

96/0625

N° d'ordre : D. U. 836

EDSPIC : 119

UNIVERSITE BLAISE PASCAL - CLERMONT II

ECOLE DOCTORALE

SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

**Thèse**

Présentée par

**Christophe DEBAIN**

Formation Doctorale :

Diplôme d'Etudes Approfondies d'Electronique et Systèmes

Pour obtenir le grade de

**DOCTEUR D'UNIVERSITE**

**SPECIALITE :**

**LOIS DE COMMANDE POUR LE CONTROLE ET LA  
MOBILITE DE MACHINES AGRICOLES.**

Soutenue publiquement le 13 Septembre 1996 devant le jury :

Monsieur Jean GAILICE  
Monsieur Alain FOURNIER  
Monsieur Patrick RIVES  
Monsieur Michel BERDUCAT  
Monsieur Pierre BONTON  
Monsieur Philippe MARTINET  
Monsieur Jean Philippe CATROS

Président  
Rapporteur  
Rapporteur  
Examineur  
Examineur  
Examineur  
Invité

N° d'ordre : D. U. 836

EDSPIC : 119

**UNIVERSITE BLAISE PASCAL - CLERMONT II**

**ECOLE DOCTORALE**

**SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND**

**Thèse**

Présentée par

**Christophe DEBAIN**

Formation Doctorale :

Diplôme d'Etudes Approfondies d'Electronique et Systèmes

Pour obtenir le grade de

**DOCTEUR D'UNIVERSITE**

**SPECIALITE :**

**LOIS DE COMMANDE POUR LE CONTROLE ET LA  
MOBILITE DE MACHINES AGRICOLES.**

Soutenue publiquement le 13 Septembre 1996 devant le jury :

Monsieur Jean GALLICE  
Monsieur Alain FOURNIER  
Monsieur Patrick RIVES  
Monsieur Michel BERDUCAT  
Monsieur Pierre BONTON  
Monsieur Philippe MARTINET  
Monsieur Jean Philippe CATROS

Président  
Rapporteur  
Rapporteur  
Examineur  
Examineur  
Examineur  
Invité

**A Cathy**

## **REMERCIEMENTS**

Ce manuscrit décrit des travaux de recherche concernant un système d'aide à la conduite d'engins agricoles qui a été développé au Cemagref (Institut de la Recherche pour l'Ingénierie de l'Agriculture et de l'Environnement) de Clermont-Ferrand dirigé par Monsieur Didier MECHINEAU en collaboration avec le LASMEA (Laboratoire des Sciences et Matériaux pour l'Electronique et, d'Automatique) de Clermont-Ferrand, URA 1793 du CNRS, dirigé par Monsieur Marc RICHETIN.

Je remercie Monsieur Jean GALLICE, Professeur à l'Université Blaise-Pascal pour avoir accepté la présidence de mon jury de thèse, pour m'avoir accueilli dans son équipe et pour les conseils qu'il m'a donnés au cours de cette thèse. Je remercie également Monsieur Philippe ZWAENEPOEL, Directeur de Recherche au Cemagref de Montoldre, pour son accueil au sein de la Division Technologie du Machinisme Agricole.

Je tiens à remercier Monsieur Patrick RIVES, Directeur de Recherche à l'INRIA de Sophia Antipolis et Monsieur Alain FOURNIER, Professeur au LIRMM à Montpellier pour avoir accepté de rapporter ma thèse.

Je remercie également Monsieur Jean-Philippe CATROS, Directeur de Recherche de la société THOMSON à Toulouse, pour l'intérêt qu'il a porté à ce travail en acceptant de participer à son jugement.

Je remercie tout particulièrement Monsieur Pierre BONTON, Professeur à l'Université Blaise-Pascal, Monsieur Philippe MARTINET, Maître de conférence à l'Université Blaise-Pascal et Monsieur Michel BERDUCAT, Ingénieur de Recherche au Cemagref pour l'encadrement qu'ils m'ont prodigué et la grande sympathie qu'ils m'ont témoignée.

Je suis tout particulièrement reconnaissant envers Monsieur Djamel KHADRAOUI pour l'aide et l'amitié qu'il m'a apportées tout le long de cette thèse.

Enfin, je remercierai tout le personnel du Cemagref qui a participé activement au succès de ce projet et à la rédaction de cette thèse, ainsi que tous les membres de ma famille dont le dévouement et le soutien m'ont été précieux.

## ABSTRACT

This doctoral thesis shows the different methods we use to control two agricultural vehicles on a line built by a mowing vegetation. We describe an help guidance system which has two parts. The first one gives some information about the environment of the robots using a CCD camera and an non supervised image processing algorithm. The second one deals with the control algorithms and the vehicles models. It is developed in this thesis.

The first chapter describes the environment of the vehicles and its consequences about some control laws which are used in the literature.

The second chapter proposes a cinematic model for the two vehicles which makes simple the application of control laws calculated for a two wheels axle.

The third chapter develops the theories of two control laws. The first one defines a trajectory between two positions of the vehicles. These positions are calculated in the space of the scene. The second one is deduced from the vision based control theory and applied to the control of non-holonomic mobile robots in the image space.

The fourth chapter shows the result obtained with the help guidance system and proposes an original approach of the model and the correction in the image space of the sliding effects induced by a transverse slope of the ground.

The last chapter gives some elements of the future work about the help guidance system robustness and its application to other vehicles.

KEY WORDS : Mobile Robots, Guidance system, Vision Based Control, Texture, Markov Field, Regions segmentation, Agriculture

## **SOMMAIRE**

<b><u>CHAPITRE I CONTEXTE DE NOS APPLICATIONS</u></b>	<b>1</b>
1. POURQUOI UN GUIDAGE PAR TRAITEMENT D'IMAGES ?	3
1.1 A QUEL UNIVERS NOS APPLICATIONS APPARTIENNENT-ELLES ?	4
1.2 COMMENT CHOISIR LE CAPTEUR ?	4
2. QUELQUES LOIS DE COMMANDE UTILISEES	8
2.1 LOIS DE COMMANDE DANS L'ESPACE DE LA SCENE	8
2.1.1 Commande floue	9
2.1.2 Commande par retour d'état d'un chariot mobile	11
2.1.3 Commande par reconstruction de trajectoire	14
2.2 LOIS DE COMMANDE DANS L'ESPACE DU CAPTEUR	14
2.2.1 Commande par réseaux de neurones	15
2.2.2 Commande Référencée Vision (CRV)	18
2.2.3 Asservissement visuel utilisant un modèle 3D de la scène	19
3. CHOIX D'UNE LOI DE COMMANDE POUR NOTRE APPLICATION	21
<b><u>CHAPITRE II MODELISATION DES VEHICULES UTILISES</u></b>	<b>23</b>
1. MODELES DE ROBOTS MOBILES	24
2. MODELISATION DU ROBOT DE TONTE	27
3. MODELISATION DE LA MOISSONNEUSE-BATTEUSE	29
<b><u>CHAPITRE III ASSERVISSEMENT D'UN ROBOT MOBILE A PARTIR D'INFORMATIONS VISUELLES</u></b>	<b>33</b>
1. LE TRAITEMENT D'IMAGES	35
1.1 ATTRIBUTS DE L'IMAGE	35
1.2 SEGMENTATION DE L'IMAGE	36
1.3 APPLICATION : RECHERCHE DE LA LIMITE DE FAUCHE	38
1.4 IMPLANTATION « TEMPS REEL »	41
2. ASSERVISSEMENT PAR RECONSTRUCTION DE TRAJECTOIRE	44
2.1 CALIBRAGE DE LA CAMERA	44
2.1.1 Modèle géométrique de la caméra	44
2.1.2 Intérêt du calibrage	46
2.1.3 Aperçu de quelques méthodes de calibrage	47
2.1.4 Notre application avec la méthode de Toscani	48
2.1.5 Résultats	53
2.2 RECONSTRUCTION DE LA TRAJECTOIRE	54
2.2.1 Calcul de la trajectoire	55

<b>3. ASSERVISSEMENT PAR COMMANDE REFERENCEE VISION</b>	<b>60</b>
3.1 QUELQUES DEFINITIONS GENERALES	61
3.2 MODELISATION DE LA SCENE	61
3.2.1 Calcul de la matrice d'interaction pour une primitive de type point	63
3.2.2 Matrice d'interaction pour une primitive de type droite.	66
3.3 L'APPROCHE FONCTION DE TACHE	69
3.3.1 Une commande proportionnelle	70
3.3.2 Une commande proportionnelle intégrale	71
3.4 APPLICATION DES DEUX LOIS DE COMMANDE REFERENCEE VISION A NOS VEHICULES.	73
3.4.1 Calcul de la matrice d'interaction pour $\underline{S}=\underline{S}^*$	73
3.4.2 Expression des lois de commande pour l'essieu rigide	75
<b><u>CHAPITRE IV SIMULATIONS ET ESSAIS</u></b>	<b><u>79</u></b>
<b>1. SIMULATION DES DEUX LOIS DE COMMANDE</b>	<b>80</b>
1.1 LA COMMANDE PAR RECONSTRUCTION DE TRAJECTOIRE	81
1.1.1 Le simulateur	82
1.1.2 Résultats	83
1.1.2.1 Influence du bruit sur la commande trajectoire	85
1.2 LA COMMANDE REFERENCEE VISION (CRV)	87
1.2.1 Le simulateur	89
1.2.2 Résultats	90
1.2.2.1 Influence de $\lambda$ et $\beta$	96
1.2.2.2 Influence du bruit sur la loi de commande référencée vision (CRV)	103
<b>2. ESSAIS REELS</b>	<b>106</b>
2.1 IMPLANTATION SUR LA TONDEUSE	107
2.1.1 Implantation de l'ensemble contrôle-commande	108
2.1.1.1 Traitement des données visuelles	109
2.1.1.2 Présentation de l'architecture	110
2.1.1.3 Implantation du correcteur bas niveau	111
2.1.1.4 Mise en oeuvre informatique de l'asservissement	112
2.1.2 Essai réel	112
2.2 IMPLANTATION DE LA CRV SUR LA MOISSONNEUSE-BATTEUSE	114
2.2.1 Présentation de l'architecture	115
2.2.2 Essais réels	115
2.2.2.1 Influence du bruit sur les mesures visuelles	119
2.3 CONTROLE DE LA MOISSONNEUSE-BATTEUSE DANS DES CONDITIONS...	120
2.3.1 Terrain en pente	120
2.3.2 Autres résultats avec la CRV intégrale	130

---

<b><u>CHAPITRE V PERSPECTIVES</u></b>	<b><u>139</u></b>
<b>1. PLATE-FORMES MOBILES ARTICULEES</b>	<b>140</b>
1.1 MODELE CINEMATIQUE	142
1.2 APPLICATION DE LA CRV A UNE PLATE-FORME ARTICULEE	144
1.3 SIMULATION	146
<b>2. COMMANDE MULTI-CAPTEURS</b>	<b>148</b>
2.1 TORSEUR D'INTERACTION POUR UN TELEMETRE LASER	149
2.2 COUPLAGE CAMERA TELEMETRE LASER	152
<b>3. INTERACTION VISION-COMMANDE</b>	<b>154</b>



## INTRODUCTION

La mécanisation a profondément changé le déroulement des travaux agricoles au cours du 20<sup>ème</sup> siècle. Aujourd'hui on conçoit difficilement, en Occident, l'utilisation de boeufs ou de chevaux pour labourer un champ, et encore moins celle d'une faux pour récolter des céréales. Ce mouvement n'a cessé de se développer. Il s'est accompagné d'un accroissement de la puissance et de la taille des machines agricoles. La diversification des activités des agriculteurs et leurs soucis réels de productivité montrent actuellement les limites d'une telle approche car l'homme n'est pas toujours adapté aux nouvelles conditions de travail imposées par ces machines de taille démesurée (coupes de huit mètres sur les moissonneuse-batteuses, vitesse des véhicules sur des terrains accidentés...). Aussi depuis quelques années les constructeurs tendent à automatiser certaines tâches afin de soulager le conducteur ou de lui donner plus d'efficacité. Dans cette optique, le Cemagref<sup>1</sup> participe à de nombreux programmes de recherche tels que CITRUS ou MAGALI qui visent à développer ou à automatiser des machines agricoles. Ce mémoire s'inscrit dans deux de ces projets concernant l'entretien des espaces naturels et la récolte des céréales.

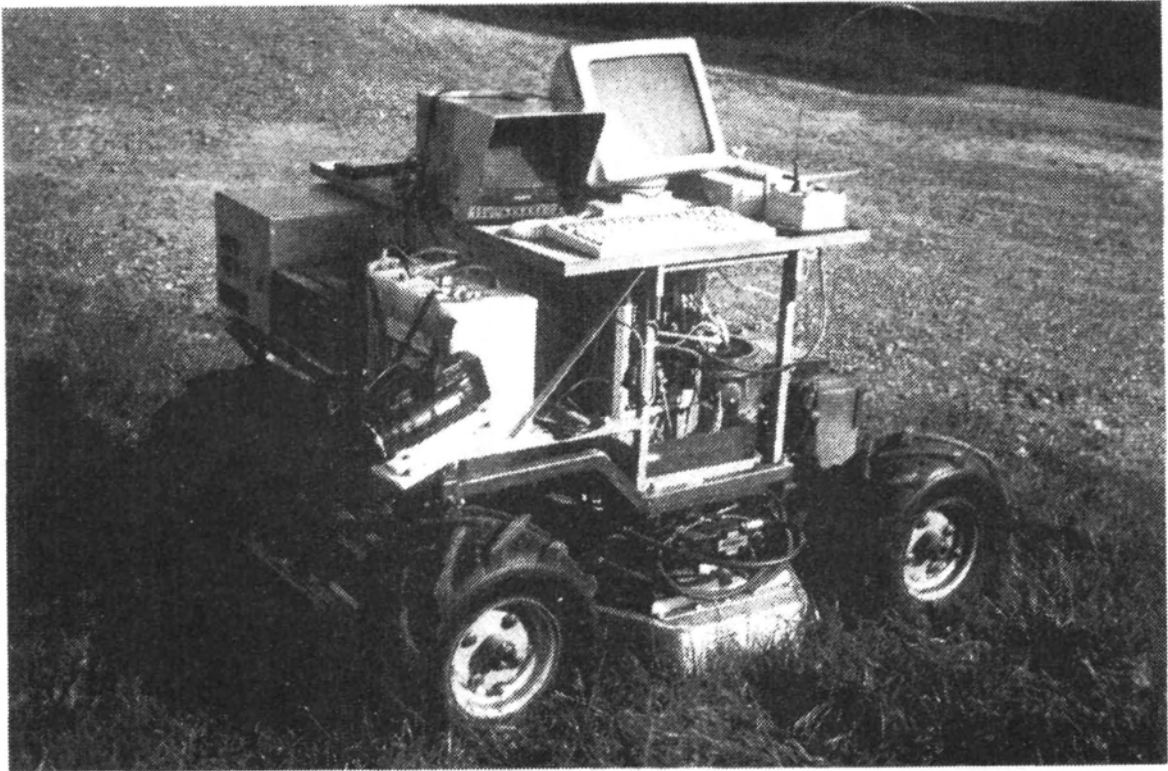
L'entretien des espaces naturels est un élément important de la politique de gestion du territoire. Il concerne les surfaces en jachère et les zones forestières mais aussi les bords de routes et les parcs publics. L'objectif principal de ce projet, regroupant : le Cemagref, le LASMEA<sup>2</sup>, la société Guérin-Courde et le Ministère de la Recherche, était de montrer la faisabilité d'un système d'aide à la conduite pour un engin d'entretien des espaces naturels. Pour cela nous avons instrumenté un prototype ayant pour base un engin à 4 roues motrices de la société Guérin-Courde (*cf photo*). Cette tondeuse professionnelle était à l'origine guidée au moyen d'une télécommande et la qualité de son travail dépendait pour beaucoup de la proximité de l'opérateur et de sa dextérité. Afin de diminuer la charge de l'opérateur, et d'accroître l'efficacité de l'engin, nous avons développé et implanté un système « d'aide à la conduite » permettant à la machine de suivre la limite de fauche qu'elle a elle-même formée lors d'un passage précédent. Nous utilisons pour cela la vision afin de percevoir l'environnement du robot,

---

<sup>1</sup> Centre de recherche pour l'ingénierie de l'agriculture et de l'environnement

<sup>2</sup> Laboratoire des Sciences et Matériaux pour l'Electronique et, d'Automatique

et un correcteur inspiré de la Commande Référencée Capteur [Sam91] développée à l'INRIA<sup>3</sup> pour corriger sa trajectoire.



*Photo de la tondeuse après implantation du système d'aide à la conduite*

Le deuxième projet vise à porter le système développé pour la tondeuse sur des machines agricoles afin d'automatiser la tâche de guidage et de permettre ainsi à l'opérateur de concentrer son attention sur le travail des outils rattachés à l'engin. Les premiers travaux agricoles que nous avons considérés, concernent la récolte des céréales. Dans ce cadre, nous avons instrumenté une moissonneuse-batteuse en lui ajoutant l'électronique dédiée au système « d'aide à la conduite » .

Ces deux projets comprennent principalement la mise au point d'un système visant à asservir automatiquement un véhicule agricole sur un front de végétation qu'il a lui même précédemment créé. Les travaux de recherche entrepris lors de cette étude ont conduit à une première

---

<sup>3</sup> Institut National de Recherche en Informatique et en Automatique

thèse, concernant la perception de l'univers des machines agricoles, puis à une seconde traitant de leur commande et présentée dans ce manuscrit.

I. Le premier chapitre permet de situer notre application et introduit quelques lois de commande utilisées en robotique mobile. Il ne se veut pas exhaustif mais souhaite donner un éclairage sur le contexte dans lequel ces travaux ont été réalisés.

II. Le deuxième chapitre concerne la modélisation cinématique de nos machines agricoles. Il s'appuie sur le modèle simple de l'essieu rigide en le généralisant à deux types de véhicules correspondant aux deux prototypes que nous avons présentés.

III. Le troisième chapitre présente la théorie associée aux modules de perception et de commande.

Une première partie expose brièvement les principes de l'algorithme de traitement d'images utilisant les champs de Markov pour segmenter des images texturées. Des précisions sont alors données sur l'architecture retenue ainsi que sur le temps de traitement d'une image afin d'évaluer les conséquences de cette approche sur la mobilité des véhicules.

Une deuxième partie concerne une loi de commande s'inspirant des travaux de Kehtarnaz [Keh91] qui reconstruit une trajectoire permettant au véhicule de rejoindre une position souhaitée.

Enfin une troisième partie est consacrée à des rappels sur la Commande Référencée Vision [Chau90] puis à son application à la robotique mobile.

IV. Le quatrième chapitre est composé de différents résultats expérimentaux obtenus en simulation ou avec les véhicules lorsque cela était possible. Il est tout d'abord destiné à montrer la faisabilité et la robustesse de nos lois de commande appliquées à des machines agricoles. Puis nous présentons une approche originale concernant l'analyse de quelques phénomènes perturbant la trajectoire des engins (pente du terrain, glissements...).

V. Le cinquième chapitre présente les perspectives en matière de robustesse de la perception des informations et du contrôle des véhicules. Il traite également de l'application de nos travaux à un troisième type de machines agricoles.

# CHAPITRE I

## CONTEXTE DE NOS APPLICATIONS

## INTRODUCTION

La robotique mobile apparut dans les années 60 au sein des laboratoires de recherche américains et russes. Cependant, les premiers démonstrateurs motivés par une demande civile et militaire datent des années 80. On trouve alors de nombreux projets concernant la robotique intérieure et extérieure. Citons, par exemple, le projet américain « ALV -Autonomous Land Vehicle- » [Wax85] [Turk88] en 1984, ou les travaux du professeur Dickmanns [Dick85] concernant la mobilité autonome, en milieu extérieur.

Les premiers travaux rencontrés dans le domaine agricole se situent dans les années 70 [Jul71]. Basés sur des modules de perception relativement simples, ils permettent aux robots d'évoluer dans des milieux bien structurés. Cependant, l'ensemble de ces travaux ne mentionne pas d'application réelle atteinte. De plus, ces études sont plus motivées par l'intérêt de la recherche que par une réelle demande d'automatisation des machines agricoles.

Depuis la fin des années 80, la robotique mobile semble connaître un nouvel essor. Des applications multiples apparaissent, tels que l'entretien des espaces naturels [Tan90], l'intervention dans des bâtiments [Cha92] [Hof95], l'automatisation de la conduite automobile [Dick87] [Chap91] [Keh91] [Jur92] [Jur94] [Kha95b] [Mas95], l'exploration de nouveaux milieux [Gir96], les travaux publics [Mäk95] ou l'automatisation des machines agricoles [Kla94] [Hum94]. De nombreux projets Européens ont vu le jour dans ce contexte tel que Prometheus, Citrus, Autofarm,... L'étendue des exemples rencontrés montre que tous ces travaux sont multi-disciplinaires et la mobilité y est présente sous diverses formes. En effet, elle peut être soit entièrement automatique [Rin95], soit téléopérée [Tho91], soit une simple indication donnée à l'opérateur, ou une synthèse de ces différentes approches [Der94].

