RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur

et de la Recherche Scientifique

École Nationale Polytechnique

Département d'Automatique

Mémoire de MASTER en AUTOMATIQUE

Thème :

Commande par logique floue appliquée sur une caméra a 2ddl pour le suivie d'un objet

Réalisé Par :
BOUSRI Abderraouf

Encadreur :
Mr.Omar STIHI.

Juin 2015

Ecole Nationale Polytechnique – Rue des frères Oudek, Hacen Badi, El Harrach, ALGER 16200.

العنوان: من نوع غامض في كاميرا تحتوي على اثنين من درجات الحرية لمتابعة تحركات جسم .

يهدف هذا العمل الى التحكم في كاميرا تحتوي على اثنين من درجات الحرية عن طريق انشاء تحكم من نوع غامض مما يسمح لها بمتابعة تحركات جسم يمثله مستوي على سطح صورة الكاميرا.

المفاتيح : السيطرة على كاميرا ,تحكم مرئي ,المنطق الغامض , متابعة تحركات جسم.

Titre : Commande par logique floue appliquée sur une caméra a 2ddl pour le suivie d'un objet.

Résumé :

Le but de ce travail est de munir une caméra a deux degrés de liberté d'une commande par logique floue, lui permettant de suivre les mouvements d'un objet détecté représenté par une surface sur le plan de l'image.

Mots Clés : commande d’une caméra, asservissement visuel, logique floue, suivie d’un objet.

Title : Control by fuzzy logic applied to a camera with two degrees of freedom to follow an object.

Abstract :

The purpose of this work is to equip a camera wich has two degrees of freedom with a fuzzy control, allowing it to follow the movements of a detected object

Key words : fuzzy logic, Image based control, control of a camera, visual servoing, fuzzy logic.

Remerciements

Au terme de ce travail,

Nous adressons nos remerciements à l'ensemble des enseignants de l'École Nationale Polytechnique spécialement ceux du département du Génie Automatique, pour leur encadrement, appuie scientifique, disponibilité ainsi que pour tout le savoir qu'ils nous ont transmis durant tout au long de notre formation.

Nous remercions aussi Monsieur El-Madjid BERKOUK, enseignant à l'École Nationale Polytechnique, d'avoir accepté d'examiner et évaluer ce travail.

Nous tenons également à remercier Monsieur H.CHKIREB, enseignant à l'École Nationale Polytechnique, de nous avoir fait l'honneur de présider ce jury.

Nous remercions également tous ceux qui nous ont soutenus et ont contribués de loin ou de près à la réalisation de ce travail.

A Maman.
A ma famille.
A Selman.

Abderraouf Bousri.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Depuis la nuit des temps l'homme a toujours essayé de réduire ses efforts en exploitant autrui, les circonstances et les faits historiques ainsi que sa curiosité lui ont permis d'inventer, de développer et ainsi exploiter des machines.

Le but de ce travail est la synthèse d'une commande pour permettre a une caméra a 2ddl (commandable en tangage et en lacet) de suivre les mouvements d'un objet détecté, pour cela nous nous sommes orientés à la commande par logique floue car elle nous permet d'implémenter le raisonnement et ainsi les bons réflexes qu'un être humain aurait eu lors d'une situation déterminée.

Nous commencerons par présenter l'objectif de ce travail ainsi que les outils utilisés. nous présenterons par la suite une méthode de détection d'objet de forme quelconque avec une caméra pour pouvoir ensuite réaliser un asservissement visuelle 2D avec une commande par logique floue appliqué pour faire orienter cette caméra.

Chapitre 1

Détection d'un objet via une caméra USB

l'objectif de ce travail est de pouvoir détecter un objet quelconque via une caméra USB, cet objet représente une surface dans le plan de l'image, seulement pour pouvoir le détecter nous nous somme orientés vers la vers une bibliothèque *VISP* [3] qui présente des fonctions prédéfinit qui peuvent être utilisées pour la détection du contour (bord) d'un objet détecté par cette camera.

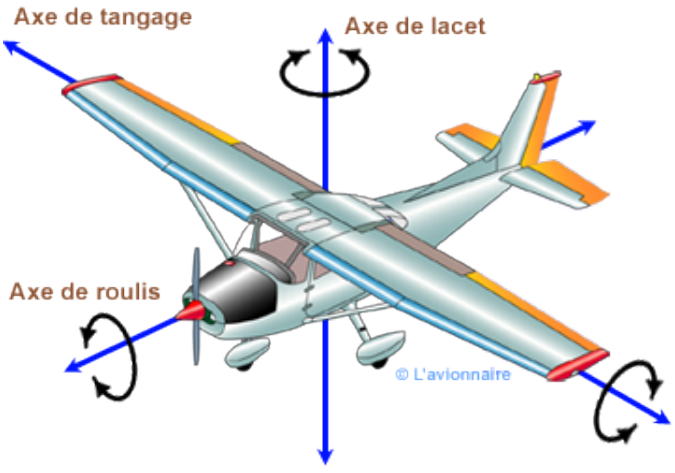


FIGURE 1.1: Notation des différentes rotations possibles.

cette caméra est une caméra a deux degrés de liberté, c'est a dire qu'elle peut effectuer seulement deux types de mouvement représentés par les deux rotations par rapport a l'axe transversale et verticale (Tengage et Lacet -fig-1.1), la figure suivante représente les différentes types de caméras qui sont capable de faire ce type de mouvement.

Types de caméra qu'on peut leur appliquer ce type de commande.

La première tache du programme est de convertir une image en couleurs en niveau de gris, sachant que dans une image numérique, le niveau de gris représente la luminosité d'un pixel, lorsque les valeurs de ses composantes de couleur sont identiques. Pour convertir une image couleur en niveau de gris il faut remplacer, pour chaque pixel les trois valeurs représentant les niveaux de rouge, de vert et de bleu, en une seule valeur représentant la luminosité.

Après l'affichage de l'image en niveau de gris nous sélectionnerons avec le curseur un premier point présent sur la surface qui représente la projection de l'objet dans le plan de l'image.

Remarque : l'objet ou bien la surface détectée doit avoir un niveau de gris quasi-constant (ce niveau de gris doit appartenir a interval définit par Niveau de gris Maximum et minimum

Aprés avoir sélectionné une surface, le programme de detection colore le contour de la surface détectée, ce qui permet de suivre le mouvement de cette dernier en temps réelle.

avec l'aide des fonctions définit dans cette bibliotheque nous pouvons récuperer les coordonées du centre de gravité de cette surface dans le plan de l'image en pixels.

