qui laisse entrer la lumière comme indique la figure (3.1).

L'image vient se former sur la face opposée au trou et celle-ci peut être capturée en y plaçant un support photosensible (papier photographique).

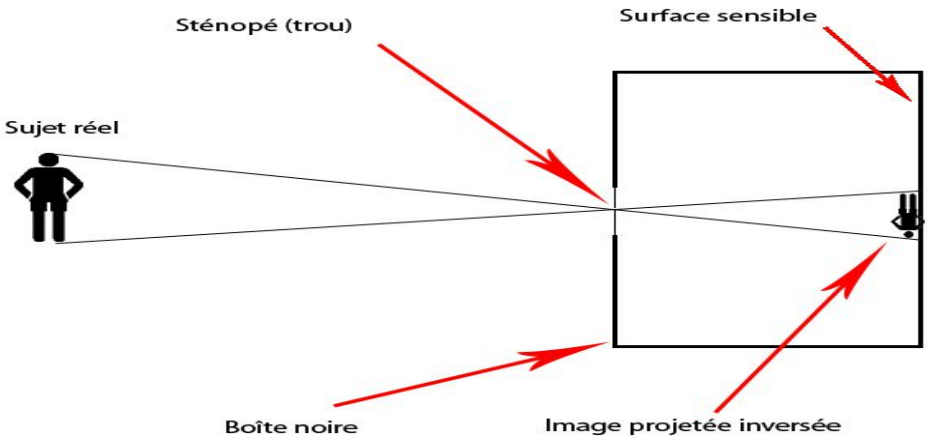


Figure (3.1) : Formation d'une image dans un sténopé

Une caméra perspective peut en effet être modélisée grâce au modèle du sténopé comme l'illustre la figure (3.2). Ce modèle associe à la caméra un repère cartésien ; ce repère dont son origine se situe sur le **centre optique C** est défini comme $\mathbf{R_c = [O_c\ X_c\ Y_c\ Z_c\ ]}$**.** La caméra est représentée par un **plan image**, le plan image est parallèle aux axes $\mathbf{X_c}$ et $\mathbf{Y_c}$**.** Il est situé à une distance **f** de l'origine **C** appelée **distance focale f** de ce plan. La droite passant par le centre de projection et perpendiculaire au plan image est **L'axe optique**. L'intersection de l'axe optique avec le plan image est appelé **point principal.**

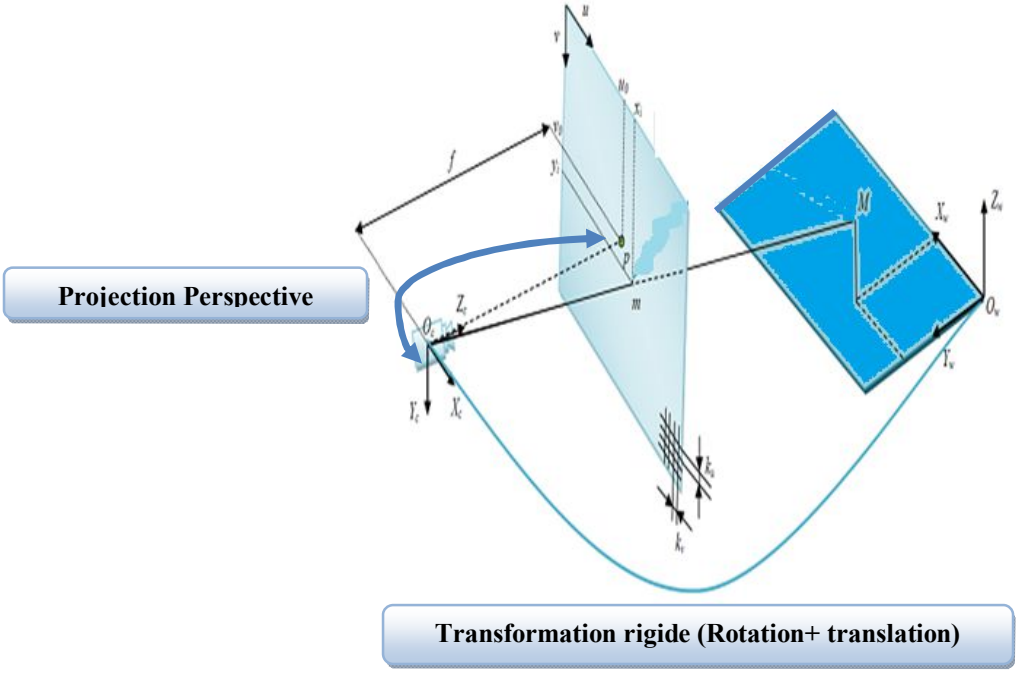


Figure (3.2) : Modèle géométrique du sténopé.

III.2.3-Transformation d'un point de l'espace sur le plan d'image :

On considère la droite passant par **M** et **C** comme étant le trajet de la lumière perçue par la caméra. La projection du point **M** de la scène sur le plan image est **m** résultant aussi de l'intersection de la droite **(CM)** avec le plan image.

La projection du point **M** de l'espace sur le plan image peut être décomposée en trois transformations élémentaires successives :

- ✓ **Transformation rigide T**
- ✓ **Projection perspective P**
- ✓ **Transformation affine A**

III.2.3.1.La transformation rigide "T" :

La transformation rigide permet d'exprimer le point ***M*** de l'espace dans le repère caméra R_c telle que les coordonnées du point $M = [X_m\ Y_m\ Z_m\ 1]^T$ sont en général exprimé dans un repère arbitraire, appelé repère monde. Le repère caméra est le repère où l'origine se situe sur le centre optique **C,** voir la figure (3.3)

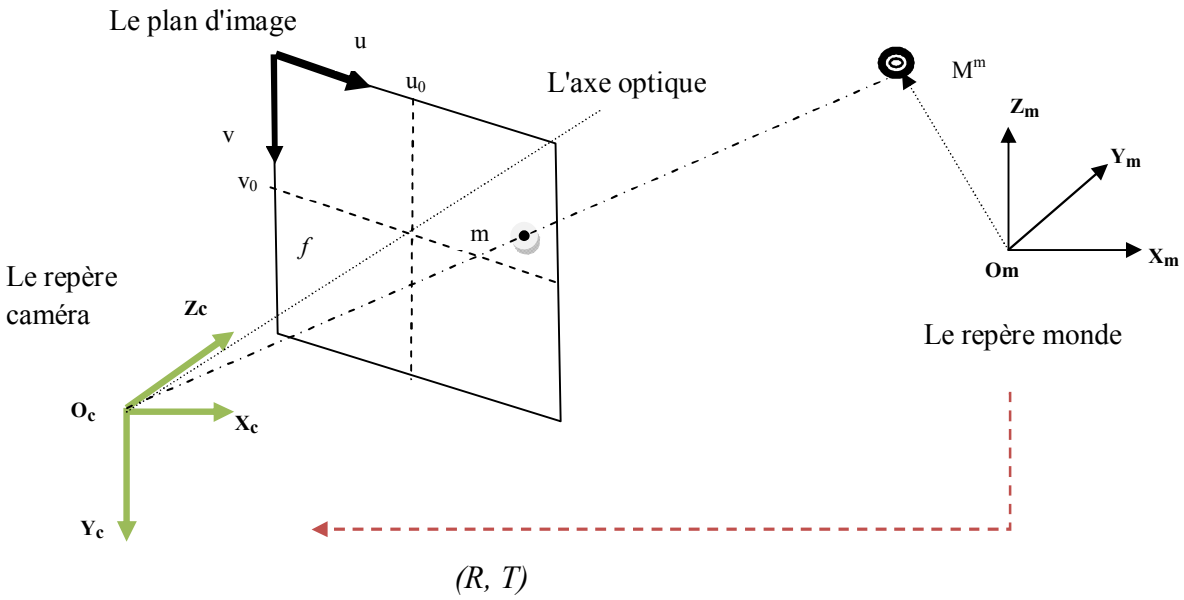


Figure (3.3) : Transformation Rigide

Les coordonnés $\boldsymbol{M}= [\boldsymbol{Xc\ Yc\ Zc\ 1}]^T$ du point M dans le repère caméra sont obtenues par la relation suivante :

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathbf{R}* \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} + \mathbf{t} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}_{\mathbf{3*1}}^{\boldsymbol{T}} & \mathbf{1} \end{bmatrix} * \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T}* \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \qquad (3.1)$$

Tel que **t** est le vecteur de translation entre le repère monde et le repère caméra de dimension (3*1).

$$\mathbf{t}=\begin{bmatrix} tx \\ ty \\ tz \end{bmatrix}$$

Et **R** est la matrice de rotation entre ces deux repères de dimension (3*3). Cette matrice de rotation peut être obtenue par le produit matriciel de trois matrices de rotation simple. Une rotation α autour d'un axe Δ sera exprimée par Rot (Δ, α). Comme le montre les équations suivantes, trois matrices de rotation simple peuvent être exprimées.

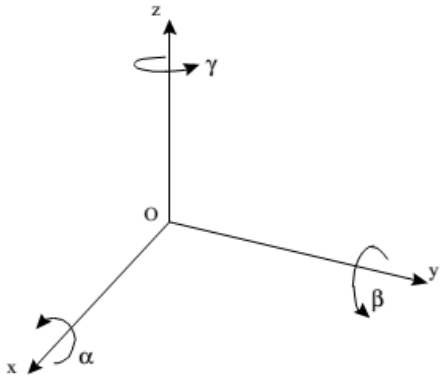


Figure (3.4) : Les trois rotations simples.

$$\text{Rot(X,}\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}; \text{Rot(Y,}\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix};$$

$$\text{Rot (Z,}\gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}= \text{Rot(X,}\alpha) * \text{Rot(Y,}\beta)* \text{Rot (Z,}\gamma)$$

Les paramètres de cette transformation sont appelés **paramètres extrinsèques** de la caméra.

III.2.3.2.Projection perspective "P" :

La projection perspective **P** transforme un point de 3D de la caméra **Rc** en un point 2D du plan image **Ri** de coordonnées **[x y 1]** ^{T} **;**

Les coordonnées du point **M** sont exprimées dans le repère attaché à la caméra de coordonnées $[X_c\ Y_c\ Z_c]^{T}$; le plan x-y de ce repère est parallèle au plan image et l'axe z est croisé avec l'axe optique. L'origine de ce repère se trouve en Oc. D'après le théorème de Thales les coordonnées du point projeté sont :

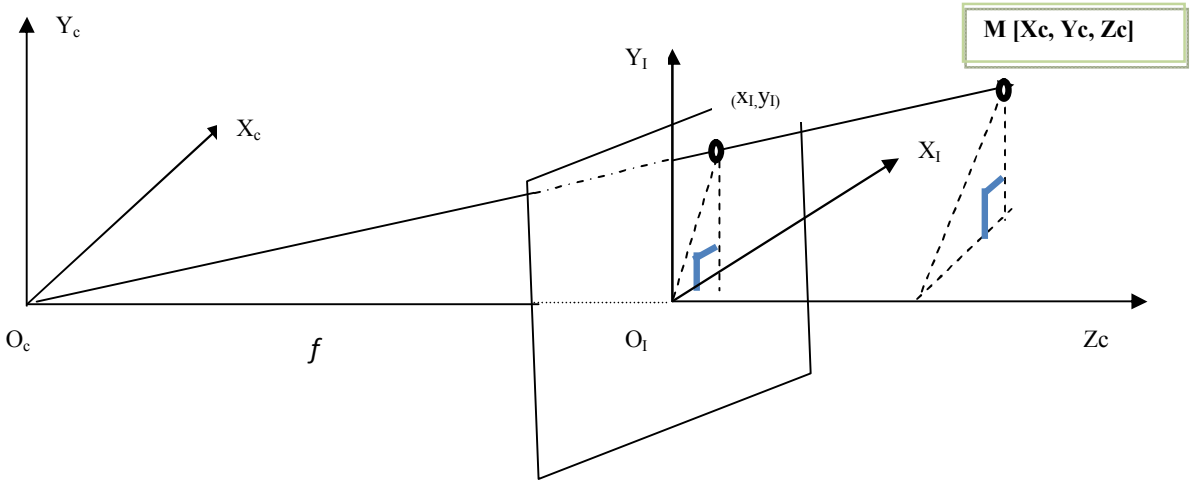


Figure (3.5) : La projection Perspective

$$\frac{f}{zc} = \frac{X_I}{Xc} = \frac{y_I}{Yc}$$

$$X_I = f*\frac{Xc}{zc} \quad ; \quad y_I = f*\frac{yc}{zc} \qquad (a)$$

Cette transformation peut s'écrire :

$$\begin{bmatrix} X_I \\ y_I \\ 1 \end{bmatrix} = = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = P* \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \qquad (3.2)$$

Ou f est la distance focale

III.2.3.3.La transformation affine "A" :

La transformation affine A permet de convertir les coordonnées images métriques

$[x_I\ y_I\ 1\ ]^T$ en coordonnées images discrètes [u v 1]^T.