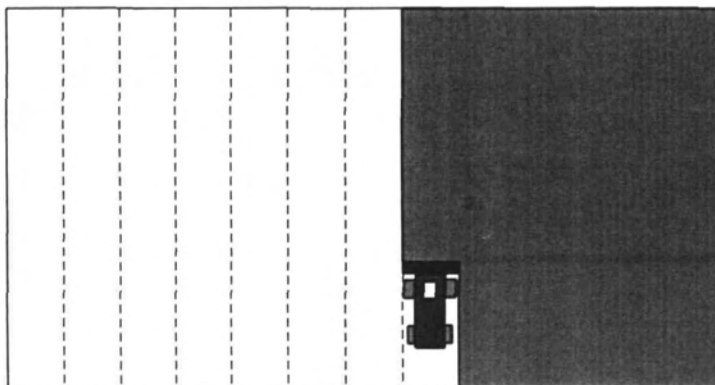
## 1. POURQUOI UN GUIDAGE PAR TRAITEMENT D'IMAGES ?

Le guidage d'un engin mobile nécessite une analyse rigoureuse des conditions dans lesquelles il évoluera. Il est évident qu'un véhicule lunaire n'aura pas les mêmes caractéristiques, ni les mêmes capteurs, qu'un manipulateur évoluant sur un site industriel. La première étape de nos projets, a été d'identifier les engins et leur environnement en fonction des applications visées. Pour les véhicules, l'analyse avait été conduite bien avant que l'on ne songe à automatiser l'entretien des « espaces naturels » (notre première application). De même, cela fait presque cinquante ans que les moissonneuses-batteuses sont utilisées pour la récolte du blé. Nous n'avons donc pas eu à nous préoccuper de la mobilité et de la locomotion de nos véhicules, qui se montraient déjà très efficaces quant à l'application de ces tâches. Par contre, une étude de leur environnement a été nécessaire afin de déterminer les moyens à mettre en œuvre pour leur « guidage automatique ».

L'examen des différentes situations rencontrées lors de travaux agricoles [Mec92] montre que l'environnement des engins agricoles est plus ou moins complexe. Il peut, toutefois, se décliner en plusieurs grandes catégories dont les deux premières sont :

- **Environnement structuré** : les alignements de plantes rencontrés en arboriculture ou en grandes cultures structurent l'univers.
- **Environnement non structuré** ou insuffisamment structuré (comme par exemple les surfaces déchaumées ou juste ensemencées).

Il existe enfin un troisième type d'environnement que l'on ne peut classer dans aucune des deux catégories précédentes : les **environnements structurés par le passage de la machine agricole**. Par exemple, lorsqu'un engin fauche de l'herbe, il laisse une interface fauchée/non-fauchée qui servira de guide au passage suivant (*Figure I 1*).



*Figure I 1 : exemple d'environnement structuré itérativement*

Cette troisième catégorie semblerait faire partie des environnements structurés. Cependant, une différence importante les distingue. En effet, si le guidage de la machine est considéré comme parfait, alors la ligne tracée au passage N structurera parfaitement l'environnement du passage N+1. Par contre, si pour une raison quelconque le passage N est déformé, voire incorrect, il déstructurera l'environnement du passage N+1. S'il est exact que, dans les deux cas, le passage de l'engin définit une structure de l'environnement, le guidage des véhicules en univers structuré suppose que cette structure soit représentative. Relativement à nos applications, nous distinguerons donc la troisième catégorie de la première et nous la nommerons **environnement structuré itérativement**.

## 1.1 A quel univers nos applications appartiennent-elles ?

Les deux applications présentées dans ce mémoire visent à faucher de la végétation. Dans le cadre de l'aide à la conduite, qui ne représente qu'une partie du guidage autonome, nous considérerons uniquement le déplacement de l'engin par rapport à la ligne qu'il a précédemment créée. Nous n'aborderons pas les situations en bout de raies ou en début de chantier qui nécessitent un repérage absolu de la machine -et/ou- une cartographie préétablie du chantier rural considéré.

Le système décrit n'a pour ambition que de guider localement un véhicule agricole par rapport à une ligne supposée présente dans l'environnement perceptible du robot. Le passage du véhicule permet la création d'une autre ligne qui sera suivie au passage N+1 ; il appartient donc à la catégorie « univers structuré itérativement » .

## 1.2 Comment choisir le capteur ?

Le choix du -ou des- capteurs de perception d'un robot mobile dépend étroitement de son environnement. La mobilité en environnement non structuré nécessitera, par exemple, une localisation de l'engin. Celle-ci pourra être réalisée au moyen de systèmes hyperfréquence de type AXYLE<sup>1</sup> ou GPS<sup>2</sup> combinés ou non à une navigation à l'estime (un tel pilotage permet de palier aux imperfections des systèmes globaux). Cependant, la détection d'obstacles et l'analyse de l'état du sol présent devant le robot ne pourra se faire que par l'intermédiaire d'une perception à courte distance. Dans ce cas, les capteurs utilisés seront soit une ou plusieurs caméras, soit des imageurs lasers bidimensionnels ou tridimensionnels ou encore une ceinture de capteurs ultrasons.

---

<sup>1</sup> Accurate XY Location Equipment

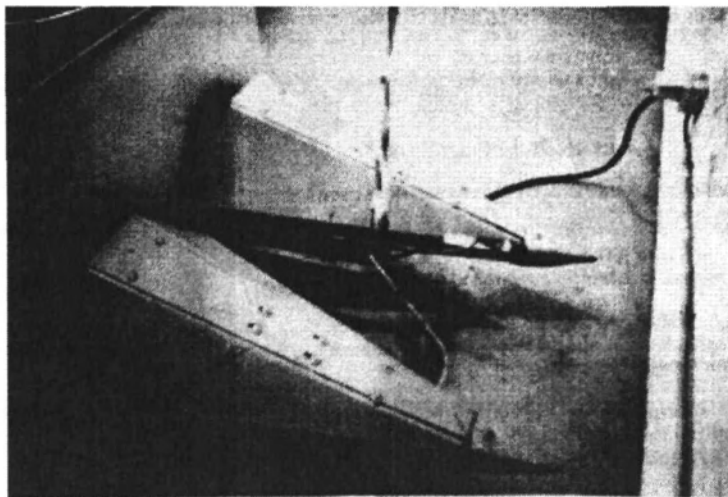
<sup>2</sup> Global Positioning System

Pour nos applications, où l'on ne s'intéresse qu'à un guidage local dans un univers structuré itérativement, nous n'utiliserons pas de localisation absolue mais seulement une perception courte ou moyenne distance de la structure de l'environnement du robot. Ce milieu est représenté par l'interface entre la végétation fauchée et la végétation non fauchée (***Photo I 1***).



***Photo I 1 : vue d'une interface fauchée/non-fauchée de blé***

On pourrait envisager de détecter cette interface à l'aide de capteurs mécaniques tels que ceux utilisés par la société CLAAS pour suivre les rangs de maïs :



***Photo I 2 : vue des palpeurs de végétation sur les carters des becs cueilleurs***



Cependant, la végétation considérée dans nos applications (herbes, céréales) est beaucoup plus fragile que ne l'est une tige de maïs. De plus, l'interface à détecter est relativement bruitée, ce qui n'est pas le cas de celle formée par les rangs de maïs.

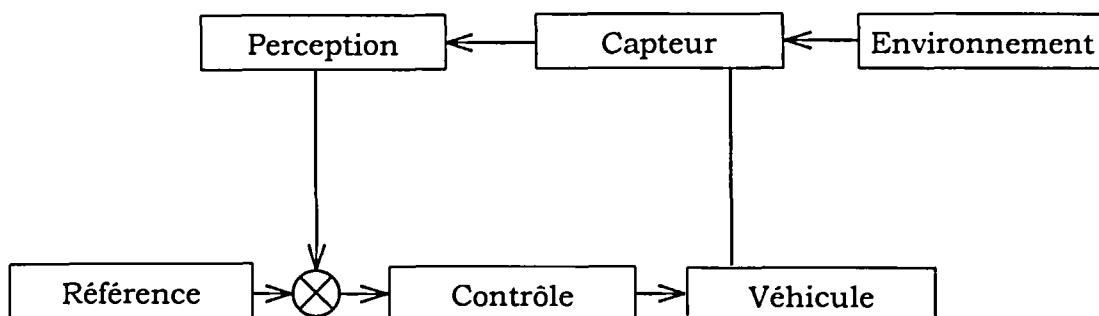
Enfin, ces capteurs ne donnent que l'écart entre la position courante de la machine et celle désirée. Or, la ligne formée par l'interface peut être modélisée localement par une droite, procurant une information de distance mais aussi de cap. En conséquence, il est intéressant de chercher un capteur fournissant ces deux informations. Parmi les solutions envisageables, citons l'imageur laser tridimensionnel ou la combinaison d'une caméra et d'un plan laser. Cependant, ces capteurs sont relativement onéreux et donc peu compatibles avec des applications agricoles. Aussi, nous sommes-nous orientés vers une caméra comportant un objectif asservi sur la luminosité et un contrôle automatique des gains.

L'intérêt de la caméra pour nos applications est de fournir une image contenant la droite qui symbolise l'interface fauchée/non-fauchée. En effet, la projection perspective réalisée par ce capteur conserve, dans la plupart des cas, la linéarité d'une primitive 3D. Bien que cela ne soit pas rigoureusement exact à cause des phénomènes de distorsion des objectifs, nous considérerons que cette hypothèse est vraie dans notre cas.

Il est évident que la projection réalisée par le capteur optique engendre une perte d'informations. En effet, il sera difficile de connaître les coordonnées 3D de la droite décrivant l'interface. Cependant, nous considérerons que la liaison entre la droite de l'image et la droite 3D est suffisamment robuste pour que nous puissions exploiter les coordonnées 2D de l'interface détectée dans l'image.

Ayant choisi notre capteur, il nous faut rechercher un algorithme de traitement d'images capable de détecter l'interface fauchée/non-fauchée, et suffisamment robuste pour s'adapter à l'ensemble des conditions naturelles que nos machines agricoles rencontreront. Ce travail a fait l'objet d'une thèse [Der94] qui présente un algorithme Markovien de segmentation en région d'images texturées. Le principe en sera donné au Chapitre III.

La comparaison de l'environnement perçu (la droite symbolisant l'interface fauchée/non-fauchée) avec un modèle de référence (position de l'interface par rapport à la machine lorsque celle-ci est asservie) suscite une action sur le déplacement du robot. Cet asservissement est représenté par le schéma suivant :



**Figure I 2 : schéma général de l'asservissement**

Cette boucle d'asservissement, et en particulier la comparaison avec le modèle, peut être exprimée dans deux espaces :

- Le premier, qui paraît le plus évident, est l'espace opérationnel du robot. Cependant, le capteur caméra ne donne pas directement d'information dans le repère de la scène. Celles-ci sont projetées sur le plan image et la projection obtenue n'étant pas une bijection, il ne sera pas toujours aisé de reconstruire les coordonnées 3D des points de l'image.
- Le deuxième espace utilisé pour analyser la situation du robot mobile est celui du capteur. Dans ce cas, il n'y aura aucun problème pour traiter les informations perçues par le capteur. Par contre, le modèle correspondant au bon asservissement de la machine devra être projeté dans le plan image. La seconde difficulté sera de calculer des lois de commande dans cet espace qui n'est pas « naturellement » utilisé par les automaticiens.

Dans un premier temps, l'asservissement de nos engins sera fait dans l'espace de la scène en reconstruisant les coordonnées 3D des points image. Pour cela, la caméra sera calibrée en émettant l'hypothèse que les points de contact de la machine avec le sol et les points de l'interface sont dans le même plan. Cependant, les engins agricoles évoluent sur des terrains relativement cahoteux et l'interface fauchée/non fauchée est souvent très irrégulière. Ces conditions de fonctionnement ont au moins deux conséquences :

1. L'hypothèse précédente (un même plan pour les points de contact de la machine avec le sol et l'interface fauchée/non-fauchée) n'est pas toujours vérifiée.
2. On constate un bruit de l'ordre de  $\pm 10$  pixels sur la détection de l'interface dans l'image. Ce bruit a des conséquences importantes sur la reconstruction de la scène qui utilise un système linéaire amplifiant le bruit des données 2D.

Comme nous l'avons vu, l'asservissement dans l'espace du capteur introduit des difficultés dans l'expression du modèle comportemental du véhicule dans son environnement. A l'inverse, le contrôle de robots mobiles dans l'espace de la scène semble plus classique, mais il nécessite une reconstruction 3D de l'environnement. Celle-ci peut introduire de nombreuses erreurs ou être coûteuse en temps de calcul si des techniques sophistiquées de reconstruction sont utilisées.

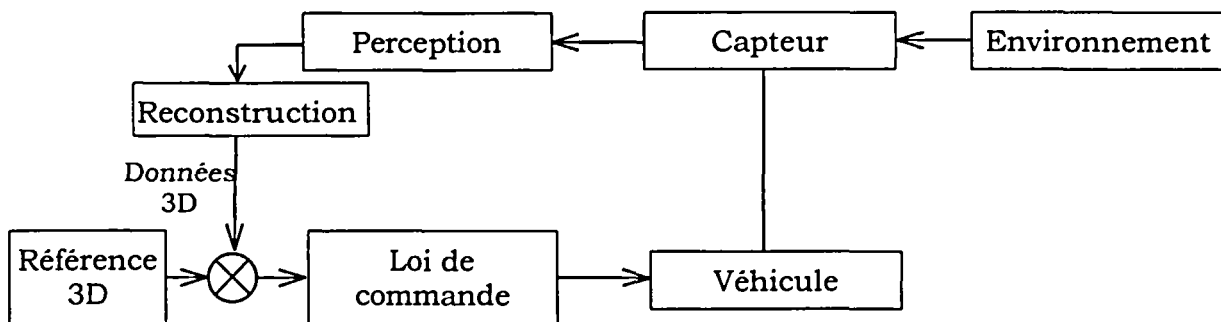
## 2. QUELQUES LOIS DE COMMANDE UTILISEES

De nombreuses lois de commande sont utilisées pour la robotique mobile. Le choix va du simple correcteur proportionnel aux lois de commande basées sur des modèles dynamiques. Citons par exemple : les commandes référencées vision, les commandes par reconstruction de trajectoire, les commandes dans l'espace d'état, les commandes par réseaux de neurones ou encore des commandes utilisant la logique floue.

Parmi ces correcteurs, beaucoup sont exprimés dans l'espace de la scène mais d'autres sont calculés directement dans l'espace du capteur.

### 2.1 Lois de commande dans l'espace de la scène

Pour cette première approche, les données issues du module de perception sont utilisées pour reconstruire l'environnement du robot ou au moins la partie de son environnement qui est nécessaire à sa mobilité. Puis la position du robot est comparée avec une référence afin de calculer une loi de commande exprimée dans l'espace de la scène. Donnons le schéma général décrivant l'asservissement d'un robot mobile dans la scène :

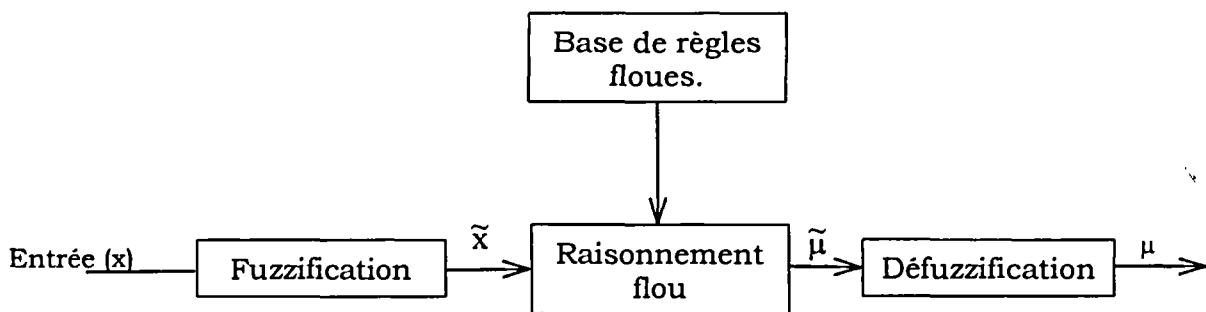


***Figure I 3 : schéma d'asservissement dans l'espace opérationnel***

### 2.1.1 Commande floue

Le principe de la commande floue est assez simple : il s'agit de reproduire le comportement d'un opérateur humain au moyen de règles qui manipulent des données imprécises (rapide, lent, long, petit, grand...). Il est basé sur la théorie des ensembles flous développée par ZADEH [Zad65] en 1965 à l'Université de Berkley. L'appartenance d'une variable à un ensemble flou dépend des frontières de cet ensemble. Elle sera représentée non pas par une variable binaire mais par une probabilité d'appartenance variant de 0 à 1. L'étape consistant à traduire l'entrée d'un système en terme de probabilité d'appartenance à des ensembles flous est appelée « Fuzzification ».

A partir de ces ensembles flous et des règles d'action fonction de l'appartenance d'une variable à un ensemble flou, il est possible de définir des raisonnements flous. La dernière étape consiste à traduire la « sortie floue » en valeur numérique compatible avec l'entrée du système ; elle est appelée défuzzification.

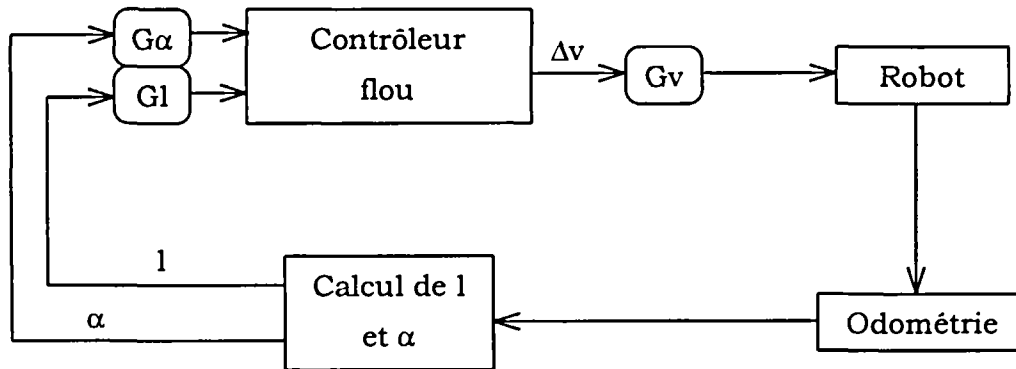


**Figure I 4 : exemple de correcteur flou**

#### Exemple de contrôle d'un robot mobile par logique floue :

Bourdon dans [Bou92] donne un exemple de commande floue permettant de contrôler un robot mobile. Le véhicule doit suivre une trajectoire prédéfinie par la donnée de sa position et de son orientation en fonction du temps. Le robot estime alors, par odométrie, la distance entre sa trajectoire et celle désirée ( $l$ ) ainsi que la différence d'orientation ( $\alpha$ ). Ces deux grandeurs représentent les entrées de son contrôleur flou, alors que la différence de vitesse des deux roues motrices en est la sortie. Les règles sont établies en fonction de l'expérience acquise par l'opérateur.

Le schéma d'asservissement est le suivant :



**Figure I 5 : schéma d'un contrôleur flou pour un robot mobile**

Les premiers résultats donnés concernent des trajectoires rectilignes ou en forme de cercle. Des simulations, ainsi que des essais réels, montrent de bonnes performances de l'algorithme de commande. Les performances du correcteur flou sont comparées avec celles d'un correcteur par retour d'état et montrent des temps de réponse légèrement inférieurs.

Pour le guidage d'un robot mobile à quatre roues, Miyata utilise également la commande floue pour la planification de la trajectoire et le guidage automatique de l'engin [Miy92]. Des considérations géométriques, ainsi que des réflexes calqués sur ceux d'un opérateur humain, lui permettent de définir ses règles et ses sous-ensembles flous.

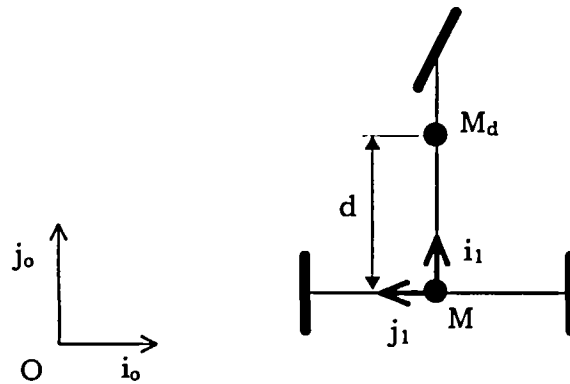
La commande floue est une méthode intéressante à utiliser lorsqu'il n'existe pas de modèle du processus : si le système est trop complexe, ou si l'on n'a pas pris le temps de le modéliser. Elle peut être mise en oeuvre si le processus est facilement commandable par un opérateur humain et si on ne cherche pas de grandes performances de l'asservissement.

*Dans le cadre de nos applications, on pourrait envisager une commande floue utilisant la position et l'orientation de la droite détectée dans l'image comme variable d'entrée. La sortie serait donnée par la direction du véhicule. Les règles floues seraient déduites de l'enregistrement des données visuelles et de l'angle de direction lors de la conduite du véhicule par un opérateur humain.*

## 2.1.2 Commande par retour d'état d'un chariot mobile

Dans [Sam90] Claude Samson démontre l'existence d'un retour d'état stabilisant continu en position et en orientation pour un robot mobile dans le cas de la poursuite d'une trajectoire décrite par « un chariot virtuel ». La démonstration suppose que la vitesse du chariot virtuel ne s'annule pas. Si cette condition nécessaire n'est pas satisfaite, l'auteur montre qu'il n'existe pas de retour d'état stabilisant continu permettant d'effectuer cette fonction.