Il faut noter que dans le plan de l'image les coordonnées sont en pixel, ainsi que les axes sont orientée comme le montre la figure 1.2.

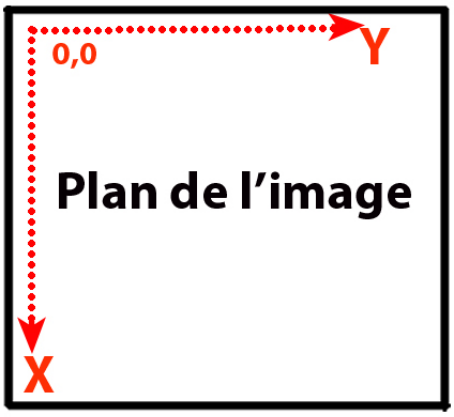


FIGURE 1.2: orientation des axes dans le plan de l'image.

la figure suivante (fig 1.3) montre la position d'un objet détecté dans le plan de l'image.

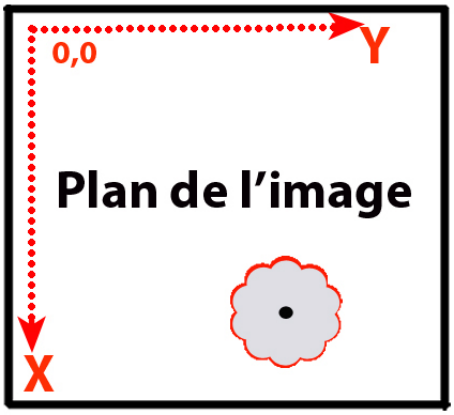


FIGURE 1.3: orientation des axes dans le plan de l'image.

Nous utilisons dans ce travail une camera USB de type **logiteque** (fig : 1.4).

Pour simplifier le calcule de la commande, on doit faire une transformation de repère pour qu'on puisse ramener ce qu'on voit (centre de gravité de la surface) vers ce qu'on

FIGURE 1.4: Logitech USB Camera.

doit voir (centre du plan de l'image), pour cela une transformation simple est a appliquer (voir fig 1.5)

$$\begin{cases} X = x - h/2 \\ Y = y - L/2 \end{cases} \tag{1.1}$$

Où :

- h est la hauteur du plan de l'image en pixels (h = 320 pixels).
- L est la largeur du plan de l'image en pixels (L = 240 pixels).

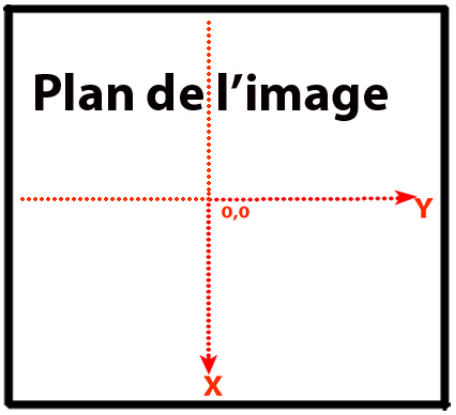


FIGURE 1.5: transformation du repère du plan de l'image.

donc l'objet détecté a comme projection une surface de même niveau de gris et de centre de gravité de coordonnées connus (x,y) en pixel (voir fig 1.6), ces paramétres doivent étre ramenés a la référence qui a comme coordonnés (0,0) pour que la caméra ne perd pas la surface de vue.

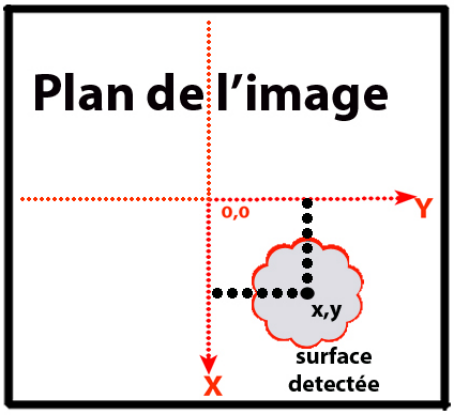


FIGURE 1.6: paramétres utilisés pour le calcule de la commande.

1.1 Résultats

les figures suivantes montrent les résultats obtenus par l'application du programme de détection des objets

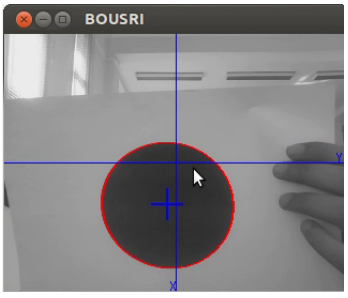


FIGURE 1.7: Résultat de detection d'un objet.

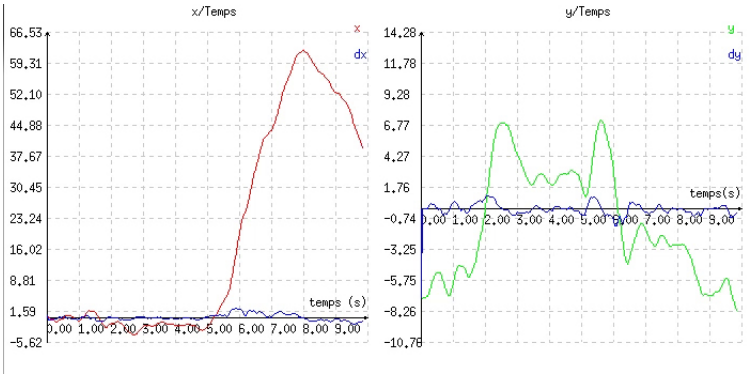


FIGURE 1.8: Résultat d'affichages des differentes parametres en temps réel.

Chapitre 2

Application de la Logique Floue pour le suivie de surface

Au départ théorie, la logique floue s'affirme comme une technique opérationnelle. Utilisée à côté d'autres techniques de contrôle avancé, elle fait une entrée discrète mais appréciée dans les automatismes de contrôle industriel. La logique floue ne remplace pas nécessairement les systèmes de régulation conventionnels ; elle est complémentaire. Ses avantages viennent notamment de ses capacités à :

- Formaliser et simuler l'expertise d'un opérateur ou d'un concepteur dans la conduite et le réglage d'un procédé,
- Donner une réponse simple pour les procédés dont la modélisation est difficile.
- Prendre en compte sans discontinuité des cas ou exceptions de natures différentes, et les intégrer au fur et à mesure dans l'expertise.
- Prendre en compte plusieurs variables et effectuer de la « fusion pondérée » des grandeurs d'influence.