Les coordonnées pixelliques (u, v) dépend des dimensions (I_x, I_y) d'un pixel ainsi que des coordonnées pixelliques (u_0, v_0) du point principal dans l'image et θ traduit la non orthogonalité potentielle des lignes et des colonnes de l'image comme le montre la figure (3.6)

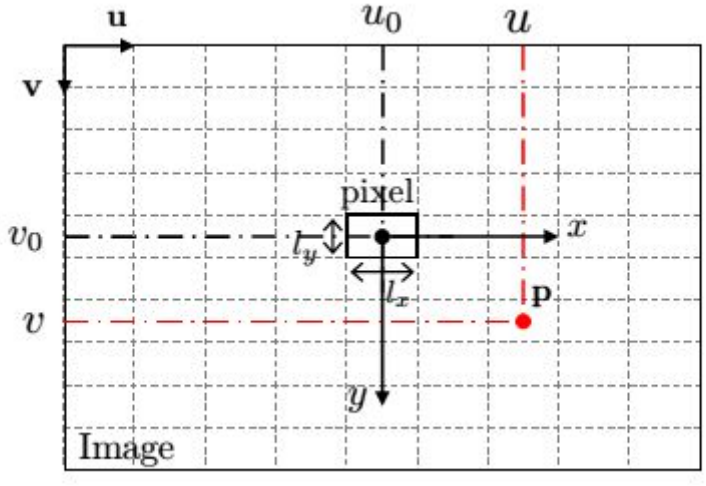


Figure (3.6) : Les coordonnées pixelliques

$$u = u_0 + k_u x_I$$
$$v = v_0 + k_v y_I$$

On pose : $k_u = \frac{1}{I_x}$, $k_v = \frac{1}{I_y}$.

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix} = \begin{bmatrix} k_u & -k_u cos\theta & u_0 \\ 0 & k_v sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_I \\ y_I \\ 1 \end{bmatrix} = A * \begin{bmatrix} X_I \\ y_I \\ 1 \end{bmatrix} \qquad (3.3)$$

Avec : $\boldsymbol{I_x}$: Largeur d'un pixel

$\boldsymbol{I_y}$: Hauteur d'un pixel

s est un facteur d'échelle qui correspond à la profondeur du point 3D observé.

En pratique, on considère souvent que les lignes et les colonnes de l'image sont perpendiculaires, et donc que $\theta = \frac{\pi}{2}$. Donc l'équation (3.3) peut alors se simplifier :

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_I \\ y_I \\ 1 \end{bmatrix} = A * \begin{bmatrix} X_I \\ y_I \\ 1 \end{bmatrix} \qquad (3.4)$$

D'après les relations (3.2) et (3.4), un point M de l'espace, de coordonnées métriques ^{c}**M=[Om Xm Ym Zm]** dans R_c se projette dans le plan image selon la relation suivante :

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3.6)$$

En posant $\rho_u = fk_u$ et $\rho_v = fk_v$, le passage d'un point M de l'espace à un pixel dans l'image peut être uniquement écrit avec la matrice extrinsèque et la matrice intrinsèque :

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix} = \begin{bmatrix} \rho_u & 0 & u_0 & 0 \\ 0 & \rho_u & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \mathrm{K} * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3.7)$$

L'équation (3.7) contient toutes les grandeurs physiques de la caméra :

- La distance focale f
- Les coordonnées du point principal (u_0, v_0)
- Le facteur d'échelle(k_u, k_v)
- θ le paramètre d'oblicité

Ces paramètres appelés **paramètres intrinsèques** ; qui doivent être estimés par le calibrage de la caméra. Donc la projection m= [u v 1] dans le plan image du point M=[**Om Xm Ym Xm**] de l'espace et obtenue par la relation :

$$\mathbf{m = K*T*M = P_{proj}*M} \quad (3.8)$$

III.3. Calibration D'une Caméra :

Le calibrage consiste à estimer les paramètres intrinsèques et extrinsèques d'un modèle de caméra à partir d'un ensemble de points 3-D et de leur image. Il s'agit donc d'estimer les éléments de la matrice (3.7) ; Cette étape est incontournable pour de nombreuses applications de vision par l'ordinateur.

Le calibrage de la caméra a été préalablement traité par la communauté de la photogrammétrie. Par conséquent, de nombreuses méthodes de calibrage ont été proposées dans la littérature. Ces approches sont généralement classifiées en deux catégories :

III.3.1.Calibrage avec un objet 3D de référence ou une mire :

Cette technique utilise l'observation d'objets en 3D avec des coordonnées connues. Les objets de calibrage (mire) (figure (3.7)) sont généralement des points répartis sur des plans orthogonaux ou sur un plan translaté dans la direction de sa normale. Le calcul peut alors être effectué de façon relativement simple [dib11].

III.3.2.Calibrage automatique (ou auto-calibrage) :

Dans cette technique, Le mouvement connu de la caméra filmant une scène statique est utilisé pour poser des contraintes sur les paramètres intrinsèques prenant en compte la rigidité des objets filmés en utilisant uniquement les informations de l'image [dib11].

Nous avons opté pour la première famille ; sachant qu'elle est disponible sur internet " **camera calibration toolbox for matlab**". Elle consiste à calibrer la caméra à partir de plusieurs images d'une mire. Prises sous des points de vue différents comme indique la figure (3.8).

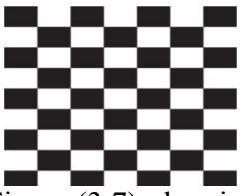


Figure (3.7) : la mire

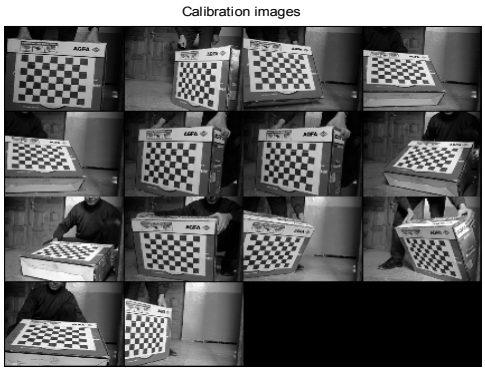


Figure (3.8) : différentes positions d'une mire du calibrage

Les positions des coins de chaque carré de la mire sont alors extraites puis raffinées en cas de distorsion. Ensuite les paramètres de la caméra sont déterminés par optimisation non linéaire.

III.4. Résultat de la calibration de la caméra :

Nous avons utilisé une webcam, type **Logitech 720 hp** (figure (3.9)) pour avoir un modèle et procéder à l'expérimentation. La caméra est placée sur un robot mobile qui se trouve au niveau du département d'électronique pour déterminer les paramètres intrinsèques de la caméra obtenus lors de la phase de calibrage. Leurs valeurs sont résumées dans le tableau (3.1)

Figure (3.9): webcam logitech 720 hp

Les Paramètres intrinsèques de la caméra		
f	f =35 mm	La distance focale
(u_0, v_0)	u_0 =553pixels v_0 =476 pixels	Les coordonnées du point principal (projection du centre optique)
(fk_u, fk_w)	(1751mm, 1750mm)	La distance focale/ Les dimensions d'un pixel
θ	0	Le paramètre d'oblicité

Table (3.1) : les résultats du calibrage

Les paramètres extrinsèques :

La figure suivante présente le repère de la caméra $R_c\ (O_c, X_c, Y_c, Z_c)$. La pyramide rouge correspond au champ de vue effectif de la caméra défini par le plan d'image.

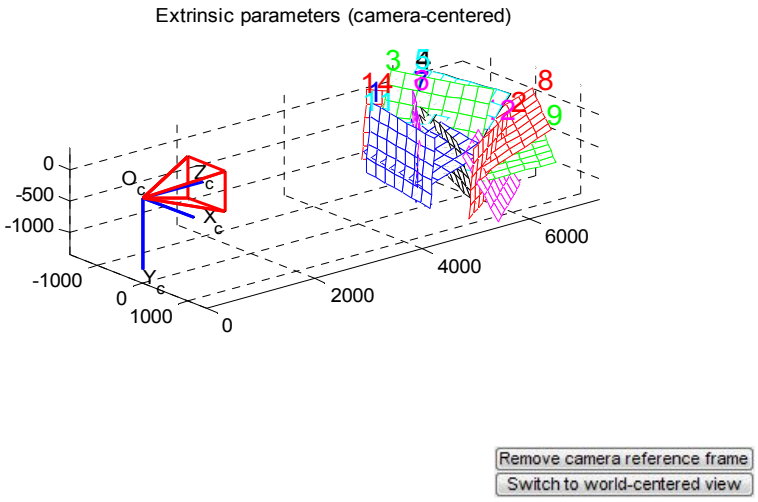


Figure (3.10) : Le champ de vue effectif de la caméra

Sur cette nouvelle figure, chaque position et orientation de la caméra sont représentées par une pyramide verte par rapport à chaque image de calibration.

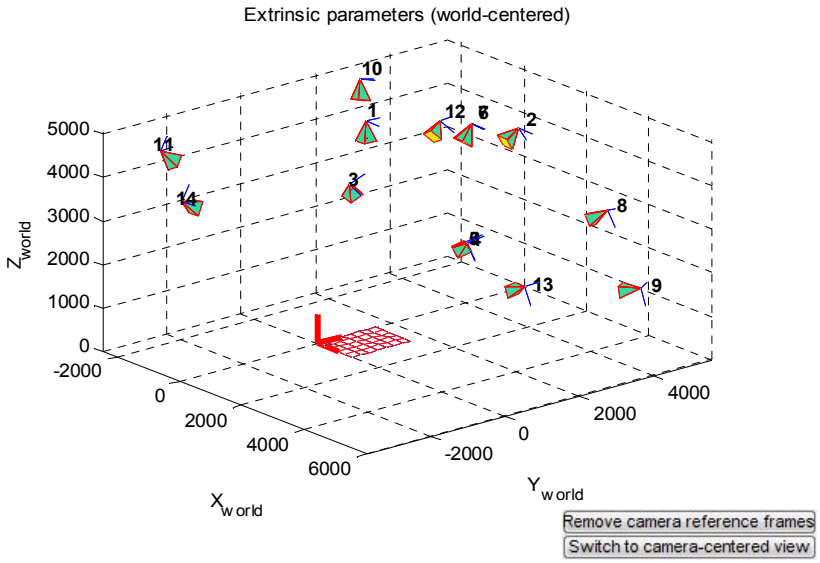


Figure (3.11) : Estimation relative de la pose d'une caméra par rapport à chaque image de calibration

III.5. Définition des repères :

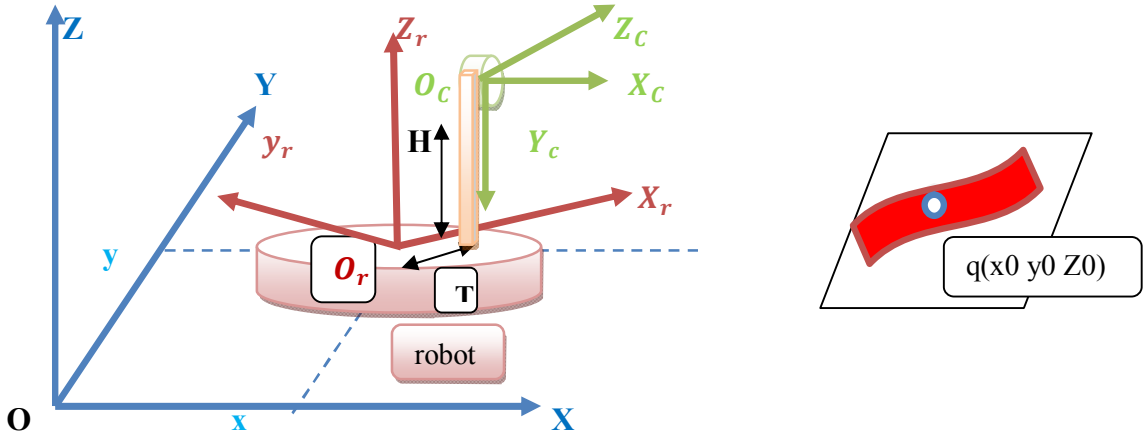


Figure (3.12) : Modèle du système robotique

La modélisation du robot s'appuie sur différents repères Figure (3.11) définis comme suit :

$R(O, X\,Y\,, Y)$: Le repère lié à la scène (repère monde)

$R_r\ (O_r, X_r, Y_r\,, Z_r)$: Le repère lié à la base mobile

$R_c\ (O_c, X_c, Y_c\,, Z_c)$: Le repère lié à la caméra

Pour la base mobile ; la position est représentée par les coordonnées (x,y) du point O_r dans le repère R_r tandis que l'orientation est donnée par l'angle θ. la position de la caméra dans le repère R_r est décrite par le vecteur $\boldsymbol{O_cO_r}$**=(T,0,H) .**

$q_b\ (X_0, Y_0\,, Z_0, 1)$: Un point de trajectoire est exprimé dans le repère monde. Sa projection dans le plan image est $q_I(u_I, v_I, 1)$. Alors, comme nous l'avons vu la relation : $\boldsymbol{q_I}$ **=K*** $\boldsymbol{T_b^c}$ ***** $\boldsymbol{q_b}$

Après calibrage, la matrice K est déterminée et le problème consiste ensuite à trouver la matrice $\boldsymbol{T_b^c}$ **.**

La relation entre le point de l'espace dans le repère monde et repère caméra est :

$$\boldsymbol{q_c} = \boldsymbol{T_r^c} * \boldsymbol{T_b^r} * \boldsymbol{q_b}$$

III.6. Résultats de Simulations et discussion :

Pour effectuer un asservissement visuel avec les paramètres obtenus, Nous avons procéder à une manipulation qui consiste à fournir au robot mobile les coordonnées échantillonnées d'une trajectoire que nous avons tracé sur une surface bien illuminée. Pour vérifier l'exactitude des coordonnées mesurées, nous avons choisi deux points sur la trajectoire à poursuivre en calculant leurs valeurs et les comparer avec celles mesurées (figure (3.13)). Les résultats de comparaison sont très satisfaisants et l'erreur est presque inexistante. Les paramètres extrinsèque des points sont présentés dans le tableau (3.2), avec :

- la position de la caméra par rapport au repère monde est :
  $[x, y, z]^T = [20cm, 25cm, 30\ cm]^T$
- L'angle de rotation a été choisi entre le repère caméra et le repère monde.