L'étude proposée s'appuie sur un modèle d'état du chariot exprimé dans un repère mobile lié à son axe [Sam91a] [Ait93]. Les variables de commande considérées sont la vitesse du centre du chariot ainsi que sa vitesse angulaire. Donnons les équations du modèle du robot dans un repère mobile  $(i_1, j_1)$  lié à l'axe du chariot :



**Figure I 6 : paramètres du chariot**

on a :

$$\begin{aligned} \vec{MM}_d &= d \cdot \vec{i}_1 \\ \vec{MO} &= x \cdot \vec{i}_1 + y \cdot \vec{j}_1 \end{aligned}$$

En considérant le vecteur de configuration :

$$X = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

et le vecteur de commande :

$$U = \begin{pmatrix} v \\ w \end{pmatrix}$$

$\theta$  étant l'orientation du chariot,  $v$  sa vitesse moyenne et  $w$  sa vitesse de rotation par rapport à l'axe  $k$ .

On obtient alors :

$$\dot{X} = B(X) \cdot U$$

avec :

$$B(X) = \begin{pmatrix} -1 & y \\ 0 & -(d+x) \\ 0 & 1 \end{pmatrix}$$

Dans ce cas, le linéarisé tangent du système (considéré à l'origine) n'est pas

commandable puisque  $\dot{X} = B(0) \cdot U$  avec :  $B(0) = \begin{pmatrix} -1 & 0 \\ 0 & -d \\ 0 & 1 \end{pmatrix}$  et  $B(0)$  non carré.

L'auteur montre alors qu'il n'existe pas de retour d'état stabilisant continu  $U(X)$  pour le chariot mobile à deux roues motrices. C'est-à-dire qu'il n'existe pas de commande continue  $U(X)$  qui fait converger en même temps  $x$ ,  $y$ ,  $\theta$  à zéro quelles que soient les conditions initiales.

Dans cette première approche, l'auteur cherche à asservir le chariot par rapport à une position fixe, c'est-à-dire à vitesse finale nulle. Il a montré que ce n'était pas possible par un retour d'état stabilisant continu. Cependant, une seconde approche est proposée où l'on essaie d'asservir le robot mobile par rapport à un « chariot virtuel » en mouvement. Il montre alors qu'il est possible de déterminer un retour d'état stabilisant continu  $U(X)$  permettant de commander le chariot mobile.

L'équation d'état du système est alors donnée par :

$$\dot{X} = A(X, t) \cdot X + B(X) \cdot U$$

avec :

$$A(X, t) = \begin{pmatrix} 0 & \dot{\theta}_r(t) & h_1(\theta, t) \\ -\dot{\theta}_r(t) & 0 & h_2(\theta, t) \\ 0 & 0 & 0 \end{pmatrix}$$

$$B(X) = \begin{pmatrix} -1 & y \\ 0 & -(d+x) \\ 0 & 1 \end{pmatrix}$$

$$h_1(\theta, t) = v_r(t) \cdot \frac{\cos\theta - 1}{\theta} + d \cdot \dot{\theta}_r(t) \cdot \frac{\sin\theta}{\theta}$$

$$h_2(\theta, t) = \dot{\theta}_r(t) \cdot \frac{\cos\theta - 1}{\theta} - v_r(t) \cdot \frac{\sin\theta}{\theta}$$

$\theta_r$  et  $v_r$  étant l'orientation et la vitesse du « chariot virtuel de référence ».

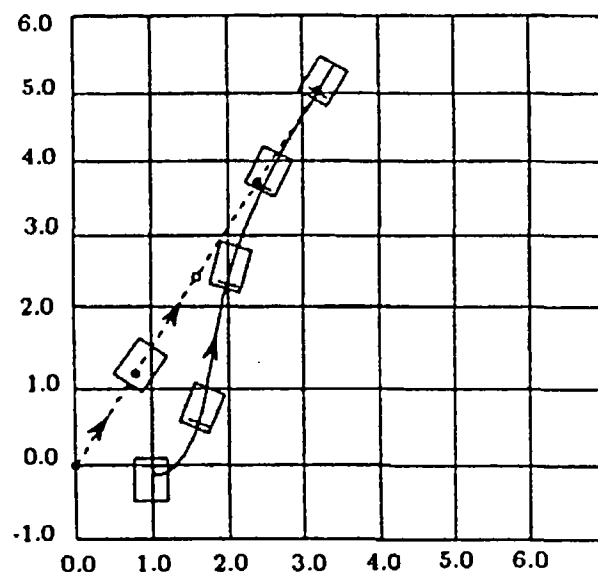
L'équation de commande est de la forme  $U=K.X$  avec  $K$  une matrice de gains choisie de telle façon que la matrice d'état du système en boucle fermée  $A(0) + B(0)K$  soit « stable » (valeurs propres à parties réelles négatives).

Dans le cas où le chariot virtuel a une vitesse constante  $v_r$  et une vitesse de rotation nulle (trajectoire rectiligne uniforme),  $U$  peut être donnée par :

$$\begin{cases} v = x \\ \dot{\theta} = \text{sign}(v_r) \cdot y - \sqrt{1 + 2|v_r|} \cdot \theta \end{cases}$$

avec :  $\text{sign}(V_r) = 1$  si  $V_r > 0$  et  
 $\text{sign}(V_r) = -1$  si  $V_r < 0$ .

En utilisant cette loi de commande l'auteur donne des résultats de simulation tout à fait encourageants :



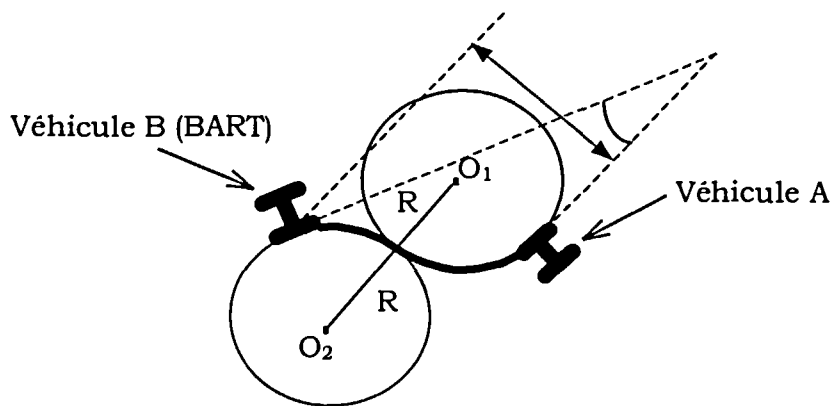
**Figure I 7 : suivi du chariot virtuel de référence**



### 2.1.3 Commande par reconstruction de trajectoire

Ces lois de commande calculent une trajectoire permettant au robot mobile de passer d'une position courante à une position désirée. L'ensemble de ces trajectoires est quasiment infini. Celles-ci sont le plus souvent modélisées par des segments de droites, des cercles [Ioo88][Keh91], des clothoïdes [Ioo88], ou des courbes B-splines [Vaz94].

Cette approche est utilisée dans le cadre du projet américain BART (Binocular Autonomous Research Team [Keh91]), qui vise à asservir un véhicule B sur un véhicule A qui le précède. Nasser Kehtarnaz utilise la stéréo vision afin de déterminer la position de B par rapport à celle de A. Grâce à la position courante de BART (B) et à sa position souhaitée (celle du véhicule précédent A), l'auteur calcule une trajectoire, composée de deux cercles, qui permet de corriger à la fois l'erreur latérale et l'erreur d'orientation (*Figure 18*). Des tests réels ont été effectués avec succès jusqu'à une vitesse d'environ 40 km/h.



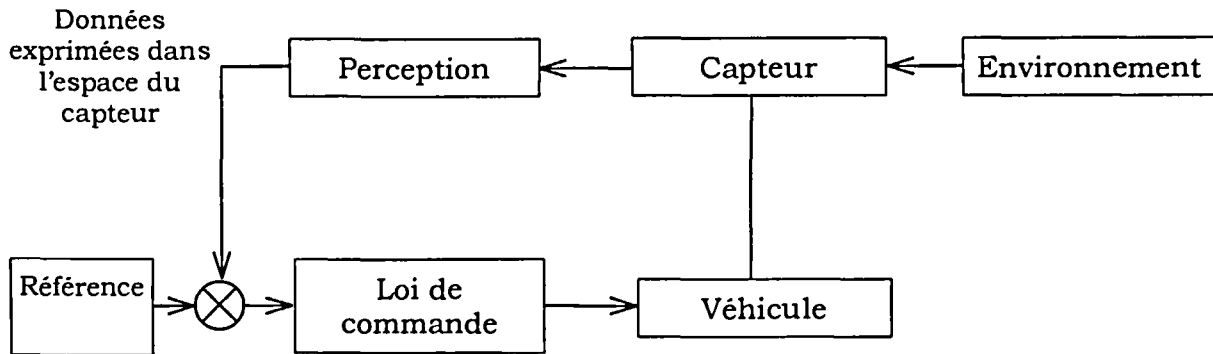
***Figure 18 : trajectoire composée de deux cercles tangents***

Cette méthode sera détaillée au chapitre III où elle sera appliquée à un véhicule mobile agricole.

## 2.2 Lois de commande dans l'espace du capteur

Les différentes lois de commande précédemment citées sont calculées à partir d'informations traduisant la position du robot -et/ou- la position qu'il doit atteindre dans l'espace. Elles nécessitent donc de percevoir l'environnement dans un repère qui lui est lié. Or, la plupart des capteurs donnent une information dans un repère issu d'un de leurs éléments (par exemple le plan image pour une caméra). Aussi, une seconde approche au problème du guidage automatique des robots mobiles est de percevoir l'environnement du robot et de contrôler sa mobilité dans l'espace du capteur.

Donnons le schéma général décrivant l'asservissement d'un robot mobile dans l'espace du capteur :



**Figure I 9 : schéma d'asservissement dans l'espace du capteur**

Dans cette approche, l'étape d'interprétation des attributs extraits du module de perception a été supprimée, ce qui a plusieurs conséquences :

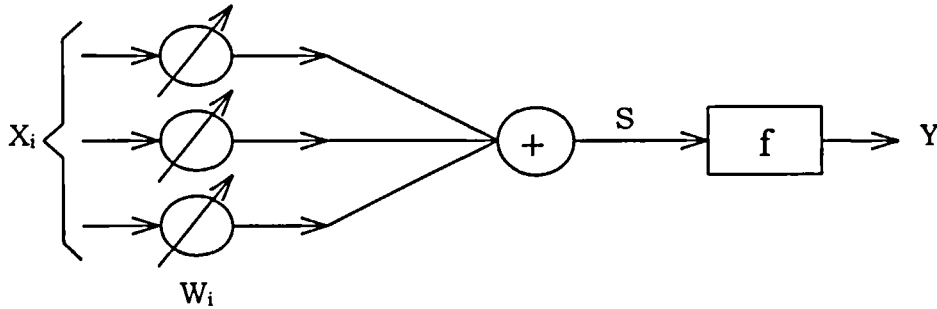
1. l'erreur sur les mesures ne sera pas amplifiée par l'étape d'interprétation,
2. les temps de calculs seront réduits,
3. le modèle traduisant le bon asservissement du robot devra être traduit dans l'espace du capteur,
4. la loi de commande devra être exprimée dans l'espace du capteur et en sera donc tributaire.

### 2.2.1 Commande par réseaux de neurones

La première modélisation d'un neurone artificiel date des années 40 ; elle s'inspire des travaux sur les neurones biologiques. Les réseaux de neurones ont ensuite connu un nouvel essor à partir de 1985 grâce à la mise au point d'algorithmes permettant l'apprentissage à partir d'un échantillon d'exemples.

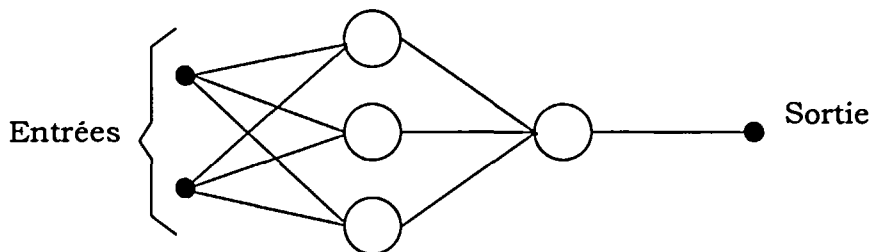
**Un neurone artificiel** peut être formalisé par un automate élémentaire caractérisé par plusieurs paramètres :

- ◆ un vecteur d'entrées  $X_i$ ,
- ◆ un vecteur de poids (coefficients synaptiques)  $W_i$ ,
- ◆ une fonction d'activation  $f$ ,
- ◆ une sortie  $Y = f\left(\sum (X_i \cdot W_i)\right)$ .



avec  $Y = f(S) = a \cdot \frac{e^{kS} - 1}{e^{kS} + 1}$  (f fonction sigmoïde)

**Un réseau de neurones** est un système multi-entrées et multi-sorties composé d'un ensemble de neurones formels connectés entre eux. D'une manière plus générale, l'architecture des réseaux de neurones peut aller d'une connectivité totale (tous les neurones sont reliés les uns aux autres) à une connectivité locale où les neurones ne sont reliés qu'à leurs plus proches voisins. Il est courant d'utiliser des réseaux à structure régulière pour faciliter leurs manipulations. Une autre architecture fréquemment utilisée est celle du réseau multicouches. Dans ce dernier, un neurone de la couche (i) est connecté à tous les neurones de la couche (i-1) et à tous les neurones de la couche (i+1). Il n'y a pas de connexions entre neurones d'une même couche. Le neurone d'une couche donnée reçoit les informations des neurones de la précédente et transmet sa sortie aux neurones de la couche suivante.



**Figure I 10 : exemple de réseau multicouches avec 2 entrées, 1 sortie et une couche cachée de 3 neurones.**

**L'apprentissage** vise à permettre au réseau de reproduire une relation d'entrée-sortie à partir d'un ensemble de n couples d'exemples  $(X^k, Y_d^k)$ ,  $k \in [1, n]$ . Le vecteur  $X^k$  est présenté à l'entrée du réseau qui doit reproduire en sortie le vecteur  $Y_d^k$ . L'apprentissage va consister à minimiser l'écart quadratique moyen E entre la sortie désirée  $Y_d$  et la sortie réelle du réseau  $Y_r$  :

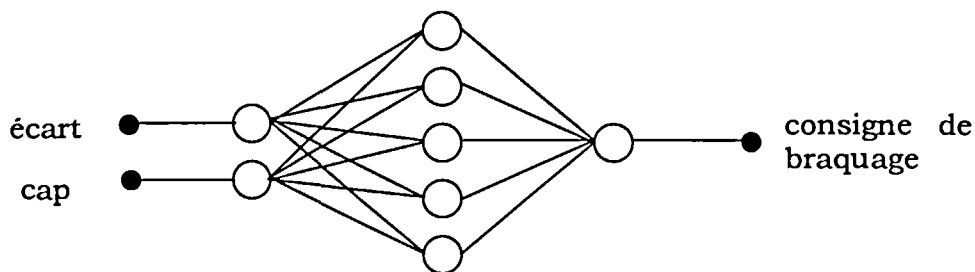
$$E = \sum_{k=1}^n E^k, \text{ avec } E^k = \sum_{i=1}^m (Y_{d,i}^k - Y_{r,i}^k)^2 \text{ (n : nombre d'exemples, m : nombre de sorties)}$$

L'erreur E est minimisée en modifiant le vecteur de poids W associé à chaque neurone.

### Application au guidage automatique d'un robot mobile agricole

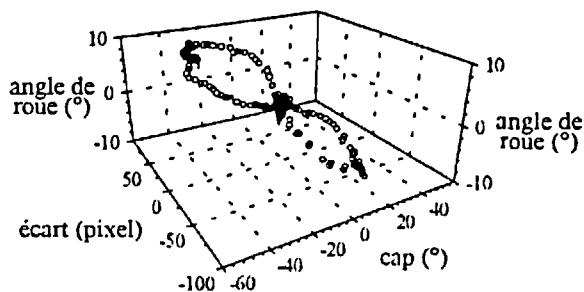
Rouveure dans [Rou94] présente une loi de commande par réseau de neurones, pour le guidage d'un robot mobile. L'application proposée vise à asservir un engin agricole sur une ligne matérialisée sur le sol. Celle-ci est détectée au moyen d'une caméra et d'un algorithme de traitement d'images utilisant des données liées au contraste. Puis un réseau de neurones calcule l'angle des roues directrices du véhicule en fonction du cap et de la position de la droite détectée dans l'image. Cette approche a l'avantage d'éviter le calibrage de la caméra ainsi qu'une modélisation cinématique du robot. Ces paramètres sont pris en compte par le réseau lors de la phase d'apprentissage qui, dans ce cas, est effectuée hors ligne.

Le réseau choisi par Rouveure, pour guider un robot mobile agricole, est un réseau multicouches de petite dimension (8 neurones). Il comporte deux entrées (écart et cap de la ligne détectée dans l'image), 1 sortie (consigne de braquage) et deux couches cachées de 2 et 5 neurones.

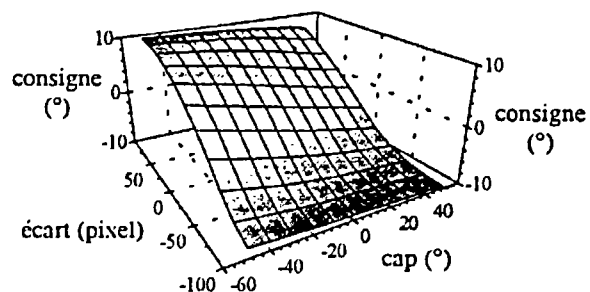


**Figure I 11 : choix de l'architecture du réseau de neurones**

L'apprentissage est réalisé à partir de l'enregistrement des entrées et des sorties du réseau lorsque la machine est conduite par un opérateur humain. Si on lui demande de parcourir dans les deux sens une trajectoire comprenant une ligne droite et un virage, on obtient la première des courbes suivantes (**Figure I 12**). A partir de ces données, le réseau interpole afin de calculer la consigne de braquage correspondant à toutes les combinaisons possibles de cap et d'écart (**Figure I 13**).



**Figure I 12 : données de l'apprentissage**



**Figure I 13 : résultat de l'apprentissage supervisé**

L'auteur mentionne de bons résultats lors du suivi d'une ligne sur terrain plat. Par contre, la méthode utilisée ne permet pas au réseau de s'adapter à des variations de modèles et de conditions de travail. Cela est dû à l'apprentissage hors ligne qui fige le réseau. Par contre, s'il était fait en ligne, il permettrait de s'adapter à des variations de la dynamique du véhicule (variation de poids, de vitesse,...) ou à des perturbations extérieures (pente du terrain, patinage des roues,...) [Nap95]. L'apprentissage constitue le principal avantage mais aussi l'inconvénient majeur de cette approche de contrôle par réseau de neurones. Il permet un étalonnage du système incluant les paramètres de la caméra et ceux de la cinématique du robot, mais il doit couvrir l'étendue des situations que la machine rencontrera. De plus, pour chaque modification de l'emplacement de la caméra (même un simple démontage) il faudra le recommencer.

### 2.2.2 Commande Référencée Vision (CRV)

Cette technique de commande date de la fin des années 80 et s'inspire des travaux effectués dans le domaine de la commande référencée capteur [Esp87] [Sam91b]. Elle vise à commander le robot dans l'espace du capteur de mesure (l'image pour une caméra) et non plus dans un espace reconstruit à partir de données mesurées. Elle est principalement basée sur la notion de fonction de tâche développée dans [Sam91b].

La commande référencée vision développée dans [Riv89] [Chau90] [Pis91] [Mot92] [Kha95b] est basée sur le même principe puisqu'elle intègre tout le formalisme de la commande référencée capteur et en particulier la notion de fonction de tâche. Elle propose d'effectuer un asservissement visuel sur un motif à atteindre dans l'image (par exemple quatre points ou une droite). Un asservissement en situation en projetant la situation de référence dans l'espace du capteur de vision est également possible si le motif est bien choisi. L'intérêt de cette technique est évident. Elle fait abstraction de toute reconstruction pouvant introduire de nombreuses erreurs ou bien nécessiter d'importants temps de calcul.

La commande référencée vision comprend plusieurs étapes :

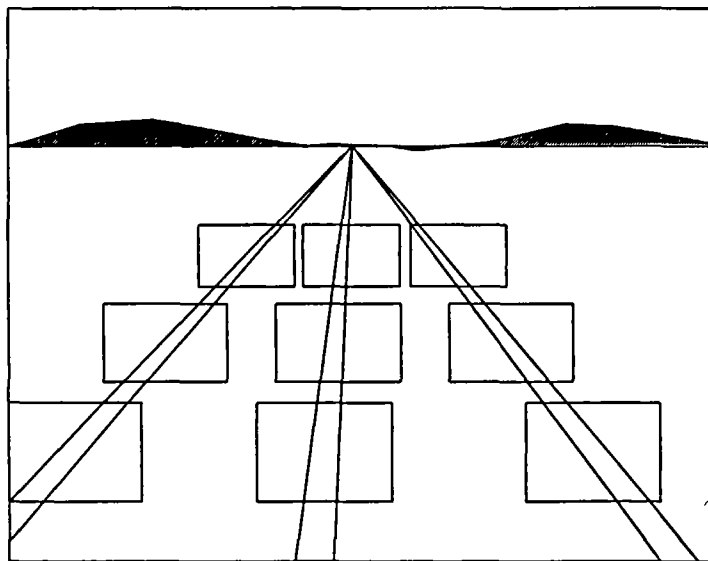
- une modélisation des liens existant entre l'image et les mouvements de la caméra,
- une étude des liaisons « virtuelles » entre la caméra et la scène (choix des primitives visuelles),
- un processus de commande qui s'inspire de la commande référencée capteur développée dans [Sam91b].