2.1 Intérêt et utilisation de la logique floue pour le contrôle

L'être humain résout souvent des problèmes complexes à l'aide de données approximatives la précision des données est, donc, souvent inutile. Ce qui fait que plutôt que de modéliser le système, il est souvent intéressant de modéliser le comportement d'un opérateur humain face au système et plutôt que par des valeurs numériques précises, le fonctionnement doit être décrit par des qualificatifs globaux traduisant l'état approximatif des variables.

Un traitement flou se fait suivant le schémas suivant :

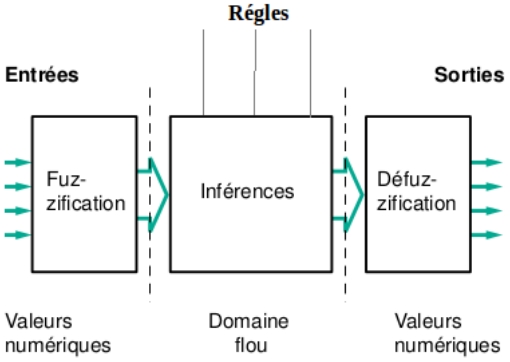


FIGURE 2.1: Régulateur floue.

les grandeurs numériques d'entrées que nous avons utilisé dans notre cas sont x, dx, y et dy tel qu'ils sont représentées sur la figure 2.2

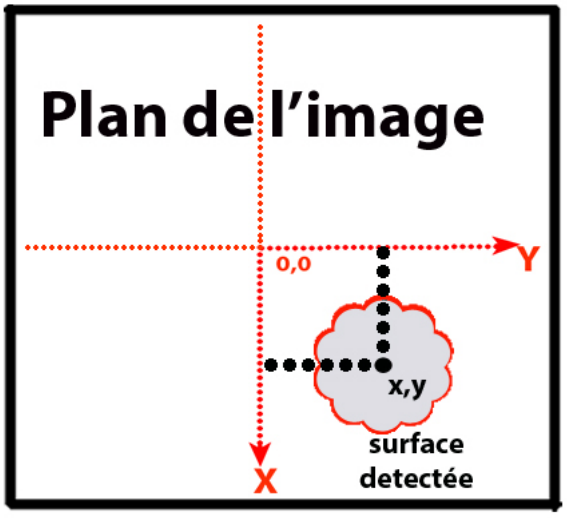


FIGURE 2.2: Paramétres utilisés pour le calcul de la commande.

le rôle du régulateur floue est de calculer la commande qu'on doit appliquer a notre systeme pour qu'il puisse suivre la surface détectée. (fig 2.3).

Pour qu'on puisse appliquer un régulateur floue, on doit d'abord transformer les variables réelles vers des variables floues, cette etape s'appelle *la fuzzification*

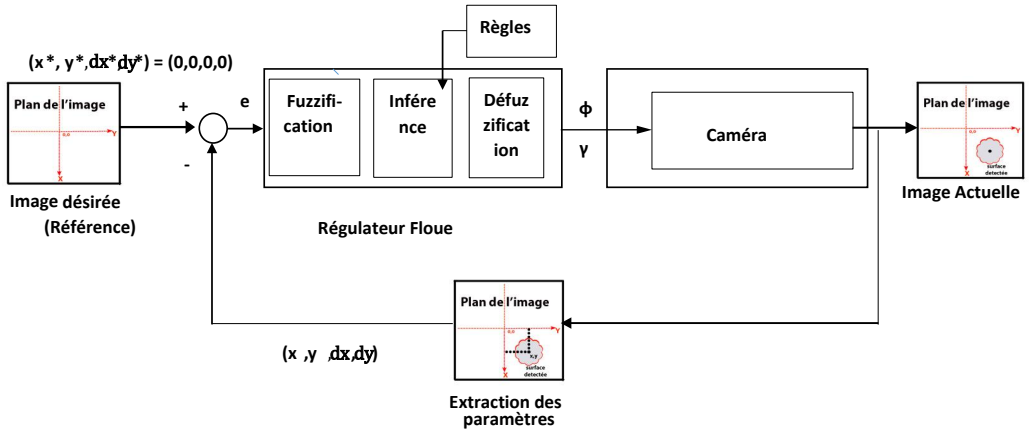
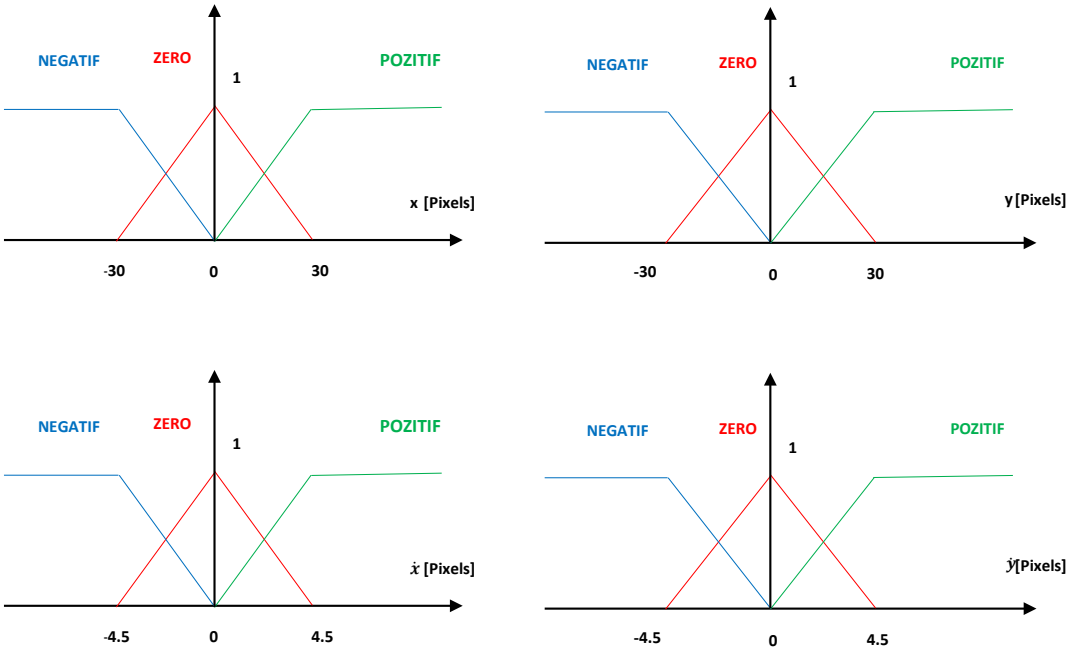