	Point 1	**Point 2**
Translation	[x, y, z] = [25cm, 175cm, 0]	[x, y, z] = [25cm, 200cm, 0]
Rotation	Rot(x,$-pi/2$)	Rot(x,$-\mathrm{pi}/2$)
Les coordonnées pixelliques	[u, v]= [680,966]	[u, v]= [680,918]

Table (3.2) : Tableau des paramètres extrinsèque des points sélectionné sur la trajectoire.

Figure (3.13) : Les coordonnées pixelliques des points cible

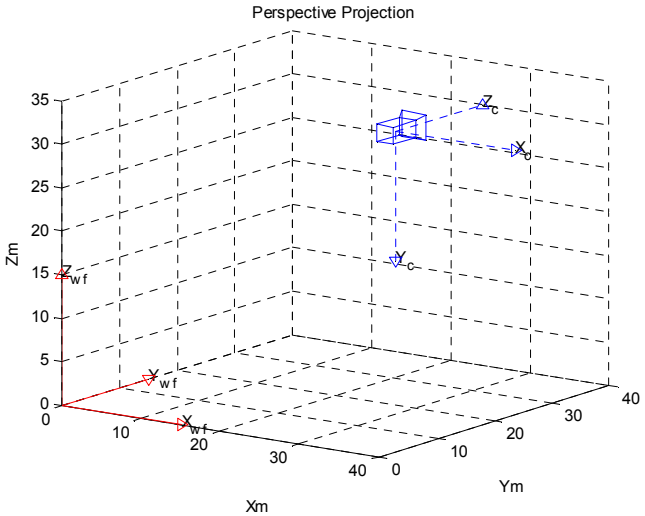


Figure (3.14) : Position du repère caméra par rapport au repère monde

Les paramètres extrinsèques de première Point :

Translation : [x,y,z] =[25cm,175cm,0]. Rotation: Rot(x,-pi/2)

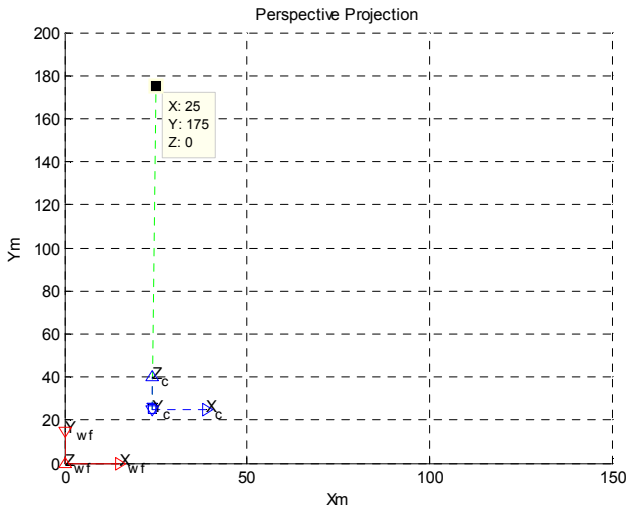


Figure (3.15) : Position du premier point cible par rapport au repère caméra et le repère monde en 2D

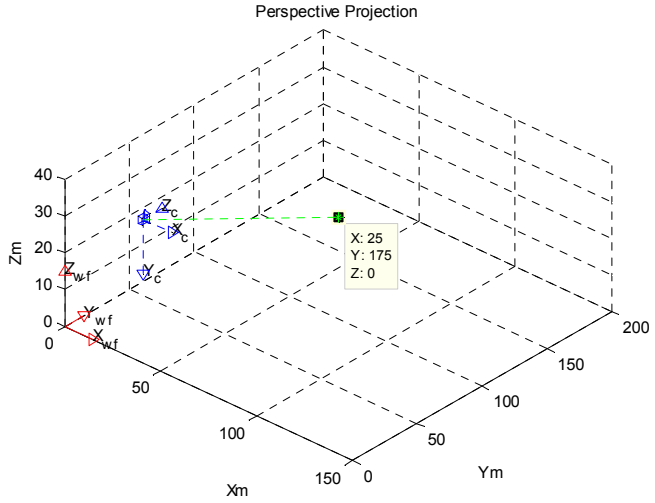


Figure (3.16) : Position de la caméra et le premier point cible par rapport au repère monde en 3D

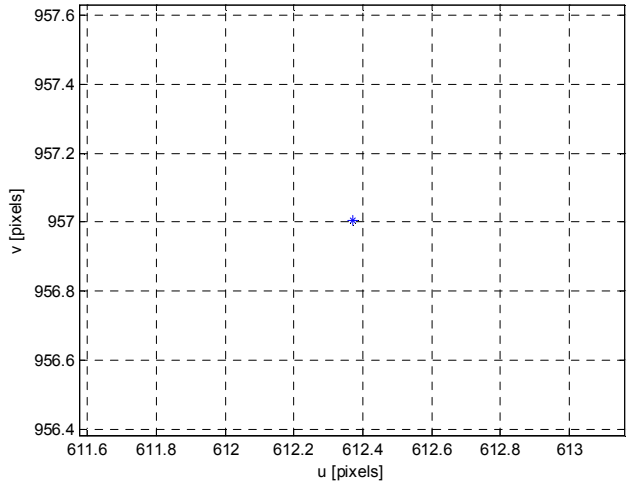


Figure (3.17) : Les coordonnées pixelliques du premier point cible après simulation

Les paramètres extrinsèques de deuxième Point :

Translation :[x,y,z] =[25cm,200cm,0]. Rotation: Rot(x,-pi/2)

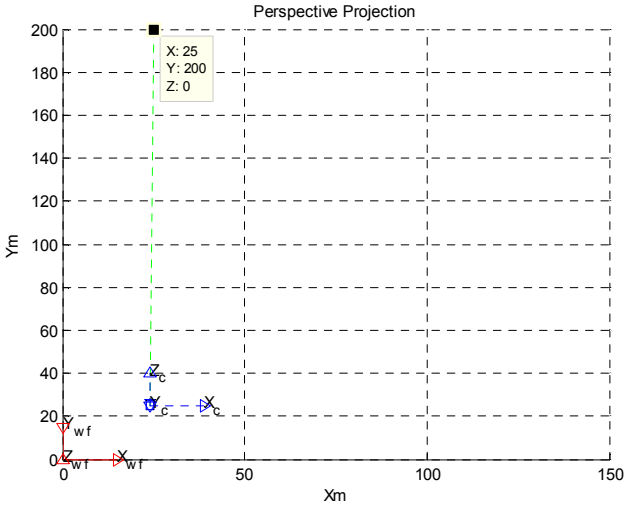


Figure (3.18) : Position du deuxième point cible par rapport au repère caméra et le repère monde en 2D

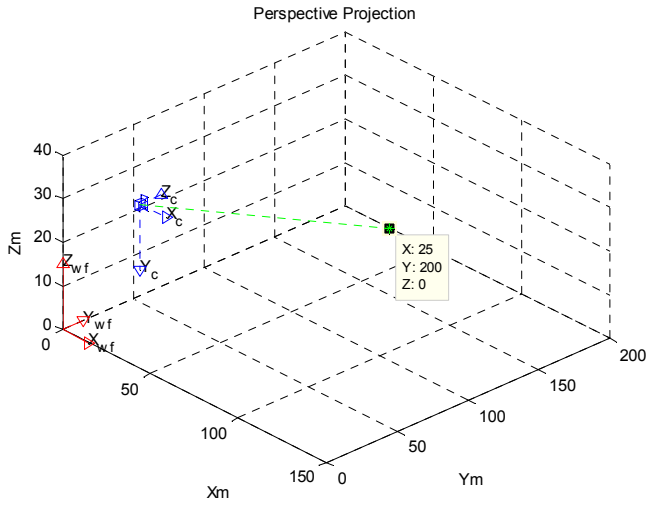


Figure (3.19) : Position de la caméra et du deuxième point cible par rapport au repère monde en 3D

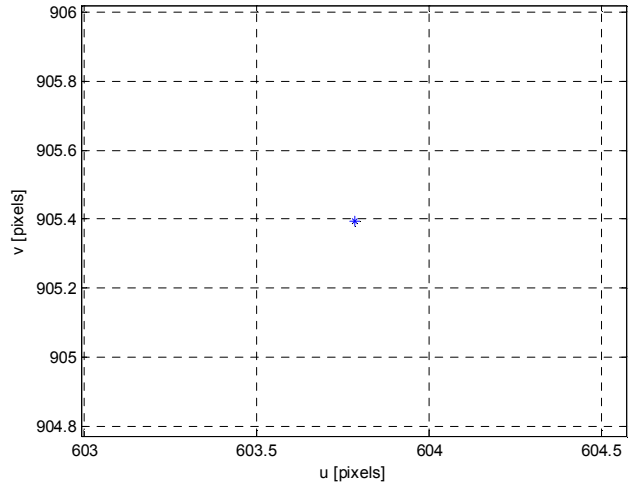


Figure (3.20) : Les coordonnées pixelliques du deuxième point cible après simulation

La figure (3.17) présente les coordonnées pixelliques du premier point choisi dont les coordonnées pixelliques réelle sont : (u, v)=(682,966). Par contre, les coordonnées pixelliques obtenus par calcul sont (u, v)=(612,957). Pour appuyiez notre démarche, on a procédé à un deuxième test avec un autre point pris sur la trajectoire de référence. La figure (3.20) présente les coordonnées pixelliques du deuxième point dont les coordonnées pixelliques réelle sont: (u,v)=(680,918). D'autre part on a calculé les coordonnées pixelliques en utilisant les matrices de transformées en obtenant ainsi (u, v)=(603,905). Les erreurs commises dans le premier cas et dans le deuxième sont successivement égales à (Δu, Δv)=(70,9) et (Δu, Δv)=(77,13). Nous pouvons expliquer cette différence au fait que la caméra n'est pas bien fixée sur son support,

ce qui provoque une erreur dans l'angle ; chose qui reste encore une des difficultés rencontrés à cause de la non disponibilité de moyens pour cela.

III.7.Conclusion :

Après avoir présenté la partie calibrage et modélisation de la caméra nous avons procédé au calibrage d'une Webcam caméra que nous avons choisi en tenant compte du prix et de la qualité. Cette étape est nécessaire pour déterminer les paramètres intrinsèques et extrinsèques de la caméra. La modélisation de la caméra présente la projection du point de l'espace 3D à un point 2D sur le plan image par trois transformations élémentaire successives : la transformation rigide, la projection perspective et la transformation affine.

CHAPITRE IV :

L'asservissement visuel

CHAPITRE IV

L'ASSERVISSEMENT VISUEL

IV. 1. Introduction :

L'asservissement visuel est une stratégie de commande qui consiste à utiliser les informations fournies par une ou plusieurs caméras pour contrôler les mouvements d'un système robotique afin de réaliser des taches de positionnement ou de suivi un objet. Nombreuses applications de la navigation visuelle d'un robot mobile par utilisation d'une, deux ou plusieurs cameras sont présentés et développés dans la vie humaine, pour la réalisation de taches de surveillance, de nettoyage, etc. La complexité des problèmes à résoudre dépend principalement de la compréhension des informations issues des capteurs visuels ainsi que la transformation de ces informations en données parfaitement utilisables par les composants décisionnels du robot. Ces informations sont des segments de droites, courbes ou régions, selon les besoins des applications envisagées. Donc l'asservissement visuel est basé sur le traitement d'image.