Dans le chapitre III nous développerons ces différents aspects que nous appliquerons à l'asservissement visuel d'un robot mobile.

Dans [Kha95a], Djamel Khadraoui décrit un exemple de commande référencée vision appliqué à un robot mobile agricole. L'auteur présente une modélisation de la scène utilisant comme primitive visuelle une droite dont la référence est verticale et située au centre de l'image. Il propose ensuite un modèle d'état incluant le modèle de la scène et le modèle cinématique de l'engin. Ceci revient à déterminer un modèle cinématique du robot dans l'espace des coordonnées image. Enfin, il synthétise une loi de commande à partir d'un placement de pôle assimilant le comportement de la machine à un système du second ordre. L'auteur mentionne des résultats expérimentaux effectués sur une machine agricole qui montrent la faisabilité et la robustesse d'une telle approche.

### 2.2.3 Asservissement visuel utilisant un modèle 3D de la scène

De nombreux systèmes d'aide à la conduite par traitement d'images utilisent un modèle de l'environnement dans lequel évolue le véhicule [Dick87]. Ceci est d'autant plus intéressant que l'univers considéré est structuré et que son modèle géométrique est connu. Citons les travaux du LASMEA<sup>3</sup> de Clermont-Ferrand [Chap91] où l'on utilise un modèle de route pour situer à la fois le véhicule dans l'espace mais aussi les zones d'intérêt de l'image où l'on est susceptible de trouver la signalisation horizontale (*Figure I 14*).



*Figure I 14 : principe de l'analyse par fenêtre*

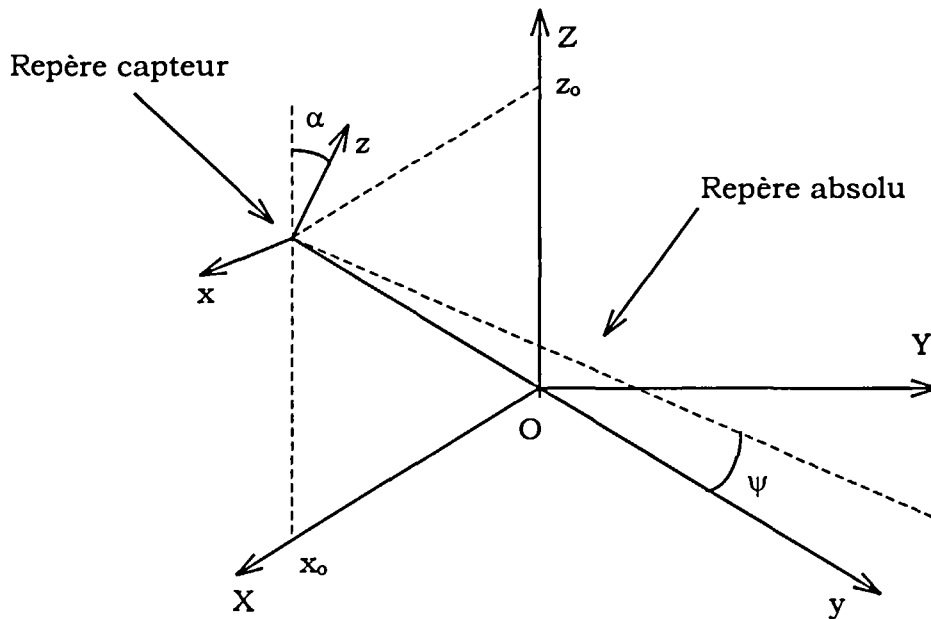
La projection de ce modèle dans l'espace image permet la réactualisation du vecteur d'état d'un filtre de Kalman qui sert à prédire la position des fenêtres. Le

<sup>3</sup> Laboratoire des Sciences et Matériaux pour l'Electronique et, d'Automatique

vecteur d'état comprend 5 paramètres traduisant la position de la caméra et la courbure de la route. A l'instant  $k$  il est donné par :

$$X(k) = (x_o(k), C(k), \psi(k), \alpha(k), z_o(k))^t$$

$C(k)$  est la courbure de la route qui est supposée plane,  $x_o$ ,  $y_o$ ,  $\alpha$ ,  $\psi$  représentent la position et l'orientation de la caméra par rapport à un repère lié à la scène.



**Figure I 15 : repère capteur et repère de la scène**

Le comportement du véhicule est décomposé en deux parties. La première concerne la cinématique du véhicule qui est approchée par un modèle bicyclette et la réaction des pneus avec le sol qui est modélisée par une accélération latérale. La seconde est liée à la direction hydraulique du véhicule (une Citroën XM) qui sera modélisée par un système du premier ordre.

Le correcteur est calculé par une méthode de placement de pôles à partir d'une équation d'état discrétisée permettant d'intégrer le modèle de la direction hydraulique. Les résultats présentés montrent un bon asservissement du véhicule jusqu'à une vitesse de 130 km/h [Jur94].

Comme dans de nombreuses applications de vision artificielle en extérieur, les prochains travaux annoncés concernent l'amélioration de la robustesse de l'algorithme de vision.

### **3. CHOIX D'UNE LOI DE COMMANDE POUR NOTRE APPLICATION**

Pour des applications agricoles, il est très difficile d'avoir un modèle représentatif du comportement cinématique et dynamique des véhicules considérés. En effet, les glissements peuvent être importants et sont dans la plupart des cas difficilement mesurables. L'évolution de l'engin sur un terrain cahoté, avec une charge et un centre de gravité variables, introduit des modifications importantes du modèle que nous avons négligées. L'utilisation des nombreux capteurs nécessaires à l'identification en temps réel du comportement des véhicules était incompatible avec le coût souhaité du système d'aide à la conduite.

De plus, de nombreuses erreurs de modélisation de la scène sont introduites par la détection de la limite de fauche. En effet, le plus souvent, nous détectons le haut de la végétation qui ne représente qu'en valeur moyenne l'emplacement de la tige à couper. Tout au long du projet, vu la précision escomptée, l'erreur introduite par la différence entre le haut et le bas de cette végétation sera négligée. Cependant, il faut noter que, pour certaines positions de la caméra et pour des végétations à hauteur importante tel que le blé, cette approximation peut aboutir à des erreurs grossières auxquelles pour l'instant nous ne nous sommes pas intéressés.

Les lois de commandes utilisées devront donc être le plus robustes possible vis-à-vis des erreurs de modélisation et des perturbations nombreuses des conditions de roulement.

Dans un premier temps, des lois de commande relativement simples basées sur un modèle cinématique simplifié de nos robots ont été retenues. Nous en présenterons deux qui seront testées sur deux engins différents. La première nécessite une reconstruction de la scène et s'inspire des travaux réalisés dans le cadre du projet (BART). La seconde loi de commande basée sur le formalisme de la commande référencée capteur a été développée et testée. Celle-ci ne nécessite pas de reconstruction de la scène puisque l'asservissement est réalisé dans l'espace du capteur. Dans les deux cas, compte tenu des faibles vitesses des robots (inférieures à 10 km/h), nous n'avons pas utilisé de modèles dynamiques de ceux-ci mais seulement des modèles cinématiques s'appuyant sur les caractéristiques géométriques des deux robots principalement considérés. Ces deux lois de commande donneront une image des différences existant entre un asservissement incluant une reconstruction de la situation du véhicule et un asservissement basé uniquement sur le type de primitive détectée par le capteur introduit directement dans la boucle de contrôle. Dans ce second cas, les erreurs de modélisation de la scène seront compensées par un apprentissage de la primitive de référence (cf. chapitre III).



## CHAPITRE II

### MODELISATION DES VEHICULES UTILISES

## INTRODUCTION

Les robots mobiles à roues sont, pour la plupart, sujets à des équations de contraintes non-holonomes [Sam93]. En effet, le plus souvent, le nombre de variables nécessaires pour caractériser l'état du robot est supérieur au nombre de ses degrés de commande. Donnons comme exemple la voiture caractérisée par trois variables (deux variables de position et une d'orientation) mais qui n'a que deux degrés de commande (une translation et une rotation d'axe verticale). A l'inverse, le chariot de supermarché, dont l'état est toujours caractérisé par les trois mêmes variables que la voiture, présente trois degrés de commande (deux translations et une rotation).

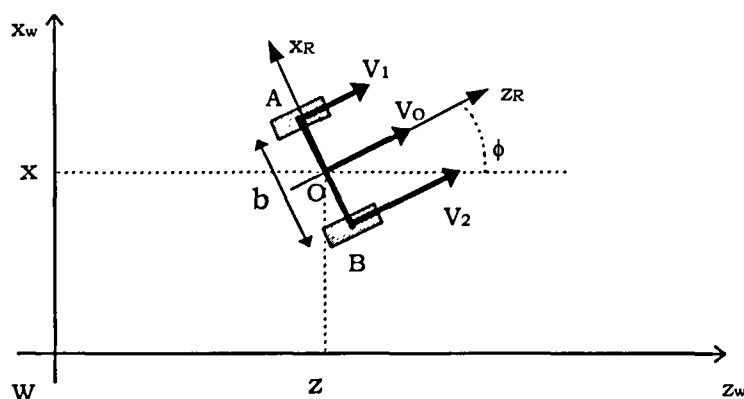
Le chariot ne présente pas de contraintes non-holonomes et on peut lui appliquer facilement les résultats obtenus pour les robots manipulateurs, alors que les véhicules de type voiture ou chenillard ont des contraintes non-holonomes dont il faudra tenir compte dans nos lois de commande.

Citons quelques travaux sur des robots mobiles omnidirectionnels, tels que ceux menés sur Uranus [Cox90] ou ceux réalisés au LIRMM [Bet93], qui ont l'avantage de posséder les trois degrés de liberté laissés par la liaison de type plan.

Dans la suite de ce travail, nous ne nous intéresserons qu'aux robots mobiles à roues caractérisés par trois variables d'état mais ayant deux degrés de liberté commandables. Ce cas de figure correspond au comportement des deux véhicules que nous considérerons. Afin de calculer un modèle cinématique de ceux-ci, nous donnerons les équations cinématiques d'un essieu à deux roues rigides puis nous les généraliserons à nos engins dont la mobilité dépend explicitement ou implicitement du comportement d'un essieu du même type. Enfin, nous tenterons de relier ces déplacements aux commandes des actionneurs des véhicules considérés.

## 1. MODELES DE ROBOTS MOBILES

Considérons un essieu rigide composé de deux roues ayant pour seul degré de liberté par rapport à l'essieu une rotation d'axe confondu avec l'essieu (**Figure II 1**). Soit A le centre de la roue gauche et B le centre de la roue droite. Notons  $V_1$  et  $V_2$  les vitesses du centre des roues gauche et droite.

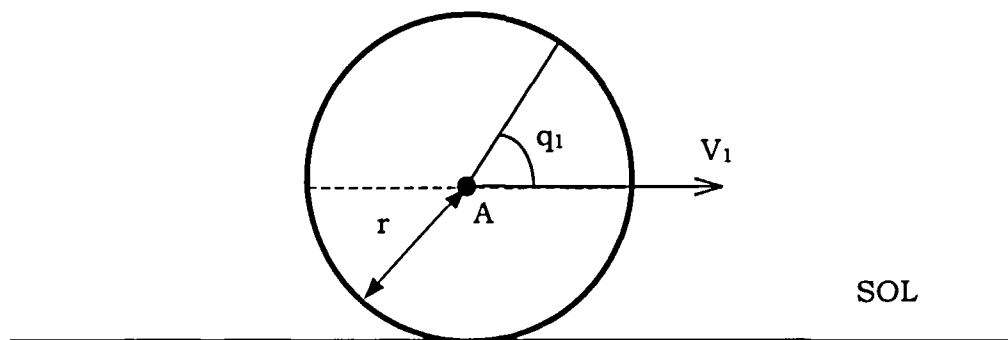


**Figure II 1 : cinématique de l'essieu rigide à deux roues**

Soit le point O situé au centre de l'essieu. Supposons que les hypothèses suivantes sont réalisées :

- l'essieu est supposé rigide et indéformable,
- il se déplace sur un sol plan,
- il y a roulement sans glissement des roues sur le sol.

Les positions angulaires des roues gauche et droite seront données par  $q_1$  et  $q_2$  et leur rayon par  $r$ . Nous noterons par  $b$  la distance entre les deux roues (**Figure II 1**)(**Figure II 2**).



**Figure II 2 : caractéristiques de la roue gauche**

Les déplacements du robot mobile sont régis par l'équation d'état donnée dans le repère cartésien  $(W, x, z)$  [Lee93] [Guld94] [Tou88b] :

$$\text{eq II 1} \quad \begin{cases} \dot{z} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \cos \phi \\ \dot{x} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \sin \phi \\ \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \end{cases}$$

Les deux premières équations ne sont pas complètement intégrables, elles contiennent une contrainte non-holonyme donnée par :

$$dx = \tan \theta \cdot dz$$

La troisième est holonome et peut être intégrée :

$$\text{eq II 2} \quad \phi = \frac{r}{b} \cdot (q_2 - q_1) + \text{cte}$$

L'équation (eq II 1) peut également s'écrire :

$$\text{eq II 3} \quad \begin{cases} v = V(O) = \frac{V_1 + V_2}{2} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \\ \dot{\phi} = \frac{V_2 - V_1}{b} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \end{cases}$$

$V(O)$  est la vitesse du centre de l'essieu. L'hypothèse roulement sans glissement implique que  $V(O)$ ,  $V_1$ ,  $V_2$  soient perpendiculaires à l'axe de l'essieu.

**L'équation (eq II 1) montre que si la vitesse moyenne du centre de l'essieu est constante, sa trajectoire ne variera qu'en fonction de son orientation  $\phi$ .** C'est-à-dire, qu'à vitesse moyenne constante, le contrôle de la trajectoire de l'essieu ne se fait que par le contrôle de son orientation. Ce fait est très important et sera largement exploité par nos lois de commande décrites au chapitre III.

Des équations (eq II 3) nous pouvons déduire que lorsque les vitesses  $V_1$  et  $V_2$  sont constantes, le point O se déplace sur un cercle de rayon R tel que [Ioo88] [Tou88b] :

$$\text{eq II 4} \quad R = \frac{v}{\dot{\phi}} = \frac{b}{2} \cdot \frac{(V_2 + V_1)}{(V_2 - V_1)} = \frac{b}{2} \cdot \frac{(\dot{q}_2 + \dot{q}_1)}{(\dot{q}_2 - \dot{q}_1)}$$

De même, tout point M de l'essieu se déplace sur un cercle dont le rayon dépend de la distance du point M au point O, et de R. Sa vitesse est alors donnée par :

$$V(M) = V(O) + \begin{pmatrix} 0 \\ \dot{\phi} \\ 0 \end{pmatrix} \wedge \vec{OM}$$

Pour caractériser la cinématique de l'essieu, nous retiendrons les trois équations donnant la vitesse moyenne et la vitesse angulaire de l'essieu ainsi que le rayon de courbure de la trajectoire que décrit son centre :

$$\text{eq II 5} \quad \begin{cases} v = \frac{r \cdot (\dot{q}_1 + \dot{q}_2)}{2} \\ \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \\ R = \frac{b}{2} \cdot \frac{(\dot{q}_1 + \dot{q}_2)}{(\dot{q}_2 - \dot{q}_1)} \end{cases}$$

Les deux plates-formes expérimentales considérées sont très différentes. La première est une tondeuse professionnelle à quatre roues motrices ; sa direction est assurée grâce à une différence de vitesse entre les roues droite et gauche. La seconde est une moissonneuse-batteuse de 15 tonnes ayant deux roues motrices à l'avant et deux roues directrices à l'arrière. Son modèle cinématique est comparable à celui d'une voiture.

Ces deux engins ont deux points communs :

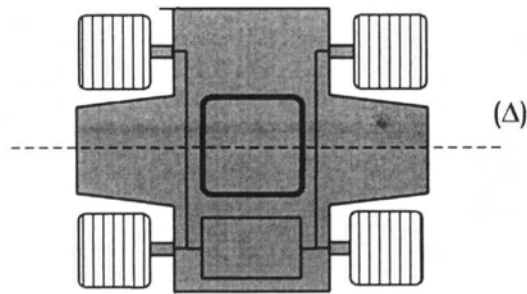
- il est possible sur les deux machines de commander la vitesse moyenne d'avancement ( $v$ ),
- pour une valeur du vecteur de commande constante la trajectoire suivie par chaque point des deux machines est un cercle.

La principale différence entre les deux véhicules est le principe de direction. Pour la tondeuse, il est basé sur une différence de vitesse alors que pour la moissonneuse-batteuse il utilise l'angle des roues arrière.

Du point de vue de la modélisation, notre approche consiste à utiliser les résultats obtenus sur l'essieu, puis à déterminer un lien entre les actionneurs de nos machines et l'essieu rigide réel ou virtuel qu'elles possèdent.

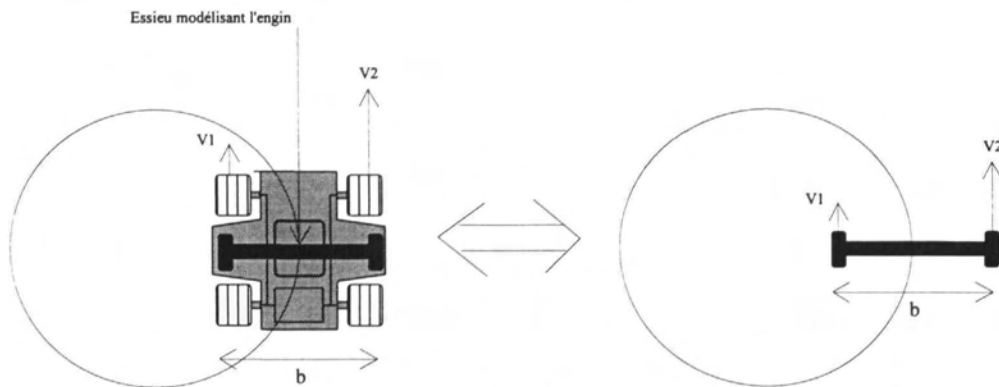
## 2. MODELISATION DU ROBOT DE TONTE

Cette plate-forme mobile (**Figure II 3**) comporte un avancement hydrostatique. Deux circuits hydrauliques indépendants contrôlent la vitesse des roues droite et gauche. La commande de la machine se fait par deux actionneurs. Le premier donne la valeur moyenne de débit des deux circuits, le second introduit une différence entre les circuits gauche et droit tout en conservant la valeur moyenne désirée. Ceci permet d'avoir un contrôle indépendant sur la vitesse moyenne de la machine et sur sa direction.



**Figure II 3 : Vue de dessus du robot de tonte**

De par sa géométrie et de par son mode de fonctionnement, ce véhicule est symétrique par rapport à l'axe ( $\Delta$ ) (**Figure II 3**). Son comportement peut être modélisé par celui d'un essieu rigide dont le centre serait confondu avec le centre de la machine (**Figure II 4**) [Deb95a].



**Figure II 4 : Modélisation de la tondeuse**

La vitesse des deux roues gauche de la machine sera identique à celle de la roue gauche de l'essieu équivalent ; il en est de même pour le côté droit.

D'après les équations cinématiques de l'essieu, nous avons :

$$\text{eq II 6} \quad \begin{cases} \dot{z} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \cos \phi \\ \dot{x} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \sin \phi \\ \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \end{cases}$$

Or, sur la machine considérée, les deux variables de commande sont :

$$\text{eq II 7} \quad \begin{cases} v = \frac{V_1 + V_2}{2} \\ \Delta v = V_2 - V_1 \end{cases} \quad \text{avec : } \begin{cases} V_1 = \dot{q}_1 \cdot r \\ V_2 = \dot{q}_2 \cdot r \end{cases}$$

En reportant (eq II 7) dans (eq II 6), on obtient sous forme matricielle :

$$\text{eq II 8} \quad \begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \frac{r}{2} \cdot \sin \phi & \frac{r}{2} \cdot \sin \phi \\ \frac{r}{2} \cdot \cos \phi & \frac{r}{2} \cdot \cos \phi \\ -\frac{r}{b} & \frac{r}{b} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{r} & \frac{-1}{2r} \\ \frac{1}{r} & \frac{1}{2r} \end{pmatrix} \begin{pmatrix} v \\ \Delta v \end{pmatrix}$$

avec :

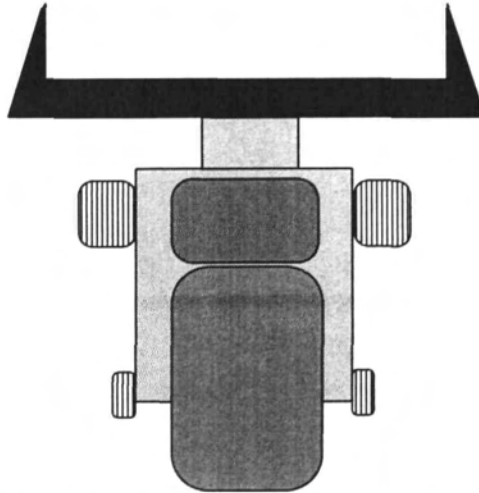
$$\text{eq II 9} \quad \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = A \cdot \begin{pmatrix} v \\ \Delta v \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & \frac{-1}{2r} \\ \frac{1}{r} & \frac{1}{2r} \end{pmatrix} \cdot \begin{pmatrix} v \\ \Delta v \end{pmatrix}$$

L'équation (eq II 8) permet d'exprimer les contraintes cinématiques de l'essieu au point O en fonction des deux actionneurs du robot qui sont v (vitesse moyenne) et  $\Delta v$  (différence de vitesse entre les roues gauche et droite).