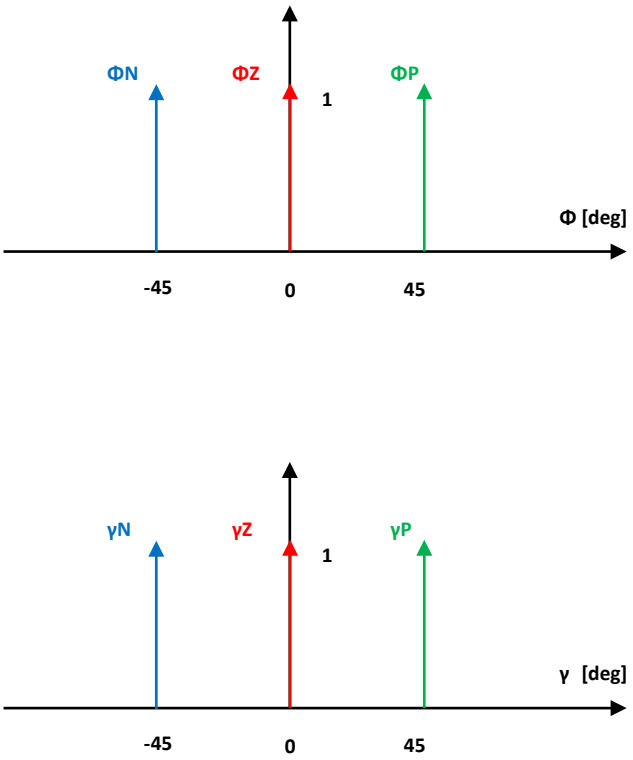
FIGURE 2.3: Schéma fonctionnel du suivie de surface.

2.2 Fuzzification

La fuzzification consiste à évaluer les fonctions d'appartenance qui vont nous permettre d'établir une relation entre la grandeur d'entrée et le degré de vérité de la variable floue correspondante ceci est illustré par les graphes des fonctions d'appartenances.

Les fonctions d'appartenances choisit pour le suivie de la surface sont les suivants :





où $\phi = 45deg$ et $\gamma = 45deg$ représentent respectivement l'angle de braquage maximale de la caméra dans la direction horizontale et verticale. donc ϕ permet d'annuler l'erreur sur y et γ annule l'erreur sur x.

Les fonctions d'appartenances ont été choisies triangulaires et symétriques pour simplifier le calcule de la commande et ainsi permettre de réduire le temps d'exécution du programme.

2.3 Énoncé des règles

Les systèmes à logique floue nécessitent une certaine expertise et connaissance en ce qui concerne le fonctionnement du système à commander exprimé sous forme d'une base de règles du type : Si ...(prémisses) ... Alors ... (conclusion).

n	Règles	y	dy	ϕ
1	RP1	P	-	P
2	RP2	Z	P	P
3	RN1	N	-	N
4	RN2	Z	N	N
5	RZ1	Z	Z	Z

TABLE 2.1: Définition des Règles Floues pour le suivie de surface (ϕ)

Une bonne énonciation des règles est une étape cruciale du bon fonctionnement de la commande par logique floue car cette étape constitue la partie ≪ raisonnement ≫ du robot, c'est pourquoi le choix des règles doit être minutieux.

Pour cela nous nous somme basé sur les décisions prises par un homme qui essaye de rejoindre la surface detecté sur le plan de l'image et de maintenir la caméra dans la bonne direction pour pouvoir garder cette surface dans ce plan.

2.4 Commande pour l'angle ϕ

le calcule de cette commande dépend des deux paramétres Y et dY.

2.4.1 Règles qui favorise un angle de braquage positif (vers la Gauche)

- RP1 Si Y est POSITIF alors ϕ est POSITIF.
- RP1 Si Y est ZERO et dy est POSITIF alors ϕ est POSITIF.

2.4.2 Règles qui favorise un angle de braquage négatif (vers la droite)

- RN1 Si Y est NEGATIF alors ϕ est NEGATIF.
- RN1 Si Y est ZERO et dy est NEGATIF alors ϕ est NEGATIF.

2.4.3 Règles qui favorise un braquage nulle

- RZ1 Si Y est ZERO et dY est ZERO alors ϕ est ZERO.

n	Règles	x	dx	γ
1	RP1	P	-	P
2	RP2	Z	P	P
3	RN1	N	-	N
4	RN2	Z	N	N
5	RZ1	Z	Z	Z

TABLE 2.2: Définition des Règles Floues pour le suivie de surface (γ)

2.4.4 Tableau des Règles

2.5 Commande pour l'angle γ

le calcule de cette commande dépend des deux paramétres x et dx.

2.5.1 Règles qui favorise un angle de braquage positif (vers la Gauche)

- RP1 Si x est POSITIF alors γ est POSITIF.
- RP1 Si x est ZERO et dx est POSITIF alors γ est POSITIF.

2.5.2 Règles qui favorise un angle de braquage négatif (vers la droite)

- RN1 Si x est NEGATIF alors γ est NEGATIF.
- RN1 Si x est ZERO et dx est NEGATIF alors γ est NEGATIF.

2.5.3 Règles qui favorise un braquage nulle

- RZ1 Si x est ZERO et dx est ZERO alors γ est ZERO.

2.5.4 Tableau des Règles

On précise que dans la commande par logique floue toutes les règles sont appliquées en même temps, cependant certaines règles seront plus représentatives de l'orientation actuelle de la caméra, elles auront donc un degrés d'influence plus important sur la commande.

2.6 Deffuzzification

A la fin de l'inférence, le braquage (commande floue) est déterminé mais il n'est pas directement utilisable pour commander le braquage de la caméra. Il est nécessaire de passer du ≪ monde flou ≫ au ≪ monde réel ≫, c'est la défuzzification. le calcule des angles de braquages réelles ϕ et γ qui representent la commande du systéme a été fait par l'intermediare de l'algorithme Min-Max via les relations suivantes :

$$\Phi = \frac{-\Phi_{max} M\Phi_N + 0 \times M\Phi_Z + \Phi_{max} M\Phi_p}{M\Phi_N + M\Phi_Z + M\Phi_P} \tag{2.1}$$

$$\gamma = \frac{-\gamma_{max} M\gamma_N + 0 \times M\gamma_Z + \gamma_{max} M\gamma_p}{M\gamma_N + M\gamma_Z + M\gamma_P} \tag{2.2}$$

2.7 Résultats

les résultats des essais en temps réel obtenus sont les suivants :