IV.2. Poursuite d'une trajectoire par vision :

La manipulation consiste à monter une caméra au dessus du robot mobile et tracer une trajectoire sur le sol avec du chatterton d'une couleur discernable (figure (4.1)). Notre choix s'est porté sur la webcam HDC270 de chez « logitech » qui présente des caractéristiques techniques suffisantes pour notre application. Elle présente des appels vidéo HD (1280x720 pixels) avec le système recommandé, une capture vidéo jusqu'à 1280x720 pixels, des photos jusqu'à 3mégapixels, un microphone intégré avec réduction des bruits, une interface USB 2.0 haut débit, un clip universel pour ordinateurs portables et écrans LCD ou CRT.

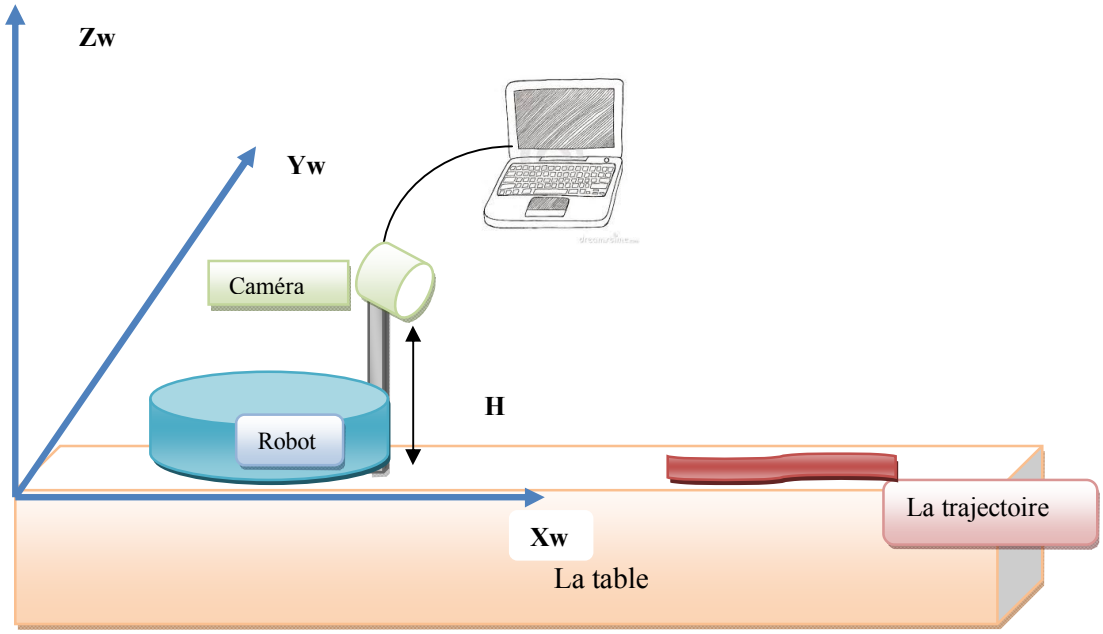


Figure (4.1) : Système de vision

IV.3. Problématique :

L'objectif de ce travail consiste à fournir au robot mobile les points échantillonnés d'une trajectoire prédéfinie captée par une caméra monoculaire montée au dessus de celui-ci. Ces points représentent l'abscisse et l'ordonnée de la courbe de la trajectoire par rapport au repère monde pour une bonne poursuite avec des erreurs très tolérables. La trajectoire est tracée sur un sol d'une façon arbitraire et très convenable pour une acquisition d'images réussie.

La phase initiale de l'algorithme développé consiste à faire une acquisition de l'image de l'environnement où évolue le robot mobile. L'image acquise contient entre autre la trajectoire à poursuivre. La deuxième phase consiste à traiter l'image acquise afin d'identifier la trajectoire tracée. Cette trajectoire représentée par une courbe dans le repère image est échantillonnée pour former le vecteur des points de référence à fournir au robot comme cibles à atteindre. Dans ce qui suit, nous allons détailler les différentes manipulations et phases de l'algorithme mise au point pour une bonne réussite de poursuite de trajectoire avec comme seul capteur : la caméra monoculaire. Le schéma synoptique de la figure (4.2) illustre d'une façon claire la commande visuelle en montrant les différents blocs constituant le système global.

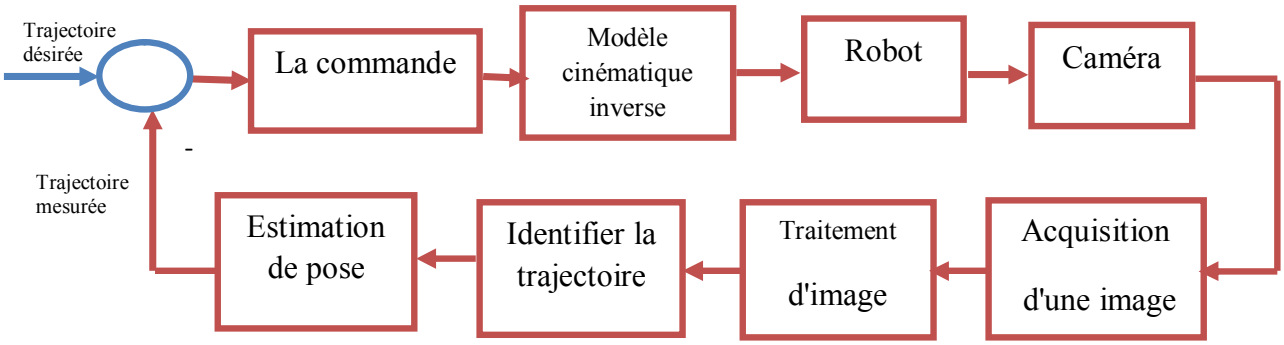


Figure (4.2) : Architecture de la commande visuelle

IV.4. Algorithme de poursuite d'une trajectoire

L'algorithme de poursuite d'une trajectoire par le robot mobile en utilisant la webcam :

- Acquisition de l'image par la webcam
- Identifier la courbe de la trajectoire
- Échantillonner la trajectoire
- Faire une transformation perspective inverse de l'image repère robot puis une transformation inverse dans le repère monde.
- Appliquer la commande cinématique développée dans le chapitre 2.

IV.4.1. L'acquisition et traitement d'image :

L'acquisition est la première étape pour le traitement et l'analyse de l'image. Pour un bon traitement, on doit s'assurer d'être entouré de meilleures conditions pour de bonne prise d'images. Le système d'acquisition est composé par :

- Une caméra couleur de type webcam ***logitech 720 HP.***
- Un support de caméra constitué d'une tige d'une longueur de 28 cm fixé sur un point en avant de la base mobile du robot.
- un ordinateur PC.

L'image acquise est représenté sous forme des pixels contenant un très grand nombre de données. Étant donné que l'image obtenue est contaminée par le bruit, il s'avère nécessaire de faire un prétraitement pour s'en débarrasser et améliorant ainsi sa qualité.

IV.4.2. Extraction des caractéristiques :

Avec l'étape de prétraitement, on obtient une image dotée de bonne qualité qui permet de faire une extraction des caractéristiques qui serviront à identifier la trajectoire à poursuivre.

IV.4.3.Algorithme de traitement d'image :

La segmentation de l'image est la base de tous les systèmes de vision pour l'extraction des caractéristiques ou d'une carte de l'environnement. Donc il faut traiter l'image pour détecter et caractériser les zones d'intérêts. La figure (4.3) présente l'algorithme de traitement d'image qui permet d'isoler la trajectoire désirée.

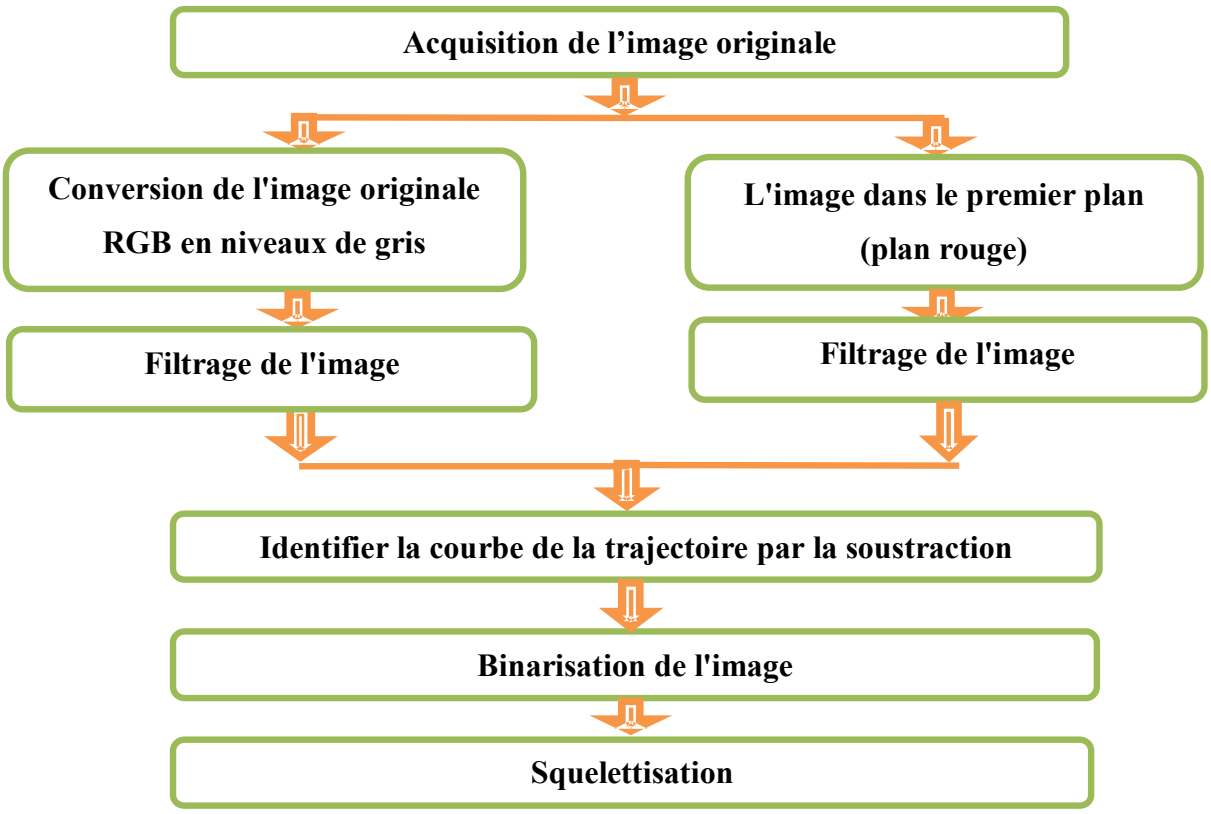


Figure (4.3) : Algorithme de traitement d'image

IV.4.3.1. Convertir l'image RGB en niveaux de gris :

Chaque pixel de l'image RGB peut être décomposé à un ensemble de trois valeurs de 8 bits, correspondant à la quantité (intensité) de rouge, vert et bleu (en anglais : *Red, Green, Blue)*. Chaque composante de couleur de l'image est désigné par un plan image ou d'un canal. Les écrans 16 millions de couleurs sont des écrans RVB à 8 bits, soit 256 couleurs par teinte de base La réduction de chaque pixel a une valeur unique de l'ensemble original de trois

valeurs rétrécit le nombre d'opérations par les deux tiers et simplifie encore le traitement de l'image. Ce processus est connu comme la conversion en niveaux de gris, aussi la résultante des pixels de l'image sera stockée sous un numéro unique de 8 bits, souvent représenté par une valeur de gris entre le noir et blanc.

Les images en niveaux de gris sont très utilisables, parce que les images en niveaux de gris sont tout à fait suffisantes pour de nombreuses tâches et il n'y a donc pas besoin d'utiliser des images couleurs plus complexes et plus difficiles.

Il existe trois méthodes pour convertir une image RVB en niveaux de gris. Un pixel gris a ses trois valeurs RVB identiques. Une méthode simple pour convertir une image couleur en niveau de gris peut être calculé par la moyenne des trois composantes RVB et d'utiliser cette valeur moyenne pour chacune des composantes:

$$\text{Gris} = \frac{R+V+B}{3} \tag{4.1}$$

Comme existe d'autre formule :

$$\text{Gris} = 0.299*R + 0.587*V + 0.114*B \tag{4.2}$$

$$\text{Gris} = 0.212*R + 0.715*G + 0.072*B \tag{4.3}$$

A noter que la formule utilisée par ***Matlab*** est celle donnée par l'expression (4.2).

IV.4.3.2. Filtrage de l'image :

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites). Le bruit est un ensemble de pixels qui ne font pas partie de la scène à traiter. Il est injecté par plusieurs sources et pour s'en débarrasser on a besoin de faire un filtrage avant tout traitement de l'image.

Plusieurs méthodes de filtrage ont été développées dans la littérature suivant le type et l'intensité du bruit présent dans l'image. Les plus simples des méthodes sont basées sur le filtrage linéaire stationnaire qui est au fait invariant par translation comme le filtre moyenneur, filtre smooth. Mais les limitations de ces techniques (en particulier leur mauvaise conservation des transitions) a conduit au développement des filtres "non-linéaire" pour pallier aux insuffisances des filtres linéaires : principalement la mauvaise conservation des contours [filtres]. Ils ont le défaut d'infliger des déformations irréversibles à l'image. Les filtres non linéaires sont basés sur des bases mathématiques ou empiriques différentes comme : le filtre nagao, FAS ou le filtre médian que nous avons choisi pour notre application.