La matrice A permet d'exprimer les vitesses angulaires des roues de l'essieu considéré en fonction des actionneurs de la machine. Nous utiliserons ce principe tout au long de notre étude. Nous montrerons ainsi que de nombreux robots mobiles peuvent être modélisés au moyen d'un essieu rigide dont les équations cinématiques sont bien connues et d'une matrice « A » que nous appellerons matrice Jacobienne, reliant les vitesses angulaires des roues de l'essieu rigide à la commande des actionneurs du véhicule.

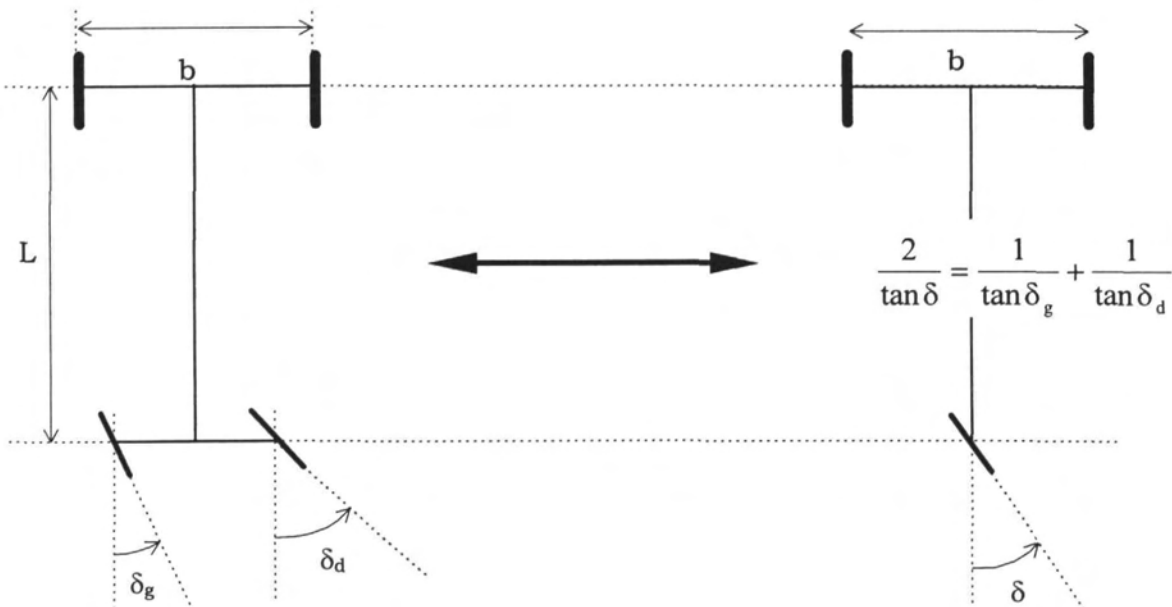
### 3. MODELISATION DE LA MOISSONNEUSE-BATTEUSE [DEB95B]

La deuxième plate-forme que nous utiliserons lors de nos expérimentations est une moissonneuse-batteuse dont l'avancement et la direction sont hydrostatiques (*Figure II 5*).



*Figure II 5 : vue de dessus de la moissonneuse-batteuse*

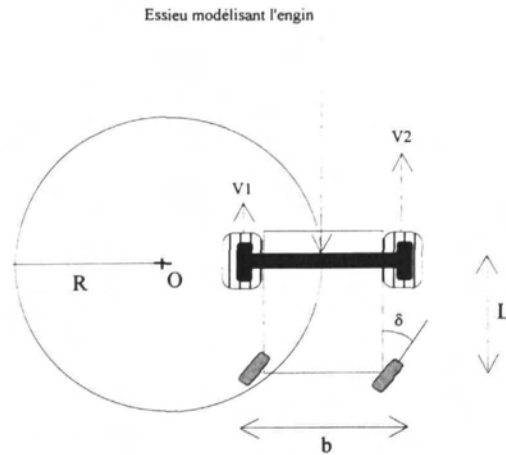
Le modèle cinématique d'un tel engin est tout à fait comparable à celui d'une voiture [Far93]. Nous ferons l'approximation qui consiste à remplacer les deux roues directrices par une seule placée au milieu de l'essieu et dont l'angle de braquage est donné par la relation de la figure suivante (*Figure II 6*).



*Figure II 6 : Approximation du véhicule à 4 roues par un tricycle*



De même que pour la tondeuse, considérons un essieu rigide de la machine. Dans ce cas, il est réel puisque nous prendrons l'essieu moteur qui se trouve à l'avant de l'engin (**Figure II 7**).



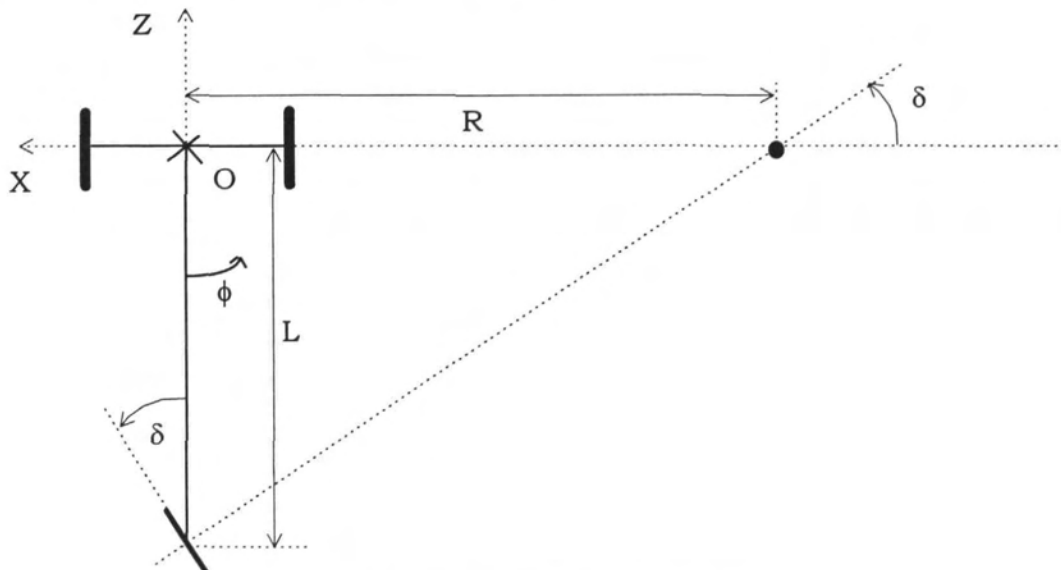
**Figure II 7 : Trajectoire du véhicule à 2 roues directrices**

Cet essieu a les mêmes contraintes cinématiques que celui décrit au premier paragraphe :

eq II 10

$$\begin{cases} \dot{z} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \cos \phi \\ \dot{x} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \sin \phi \\ \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \end{cases}$$

Déterminons la matrice A qui permet de lier les vitesses angulaires des roues de l'essieu rigide aux actionneurs du véhicule.



**Figure II 8 : Géométrie du tricycle**

Donnons la relation trigonométrique reliant R (rayon de courbure de la trajectoire de O) et  $\delta$  l'angle des roues directrices (**Figure II 8**) :

$$\text{eq II 11} \quad \tan \delta = -\frac{L}{R}$$

Le deuxième actionneur est la commande de la vitesse moyenne d'avancement. Elle est appliquée à l'essieu moteur, c'est-à-dire à l'essieu rigide considéré pour calculer le modèle cinématique. Elle est donnée par :

$$\text{eq II 12} \quad v = \frac{V_1 + V_2}{2}$$

En reportant (eq II 12) et (eq II 11) dans (eq II 10), nous obtenons :

$$\text{eq II 13} \quad \begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \frac{r}{2} \cdot \sin \phi & \frac{r}{2} \cdot \sin \phi \\ \frac{r}{2} \cdot \cos \phi & \frac{r}{2} \cdot \cos \phi \\ -\frac{r}{b} & \frac{r}{b} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{r} & \frac{b}{2Lr} \\ \frac{1}{r} & \frac{-b}{2Lr} \end{pmatrix} \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

avec :

$$\text{eq II 14} \quad \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = A \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & \frac{b}{2Lr} \\ \frac{1}{r} & \frac{-b}{2Lr} \end{pmatrix} \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

L'équation (eq II 13) donne le modèle cinématique du véhicule en fonction de ses actionneurs ( $\delta$  et  $v$ ). Il est intéressant de noter que l'on retrouve l'équation cinématique de l'essieu rigide donnée en première partie, associée à la matrice A et aux actionneurs du véhicule.

### Conclusion

La modélisation cinématique que nous avons choisie consiste à séparer le comportement de l'engin de celui de ses actionneurs. Chaque modèle comporte deux parties bien distinctes. La première, qui est commune aux deux véhicules, est représentée par les équations de mouvement du classique essieu à deux roues au moyen des variables ( $x, z, \theta$ ). La deuxième, que nous appellerons matrice jacobienne A, est propre au véhicule et devra être calculée pour chaque robot mobile. Cette matrice permet d'établir une équation entre la vitesse angulaire des roues d'un essieu rigide du véhicule et le contrôle de ses actionneurs.

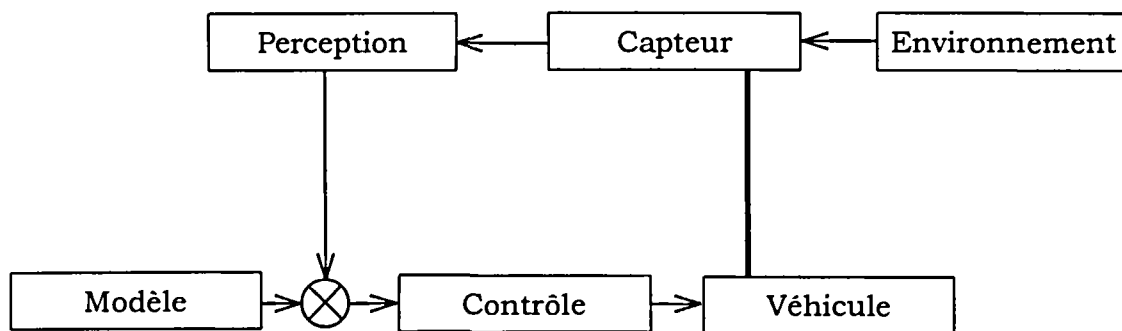
Dans la suite de notre étude, nous présenterons des lois de commande calculées pour l'essieu rigide. L'intérêt de l'approche cinématique présentée est de pouvoir appliquer facilement les résultats que nous obtiendrons à différents types de robots mobiles.

## CHAPITRE III

# ASSERVISSEMENT D'UN ROBOT MOBILE A PARTIR D'INFORMATIONS VISUELLES

## INTRODUCTION

L'aide au guidage d'engins agricoles que nous présentons pose deux problèmes majeurs. Le premier est la perception de l'environnement, le second est le contrôle des véhicules par rapport aux informations perçues. L'énoncé de ces deux difficultés montre bien le lien qui unit la seconde partie à la première. Nous ajouterons une boucle de retour permettant de corriger les différentes erreurs et dérives du système dues à des modélisations simplifiées et à des données imprécises. Le schéma général de l'application est donc :



**Figure III 1 : schéma général de l'asservissement**

Dans la suite de ce chapitre nous donnons quelques éléments du module de perception basé sur un capteur caméra et un algorithme de segmentation d'images non supervisée. Puis, nous détaillons les deux lois de commandes que nous avons utilisées pour contrôler nos robots :

- ♦ La première construit une trajectoire permettant à l'engin de passer de sa position courante à une position désirée.
- ♦ La seconde est basée sur la théorie de la commande référencée capteur et proposera deux fonctions de tâche adaptées à nos applications.

## 1. LE TRAITEMENT D'IMAGES

Le capteur de perception retenu pour nos applications est une caméra CCD dont l'ouverture de l'objectif et le gain interne sont asservis sur la luminosité. L'image obtenue comporte principalement deux régions : une zone de végétation fauchée et une zone de végétation non-fauchée. Le but de l'algorithme développé au Cemagref par M. Derras [Der94] est d'extraire les coordonnées de la primitive symbolisant la limite entre la zone fauchée et la zone non-fauchée. Celle-ci sera modélisée par une droite dont on donnera la pente et l'ordonnée à l'origine.

### 1.1 Attributs de l'image

Avant de segmenter l'image en régions homogènes M. Derras la découpe en sites de  $16 \times 16$  pixels appelés « texels ». Pour chacun de ces sites, deux paramètres liés aux histogrammes locaux sont calculés :

- le premier est le maximum de l'histogramme qui traduit la nuance de gris prédominante dans le texel,
- le second est le moment d'ordre 2 qui rend compte de l'amplitude de cette prédominance.

Puis, il utilise les matrices de coocurrences [Har79][Chen79] qui retranscrivent la distribution spatiale des variations de niveaux de gris entre pixels voisins. A partir de ces matrices, il extrait deux paramètres parmi les 14 facteurs proposés par Haralick [Har73] :

- le premier est le coefficient d'homogénéité (H) qui donne une idée de l'état de surface de la zone étudiée,
- le deuxième est l'entropie (E) qui reflète une notion d'ordre.

Pour chaque texel, on obtient donc quatre descripteurs qui vont permettre d'effectuer une classification des 1024 sites de l'image.

## 1.2 Segmentation de l'image

A partir de l'image vectorielle obtenue précédemment, M. Derras a développé une méthode de segmentation en région. La multiplicité des situations rencontrées dans le monde agricole rend difficile l'utilisation d'hypothèse a priori sur le résultat de la segmentation. Aussi, s'est-on orienté vers une méthode de segmentation non supervisée basée sur la théorie des champs de Markov [Che89][Su190]. Cette méthode présente l'intérêt d'être bien adaptée au traitement de séquences d'images puisque le traitement de l'image n+1 intègre les résultats de l'image n. Ceci est essentiel lorsqu'on traite des séquences d'images avec, à terme, un objectif « temps réel ».

Toute la difficulté réside dans la segmentation de l'image (étiquetage des régions distinctes) à partir de la seule connaissance de ses divers descripteurs [Der87] [Gen90]. Traduit sous forme mathématique, le problème consiste à maximiser la probabilité a posteriori  $P(X/Y)$ , c'est-à-dire la probabilité de l'objet cherché conditionnellement aux mesures effectuées, avec :

- X résultat de la segmentation (champ des étiquettes),
- Y descripteur de l'image.

En appliquant le théorème de Bayes, cette probabilité a posteriori peut s'exprimer sous la forme :

$$\text{eq III 1} \quad P(X/Y) = \frac{P(X) \cdot P(Y/X)}{P(Y)}$$

avec :

- $P(X)$  probabilité a priori sur la connaissance du champ des étiquettes,
- $P(Y/X)$  probabilité conditionnelle de la mesure par rapport à l'une des segmentations possibles. Cette dernière décrit entièrement le processus de transposition "données-étiquettes",
- $P(Y)=1$  (les données sont connues).

L'image se présente donc sous la forme d'un champ de données (4 paramètres par région élémentaire 16.16 pixels) que l'on cherche à modéliser par un champ des étiquettes qui, à terme, sera la segmentation de l'image. En s'appuyant sur la théorie des champs de Markov, on limite les effets de chaque élément sur l'ensemble (effet global) à une interaction locale entre sites voisins. Ces sites voisins sont reliés entre eux par des "cliques" d'ordres divers qui définissent la forme et la taille des systèmes de voisinages considérés.

On cherche ensuite à exprimer sous une autre forme cette première équation de Bayes. Pour cela, le théorème de Hammersley-Clifford permet de faire intervenir les distributions de Gibbs qui s'expriment de la manière suivante :

$$P(\omega) = \left(\frac{1}{Z}\right) \cdot \exp\{-U(\omega)\} \quad \text{avec } \omega \in \Omega$$

avec :

- Z constante de normalisation,
- U( $\omega$ ) fonction d'énergie,
- $\omega$  état particulier du système,
- $\Omega$  ensemble des états possibles.

Par conséquent, chacun des termes de l'équation (eq III 1) peut s'exprimer sous la forme d'une distribution de Gibbs. Donc :

$$P(X/Y) = P(X) \cdot P(Y/X) \quad \text{devient}$$

$$\left(\frac{1}{Z}\right) \cdot \exp\{-U(X/Y)\} = \left(\frac{1}{Z}\right) \cdot \exp\{-U(X)\} \cdot \left(\frac{1}{Z}\right) \cdot \exp\{-U(Y/X)\}$$

En utilisant les logarithmes népériens, l'expression se réduit alors à :

$$U(X/Y) = U(X) + U(Y/X)$$

L'objectif est de réaliser le calcul des deux termes de la fonction pour minimiser l'énergie  $U(X/Y)$  avec :

- U(X) contraintes liées au champ des étiquettes (information a priori),
- U(Y/X) fonction d'énergie liée aux interactions "données-étiquettes".



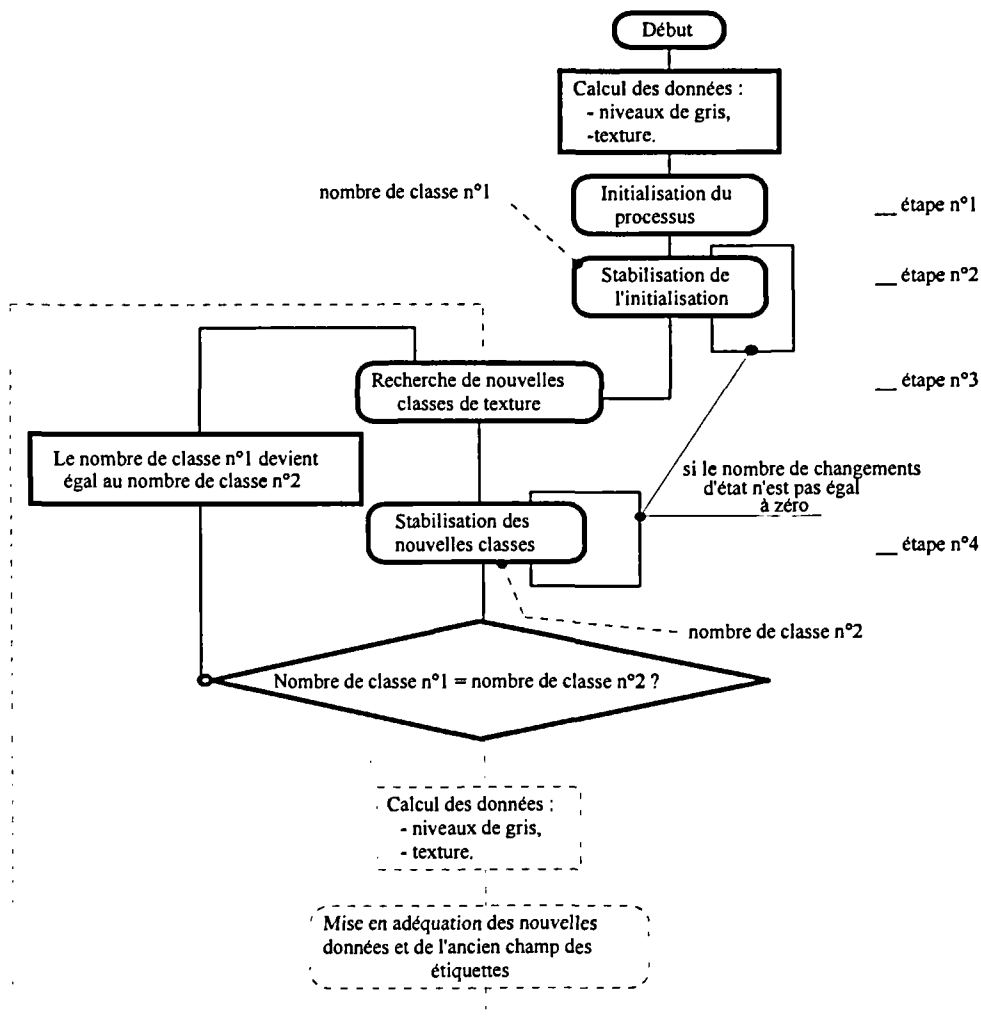
### 1.3 Application : recherche de la limite de fauche

Pour mettre en oeuvre cette méthode de segmentation de l'image en utilisant la modélisation Markovienne, M. Derras se sert de la théorie du « recuit simulé » et plus particulièrement de l'algorithme déterministe I.C.M. (Iterated Conditional Mode). Pour chaque site, les différentes valeurs possibles de la fonction d'énergie sont calculées et seul l'état correspondant à **l'énergie minimale** est retenu.

Les deux termes de la fonction d'énergie a posteriori  $U(X/Y)$  seront donnés dans la pratique par :

- $U(X)$  fonction dite de "pénalité" liée au voisinage du site étudié,
- $U(Y/X)$  distance du QUI-2 entre chaque site et une référence générale dans l'image.

Le processus se décompose alors en quatre grandes étapes résumées par la figure suivante :



**Figure III 2 : description du processus de segmentation**

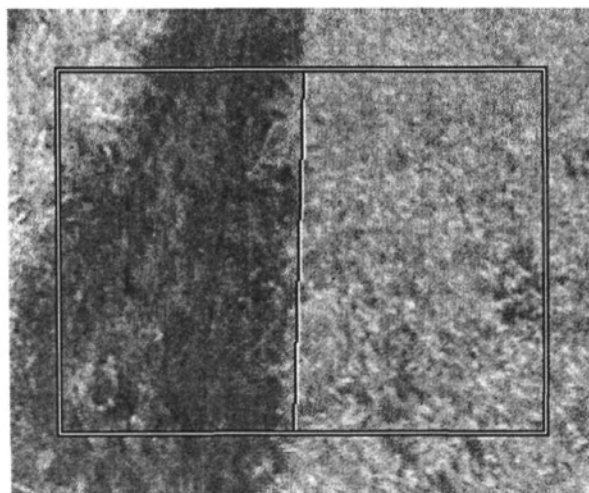
La figure (**Figure III 2**) montre les différentes étapes du processus de segmentation Markovien. L'étape n°1, correspondant à l'initialisation du processus, démarre sans aucune connaissance sur la forme ou le nombre de textures dans l'image. Par conséquent, l'auteur forme l'hypothèse que l'image est constituée d'une seule région. Le champ des étiquettes est donc initialisé à 0 et un premier découpage est réalisé à partir de la comparaison de l'énergie locale de chaque site avec la valeur de l'énergie globale moyenne calculée sur la totalité de l'image. A l'issue de l'initialisation, on obtient une classe dans le cas d'une image homogène, deux classes dans les autres cas.