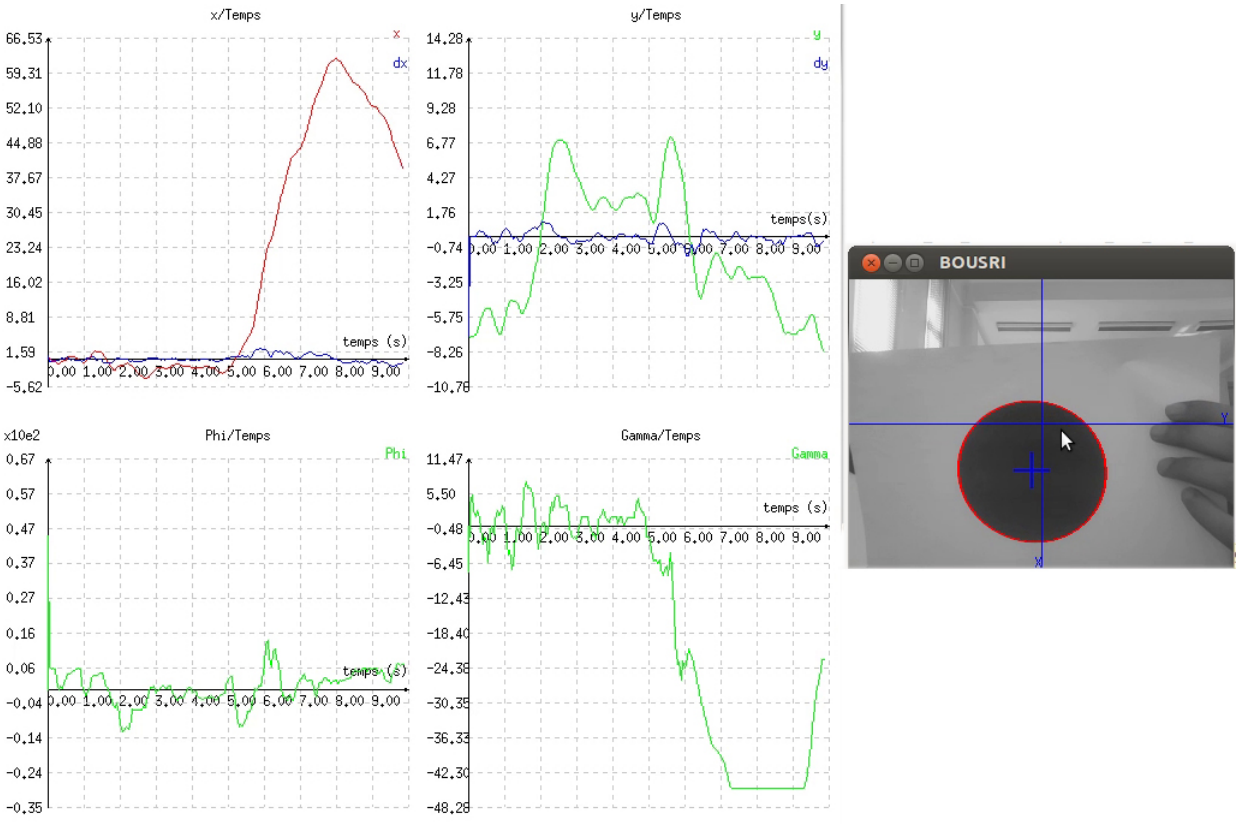
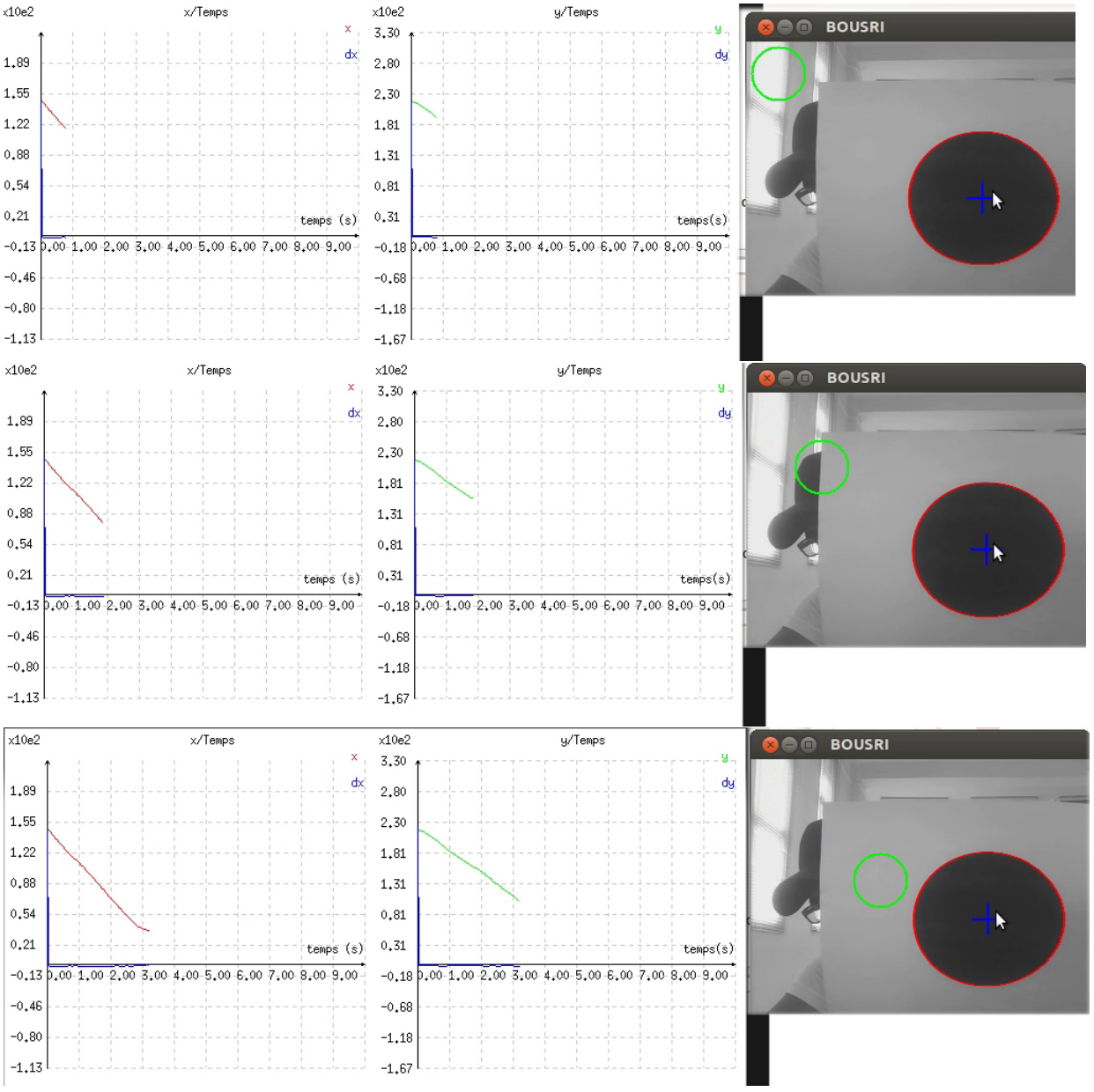