Filtre médian

Le filtre médian dit également filtre spatial à fenêtre glissante. Il est particulièrement efficace contre le bruit dans des images à niveaux de gris, il remplace les valeurs centrales de chaque fenêtre par la valeur médiane de toutes les valeurs de pixels dans la fenêtre. Ainsi, si plusieurs pixels voisins sont bruités, on peut corriger le pixel courant. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. Par contre, ce filtre est couteux en terme de calculs car il nécessite d'effectuer un tri des voisins pour chaque pixel. Le noyau est généralement carré, mais peut être de n'importe quelle forme. Comme illustration, on prend l'exemple suivant d'un filtre médian d'une fenêtre d'analyse 3x3.

5	**3**	**6**
2	**11**	**9**
8	**4**	**7**

Valeurs non filtrés

Les niveaux de gris sont classés par ordre croissant dans l'ordre croissant :
2, 3, 4, 5, **6,** 7, 8, 9, 11
La valeur médiane **6** est choisie

*	*	*
*	**6**	*
*	*	*

Filtre médian

La valeur de centre (précédemment 11) est remplacé par la médiane de l'ensemble des neuf valeurs : **6.**

IV.4.3.3. Identifier la courbe de la trajectoire :

Cette étape requise pour arriver à détecter la trajectoire désirée consiste à rechercher les couleurs des objets dans l'image. Elle permet d'identifier des objets qui se trouvent dans l'environnement à partir de leur couleur et de supprimer les éléments extérieurs qui ne sont pas importants. La technique utilisée pour isoler la trajectoire est l'opérateur de soustraction.

L'opérateur de soustraction prend deux images en entrée et produit en sortie une troisième image dont les valeurs des pixels sont simplement les valeurs des pixels de la première image

moins ceux correspondants à la deuxième image. Il est également souvent possible d'utiliser une seule image comme entrée et soustraire une valeur constante à tous les pixels. Par ailleurs, on peut trouver certaines versions où l'opérateur génère une sortie résultante de la différence entre les valeurs absolues des pixels plutôt qu'une sortie signée.

- La soustraction des deux images est effectuée sans détour en un seul passage. Les valeurs des pixels de sortie sont données par :

$$Q(i,j) = P_1(i,j) - P_2(i,j) \qquad (4.4)$$

- Ou si l'opérateur calcule les différences absolues entre les deux images d'entrée :

$$Q(i,j) = |P_1(i,j) - P_2(i,j)| \qquad (4.5)$$

- Ou si elle est simplement souhaitable de soustraire une valeur constante C à partir d'une seule image :

$$Q(i,j) = P_1(i,j) - C \qquad (4.6)$$

Si les valeurs des pixels dans les images d'entrées sont des vecteurs plutôt que des valeurs scalaires (par exemple les images en couleurs), alors les composantes individuelles (par exemple les composantes rouge, vert et bleu) sont simplement soustraites séparément pour produire la valeur de sortie.

Lorsque les valeurs des pixels de la sortie sont négatives, on peut avoir plusieurs implémentations de l'opérateur. Certains peuvent fonctionner avec des formats d'image qui soutiennent les pixels de valeurs négatives. La façon avec laquelle ils sont affichés est déterminée par la carte des couleurs de l'écran "*colormap*". Si le format de l'image ne supporte pas les nombres négatifs alors souvent ces pixels sont mis à zéro (c'est-à-dire typiquement noir). Sinon, l'opérateur peut traiter les valeurs négatives, de sorte que par exemple (-30) apparaît à la sortie 226 (en supposant que les valeurs sont codés en 8 bits).

Dans notre cas les deux images d'entrée sont l'image dans le premier plan (le plan rouge) et l'image originale en niveau de gris, le format de l'image ne supporte pas les nombres négatifs donc si la différence entre les pixels est inférieure à 0 ces valeurs sont remplacées par des valeurs en nivaux de gris codé en 8 bits.

IV.4.3.4. Binarisation de l'image :

Les images binaires sont des images dont les pixels ne peuvent avoir que deux valeurs d'intensité possibles. Ils sont normalement affichés en noir et blanc. Numériquement, les deux valeurs sont souvent 0 pour le noir, et 1 ou 255 pour le blanc.

Le principe de « binariser » une image est de comparer chaque pixel de l'image à un seuil et affecter la valeur noire à ce pixel si la valeur d'origine est inférieure au seuil ou bien la valeur blanche dans le cas contraire. La dynamique de l'image est alors réduite à deux luminosités.

IV.4.3.5. Squelettisation :

La squelettisation est une transformation largement utilisée dans les champs d'analyse des images et de reconnaissance de forme. La squelettisation dans le plan désigne un processus qui transforme un objet 2D en une représentation d'une ligne 2D. Le concept de squelette a été introduit par Blum en 1961, sous le nom de "la transformation d'axe médian".

L'idée de la squelettisation est de réduire un objet à une représentation de moindre dimension qui capte les principales caractéristiques de l'objet original. Les squelettes peuvent être utilisés pour visualiser des aspects importants d'un objet.

Le squelette d'une forme est défini comme l'ensemble des points dont la distance au point le plus proche du contour est localement maximum [David *et al*]. Ainsi, le squelette représente la ligne centrée dans la forme. Par exemple, le squelette d'un cercle est le centre du cercle et le squelette d'un carré est donné par ses diagonales.

Différentes méthodes peuvent être utilisées pour définir un squelette. Nous allons utiliser la squelettisation basée sur l'amincissement. Elle est donnée par une méthode bien connue qui s'appelle la morphologie mathématique. C'est une discipline basée sur la théorie des ensembles.

Exemple de squelettisation :

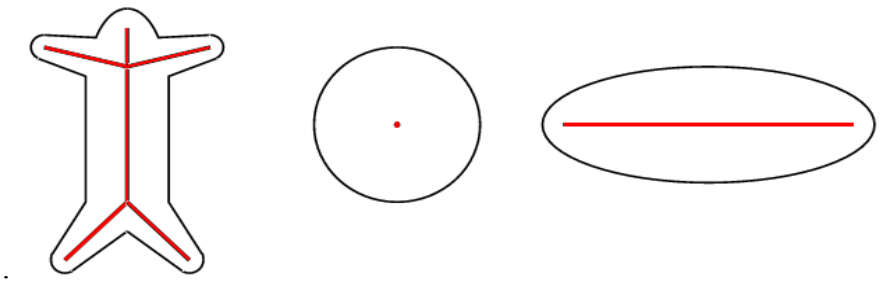


Figure (4.4) : Exemple d'une squelettisation

L'opérateur morphomathématique :

La *morphologie mathématique* est une théorie de traitement *non linéaire* de l'information apparue en France dans les années 60 [G. Matheron & J. Serra, Ecole des Mines de Paris], et qui est aujourd'hui très largement utilisée en *analyse d'images*. Contrairement au traitement linéaire des images, la morphologie mathématique ne s'appuie pas sur le traitement du signal, mais repose sur la *théorie des ensembles*, ce qui en fait une discipline relativement « auto-contenue » et formant un tout cohérent [Morph].

La Morphologie mathématique est un outil utile pour la segmentation et le traitement d'image. Initialement appliqués sur les images en noir et blanc par Matheron et Serra en 1965, elle a été ensuite étendue à des images en niveaux de gris par Dougherty en 1978. Pour l'appliquer à des images en couleurs, il suffit alors de traiter chaque couleur séparément.

La morphologie mathématique est un outil pour extraire les composants utiles dans la représentation et la description de la région de forme image, tels que les frontières, les squelettes etc. La langue de la morphologie mathématique est la théorie des ensembles. Elle utilise un ensemble de centre x, de géométrie et de taille connues, appelé **élément structurant**, c'est à dire des configurations élémentaires de pixels à rechercher dans l'image.

En raison de la complexité de la morphologie à l'échelle de couleur, l'image couleur est transformée en une image binaire par la segmentation. Les opérateurs dans le domaine de la morphologie binaire sont la dilatation, l'érosion et deux opérations complémentaires : la fermeture et l'ouverture sont utilisées pour le prétraitement des données.

Éléments structurant :

Un élément structurant est tout simplement est une image binaire ou un masque binaire (constitué de pixels blancs et noirs) qui nous permet de définir arbitrairement des structures de voisinage que l'on souhaite.

Squelettisation basée sur l'amincissement :

L'amincissement ou « *thinning* » est une méthode qui consiste à éplucher la forme jusqu'à obtenir un ensemble de points connectés d'un pixel qui conserve la topologie de la forme : le squelette [Aurore13]. Donc elle consiste à réduire une forme en un ensemble de courbes centrées dans la forme d'origine. L'importance de la squelettisation vient du fait que dans le processus de perception visuelle.

L'application de l'amincissement a été introduit par Harry Blum dans les années soixante, de but de créer un nouveau descripteur de formes. La squelettisation sert à la reconnaissance de formes, la modélisation des solides pour la conception et la manipulation de formes, l'organisation de nuages de points, la planification de trajectoire, la détection de réseaux routiers, les animations, etc.

Plusieurs algorithmes d'amincissement sont proposés dans la littérature, L'algorithme d'amincissement que nous avons adopté est de Lam, Seong-Whan Lee, et Ching Y. Suen [Lam *et al* 92] qui proposent une implémentation morphologique de leur algorithme d'amincissement

IV.5. Résultats de simulations de traitement d'image :

Figure (4.5) :l'image originale

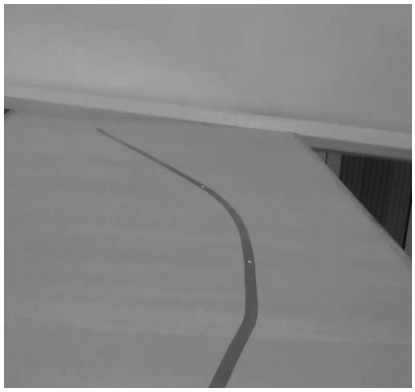
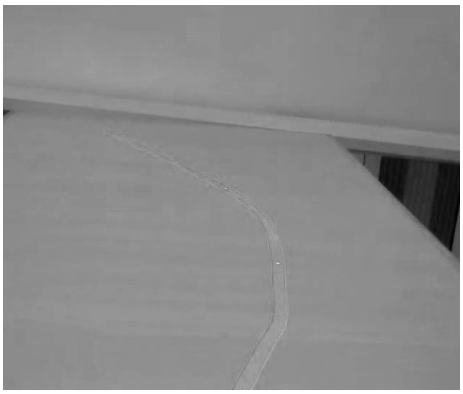


Figure (4.6) : l'image en nivaux de gris



Figue (4.7) : Le plan rouge de l'image

Figure (4.8) Identification de la trajectoire

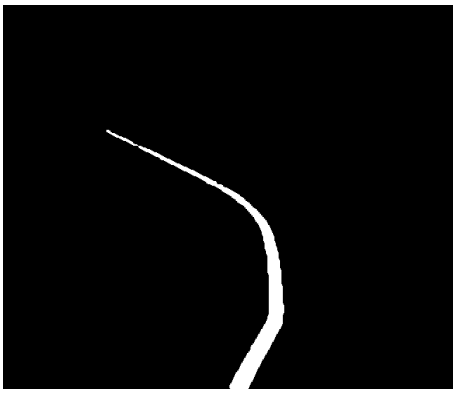


Figure (4.9) : l'image après filtrage et binarisation

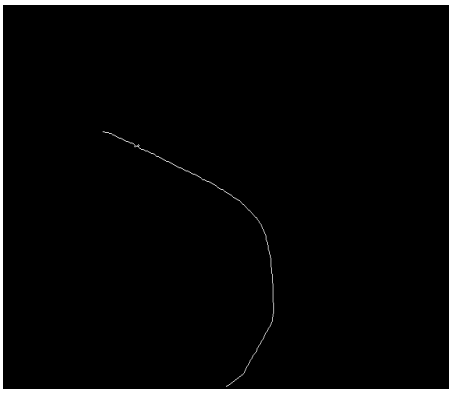


Figure (4.10) : la squelettisation de la trajectoire

D'après les résultats obtenus, on peut déduire que l'algorithme de traitement d'image proposé est capable d'isoler et d'identifier la courbe de la trajectoire désirée. L'idée de la squelettisation permet d'éliminer les " *outliers* " permettant d'obtenir les coordonnées des points de la trajectoire les plus favorables.

IV.6. La transformation perspective et inverse d'un point :

L'algorithme de squelettisation que nous avons appliqué permet de mincir la courbe de la trajectoire que le robot mobile est supposé poursuivre. Ceci nous amène à pouvoir échantillonner la trajectoire en recueillant les points adéquats et connaître ainsi leurs coordonnées dans le système de coordonnées image. La première transformation est la transformation perspective inverse qui transforme un point dans le système de coordonnées image à un point dans le système de coordonnées robot (RCS). La deuxième permet de transformer un point dans le système de coordonnées robot à un point dans le système de coordonnées monde (WCS). [Hartley *et al*03] décrit les processus de transformation perspective directe et inverse et qui sont à la fois modélisés dans [Duda *et al*73].