L'étape n°2 sert à mettre de "l'ordre" dans le champ des étiquettes. Les fonctions d'énergie relatives aux différentes étiquettes sont calculées et l'état de chaque site étudié est alors modifié en fonction de la région fournissant la valeur d'énergie la plus faible. Ce procédé est réitéré au cours de plusieurs balayages de l'image jusqu'à ce qu'il n'y ait plus de changement dans le champ des étiquettes.

Les étapes n°3 et 4 sont, en fait, des généralisations des deux premières étapes à l'ensemble des classes détectées. Le processus s'arrête lorsque le nombre de classes ne change plus.

Les champs de Markov trouvent leur pleine justification dans le cadre d'une utilisation en dynamique. L'idée consiste à réutiliser, comme information a priori, le champ des étiquettes issu de la segmentation de l'image n pour traiter l'image suivante n+1. Ceci a pour principal objectif de limiter les temps de calcul à une simple réactualisation du champ des étiquettes. Dans notre cas, seule la première image d'une séquence est traitée entièrement. Ensuite, toujours dans le but d'obtenir un gain supplémentaire en temps de calcul, la segmentation est localisée sur une zone d'intérêt (quart de l'image) suffisante pour assurer le guidage futur de l'engin.

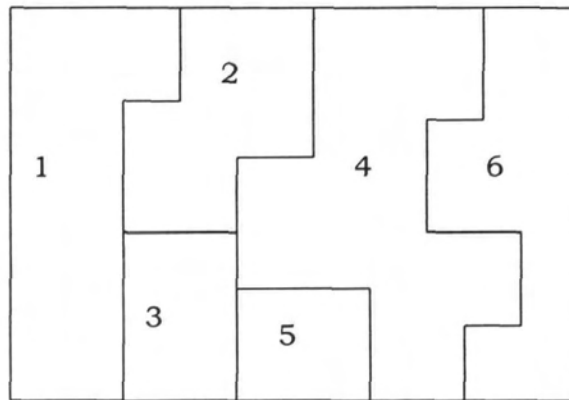
La figure suivante montre un exemple de résultat de détection d'une limite de fauche. On observe un positionnement correct de l'interface dont les coordonnées sont envoyées au programme de commande.



**Figure III 3 : résultat du traitement d'une image naturelle**

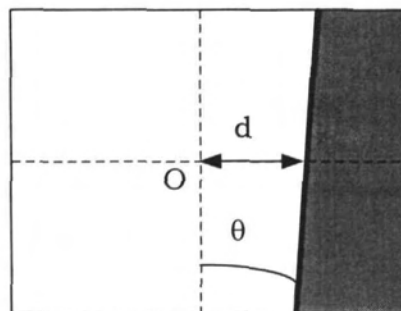
Remarques

- 1) Dans la majorité des cas, la segmentation de l'image donne un nombre de régions supérieur à 2 (**Figure III 4**). Cela peut être dû à des zones d'ombre ou à des variations du sol et de la végétation. Aussi, obtient-on une image avec de nombreuses frontières parmi lesquelles il faut distinguer l'interface fauchée/non-fauchée. Aujourd'hui, nous employons des critères géométriques (on sélectionne l'interface la plus proche de la précédente) pour choisir la frontière symbolisant le front de coupe. Cette technique introduit de nombreuses erreurs car elle ne tient compte ni des mouvements des véhicules ni des irrégularités de la végétation. Afin de rendre plus robuste ce choix, nous envisageons de fusionner le résultat de la segmentation Markovienne avec d'autres algorithmes de traitement d'images [Der95] ou bien d'utiliser un deuxième capteur dont la fonction serait de lever le doute sur le choix de la « bonne frontière ».



**Figure III 4 : résultat d'une segmentation donnant plus de deux régions**

- 2) Les coordonnées de la droite modélisant l'interface fauchée/non-fauchée seront représentées par l'angle de la pente de la droite par rapport à la verticale ( $\theta$ ) et par son ordonnée à l'origine ( $d$ ) (**Figure III 5**).

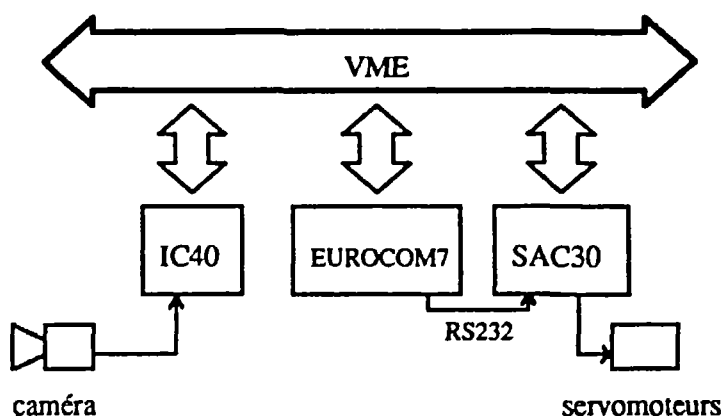


**Figure III 5 : coordonnées  $(d, \theta)$  de la droite**

## 1.4 Implantation « temps réel » [Der94]

L'algorithme de traitement d'images présenté a été développé sur une station SUN SPARC1e comportant des cartes de traitement d'images de la société IMAGING TECHNOLOGY, le tout intégré dans un premier rack VME. Le temps de traitement sur cette architecture est de l'ordre de 500 ms. Cependant, ce système n'étant pas embarquable sur nos véhicules, il nous fallait une architecture matérielle capable d'accueillir notre programme et susceptible d'être installée directement sur la machine. Les conclusions d'une étude menée en 1992 [Boi92] nous ont conduits à choisir un matériel proposé par la société ELTEC, fonctionnant sous contrôle du système d'exploitation **OS9**. L'utilisation d'une seule carte s'avérant insuffisante quant au temps de calcul (plus d'une seconde), une solution multiprocesseurs a été adoptée.

Les véhicules utilisés dans nos applications évoluent à une vitesse de 3 à 6 km/h. Ceci représente plus d'un mètre parcouru à la seconde. Avec des temps de calcul de l'ordre de la seconde la période d'échantillonnage était trop importante. C'est pourquoi il a été décidé d'implanter l'algorithme sur une **architecture pipeline** comportant trois cartes de la société ELTEC (IC40, Eurocom7 et SAC30) reliées physiquement par un **bus VME**. Les deux premières cartes sont dédiées au traitement d'images, la troisième est réservée à la commande du robot. Des drivers de communication appropriés ont permis d'établir les liens d'échange d'informations entre les différentes cartes via le bus fond de panier ou via une liaison série en face avant des cartes comme le montre la figure suivante :



*Figure III 6 : Architecture parallèle de la tondeuse*

En ce qui concerne les détails de la « parallélisation », nous renvoyons aux travaux de M. Derras dans [Der94]. Cependant, nous retiendrons que le schéma de « parallélisation » choisi est du type pipeline avec une double synchronisation.

L'organisation est la suivante :

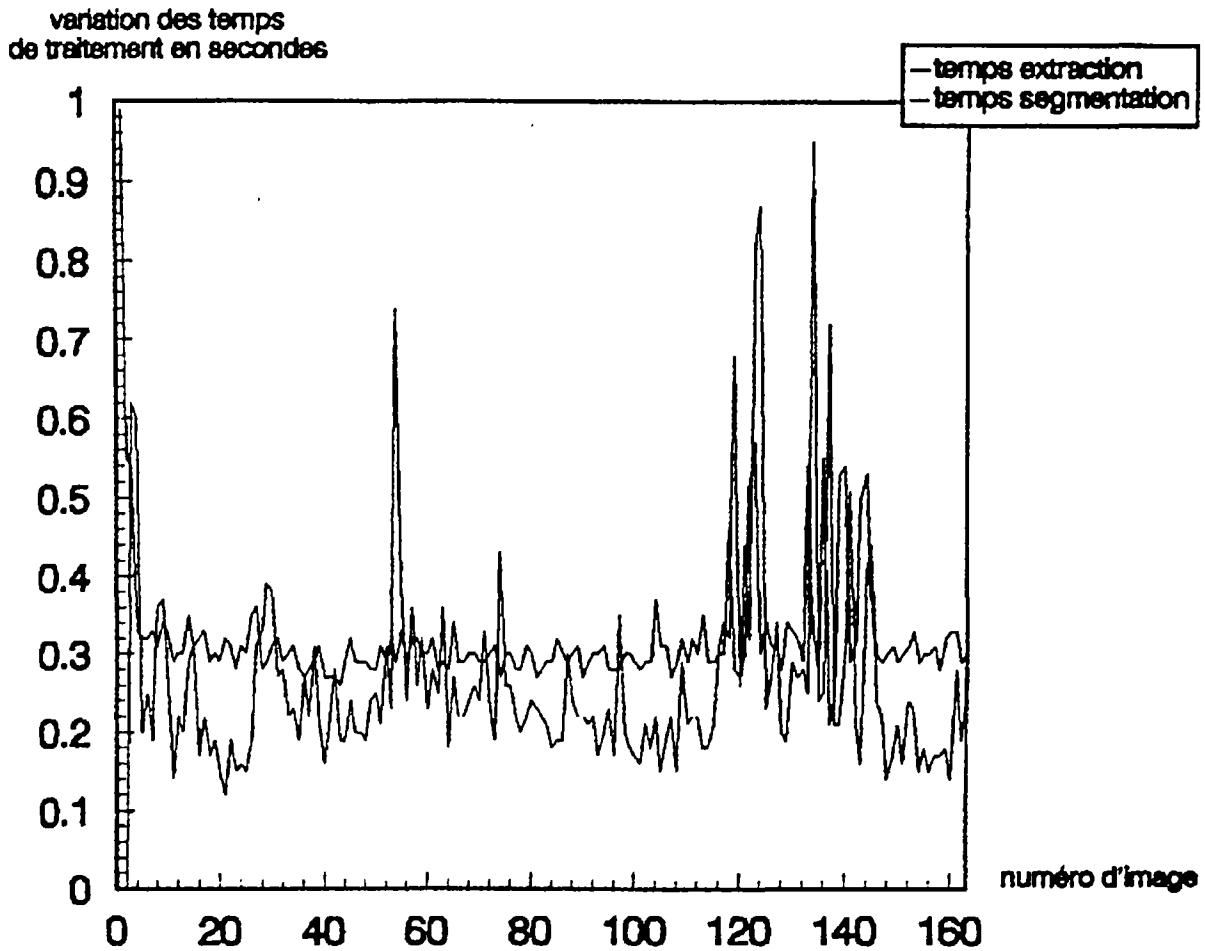
- la première carte (IC40) permet la digitalisation de l'image et calcule les différents attributs de texture et de niveau de gris,
- par l'intermédiaire d'une zone de mémoire partagée, elle transmet la pseudo-image vectorielle (une image ayant ses sites paramétrés par 4 descripteurs) à la seconde carte (Eurocom7).
- l'algorithme de segmentation Markovienne est entièrement implanté sur la deuxième carte. Il définit les différentes régions de l'image en fonction des descripteurs qu'il a reçus. Puis, il envoie les coordonnées de la limite de fauche à la troisième carte (SAC30). Cette dernière a une double tâche :
  - elle assure un contrôle bas niveau de l'engin au moyen d'un correcteur PI numérique à gains adaptatifs,
  - en parallèle, elle calcule la commande à partir des informations visuelles.

### **Conséquences :**

Grâce à cette architecture, « le temps de calcul » pour une image est de l'ordre de 300 ms. En fait, le rafraîchissement des informations visuelles se fait toutes les 300 ms, mais il s'écoule 600 ms entre l'acquisition de l'image et la fin de son traitement. Ceci est dû à l'architecture pipeline qui introduit un retard pur égal à la somme des temps de calcul de chaque carte. La période d'échantillonnage de notre application sera donc de 300 ms avec un retard pur de 600 ms.

### **Remarques :**

- L'architecture présentée n'est pas tout à fait identique sur les deux véhicules. Pour la moissonneuse-batteuse la carte SAC30 est remplacée par un PC. Les données visuelles lui sont transmises via une liaison série (RS232). Il assure le contrôle bas niveau de l'angle des roues directrices et calcule la loi de commande permettant d'asservir l'engin.
- **L'algorithme de segmentation utilise la théorie des champs de Markov. De par son principe, il n'est pas déterministe en ce qui concerne les temps de calcul. En effet, sur la figure (*Figure III 7*) nous constatons que ceux-ci peuvent varier de 300ms à une seconde.**



*Figure III 7 : séquence de fauche d'herbe*

### Conclusion

Nous avons vu que la perception des informations visuelles se faisait grâce à une caméra CCD et à un algorithme de segmentation d'images en régions utilisant des données de contraste et de texture. Ce dernier a été « parallélisé » et porté sur une architecture de type pipeline. La période d'échantillonnage de notre application est conditionnée par le temps de calcul du traitement d'images. Celle-ci **n'est pas fixe** et peut varier de 300 %. De plus, l'architecture retenue en 1992 introduit un retard pur au moins égal à 600 ms. Le temps de calcul de la partie commande (haut et bas niveau) est négligeable et ne **nécessitera donc pas un matériel performant**.

## 2. ASSERVISSEMENT PAR RECONSTRUCTION DE TRAJECTOIRE

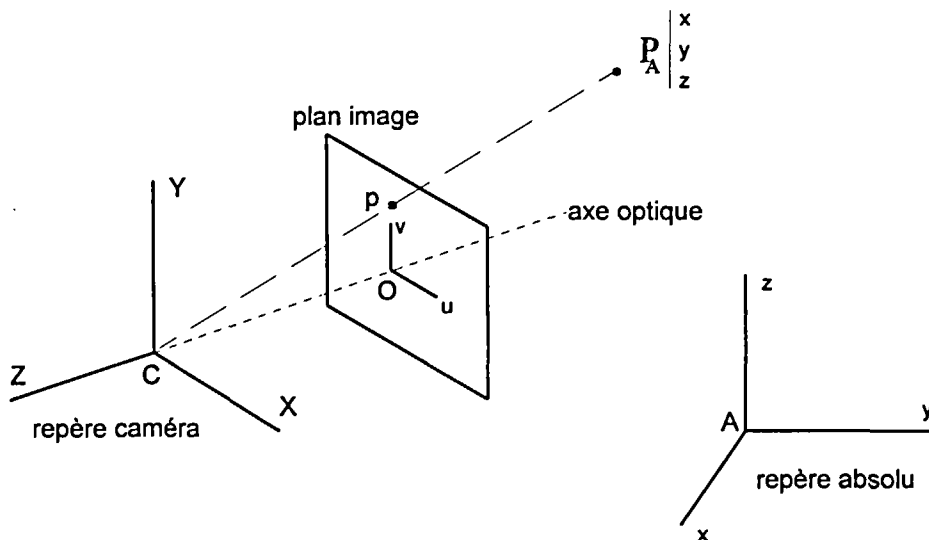
Le principe de cette première loi de commande est de déterminer, à partir d'une position initiale de notre engin et de sa position finale désirée, une trajectoire lui permettant de passer de l'une à l'autre. Cependant, les seules informations dont nous disposons sont les coordonnées de l'interface détectée dans une image prise 3 m devant la machine. Il faut donc, dans un premier temps, reconstruire la position 3D de l'interface qui correspond à la position finale (ou désirée) de l'engin.

### 2.1 Calibrage de la caméra

Les informations visuelles issues du programme de traitement d'images sont des informations 2D. Elles ne rendent compte de ce qui se passe dans la scène que par l'intermédiaire d'une projection de celle-ci sur la matrice CCD. Il nous faudra donc calculer la matrice de projection inverse afin de retrouver les coordonnées 3D des points extraits de l'image. Pour cela, nous allons présenter différentes méthodes permettant de calibrer notre caméra et retenir celle qui nous semble la mieux adaptée à notre application.

#### 2.1.1 Modèle géométrique de la caméra

Le modèle géométrique de la caméra dépend du type de projection utilisé. Nous modélisons la caméra par une projection perspective en utilisant le modèle sténopé dont la précision est suffisante pour nos applications (**Figure III 8**).



**Figure III 8 : modèle sténopé de la caméra**

Le modèle sténopé est un schéma de projection perspective utilisant comme centre de projection **C**, le centre optique de l'objectif et comme plan image **P**, le plan de la surface photosensible de la caméra. La projection orthogonale du centre **C** sur le plan image **P** est le point **O**. La distance entre le plan image et le centre optique **CO** est la distance focale notée **f**. Le plan **P** est caractérisé par les axes **(Ou)** et **(Ov)** définissant les directions de balayage vertical et horizontal de la caméra. Le repère de la caméra est défini par les axes **(CX)** et **(CY)** parallèles aux axes **(Ou)** et **(Ov)** et l'axe **(CZ)** perpendiculaire au plan image. Cet axe est assimilé à l'axe optique.

En pratique, le repère caméra **(C, X, Y, Z)** n'est pas accessible dans le sens où ni la position exacte du centre optique, ni celle des axes ne sont connues. Il est donc préférable de repérer les points de la scène dans un repère absolu **(A, x, y, z)** appelé aussi repère égocentrique.

Notons **(x, y, z)** les coordonnées tridimensionnelles d'un point **P** dans le repère absolu. Soit **(X, Y, Z)** les coordonnées tridimensionnelles de ce même point **P** dans le repère caméra. Le point **P** de l'espace se projette sur le plan image en **p(u', v')**. La projection n'est pas une application bijective : il y a donc une infinité de points ayant comme projeté le point **p**. Pour chaque point image **p**, il correspond une infinité d'antécédents dans l'espace constituant une droite sur laquelle se situe le point **P**.

Pour estimer la position dans l'espace d'un point à partir d'informations bidimensionnelles, il est nécessaire d'avoir une autre information. C'est la raison pour laquelle on utilise, généralement, un autre capteur actif (ex : un télémètre) ou passif (ex : une 2<sup>me</sup> caméra<sup>1\*</sup>) ou bien encore une séquence d'images pour obtenir l'information profondeur.

Nous verrons par la suite que, pour notre application, un deuxième capteur n'est pas nécessaire car tous les points de l'objet à localiser (la limite de fauche) sont au sol, donc appartiennent au plan  $z=0$  (hypothèse retenue lors de l'analyse de l'environnement de nos robots).

---

<sup>1\*</sup> A partir de deux vues d'une scène, provenant de deux caméras ou d'une seule caméra en deux localisations, la position d'un point dans l'espace peut être déterminée par intersection des deux droites définies par les vues (triangulation).



### 2.1.2 Intérêt du calibrage

La procédure de calibrage est une étape nécessaire pour obtenir précisément la correspondance entre les coordonnées tridimensionnelles d'un point de la scène et les coordonnées pixel du point correspondant dans l'image numérisée. Calibrer une caméra consiste à déterminer la relation liant les coordonnées tridimensionnelles d'un point de l'espace et les coordonnées bidimensionnelles, en modélisant le processus de projection. Cette modélisation nécessite la détermination de deux groupes de paramètres :

- **les paramètres extrinsèques** correspondant à la transformation rigide du repère rétinocentrique (associé à la caméra) au repère absolu (associé à la scène).
  
- **les paramètres intrinsèques**, appelés éléments propres de la caméra, définissent les caractéristiques géométriques et optiques de celle-ci. Ce sont :
  - les coordonnées du centre optique dans l'image,
  - les facteurs d'échelle de l'image sur l'axe x et y (notés respectivement  $e_x$  et  $e_y$ ),
  - la distance focale  $f$ .

La procédure de calibrage n'utilise pas, si possible, les données fournies par le constructeur. La procédure de calibrage estime elle-même les paramètres de la caméra à partir d'un ensemble de points de référence<sup>2\*</sup> parfaitement connus dans la scène et de leurs correspondants dans l'image.

Notons également que, sans calibrage de la caméra, la position exacte de la projection du centre optique n'est pas parfaitement connue. Cette méconnaissance peut entraîner des erreurs supplémentaires sur l'estimation des coordonnées tridimensionnelles.

---

<sup>2\*</sup> appelés aussi points de calibrage.

### 2.1.3 Aperçu de quelques méthodes de calibrage

De nombreuses techniques de calibrage sont présentées dans la littérature. Une synthèse de ces méthodes est donnée dans l'article de Stai [Sta89].

Les techniques les plus anciennes consistent à résoudre les équations non linéaires caractérisant la transformation des coordonnées tridimensionnelles en coordonnées images bidimensionnelles par des méthodes d'optimisation non linéaires. L'inconvénient de ces techniques est qu'elles sont très intensives en calcul et qu'elles dépendent énormément de l'estimation initiale des paramètres.