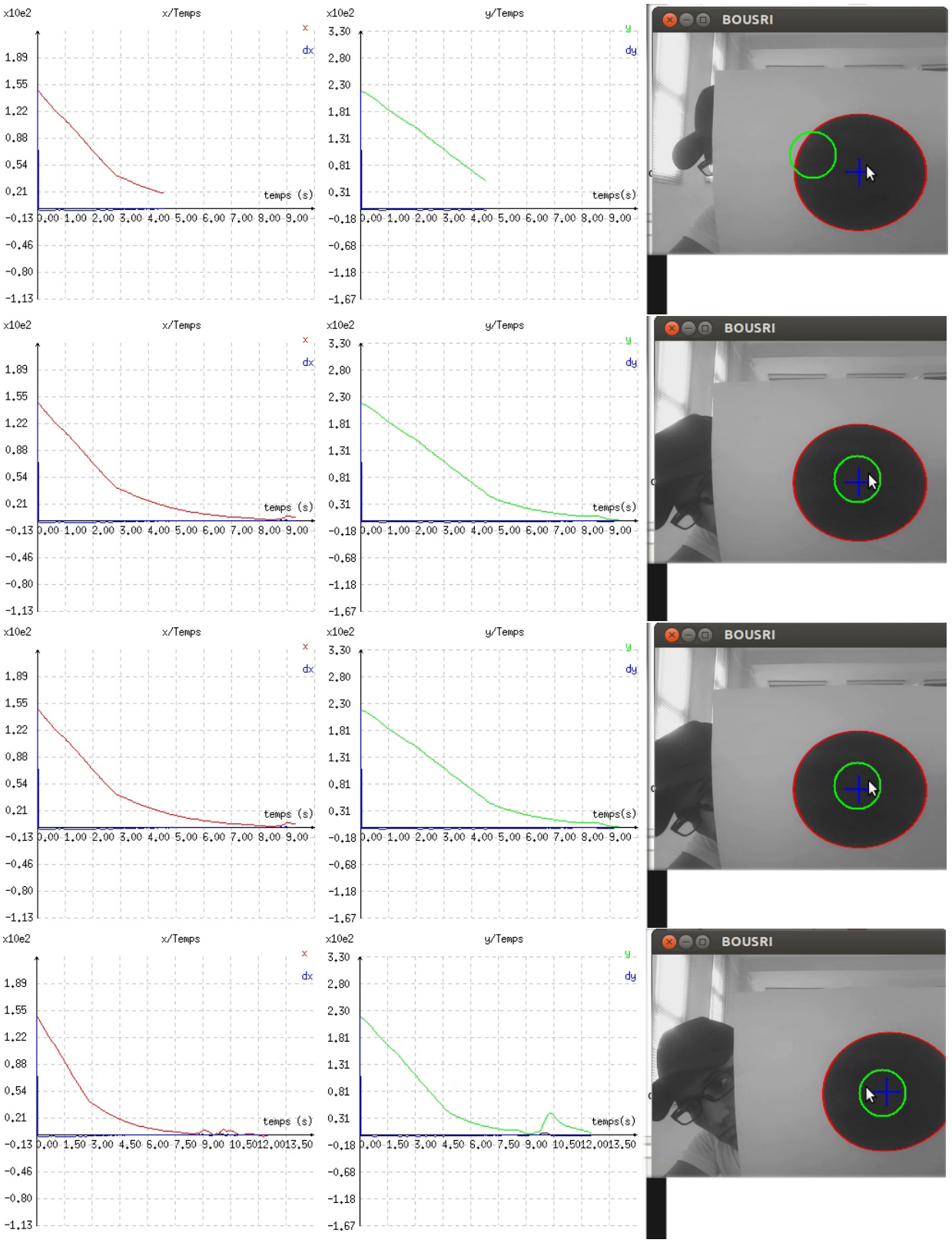
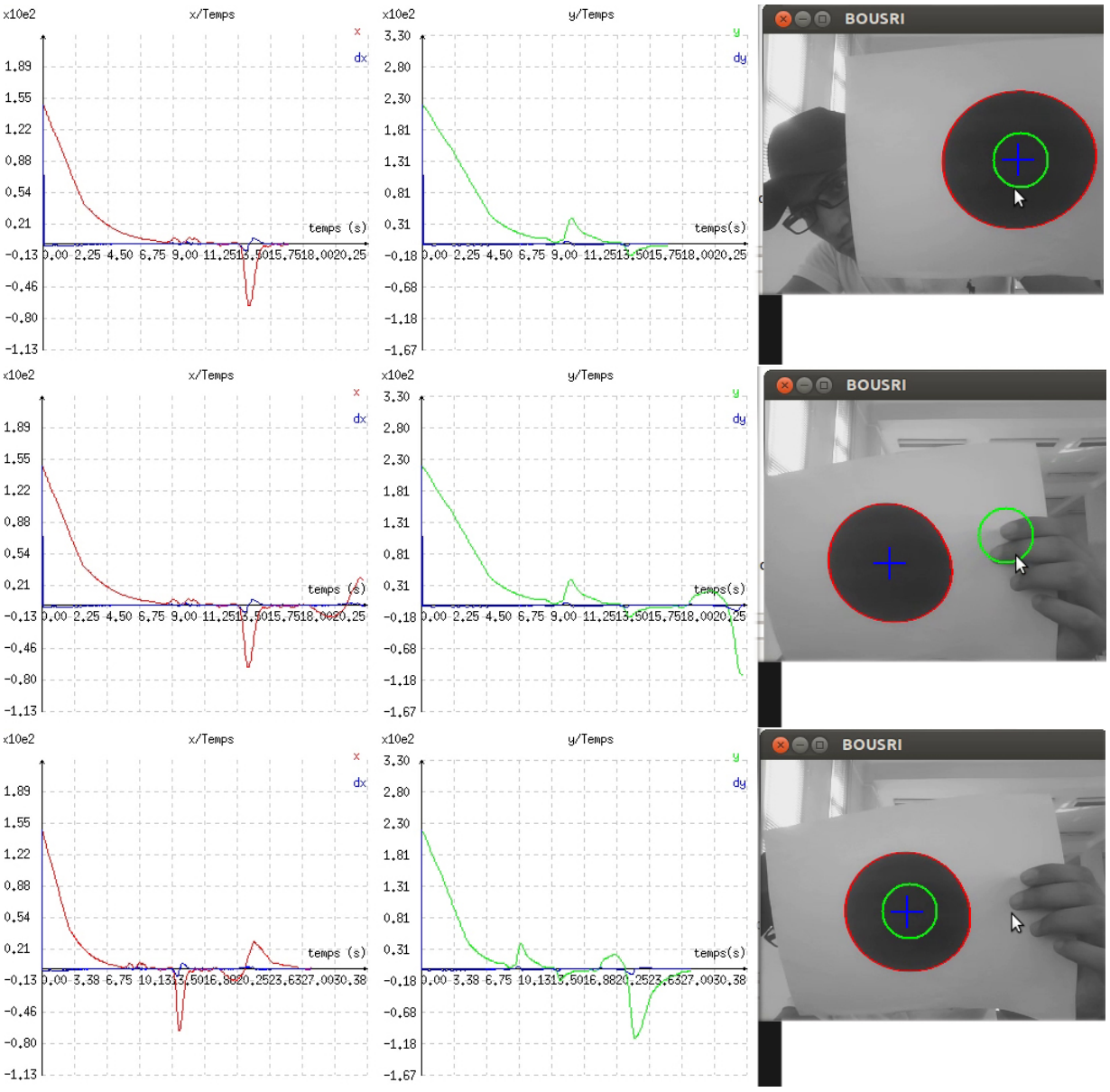