IV.6.1. La transformation perspective d'un point :

La transformation perspective est la méthode de la cartographie des points en trois dimensions sur un plan bidimensionnel appelé le plan de projection. La transformation perspective directe est l'approximation de premier ordre dans le processus de la prise de vue. La ligne qui relie un point du monde avec la lentille coupe le plan d'image et définit son point

unique de l'image. La transformation perspective inverse spécifie la ligne droite sur laquelle le point dans le repère monde correspond à un certain point de l'image.

La première transformation est la transformation perspective inverse. Cette étape consiste à récupérer les cordonnées de la trajectoire dans le système de coordonnées robot (RCS) en utilisant le principe de la vue d'œil d'oiseau. Pour mener cette étude, la surface de la trajectoire est plane ; le robot possède trois dimensions et le système de coordonnées est défini comme l'indique la figure (4.11). La position du point **P** de la trajectoire est exprimée par les coordonnées (Xr, Yr, Zr) dans le système de coordonnées de robot (RCS). Dans ce cas, les (x_I, y_I) sont les coordonnées du point **P** correspondant dans le plan d'image.

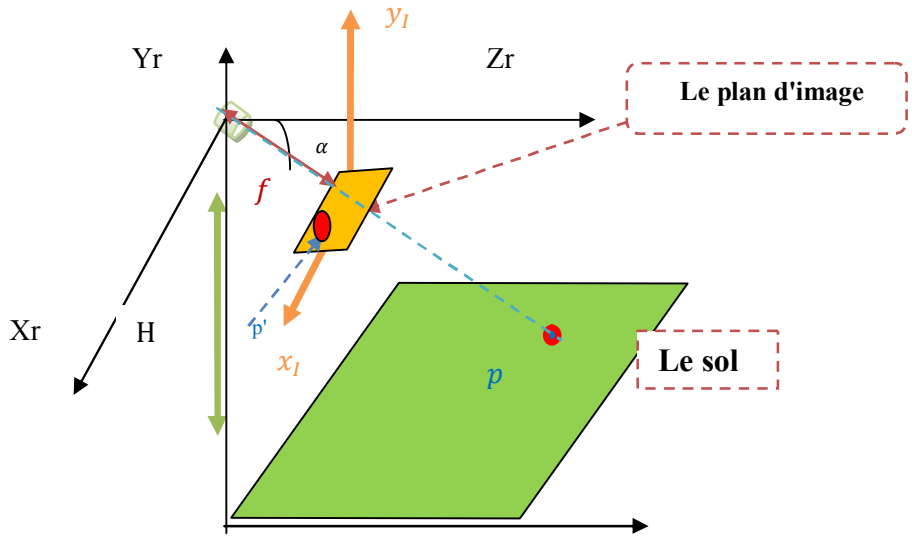


Figure (4.11) : Relation entre RCS et ICS

Dans l'analyse qui suit, la webcam est montée sur le robot. Les coordonnées du système robot RCS et les coordonnées du système de projection de l'image ICS sont fixé sur la caméra comme illustre la figure (4.12).

Le point du centre de la lentille de la webcam est à l'origine dans le système de coordonnées du robot RCS, et l'axe Zr est l'axe d'orientation du robot. L'angle α représente l'angle d'inclinaison de la caméra par rapport à l'axe Xr. (x, y) la position du robot dans le repère monde.

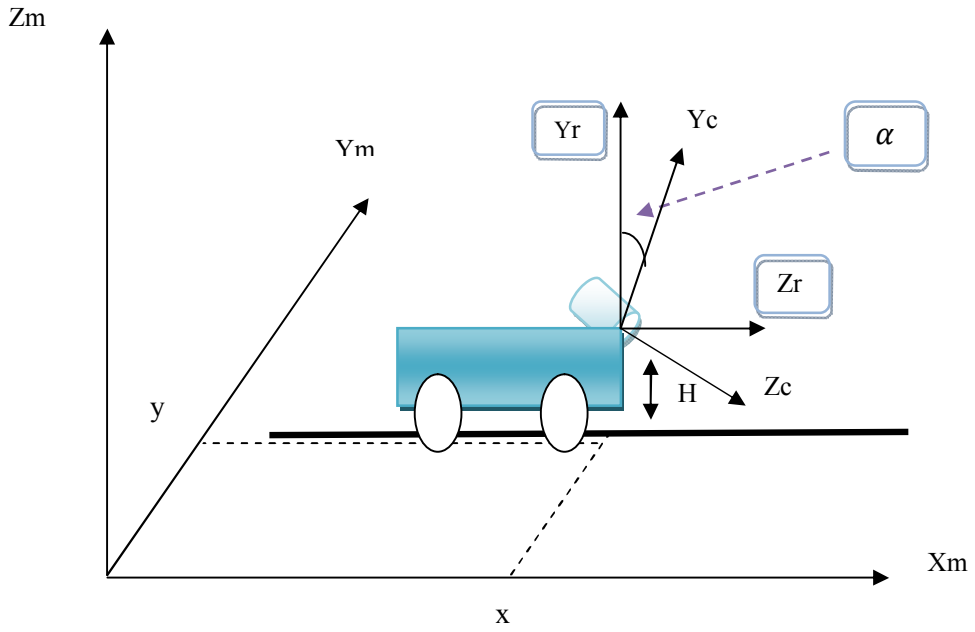


Figure (4.12) : Angle d'inclination de la Camera

La relation perspective entre RCS et le système des coordonnées image (ICS) est représenté par la figure (4.11) ; Le point p_c (x_c, y_c, z_c) d'un point statique p (X, Y, Z) dans Le repère caméra et le point de perspective p' (x_I, y_I) représente le point p (X, Y, Z) dans ICS.
La projection perspective transforme le point p_c (x_c, y_c, z_c) dans le plan image p' (x_I, y_I) :

$$x_I = f*\frac{Xc}{zc} \quad ; \quad y_I = f*\frac{yc}{zc} \qquad (4.7)$$

α est l'angle d'inclinaison de la caméra par rapport à l'axe X_r. Donc, la relation entre le repère caméra et repère robot est :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & sin\alpha \\ 0 & -sin\alpha & cos\alpha \end{bmatrix} * \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix}$$

$$\begin{cases} x_c = X_r & (4.8) \\ y_c = cos\alpha * Y_r + sin\alpha * Z_r & (4.9) \\ z_c = -sin\alpha * Y_r + cos\alpha * Z_r & (4.10) \end{cases}$$

D'après les équations (4.8), (4.9), (4.10) les relations dans (4.7) deviennent :

$$\boldsymbol{x_I = \frac{f * X_r}{Z_r * cos\alpha - Y_r * sin\alpha}} \quad (4.11)$$

$$\boldsymbol{y_I = \frac{f * (Y_r * cos\alpha + Z_r * sin\alpha)}{Z_r * cos\alpha - Y_r * sin\alpha}} \quad (4.12)$$

Avec :

f : la distance focale et α l'angle d'inclinaison.

(x_I, y_I) : Les coordonnées 2-D d'un point dans le plan d'image.

(x_c, y_c, z_c) : Les coordonnées 3-D d'un point dans le repère caméra.

(X_r, Y_r, Z_r) : Les coordonnées 3-D d'un point dans le repère robot

IV.6.2.Transformation perspective inverse de l'image à l'espace 3D RCS :

La trajectoire à vue d'œil d'oiseau est obtenue par la transformation perspective inverse des données d'image aux coordonnées d'espace 3-D réel en utilisant les relations perspectives entre les systèmes de coordonnées. Si la hauteur de la caméra par rapport à la base du robot mobile est connue, alors les points cibles de la trajectoire peuvent être calculés comme suit :

D'après la relation (4.12) :

$$y_I = \frac{f * (Z_r * sin\alpha + Y_r * cos\alpha)}{Z_r * cos\alpha - Y_r * sin\alpha}$$

$y_I * (Z_r * cos\alpha - Y_r * sin\alpha) = f * (Z_r * sin\alpha + Y_r * cos\alpha)$

$Z_r * (y_I * cos\alpha - f * sin\alpha) = Y_r * (y_I * sin\alpha + f * cos\alpha)$

$$Z_r = Y_r * \frac{y_I * sin\alpha + f * cos\alpha}{y_I * cos\alpha - f * sin\alpha}$$

La relation (4.11) devient :

$$x_I = \frac{f * X_r}{-sin\alpha * \mathrm{Y_r} + cos\alpha * \mathrm{Z_r}}$$

$$X_r = \frac{x_I * (Z_r * cos\alpha - Y_r * sin\alpha)}{f}$$

$$X_r = \frac{x_I * (Y_r * (\frac{y_I * sin\alpha + f * cos\alpha}{y_I * cos\alpha - f * sin\alpha}) * cos\alpha - Y_r * sin\alpha)}{f}$$

$$X_r = Y_r * \frac{x_I}{y_I * cos\alpha - f * sin\alpha}$$

$$\boldsymbol{X_r = Y_r * \frac{x_I}{y_I * cos\alpha - f * sin\alpha}}$$

$$\boldsymbol{Y_r = -H}$$

$$\boldsymbol{Z_r = Y_r * \frac{y_I * sin\alpha + f * cos\alpha}{y_I * cos\alpha - f * sin\alpha}}$$

tels que :

H : représente l'élévation de la caméra au-dessus de la surface de la trajectoire.

IV.6.3. La transformation au repère monde :

On détermine les coordonnées cibles (X_d, Y_d, Z_d) de la trajectoire dans le repère monde (WCS) par les équations de changement des repères comme suit :

$$p_d = \mathbf{T_b^r} * p_r \qquad (4.13)$$

$\mathbf{T_b^r}$**:** C'est la transformation des points du repère robot au repère monde.

$$\mathbf{T_r^b} = \begin{bmatrix} R_z\left(\frac{\pi}{2}\right) * R_x\left(-\frac{\pi}{2}\right) * R_y\left(-\frac{\pi}{2}\right) & p_r^b \\ 0 & 1 \end{bmatrix}$$

p_r **:** les points de la trajectoire dans le repère robot.

p_d **:** les points désires de la trajectoire dans le repère monde.

p_r^b : La position du robot par rapport au repère monde.

$R_z(\theta)$ $R_x(\theta)R_y(\theta)$: Les matrices de rotation entre le repère monde et le repère robot selon les axes z, x et y.

$$\mathbf{T_b^r} = (\mathbf{T_r^b})^{-1}$$

La position de la trajectoire désiré dans le repère monde est obtenue par la multiplication des (X_r, Y_r, Z_r) par$(\mathbf{T}_\mathbf{r}^\mathbf{b})^{-\mathbf{1}}$. L'équation (4.13) devient :

$$\begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} = \mathbf{T}_\mathbf{b}^\mathbf{r} * \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} R_z\left(\frac{\pi}{2}\right) * R_x\left(-\frac{\pi}{2}\right) * R_y\left(-\frac{\pi}{2}\right) & p_r^b \\ 0 & 1 \end{bmatrix}^{-1} * \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix}$$

Avec : $[X_d\ Y_d\ Z_d]$: les coordonnées de la trajectoire désiré dans le repère monde.

Pour effectuer les tests de simulation, nous avons choisi :

- La position de robot par rapport au repère monde est (x, y, z) = (35cm,35cm,0).
- L'angle d'inclinaison α égal à 20°.