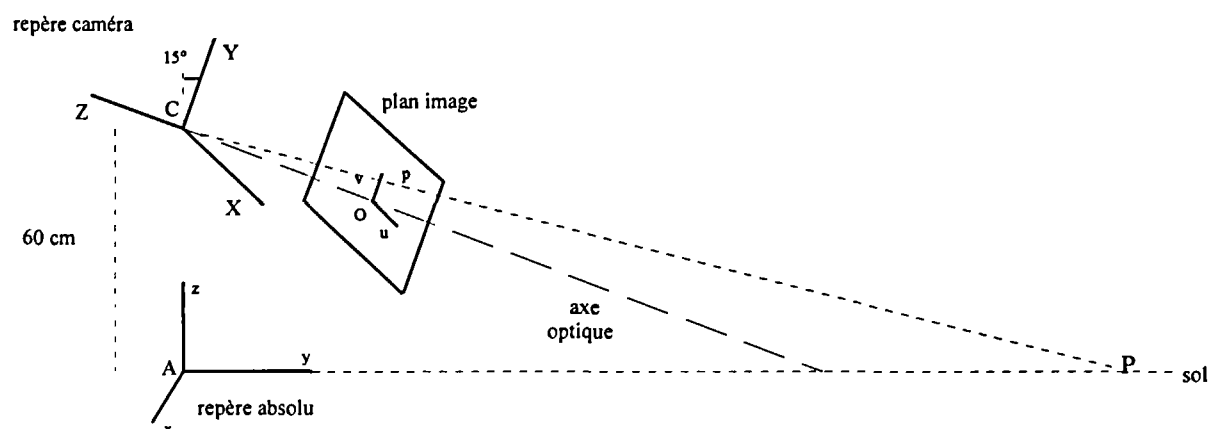
Faugeras et Toscani [Fau87] proposent une méthode qui consiste à écrire la matrice de projection perspective à l'aide des coordonnées homogènes en ignorant la distorsion de l'objectif. Ceci permet d'obtenir un système linéaire. Les coefficients de la matrice sont déterminés à partir des coordonnées tridimensionnelles d'un ensemble de points de référence et des coordonnées de leur projection dans l'image. Cette étape est suffisante pour obtenir une correspondance entre le "tridimensionnel et le bidimensionnel". Cependant, si on s'intéresse à l'extraction des paramètres de la caméra, une deuxième étape utilisant le filtre de Kalman permet d'obtenir une précision plus fine.

Une technique proposée par Stai [Sta87] consiste à déterminer les paramètres de calibrage avec des équations linéaires. Une contrainte appelée Contrainte d'Alignement Radial est introduite. Celle-ci permet de séparer les paramètres de calibrage en deux groupes et de les rendre linéairement indépendants. Dans ce cas, la distorsion de l'objectif est modélisée.

Une méthode proposée par Chaumette et Rives [Cha89], permettant également de modéliser la distorsion due à l'objectif, consiste à déterminer dans un premier temps les valeurs approchées des paramètres de calibrage par la résolution d'un système linéaire à l'aide de la méthode de Lagrange. Ces résultats étant biaisés en présence de bruit, ils sont utilisés comme estimations initiales à l'entrée d'un algorithme de minimisation non linéaire.

### 2.1.4 Notre application avec la méthode de Toscani

Dans le cas de notre application, il faut noter que l'objectif n'est pas d'extraire les paramètres exacts de la caméra mais de déterminer une relation entre les points de la scène et leur projection dans l'image. De plus, la précision recherchée, concernant la détermination de la position des points de la scène, n'est pas extrêmement sévère : nous souhaitons guider nos véhicules avec une précision de 5 centimètres sur une largeur de travail allant de 1 à 8 mètres. Nous avons donc utilisé la méthode de base, proposée par Faugeras et Toscani, d'une part, pour sa facilité de mise en oeuvre et, d'autre part, pour les résultats encourageants obtenus par l'équipe SMTI du Laboratoire d'Electronique de Clermont-Ferrand [Chap91].



**Figure III 9 : position et orientation de la caméra**

Point A : point de contact du véhicule avec le sol

Point C : centre optique de la caméra embarquée

Tous les points à localiser dans la scène sont au sol, c'est-à-dire en  $z=0$  dans le repère absolu. Ceci permet de faire correspondre à un point de l'image un seul point de la scène.

Nous avons choisi de centrer le repère absolu sur le véhicule afin de calculer les coordonnées des points image par rapport au véhicule. Une fois la caméra calibrée, dans la mesure où l'orientation et la position de celle-ci ne sont pas modifiées, nous pouvons retrouver à tout instant la position de la limite de fauche dans la scène.

Afin de relier les coordonnées des points de la scène à celles de leur projeté dans l'image, nous calculons les matrices de passage entre les différents repères considérés.

Passage du repère absolu au repère caméra

La transformation rigide entre le repère caméra et le repère absolu est définie par six paramètres :

- trois paramètres de rotation,
- trois paramètres de translation :  $t_x, t_y, t_z$ .

Considérons un point P de coordonnées  $(x, y, z)$  dans le repère absolu et  $(X, Y, Z)$  dans le repère caméra. Nous pouvons écrire la relation suivante :

eq III 2

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

où  $\begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$  est la matrice résultante des trois rotations autour des axes  $x, y, z$ .

Passage du repère caméra au repère du plan image :

Considérons  $p(u', v')$  le projeté de P de coordonnées  $(X, Y, Z)$  dans le repère caméra. En utilisant les équations de la projection perspective, nous obtenons :

$$\begin{aligned} u' &= \frac{X \cdot f}{Z} \\ v' &= \frac{Y \cdot f}{Z} \end{aligned} \quad \text{avec } f \text{ distance focale.}$$

Afin d'avoir une relation linéaire entre ces coordonnées, introduisons les coordonnées homogènes :

eq III 3

$$\begin{pmatrix} W \cdot u' \\ W \cdot v' \\ W \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Passage du repère plan image au repère de l'image numérisée :

Les coordonnées  $(u', v')$  ne sont pas directement exploitables car l'unité de mesure dans le plan image est le pixel. Il est donc nécessaire de tenir compte de la résolution du capteur. De plus, le centre de la matrice CCD n'est pas forcément la position exacte de la projection du centre optique.

Les coordonnées du point  $p(u, v)$  exprimées en pixel s'écrivent alors :

$$u = \frac{p_u + u'}{e_u} \quad v = \frac{p_v + v'}{e_v}$$

$(u', v')$  sont les coordonnées de  $p$  dans le repère image.  $e_u$  et  $e_v$  sont les facteurs d'échelle sur l'axe  $u$  et  $v$ .  $p_u$  et  $p_v$  sont les coordonnées de la projection du centre optique dans le repère image. Toutes ces valeurs sont exprimées en centimètres.

Soit

$$\begin{pmatrix} Wu \\ Wv \\ W \end{pmatrix} = \begin{pmatrix} \frac{1}{e_u} & 0 & \frac{p_u}{e_u} \\ 0 & \frac{1}{e_v} & \frac{p_v}{e_v} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} Wu' \\ Wv' \\ W \end{pmatrix}$$

Passage du repère absolu au repère image numérisée :

Le passage des coordonnées tridimensionnelles d'un point de la scène, aux coordonnées pixel du point de luminance correspondant dans l'image, se traduit par le système linéaire suivant :

eq III 4

$$\begin{pmatrix} Wu \\ Wv \\ W \end{pmatrix} = \begin{pmatrix} \frac{1}{e_u} & 0 & \frac{p_u}{e_u} \\ 0 & \frac{1}{e_v} & \frac{p_v}{e_v} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Soit

$$\begin{pmatrix} Wu \\ Wv \\ W \end{pmatrix} = T \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \text{avec : } T = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & 1 \end{pmatrix}$$

Pour trouver les coefficients  $t_{ij}$  de la matrice  $T$ , la méthode des moindres carrés sera utilisée. Nous pouvons écrire pour chaque point de l'espace :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \bar{P}^T & 0_{14} & -P^T u \\ 0_{14} & \bar{P}^T & -P^T v \end{pmatrix} \begin{pmatrix} \bar{t}_1 \\ \bar{t}_2 \\ \bar{t}_3 \end{pmatrix} \quad \text{avec} \quad \begin{aligned} P &= (x \ y \ z)^T \\ \bar{P} &= (x \ y \ z \ 1)^T \\ t_i &= (t_{i1} \ t_{i2} \ t_{i3})^T \\ \bar{t}_i &= (t_{i1} \ t_{i2} \ t_{i3} \ t_{i4})^T \end{aligned}$$

Pour  $N$  points de référence, le système s'écrit :

$$Y_{2N \times 1} = H_{2N \times 11} \cdot \Theta_{11 \times 1}$$

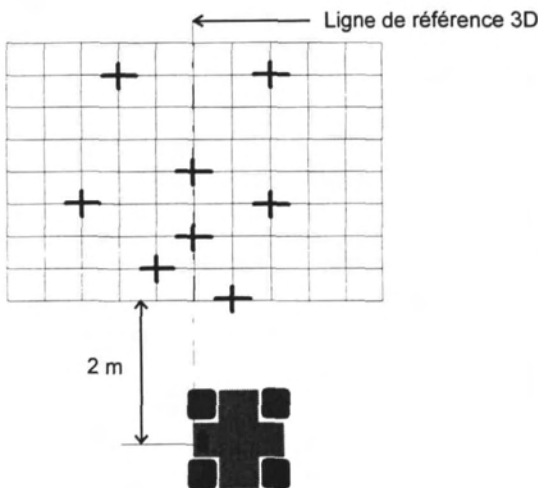
où  $\Theta$  est le vecteur des 11 coefficients  $t_{ij}$ , et  $Y$  le vecteur de mesure constitué de  $N$  couples  $(u, v)$ .

La résolution du système nécessite, en théorie, 6 points de référence non coplanaires. Nous obtenons avec la méthode des moindres carrés (cf. annexe A) :

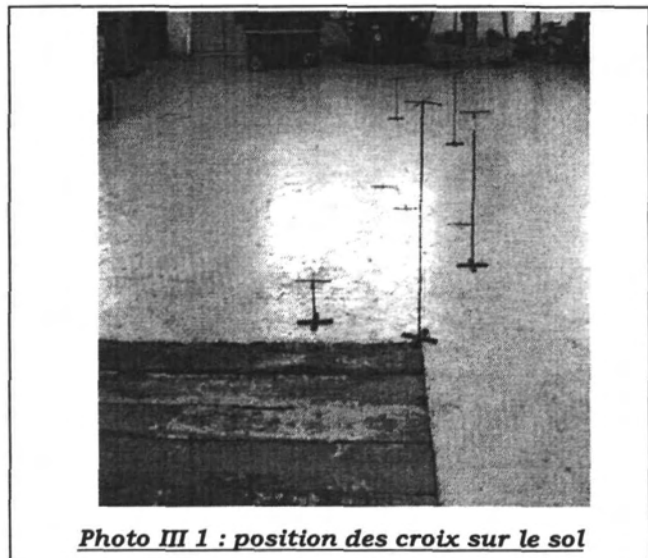
$$\Theta_{\text{optimum}} = (H^T H)^{-1} H^T Y$$

Expérimentation et localisation des points dans la scène

Pour déterminer les différents paramètres de la matrice  $T$ , nous avons utilisé une grille plaquée sur le sol (**Figure III 10 et Photo III 1**). Nous avons ensuite disposé sur celle-ci 8 croix à différentes hauteurs (0, 10, 20, 30, 40, 50 cm). Il est indispensable que ces croix ne soient pas dans le même plan afin que les différentes matrices considérées soient inversibles. De plus, il faut aligner la machine avec la ligne de référence 3D à partir de laquelle se fera l'asservissement.



**Figure III 10 : essai de calibrage**



**Photo III 1 : position des croix sur le sol**

Munis des coordonnées de ces croix dans la scène, ainsi que dans le repère image, nous sommes maintenant capables de calculer les différents paramètres de la matrice T.

*eq III 5*

$$\begin{pmatrix} W \cdot u \\ W \cdot v \\ W \end{pmatrix} = T \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Remarque

Nous avons utilisé 8 points pour déterminer les coefficients de la matrice T. En fait 6 non coplanaires sont suffisants, mais un nombre supérieur permet d'accroître la précision.

Notre but est d'estimer la position dans l'espace d'un point (x, y, z) à partir de ses coordonnées (u, v) exprimées en pixels sur l'écran. Pour cela, il faudrait déterminer la matrice inverse de T. Cependant, cette matrice traduit une projection qui est une application non bijective. Il faut donc ajouter une hypothèse supplémentaire qui consiste, dans notre cas, à supposer que le point reconstruit se trouve dans le plan z=0. Ce plan ne contenant pas l'axe optique, il est possible de déterminer une application qui donne les coordonnées des points de la scène contenus dans le plan z=0 en fonction de leurs coordonnées pixels. Nous obtenons donc :

*eq III 6*

$$\begin{pmatrix} Wu \\ Wv \\ W \end{pmatrix} = T_{z=0} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \text{avec : } T_{z=0} = \begin{pmatrix} t_{11} & t_{12} & t_{14} \\ t_{21} & t_{22} & t_{24} \\ t_{31} & t_{32} & 1 \end{pmatrix}$$

Cette nouvelle matrice est inversible :

*eq III 7*

$$T_{z=0}^{-1} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

On obtient donc :

*eq III 8*

$$\begin{aligned} x &= \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + a_{33}} \\ y &= \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + a_{33}} \\ z &= 0 \end{aligned}$$

Nous retrouvons ainsi les coordonnées des points de la scène appartenant au plan  $z=0$ , à partir de l'information bidimensionnelle fournie par l'image.

### 2.1.5 Résultats (numériques en annexe A)

Une vérification du calibrage a été réalisée avec six points, en comparant les coordonnées de ces points mesurées dans le repère absolu et celles calculées à partir des données fournies par la caméra.

Sur un champ de vision s'étalant de 1 m à 3.3 m par rapport au pied de la caméra, nous obtenons une erreur de l'ordre du centimètre. Celle-ci peut être engendrée par la modélisation employée, mais aussi par la détermination des coordonnées 2D des points de l'image que nous avons relevés de manière empirique et avec une précision sûrement supérieure au pixel. Cependant, les résultats obtenus sont suffisants pour notre application qui ne demande pas une précision inférieure à 5 cm.

Notons que la reconstruction des points de la scène se fait au moyen de la résolution d'un système linéaire qui peut amplifier un bruit introduit sur les données d'entrée. Ce phénomène n'est pas à négliger car nos véhicules évoluent sur un terrain cahoteux et l'interface détectée est souvent imprécise à cause des irrégularités de la végétation.

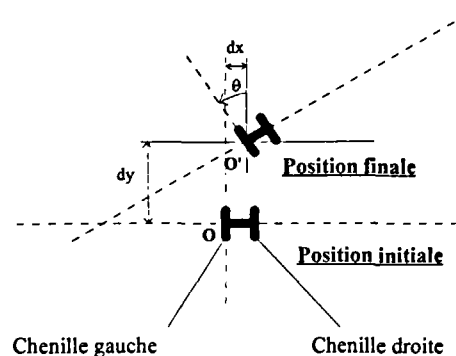


## 2.2 Reconstruction de la trajectoire

Dans cette partie, le calcul d'une trajectoire permettant de relier deux positions d'un essieu rigide est présenté. Il utilise différents paramètres (distance du centre de l'image à l'engin considéré, hauteur de la caméra ...) qui, dans l'exemple exposé, sont ajustés par rapport au premier véhicule (la tondeuse). Toutefois, la méthode employée en est indépendante ce qui permettra une généralisation facile de l'algorithme à d'autres types d'engins (tracteur, moissonneuse-batteuse, voiture ...).

La position de la caméra ainsi que sa focale et son orientation ont été choisies afin d'optimiser les performances de l'algorithme de traitement d'images. Dans ces conditions, le centre de l'image se trouve à une distance de 3 mètres de la machine. La position finale désirée se trouve donc éloignée d'une distance ( $dy$ ) de 3 mètres de la position initiale (**Figure III 11**). La vitesse moyenne de la tondeuse est de 3 km/h. Cette vitesse ne doit pas varier tout au long de la trajectoire puisque la machine évolue à vitesse moyenne constante. Nos applications utilisent un capteur caméra ; il faudra donc éviter tout changement brusque d'orientation de la machine afin que l'interface ne sorte pas du champ visuel du capteur et que les variations de l'image soient les plus faibles possibles.

Pour cela, la trajectoire calculée doit permettre de passer de la position initiale à la position finale avec les mouvements les plus amples, c'est-à-dire avec des rayons de courbure les plus grands possibles. **L'idée est de trouver l'équation de deux cercles tangents contenant respectivement les points de départ et d'arrivée.**



**Figure III 11 : deux positions de la tondeuse**

### Remarque

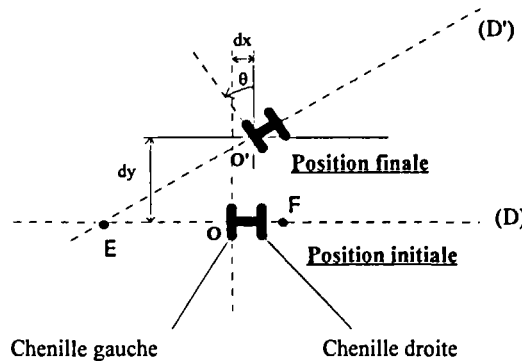
Dans la suite de la présentation, nous « approximerons » la tondeuse par l'essieu rigide virtuel qui la modélise (cf chapitre II). Le point à asservir sera le centre de la roue gauche de cet essieu.

Kehtarnaz dans [Keh91] utilise des cercles de même rayon. L'avantage de sa méthode est d'avoir une courbure constante en valeur absolue tout le long de sa trajectoire. Notre manière de calculer les deux rayons des deux cercles est très voisine mais elle permet souvent de limiter l'angle de la machine par rapport à l'interface. Ceci rend le fonctionnement du système plus doux et donc plus favorable au bon déroulement du programme de traitement d'images. Cependant, d'autres courbes telles que des clotoïdes pourraient être utilisées.

### 2.2.1 Calcul de la trajectoire :

- \* Soit O, O' les points correspondants à la position initiale et à la position finale du centre de la roue gauche,
- \* Soit (D) la droite passant par le centre des deux roues pour la position initiale et (D') la droite passant par le centre des deux roues pour la position finale,
- \* Soit  $\theta$  l'angle formé entre la direction de référence (droit devant soi) et celle que l'on devra avoir en position finale. Soit dx la dérive mesurée entre la position finale et la position initiale,
- \* Soit dy la distance moyenne parcourue le long de l'axe y. Soit E l'intersection de (D) et (D'),
- \* Soit un point F appartenant à (D) tel que :

$$OF = \frac{1}{2} O'F$$



#### 2.2.1.1 Présentation des calculs :

L'équation de la droite (D') est donnée par :  $Y = (\tan\theta)X + dy - (\tan\theta)dx$

L'intersection de (D) et (D') nous donne le point E :

$$E \left( \begin{array}{c} \frac{(\tan\theta)dx - dy}{\tan\theta} \\ 0 \end{array} \right) \text{ si } \tan\theta \neq 0;$$

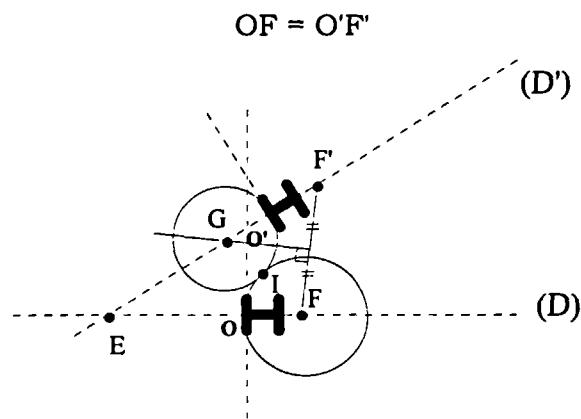
Exprimons les coordonnées du point F appartenant à (D) :

on a : 
$$OF = \frac{1}{2} O'F$$

donc :

$$F \begin{pmatrix} \frac{-dx + \sqrt{4(dx)^2 + 3(dy)^2}}{3} \\ 0 \end{pmatrix}$$

Soit F', un point de (D') tel que :



Notons X et Y les coordonnées de F' et dF la distance OF ; alors la condition précédente se traduit par :

$$(X - dx)^2 + (Y - dy)^2 = dF^2$$

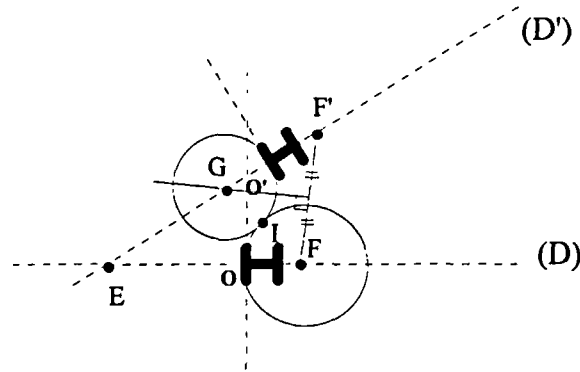
Ce qui donne :

$$X = \frac{dx - ab + ady \pm \sqrt{(dx - ab + dy)^2 - (1 + a^2)(dx^2 + b^2 - 2bdy + dy^2 - dF^2)}}{1 + a^2}$$

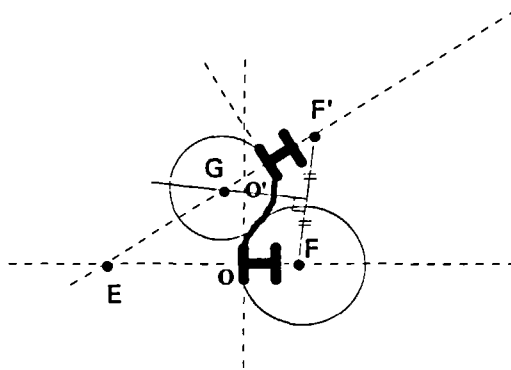
$$Y = aX + b$$

avec  $a = \tan\theta$  et  $b = dy - \tan\theta dx$ , X et Y étant les coordonnées de F'.