FIGURE 2.4: variation de la commande appliquée en temps réel en fonction des paramètres x et y.

Pour pouvoir visualiser les résultat d'une façon claire on appliqué cette commande a un cercle dans le plan de l'image (en vert fig-**??**) pour qu'il puisse suivre le centre de gravité de la surface détectée.




x10e2
x/Temps
x
dx
1.89
1.55
1.22
0.88
0.54
0.21
-0.13
-0.46
-0.80
-1.13
temps (s)
0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00
x10e2
y/Temps
y
dy
3.30
2.80
2.30
1.81
1.31
0.81
0.31
-0.18
-0.68
-1.18
-1.67
temps(s)
0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00
BOUSRI
0.00 1.50 3.00 4.50 6.00 7.50 9.00 10.50 12.00 13.50



l'erreur entre les coordonnées du centre du cercle et ceux du centre de gravité de la surface détectée est représentée sur la partie gauche de la figure ci-dessus, où les images de 1 à 5 représentent l'application de la commande a un cercle pour le suivie de la surface où l'erreur est sans perturbation pour les autres images l'erreur est avec une perturbation.

Conclusion générale

Dans notre thèse nous avons muni la caméra d'une commande par logique floue lui permettant de détecter par l'intermédiaire de la bibliotheque *VISP*,un objet de type quelconque et de suivre son centre de gravité représenté par un point dans le plan de l'image sans oscillations.

Les résultats du travail présenté dans ce mémoire permettent de dégager les perspectives suivantes :

- Améliorer le programme de détection de la trajectoire.
- Développer ces méthodes pour pouvoir passer à leurs applications à grande échelle.

Annexe A

Annexe A

A.1 La programmation orientée objet

A.1.1 Les deux grands mondes de la programmation

Il existe deux grands types de programmation :

- **Le Procédural** ou structuré comme le C où on définit des fonctions s'appelant mutuellement. Chaque fonction ou procédure est associée à un traitement particulier qui peut être décomposé en sous traitements jusqu'à obtenir des fonctions basiques. Globalement, le procédural travaille sur l'action ou le verbe. Par exemple, pour calculer la vitesse d'une voiture, la philosophie du procédural conduira à faire : `calculVitesse(voiture)` partout où cela est nécessaire. Le code faisant référence à une voiture est donc "fondu" et dispersé dans l'ensemble des programmes. [2]
- **L'Orienté Objet (ou OO)** où on manipule uniquement des objets c'est-à-dire des ensembles groupés de variables et de méthodes associées à des entités intégrant naturellement ces variables et ces méthodes. Dans cette configuration, le sujet est prépondérant : disposant d'un objet Voiture, on effectuera spontanément : `Voiture.calculVitesse()`. Tout le code touchant à l'objet Voiture se trouve ainsi regroupé. [2]
  Remarque :

  Le C++ est l'un des langages de programmation orientée objet, ou l'objet est créé via la déclaration d'une structure informatique particulière qui s'appelle **classe**, où cet objet représente une instance de cette classe.

A.1.2 Avantages de la programmation Orientée Objet

L'Orienté Objet présente d'énormes avantages :

- En général, en OO, cohabitent deux types de développeurs : ceux qui conçoivent les objets et ceux qui utilisent ces objets dans leurs programmes. De cette façon, la programmation orienté objet permet de diviser la complexité [2], car chaque utilisateur peut utiliser des objets prédéfinit qui le concerne pour les implémenter dans un programme dans le but de réaliser un projet.
- Les informations concernant un domaine étant centralisées en objets, il est facile de sécuriser le programme en interdisant ou autorisant l'accès à ces objets aux autres parties du programme.[2]

A.2 La notion de classe

Une classe permet de regrouper dans une même entité des données et des fonctions membres (appelées aussi méthodes) permettant de manipuler ces données. La classe est la notion de base de la programmation orientée objet. Il s'agit en fait d'une évolution de la notion de structure, qui apporte de nouvelles notions orientées objet absolument fondamentales. Un objet est un élément d'une certaine classe : on parle de l'instance d'une classe.

A.3 Ubuntu

Fondé sur la distribution Linux Debian, Ubuntu est un système d'exploitation open source, il est disponible gratuitement, pour les utilisateurs et aussi pour les entreprises, Ubuntu présente plusieurs avantages par rapport aux autres systèmes d'exploitation.

A.4 Asservissement visuel

L'utilisation de l'asservissement visuel en robotique mobile connaît depuis quelques années un essor important, notamment pour les applications des véhicules autonomes, la difficulté essentielle réside dans l'élaboration des lois de commande prenant en compte les contraintes non holonomes de ce type de robots [1]. L'asservissement visuel consiste à contrôler les mouvements d'un système robotique en utilisant des informations visuelles,

notées s, issues d'une ou plusieurs caméras

Les différentes techniques d'asservissement visuel sont classées en fonction des mesures utilisées en entrée du module de commande. Parmi les techniques les plus connues on distingue :

- L'asservissement visuel 3D ou *Position Based Visual servoing* (PBVS).
- L'asservissement visuel 2D ou *Image Based Visuel servoing*(IBVS).