IV.7. Résultats de simulations :

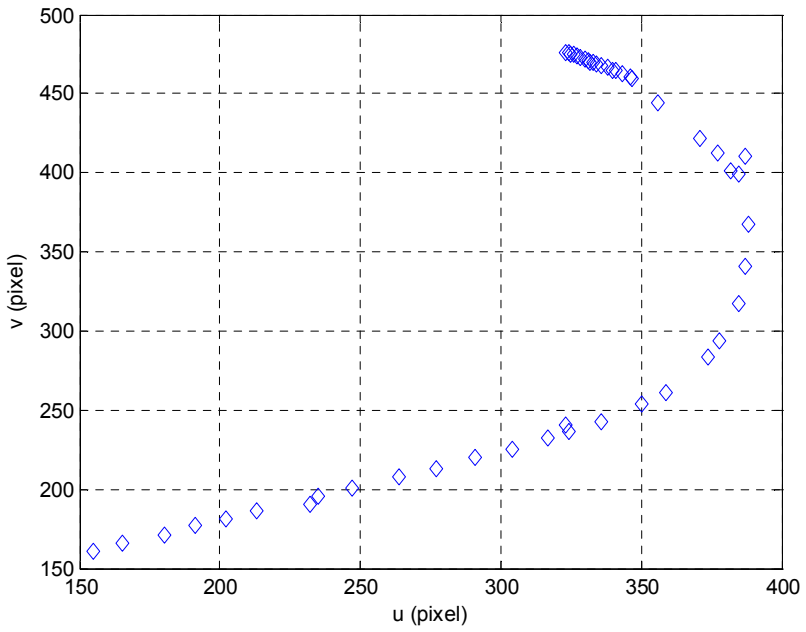


Figure (4.13) : Les points échantillonnés de la trajectoire

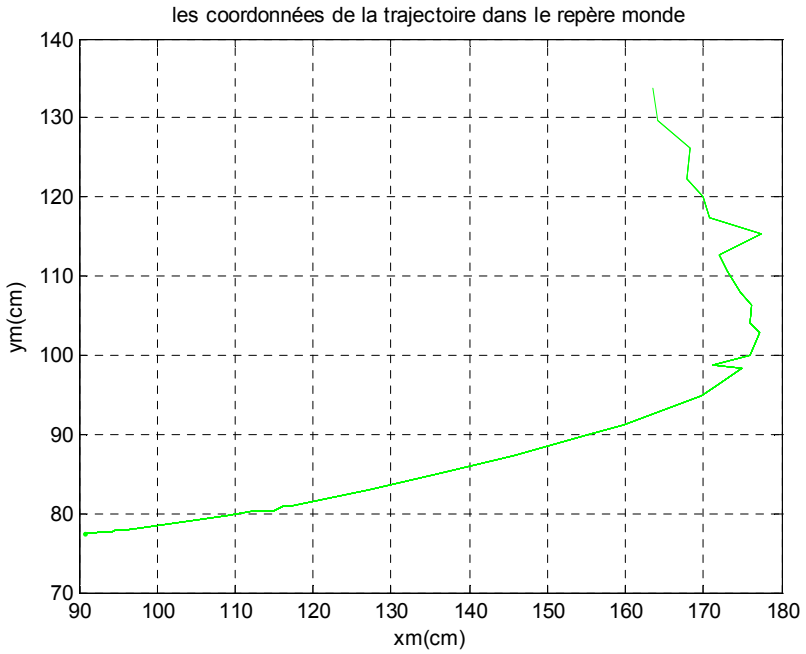


Figure (4.14) : Les coordonnées de la trajectoire dans le repère monde

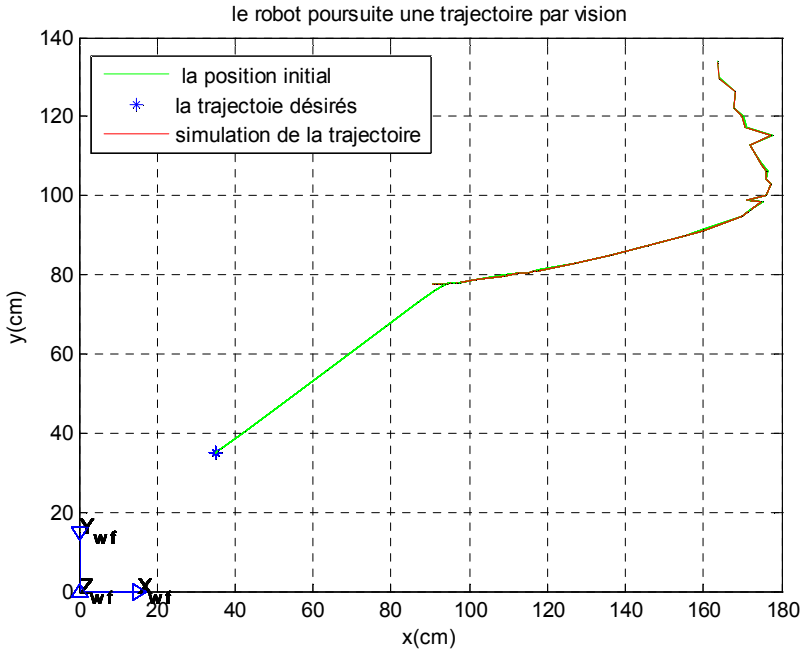


Figure (4.15) : poursuite de la trajectoire par vision

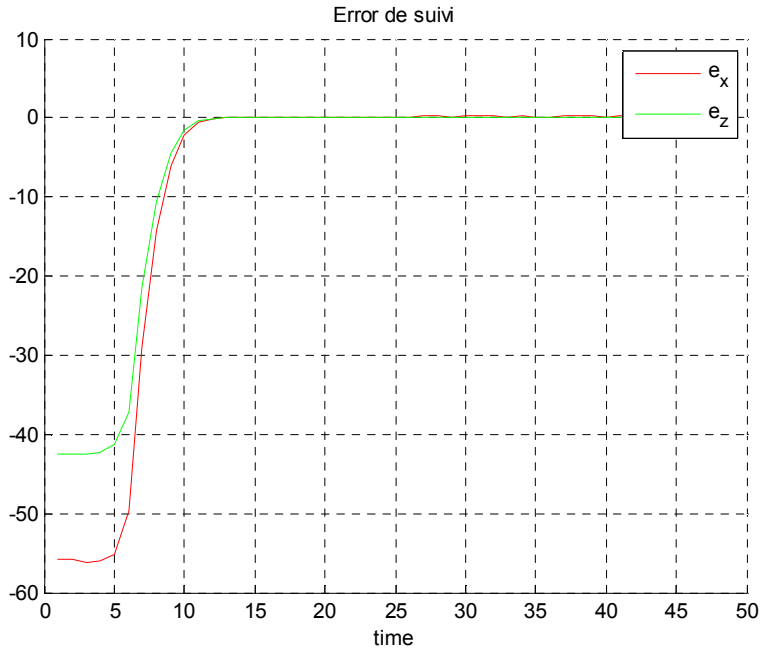


Figure (4.16) : erreur de poursuite

D'après les figures de simulation du traitement d'image (4.9) et (4.10), on remarque que l'algorithme proposé pour isoler la trajectoire, a donné de bons résultats, et l'idée de la squelettisation nous a permis de choisir d'une façon convenable, unique et non ambigüe les coordonnées de référence.

La figure (4.13) illustre les points échantillonnés de la trajectoire choisi à partir à la figure de squelettisation ; La figure (4.14) montre les transformations de la trajectoire (RCS) en système des coordonnées monde, toutes ces coordonnées sont en centimètre (cm). La figure (4.15) illustre les résultats de la simulation de la trajectoire suivie par le robot omnidirectionnel. La dernière figure, (figure (4.16)) présente l'erreur de poursuite de la trajectoire, montre que l'erreur a tendance à s'annuler après un certain temps.

IV.8. Conclusion :

Dans ce chapitre nous avons utilisé un algorithme de suivi de trajectoire par un robot mobile omnidirectionnel en utilisant la vision par ordinateur. L'algorithme proposé est principalement divisé en deux parties. La première concerne l'acquisition puis le traitement d'image adéquat qui permet une segmentation de l'image et l'obtention de la courbe de la

trajectoire à poursuivre. La seconde partie, offre au robot mobile les points de référence de la trajectoire désirée pour accomplir la tâche de poursuite. Tous les calculs permettant la représentation des points dans le repère particulier ont été développés en utilisant les transformations matricielles nécessaires à la boucle d'asservissement.

CONCLUSION GÉNÉRALE

Parmi les nombreux organes sensoriels développés par la nature et imités par la technologie : la vision. Celle-ci apporte l'information la plus riche et la plus complète sur le milieu entourant. Dans ce mémoire, nous avons tenté de présenter une méthode qui s'applique aux problèmes de la navigation visuelle d'un robot mobile. Pour ceci, nous avons proposé une méthode de navigation qui exploite uniquement les signaux émanant d'une caméra couleur, webcam, placée sur un robot mobile. Notre choix s'est porté sur le robot mobile omnidirectionnel "*Robotino*". L'objectif cherché est de concevoir une loi de commande exploitant uniquement les informations visuelles pour accomplir une tâche de poursuite de trajectoire dans une navigation « Indoor ».

Dans le premier chapitre, nous avons présenté des généralités sur les systèmes de vision en robotique, et un état de l'art sur les travaux de recherche effectués dans ce domaine pendant la dernière décennie.

Dans le deuxième chapitre nous avons présenté le model cinématique de "*Robotino*" et une méthode de commande cinématique. Les résultats de simulation attestant le succès de la méthode ont été étalés et commentés pour différents types de trajectoires de référence.

Dans le troisième chapitre nous avons présenté le calibrage et la modélisation de la caméra webcam utilisée à cet effet. Ce calibrage est nécessaire pour obtenir les paramètres intrinsèques et extrinsèques utiles à l'utilisation de la caméra.. La modélisation de la caméra est basée sur le modèle "sténopé" qui est le plus employé dans la vision par ordinateur. Il permet en effet de transformer un point 3D de l'espace à un point 2D dans le plan image. Ceci est possible grâce à trois transformations successives : la transformation perspective, la projection perspective et la transformation affine. Les résultats du calibrage accompagnés d'un test pour vérifier l'exactitude des coordonnées mesurées a aussi été illustrée.

Dans le dernier chapitre, nous avons proposé une méthode de poursuite d'une trajectoire par vision monoculaire. Les coordonnées de la trajectoire désirée sont obtenues par échantillonnage de la trajectoire désirée. Cette trajectoire est obtenue après avoir appliqué un traitement d'image adéquat qui passe d'abord par une acquisition de l'image, filtrage,

Segmentation et squelettisation. Pour qu'elles soient utiles à la commande cinématique choisie à cet effet, ces coordonnées sont transformées du système des coordonnées image au système des coordonnées monde les résultats de simulation sont présentés à la fin du chapitre.

Perspectives

Plusieurs amélioration et perspective peuvent s'envisager dans la continuité des travaux menés dans cette thèse. Comme futurs travaux nous envisageons de :

- développer le modèle dynamique du "*Robotino*".
- Proposer une loi de commande non-linéaire pour une poursuite de trajectoires obtenues à partir de séquences vidéo.
- développer un algorithme d'évitement d'obstacles.
- développer un algorithme pour Cartographie et Localisation Simultanées, connue en anglais sous le nom de **SLAM** (*Simultaneous Localization And Mapping*) ou **CML** (*Concurrent Mapping and Localization*) .

RÉFÉRENCES

[Aurore13] : A. Aurore, thèse de doctorat "Squelettisation en un balayage Application à la caractérisation osseuse ", version 1. 25 janvier 2013. tel-00781222.

[Baptiste12] : C. Baptiste, thèse de doctorat "Localisation temps-réel d'un robot par vision monoculaire et fusion multi-capteurs" , université "blaise pascal - Clermont II" , Soutenue publiquement le 14 décembre 2012.

[Barbosa04]: G. Barbosa, J.J, Thèse de doctorat " Vision panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images" , Université Toulouse III, 2004.

[Benosman01]: R. Benosman, S. B. Kang, Panoramic Vision: Sensors,Theory and Applications.NY: Spring Verlag, Monographs in computer Science 2001. ISBN 0—387-95111-3

[Benosman96] : R. Benosman, and T. Maniere, and J. Devars, Multidirectional stereovision sensor, calibration and scenes reconstruction, International Conference on Pattern Recognition (ICPR), pp. 161-165, 1996.

[Ching *et al*05]: C. Ching and, L. Jiang, T. Wang, T. Wang . Article "Kinematics Control of an Omnidirectional Mobile Robot".Deaprtment of Electrical Engineering, National Chung Hsing Universit 250,KuoKuang Road,Taichung 402, Taiwan,Processing of 2005 CACS Automatic control conference Tainan taiwan,Nov 18-19;2005.

[Courbon *et al*09]: J. Courbon,Y. Mezouar, &P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model". Intelligent Transport System (2009).

[Cyril02] : D. Cyril, thèse de doctorat : "Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels ", l'université de technologie de Compiègne, Soutenue le 22 février 2002.

[Courteille07]: F. Courteille, and J.D.Durou,P.and Gurdjos. Article "Shape From Contour for the Digitization of Curved Documents", Asian on Computer Vision (ACCV), pp. 196-205, 2007.

[Devernay97]: F. Devernay, Thèse de doctorat " Vision stéréoscopique et propriétés différentielles des surfaces", École polytechnique, 1997.

[David *et al*]: F. David, E. Mouaddib J. Salvi, " Segmentation et Décodage d'un Patron de Lumière Structurée ".CREA, Université de Picardie Jules Verne, Amiens, France, david.fofi@u-picardie.fr II, Université de Girona, Espagne, qsalvi@eia.udg.es.

[Dib11]: A. Dib ,thèse de doctorat "Commande non linéaire et asservissement visuel de robot mobile autonome", École Doctorale "science et technologie de l'information des télécommunication et des systèmes " soutenance le 21octobre 2011.

[Durand12]: P. Durand, thèse doctorat de l'université de Toulouse " Navigation référencée multi-capteurs d'un robot mobile en environnement encombré" ,soutenue 20 janvier 2012.

[Duda *et al*73] : R.Duda, and P. Hart. (1973). Pattern Classification and Scene Analysis. John Wiley and Sons Publisher. New York: John Wiley & Sons.

[Durou07]: J. D. Durou, and F. Courteille, Article "Integration of a Normal Field without Boundary Condition", International Conference on Computer Vision (ICCV), pp. 1-8, 2007.