A.4.1 L'asservissement visuel 3D

L'asservissement visuel 3D utilise en entrée de la boucle de commande des informations tridimensionnelles exprimées dans un repère euclidien, à savoir la position r de la caméra par rapport à l'objet d'intérêt, la boucle de commande consiste alors à estimer à chaque itération la position r de la caméra par rapport à cet objet, pour atteindre une position de référence r^* souhaité (voir Fig A.1), à partir des informations visuelles issues de l'image.

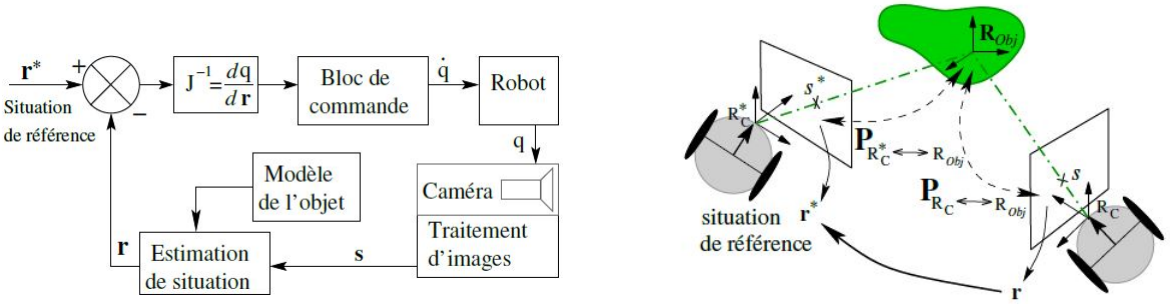


FIGURE A.1: Principe de l'asservissement 3D [**?**]

Pour l'estimation de la position de la caméra par rapport à l'objet, plusieurs méthodes existent. Ces méthodes reposent généralement sur la connaissance à priori d'un modèle 3D de l'objet et des paramètres intrinsèques de la caméra issues du calibrage de cette dernière. Malgré la simplicité de la définition des lois de commandes, l'asservissement visuel 3D présente un inconvénient majeur causé par les erreurs importantes introduites d'une part, par les paramètres intrinsèques de la caméra et d'autre part par le modèle géométrique du robot.

A.4.2 L'asservissement visuel 2D

Contrairement à l'asservissement visuel 3D, les techniques de l'asservissement visuel 2D consiste à utiliser directement les informations visuels s extraites de l'image sans passer par la phase d'estimation de r, la commande consiste alors à contrôler le mouvement de la caméra afin que les mesures dans l'image $s(t)$ atteignent une valeur désiré s^* ou suivent une trajectoire spécifié $s^*(t)$ (voir Fig A.2).

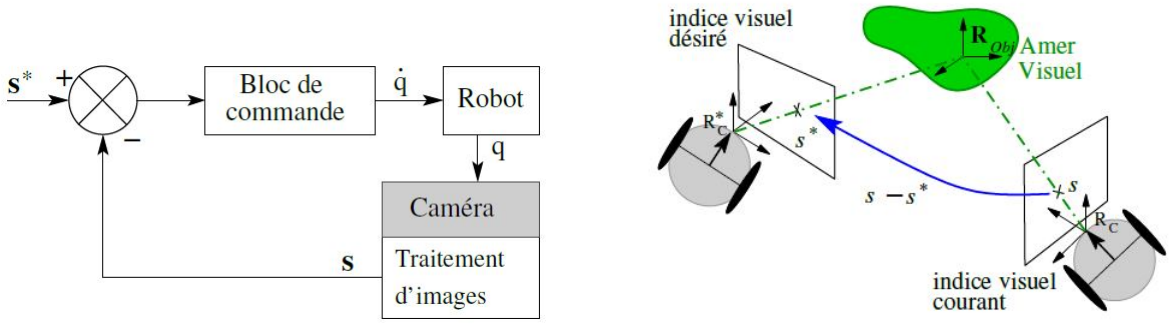


FIGURE A.2: Principe de l'asservissement 2D [**?**]

Cependant pour l'élaboration des diverses lois de commande pour un asservissement visuel 2D, l'obtention de la relation liant les informations visuels s au mouvement de la caméra est une action incontournable. Cette relation est fondamentale et elle est obtenue par la dérivation des informations sensoriels s par rapport à la situation r de la caméra, elle est définit par une matrice appelée *Jacobien de l'image* ou *Matrice d'interaction* notée L (éq A.1).

$$L = \frac{\partial s}{\partial r} \tag{A.1}$$

Les lois d'asservissements visuels 2D sont, d'une manière générale, des lois de commandes relativement rapides à calculer, puisqu'il n'y a pas de phase de reconstruction 3D. Cette simplicité représente un gain considérable au niveau du calcul de la commande, ce qui contribue à la stabilité du système à contrôler.

Il existe une approche qui permet de synthétiser des lois de commande utilisant les données visuelles, cette approche consiste à exploiter le formalisme des fonctions de tâches. Ce formalisme permet de modéliser la tâche considérée sous la forme d'une fonction, puis de synthétiser (voir Annexe) un asservissement permettant de la réguler à zéro.

A.5 Le formalisme de fonctions de tâches

Initialement conçue pour des bras manipulateurs ce formalisme ne peut être directement étendu au cas des robots mobiles non-holonomes du fait de la contrainte de roulement sans glissement. En effet, cette contrainte entrave les mouvements de la caméra, et les tâches robotiques nécessitent alors la réalisation de manœuvres qui peuvent conduire à la perte du motif visuel [**?**]. la solution consiste a introduire un degrés de liberté supplémentaire pour rendre les mouvements de la caméra holonomes par l'introduction d'un degré de liberté supplémentaire permettant à la caméra de se déplacer indépendamment du robot mobile, pour pouvoir en fin appliquer le formalisme des fonctions de tâches.

Bibliographie

[1] François Chaumette. *De la perçeption à l'action : l'asservissement visuel, de l'action à la perçeption : la vision active.* PhD thesis, Université de Reinnes 1 Institut de formation Superieure en Informatique et en Communication, Janvier 1998.

[2] Bertrand Florat. Chapitre 1 l'orienté objet. http://www.florat.net/tutorial-java/chapitre01.html, Consulté en Mai 2015.

[3] Visual Servoing Platform VISP. Bibliothèque multi plate-forme modulaire pour l'asservissement visuel. http://www.irisa.fr/lagadic/visp/visp.html, Consulté en Janvier 2015.