[Eric *et al*05] : R. Eric, and L. Maxime, D. Michel ,L. Jean-Marc. Article " Localisation par vision monoculaire pour la navigation autonome ";LASMEA, UMR 6602 CNRS et université Blaise Pascal, 24 avenue des Landais, 63177 Aubière, France, Manuscrit reçu le 13 juillet 2005.

[Eric06]: R. Eric thèse de doctorat " Cartographie 3D et localisation par vision monoculaire pour la navigation autonome d'un robot mobile" , Soutenue publiquement le 26 Septembre 2006.

[Favaro05]: P. Favaro and S. Soatto "A Geometric Approach to Shape from Defocus" IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 406-417, March 2005.

[Forsyth02]: D.A.Forsyth, Shape From Texture without Boundaries, European Conference on Computer Vision (ECCV), pp. 225-239, 2002.

[festo] : www.**festo**-didactic.com/fr-fr

[filtre] : http: //www.tsi.telecomparistech.fr /pages/ enseignement/ressources/beti/ filtres_lin_nlin /filtres.html

[Jogan *et al*00] : M. Jogan. & Leonardis, A, (2000). Robust localization using panoramic view-based recognition. IEEE International Conférence on Computer Vision and Pattern Recognition, 4136.

[Juan05]: Juan. Gavina cervantes. Thèse de doctorat "Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-

structure", l'Institut National Polytechnique de Toulouse, 15 Février 2005.

[Horn75] : Horn, B.K.P., Obtaining Shape From Shading Information, The Psychology of Computer Vision, vol. 4, pp. 115-155, 1975.

[Hartley *et al*03] : R. Hartley. And A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, ISBN: 0521623049.2003.

[IYT90]: H. Ishiguro, M. Yamamoto, Tsuji. "Omnidirectionnal stereo for making global map". Proceedings on 3th Internationalal Conférence on Computer Vision, 1990.

[Guillaume10]: C. Guillaume, thèse de doctorat " Estimation de pose et asservissement de robot par vision omnidirectionnelle", l'Université de Picardie Jules Verne, Thèse dirigée par El Mustapha Mouaddib préparée à l'UPJV au sein du laboratoire Modélisation, Information et Systèmes, soutenue le 30 novembre 2010.

[Gibson50]: J. Gibson. "The Perception of the Visual World." Houghton Mifflin, Boston, 1950.

[Klette98]: R. Klette, K. Schluns, A. Koschan, Computer Vision: Three-dimensional Data from Images, Springer, Singapore (1998).

[Lam *et al*92] : L. Lam, Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column

[Lingrand04]: D. Lingrand " Cours de traitement d'image" , Projet de recherche ISRN I3S/RR-2004-05-05-FR . informatique, signaux et systèmes des Sophia Antipolis UMR 6070 . 22 JNVIER 2004.

[Miguel *et al*04]: A.Miguel and A. Solanas ,article " 3D Simultaneous Localization and Model ing from Stereo Vision" ; Proceedings of the 2004 IEEE ;International Conference on Robotics & Automation New Orleans, April 2004.

[Michel04]: K.Michel, thèse doctorat "Perception Multi sensorielle Pour le Positionnement d'un Véhicule Autonome dédié aux Personnes Handicapés Moteurs" ; l'Université de Metz; Soutenue publiquement le 21 septembre 2004.

[Mathias08]: M. Mathias perrollaz, Docteur de l'Université´e Pierre et Marie Curie , thèse " Détection d'obstacles multi-capteurs supervisée par stéréovision ", Soutenue publiquement le 13 Juin 2008.

[Morph]: http://sparis.free.fr/Cours_Image/AI6_1Operateurs_Morphologiques . ppt

[Nalwa96]: Nalwa,V. S. A True Omnidirectional Viewer, Rapport Technique, Bell Laboratories, Holmdel, NJ 07733, USA, 1996.

[Nayar97]: Nayar, S.K., Omnidirectional Video Camera, DARPA Image UnderstandingWorkshop (IUW), pp. 235-241, 1997.

[Oualid10]: D.Oualid , thèse de doctorat "Localisation Et Guidage Du Robot Mobile Atrv2 Dans Un Environnement Naturel" , universite des sciences et de la technologie houari boumediene , Soutenue publiquement le 15 décembre 2010.

[Prados06]: E. Prados, and O.Faugeras, Rôle clé de la modélisation en Shape From Shading, Reconnaissance des Formes et Intelligence Artificielle (RFIA), pp. ,2006

[Pollefeys04]: M. Pollefeys, and L. Gool, M. Vergauwen, F. Verbiest, K. Cornelis. and Tops, J., "Visual Modeling with a Hand-Held Camera", International Journal of Computer Vision (IJCV), vol. 59, pp. 207-232, 2004.

[Rémi10]: R. Boutteau. Thèse de doctorat "Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions" _ l'Université de ROUEN "; 19 avril 2010.

[Renaud08]: M. Renaud Barate, thèse de doctorat "Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique , docteur de l'université pierre et marie curie, soutenue le 26 Novembre 2008.

[Panoscan09]: Panoscan Inc., Panoscan Panoramic Camera for High Speed Digital Capture, http://www.panoscan.com/, 2009

[Saurav *et all*10]: K. Saurav.G. Daya et Y. Sakshi.Article "Sensor Fusion of Laser et Stereo Vision Camera for Depth Estimation and Obstacle Avoidance"; Delhi Technological University ©2010 International Journal of Computer Applications (0975 - 8887)Volume 1 – No. 26

[Sylvain 04]: M. Sylvain. Article "Système embarqué de localisation et de perception pour un robot mobile" , école polytechnique de Montréal - cours séminaire ele6904, avril 2004.

[Vincent 05]: L. Vincent, thèse de doctorat " Stéréovision Embarquée sur Véhicule : de l'Auto-Calibrage `a la Détection d'Obstacles." Institut National Des Sciences Appliquées de Toulouse soutenue le 10 novembre 2005.

[viviane11]: C. Viviane.Thèse de doctorat " Contribution à la navigation d'un robot mobile par commande référencée multi-capteurs ", l'Université Paul Sabatier de Toulouse, Soutenue le 2 Décembre 2011.

[Woodham80]: R.J. Woodham, Photometric Method for Determining Surface Orientation from Multiple Images, Optical Engineering, vol. 19, pp. 139-144, 1980.

[Yagi99]: Y. Yagi. Omnidirectional Sensing and Its Applications, IEICE Transactions on Information and Systems, vol. E82-D, pp. 568-579, 1999.

Annexe A

Les opérateurs Morphologiques

L'érosion et la dilatation sont deux opérateurs de base de la morphologie mathématique.

A.1. L'Érosion :

L'effet de l'opérateur de base l'érosion sur une image binaire consiste à éroder les limites des pixels de premier plan (généralement les pixels blancs). Ainsi, les zones de pixels de premier plan se rétrécissent en taille. Mathématiquement, l'érosion de l'ensemble A par l'ensemble B est un ensemble des points *x* tels que B traduit par *x* est encore contenue dans A (Figure. A.1).

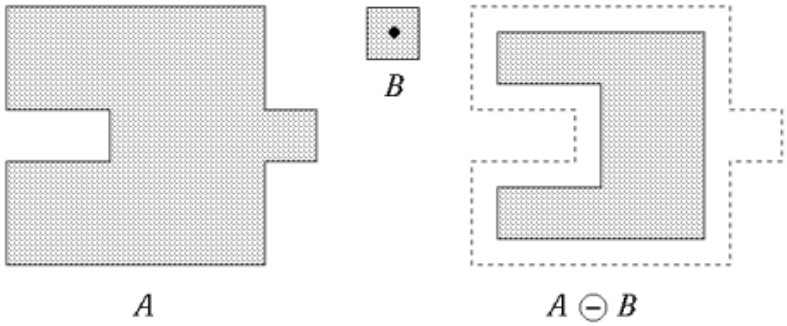


Figure (A.1) : Exemple d'érosion

A.2. La Dilatation :

La dilatation est l'autre de ces deux opérateurs de base de la morphologie mathématique. L'effet de l'opérateur de base la dilatation sur une image binaire pour agrandir les zones de pixels du premier plan (c'est-à-dire les pixels blancs) à leurs frontières. Les zones de pixels de premier plan se développent ainsi en taille.

Mathématiquement, l'érosion de A par B est l'ensemble de tous les déplacements de *x* de telle sorte que A et B se chevauchent d'au moins un élément non nul comme le montre la figure (A.2).

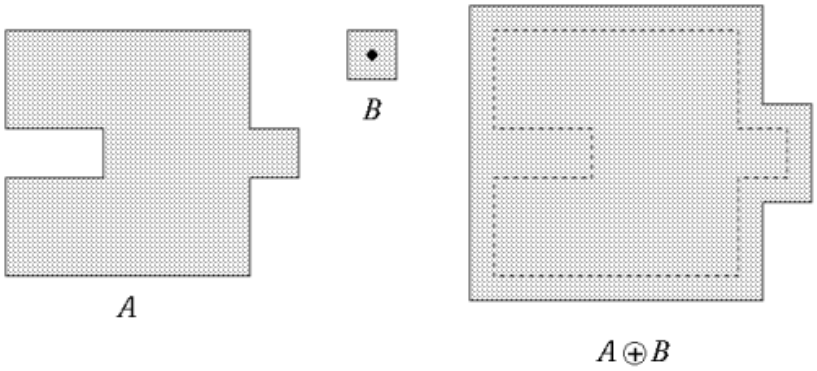


Figure (A.2) : Exemple de délitation

A.3. Ouverture et Fermeture :

L'ouverture est un opérateur composite, construit à partir de deux opérateurs de base décrits ci-dessus. L'ouverture de l'ensemble A par l'ensemble B est obtenue premièrement par l'érosion de A par B, puis la dilatation résultante par B (Figure. A.3).

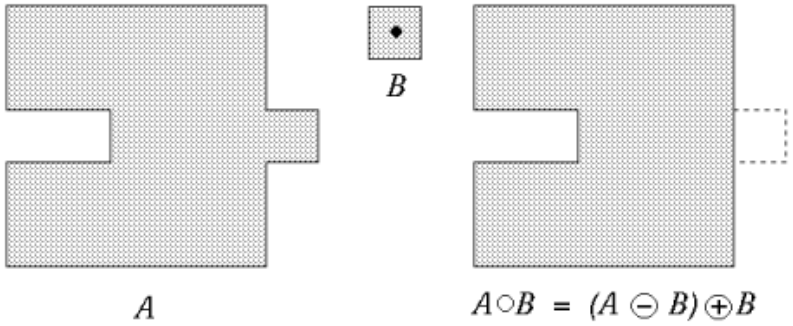


Figure (A.3) : Exemple d'ouverture

La fermeture, comme l'ouverture, est également un opérateur composite. La fermeture de l'ensemble A par l'ensemble B est obtenue premièrement par la dilatation de l'ensemble A par B, puis l'érosion résultant par B (Figure. A.4).

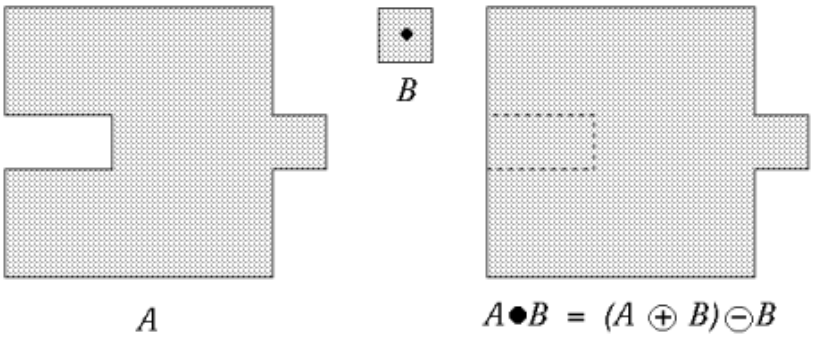


Figure (A.4) : Exemple de fermeture

A.4. Extraction limite

La limite de l'ensemble A peut être trouvée premièrement par l'érosion de A par B, puis en prenant l'ensemble de la différence entre l'original de A et l'érodé de A (Figure. A.5).

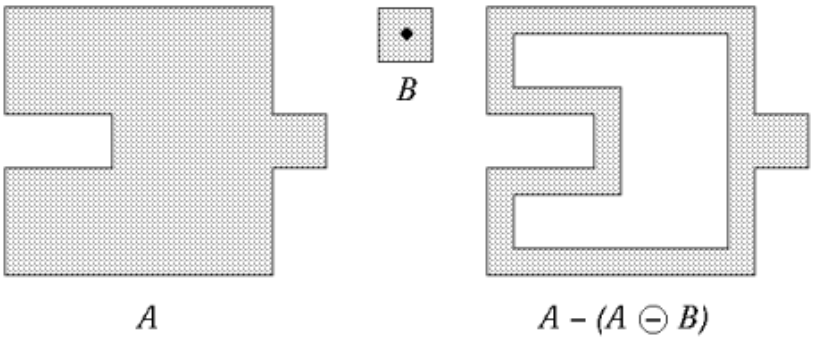


Figure (A.5) : Extraction