Le centre du deuxième cercle (G) doit être choisi tel que la distance GF soit égale à la somme des deux rayons des deux cercles représentant la trajectoire de la machine. Le deuxième cercle de centre G passe par le point O'. La distance GF doit donc être égale à la distance GF', donc G appartient à la médiatrice de F, F'. G est l'intersection de la médiatrice de F, F' et de (D').



I, point de tangence des deux cercles, est tel que G, I, F soient alignés. Les angles des deux rotations à effectuer sont donnés par les points IFO pour la rotation de centre F et IGO' pour la rotation de centre G.



On a donc déterminé l'angle et le rayon de nos deux rotations qui permettent à l'engin de passer d'une position initiale à une position finale. La tâche consiste maintenant à déterminer les vitesses des deux roues de l'essieu afin de suivre la trajectoire précédemment définie.

Pour cela, nous utilisons l'équation (chapitre II eq 5) :

eq III 9 
$$R = \frac{v}{\phi} = \frac{b}{2} \cdot \frac{(V_2 + V_1)}{(V_2 - V_1)} = \frac{b}{2} \cdot \frac{(\dot{q}_2 + \dot{q}_1)}{(\dot{q}_2 - \dot{q}_1)}$$

Dans nos applications la vitesse moyenne de l'engin est fixée à l'avance par l'opérateur :

eq III 10 
$$v = \frac{r \cdot (\dot{q}_1 + \dot{q}_2)}{2} = v_{\text{moy}}$$

A partir de (eq III 9) et (eq III 10), on obtient :

$$\text{eq III 11} \quad \begin{cases} \dot{q}_1 = \frac{v_{\text{moy}} \cdot (2R - b)}{2Rr} \\ \dot{q}_2 = \frac{v_{\text{moy}} \cdot (2R + b)}{2Rr} \end{cases}$$

La vitesse de chacune des deux roues de l'essieu rigide est donc calculée en fonction du rayon de courbure de la trajectoire à suivre et de la vitesse moyenne de l'engin.

Pour relier la vitesse angulaire des roues de l'essieu aux actionneurs de nos robots nous utiliserons la matrice A propre à chaque véhicule et décrite au chapitre II.

### 2.2.1.2 Exemple de la tondeuse :

Pour la tondeuse, l'expression de la matrice A reliant le comportement de l'essieu rigide virtuel aux actionneurs du véhicule est donnée par :

$$\text{eq III 12} \quad A = \begin{pmatrix} 1 & -1 \\ r & 2r \\ 1 & 1 \\ r & 2r \end{pmatrix} \text{ avec : } \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = A \cdot \begin{pmatrix} v \\ \Delta v \end{pmatrix}$$

Rappelons que les deux actionneurs de la tondeuse commandent v et Δv.

On a donc :

$$\text{eq III 13} \quad \begin{pmatrix} v \\ \Delta v \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ -r & r \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

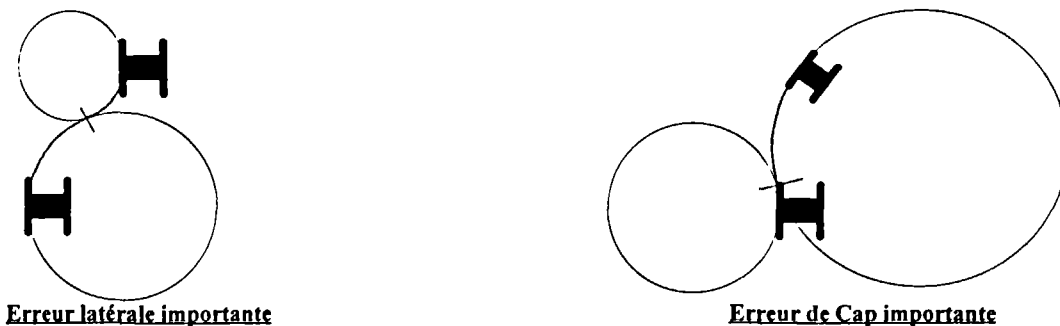
De (eq III 13) et (eq III 11), nous déduisons :

$$\text{eq III 14} \quad \begin{cases} v = v_{\text{moy}} \\ \Delta v = \frac{b \cdot v_{\text{moy}}}{R} \end{cases}$$

L'équation (eq III 14) donne l'expression de la commande des actionneurs de l'engin en fonction de la vitesse moyenne désirée et du rayon de courbure de la trajectoire.

**Remarques :**

La trajectoire décrite comporte deux arcs de cercle. Ceux-ci peuvent être de longueurs très variables suivant les configurations. Si, par exemple, l'écart latéral est important, le premier arc de cercle sera le plus long. Par contre, si l'erreur de cap est la plus grande, le deuxième arc de cercle sera prépondérant (**Figure III 12**). Dans la pratique, l'asservissement de nos véhicules sur une ligne qui est quasiment toujours droite se fait principalement grâce au cap. En effet, chaque changement de direction de la trajectoire est d'abord indiqué par une erreur de cap. La trajectoire de la machine se résumera donc, le plus souvent, au deuxième arc de cercle. Celui-ci sera parcouru en partie à chaque itération puisque la machine avance de 25 cm en 300 ms.



**Figure III 12 : deux trajectoires différentes.**

**Conclusion**

Le but de notre application est d'asservir un engin agricole sur une limite de fauche qu'il a lui-même tracée au passage précédent. La trajectoire de cette limite de fauche est donc une trajectoire de la machine. L'approche présentée permet de la reconstruire et de la suivre. De plus, le fait d'avoir une période d'échantillonnage non déterministe n'est pas rédhibitoire puisque la machine suit une trajectoire dont le temps de parcours est au moins trois fois supérieur à la période d'échantillonnage.

Cependant, cette technique utilise une reconstruction de la scène supposant que les points de l'interface se trouvent dans le même plan que celui des roues de l'engin. Or, nos véhicules évoluent sur des terrains cahoteux rendant aléatoires les résultats de la reconstruction. Nous montrerons, au chapitre IV, que la qualité d'asservissement de cette première méthode se dégrade rapidement lorsque l'on bruite les informations visuelles.

C'est pourquoi, nous avons choisi de nous orienter vers une méthode permettant d'asservir notre machine dans l'espace image. Nous éviterons ainsi les erreurs de modélisation introduites par l'étape d'interprétation et l'effet « amplification d'erreur » du système linéaire qui assure la reconstruction de la scène.

### **3. ASSERVISSEMENT PAR COMMANDE REFERENCEE VISION**

#### **INTRODUCTION**

Dans sa thèse, François CHAUMETTE écrit [Chau90] :

*« L'emploi des techniques basées sur un asservissement sur la situation de la caméra présente l'inconvénient majeur de nécessiter une phase d'interprétation des informations visuelles, de manière à reconstruire la situation courante entre la caméra et son environnement. Les problèmes inhérents à cette interprétation sont de deux sortes :*

*- les algorithmes qu'elle nécessite peuvent être extrêmement lourds dans le cas de scènes complexes, réduisant grandement la dynamique du système.*

*- les erreurs de précision de cette phase d'interprétation entraînent inévitablement des erreurs dans l'estimation de la situation courante. Ainsi, la convergence de l'algorithme peut être obtenue (c'est-à-dire que la situation estimée est égale à la situation voulue) sans que la tâche soit effectivement réalisée (situation réelle différente de la situation voulue).*

*L'approche asservissement visuel permet, elle, de s'affranchir totalement de cette phase d'interprétation et semble être optimale au niveau des temps de calcul puisque les informations extraites sont directement utilisées dans la boucle de commande ».*

La commande référencée vision, présentée par Chaumette, est l'application aux asservissements visuels du formalisme de la commande référencée capteur développé par Claude Samson Michel le Borgne et Bernard Espiau [Sam91b]. Elle permet d'assurer la régulation de la trajectoire d'un robot dans l'espace image.

Néanmoins, cette technique a été développée essentiellement pour des robots manipulateurs holonomes. La principale difficulté de notre travail a donc été de rendre l'asservissement stable pour nos véhicules qui sont non holonomes et qui ne possèdent que deux degrés de liberté.

Dans la suite de ce chapitre, nous présenterons quelques rappels concernant le formalisme de la commande référencée vision en nous appuyant sur les travaux de François Chaumette [Chau90]. Nous détaillerons la modélisation de la scène définie par le « torseur d'interaction » qui relie les mouvements de la caméra aux variations de l'image. Puis la notion de fonction de tâche développée par Claude Samson dans [Sam91b] nous permettra de calculer nos lois de commande.

### 3.1 Quelques définitions générales

- ◆ Un robot est un système mécanique capable de déplacer son effecteur (qui peut être lui-même) afin de réaliser une tâche qui lui a été assignée.
- ◆ La situation de l'effecteur, notée  $\bar{r}$  décrit la position et l'orientation de celui-ci par rapport à un repère choisi. 6 variables sont nécessaires et suffisantes pour décrire  $\bar{r}$  (3 variables pour l'orientation et 3 variables pour la position).
- ◆ Le nombre de degrés de liberté d'un robot est défini comme étant le nombre de paramètres indépendants capables de représenter l'ensemble des situations qu'il peut atteindre.
- ◆ Les mouvements d'un robot sont engendrés par l'application de forces ou de couples, notés  $\Gamma$ , qui sont transmis aux différents axes du robot par l'intermédiaire d'actionneurs. Des capteurs proprioceptifs permettent de mesurer les effets de ces actionneurs sur l'état interne du robot. Les variables, représentant cet état, sont appelées les variables articulaires et sont notées  $\underline{q}$ . Elles fournissent les positions de chaque axe par rapport à une position de référence.
- ◆ Le jacobien du robot  $J(\underline{q})$  relie les vitesses  $\dot{\underline{q}}$  dans l'espace articulaire au torseur cinématique  $\bar{r}$  exprimé dans l'espace dit opérationnel. On a alors la relation :  $\bar{r} = j(\underline{q}) \cdot \dot{\underline{q}}$
- ◆ Considérons  $\underline{S}$ , une information visuelle extraite d'une image d'une caméra, différentiable de  $SE_3$  dans  $\mathcal{R}$ .  $SE_3$  est l'espace de configuration des corps rigides. On considère que seuls les mouvements de la caméra ou des objets de son environnement peuvent faire varier la valeur de l'information visuelle.

### 3.2 Modélisation de la scène

Dans le cadre de nos applications de robotique mobile, la caméra est fixée sur le robot. Donc, seuls les déplacements du robot ou celui des objets constituant l'environnement de la caméra peuvent faire varier  $\underline{S}$ . De plus, le déplacement du robot est fonction des coordonnées articulaires  $\underline{q}$  de ces axes, donc :

$$\underline{S} = \underline{S}(\underline{q}, t)$$

Différencions  $\underline{S}$  :

eq III 15

$$\dot{\underline{S}} = \frac{\partial \underline{S}}{\partial \underline{q}} \cdot \dot{\underline{q}} + \frac{\partial \underline{S}}{\partial t}$$



$\frac{\partial \underline{S}}{\partial t}$  correspond aux mouvements des primitives visuelles décrivant l'environnement de la caméra.

Le terme  $\frac{\partial \underline{S}}{\partial \underline{q}}$  peut se décomposer sous la forme :

$$\text{eq III 16} \quad \frac{\partial \underline{S}}{\partial \underline{q}} = \frac{\partial \underline{S}}{\partial \underline{r}} \cdot \frac{\partial \underline{r}}{\partial \underline{q}}$$

$\frac{\partial \underline{r}}{\partial \underline{q}}$  est le jacobien du robot,

$\frac{\partial \underline{S}}{\partial \underline{r}}$  traduit la variation des informations visuelles en fonction des déplacements du robot. Il est appelé jacobien de la tâche, ou matrice d'interaction, ou bien encore jacobien d'image [Hut96].

Supposons que, l'environnement de la caméra soit immobile. A partir de (eq III 15) et (eq III 16) nous obtenons :

$$\dot{\underline{S}} = \frac{\partial \underline{S}}{\partial \underline{r}} \cdot \frac{\partial \underline{r}}{\partial \underline{q}} \cdot \dot{\underline{q}}$$

Chaumette montre que, pour une composante  $S_j$  de  $\underline{S}$ , nous pouvons écrire dans le repère lié à la caméra la relation fondamentale :

$$\dot{S}_j = H_j \bullet T$$

- $T$  est le torseur cinématique représentant la vitesse relative de la scène par rapport à la caméra,
- $\bullet$  est le produit de torseur,
- $H_j$  est un torseur dont l'expression dépend à la fois des caractéristiques de l'environnement et du capteur lui-même. Il caractérise complètement les interactions entre capteur et environnement. Il est appelé torseur d'interaction.

En posant  $L_j^T$  la représentation matricielle du torseur d'interaction  $H_j$  exprimée dans un repère et en un point donnés, on peut écrire :

$$\dot{S}_j = L_{S_j}^T \cdot T$$

avec :

$$L_{S_j}^T = H_j \cdot \begin{pmatrix} 0 & I_3 \\ I_3 & 0 \end{pmatrix}$$

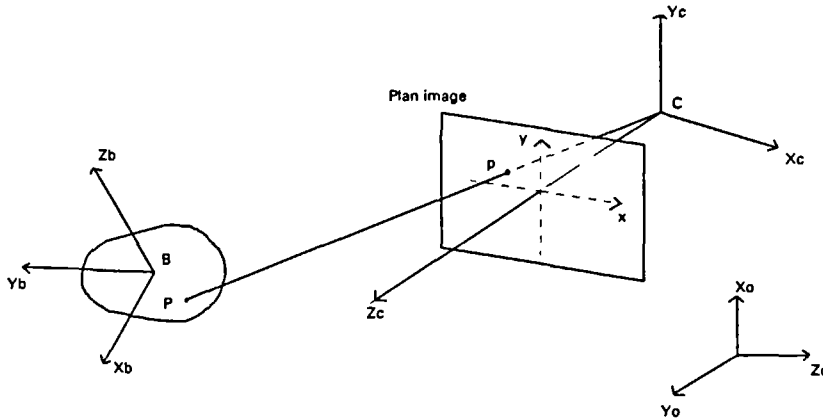
De même, en considérant l'ensemble des informations visuelles, nous obtenons :

$$\underline{\dot{S}} = L_S^T \cdot T$$

Dans la suite de ce paragraphe, nous nous proposons de calculer  $L_S^T$  pour deux primitives visuelles géométriques qui sont le point et une droite. Une méthode générale permettant le calcul de ce torseur pour de nombreuses autres primitives est donnée dans [Chau90] et [Mot92].

### 3.2.1 Calcul de la matrice d'interaction pour une primitive de type point

Soit une scène B ayant comme référentiel  $R_b$  et une caméra C ayant pour référentiel  $R_c$ , le tout étant placé dans un référentiel galiléen  $R_0$ . Soit P un point de la scène. Nous allons nous intéresser à l'expression de la vitesse de P dans les différents référentiels (**Figure III 13**).



**Figure III 13 : projection du point P dans les différents référentiels.**

$R_c$  est le référentiel lié à la caméra,  $R_b$  étant liée à la scène. Les différents éléments décrits ci-dessus sont en mouvement les uns par rapport aux autres. Exprimons la vitesse de P dans les différents référentiels :

on a :

eq III 17

$$\vec{V}_{R_b}(P) = \vec{V}_{R_b}(C) + \vec{V}_{R_c}(P) + \vec{\Omega}_{R_c/R_b} \wedge C\vec{P}$$

de même :

$$\vec{V}_{R_0}(C) = \vec{V}_{R_0}(B) + \vec{V}_{R_b}(C) + \vec{\Omega}_{R_b/R_0} \wedge B\vec{C}$$

Prenons comme hypothèse l'immobilité de la scène et de ces objets :

$$\vec{V}_{R_b}(P) = \vec{0} ; \vec{\Omega}_{R_b/R_0} = \vec{0} ; \vec{V}_{R_0}(B) = \vec{0}$$

$$\Leftrightarrow \vec{V}_{R_c}(P) = -\vec{V}_{R_b}(C) - \vec{\Omega}_{R_c/R_b} \wedge C\vec{P} \text{ et } \vec{V}_{R_0}(C) = \vec{V}_{R_b}(C)$$

on obtient donc :

$$\bar{V}_{R_c}(P) = -\bar{V}_{R_0}(C) - \bar{\Omega}_{R_c/R_0} \wedge C\bar{P}$$

avec :

$$\bar{\Omega}_{R_c/R_c} = \bar{\Omega}_{R_c/R_0} - \bar{\Omega}_{R_b/R_0} = \bar{\Omega}_{R_c/R_0}$$

Les données que nous voulons exploiter doivent être exprimées dans le plan image qui est perpendiculaire à l'axe  $Z_c$  et situé à une distance  $F$  du centre optique. Ecrivons les équations traduisant cette projection de l'univers 3D dans le plan image :

$$\text{eq III 18} \quad \begin{cases} u = F_u \frac{X_{R_c}}{Z_{R_c}} \\ v = F_v \frac{Y_{R_c}}{Z_{R_c}} \end{cases} \quad \text{avec :} \quad \begin{cases} F_u = \frac{F}{e_u} \\ F_v = \frac{F}{e_v} \end{cases}$$

$e_u$  et  $e_v$  représentent les dimensions d'un pixel de la matrice CCD de la caméra.

Différencions ces expressions afin de déterminer la vitesse du point dans le plan image :

$$\text{eq III 19} \quad \begin{cases} \dot{u} = F_u \cdot \left( \frac{\dot{X}_{R_c}}{Z_{R_c}} - \frac{X_{R_c}}{Z_{R_c}^2} \dot{Z}_{R_c} \right) \\ \dot{v} = F_v \cdot \left( \frac{\dot{Y}_{R_c}}{Z_{R_c}} - \frac{Y_{R_c}}{Z_{R_c}^2} \dot{Z}_{R_c} \right) \end{cases}$$

$$\text{avec :} \quad P_{R_c} \begin{pmatrix} X_{R_c} \\ Y_{R_c} \\ Z_{R_c} \end{pmatrix} \quad \text{et} \quad \bar{V}_{R_c}(P) \begin{pmatrix} \dot{X}_{R_c} \\ \dot{Y}_{R_c} \\ \dot{Z}_{R_c} \end{pmatrix}$$

Explicitons (eq III 17) :

$$\text{eq III 20} \quad \begin{pmatrix} \dot{X}_{R_c} \\ \dot{Y}_{R_c} \\ \dot{Z}_{R_c} \end{pmatrix} = \begin{pmatrix} -V_{R_0}(C)_x - Z_{R_c} \cdot \Omega_{R_c/R_0 y} + Y_{R_c} \cdot \Omega_{R_c/R_0 z} \\ -V_{R_0}(C)_y - X_{R_c} \cdot \Omega_{R_c/R_0 z} + Z_{R_c} \cdot \Omega_{R_c/R_0 x} \\ -V_{R_0}(C)_z - Y_{R_c} \cdot \Omega_{R_c/R_0 x} + X_{R_c} \cdot \Omega_{R_c/R_0 y} \end{pmatrix}$$

avec :

$$V_{R_0}(C) \begin{pmatrix} V_{R_0}(C)_x \\ V_{R_0}(C)_y \\ V_{R_0}(C)_z \end{pmatrix} \quad \text{et} \quad \Omega_{R_c/R_0} \begin{pmatrix} \Omega_{R_c/R_0 x} \\ \Omega_{R_c/R_0 y} \\ \Omega_{R_c/R_0 z} \end{pmatrix}$$

Remplaçons (eq III 20) dans (eq III 19) :

$$\dot{u} = F_u \cdot \left( -\frac{V_{R_0}(C)_x}{Z_{R_c}} - \Omega_{R_c/R_0 y} + \frac{Y_{R_c}}{Z_{R_c}} \Omega_{R_c/R_0 z} + \frac{X_{R_c}}{Z_{R_c}^2} V_{R_0}(C)_z + \frac{X_{R_c} Y_{R_c}}{Z_{R_c}^2} \Omega_{R_c/R_0 x} - \frac{X_{R_c}^2}{Z_{R_c}^2} \Omega_{R_c/R_0 y} \right)$$

D'après (eq III 18) on a :

$$\begin{cases} F_u \cdot X_{R_c} = u \cdot Z_{R_c} \\ F_v \cdot Y_{R_c} = v \cdot Z_{R_c} \end{cases}$$

$$\dot{u} = -\frac{F_u \cdot V_{R_0}(C)_x}{Z_{R_c}} - F_u \cdot \Omega_{R_c/R_0 y} + v \frac{F_u}{F_v} \Omega_{R_c/R_0 z} + \frac{u}{Z_{R_c}} V_{R_0}(C)_z + \frac{u \cdot v}{F_v} \Omega_{R_c/R_0 x} - \frac{u^2}{F_u} \Omega_{R_c/R_0 y}$$

$$\text{eq III 21} \quad \dot{u} = -\frac{F_u}{Z_{R_c}} V_{R_0}(C)_x + \frac{u}{Z_{R_c}} V_{R_0}(C)_z + \frac{u \cdot v}{F_v} \Omega_{R_c/R_0 x} - \left( F_u + \frac{u^2}{F_u} \right) \cdot \Omega_{R_c/R_0 y} + v \frac{F_u}{F_v} \Omega_{R_c/R_0 z}$$

De même :

$$\dot{v} = F_v \cdot \left( -\frac{1}{Z_{R_c}} V_{R_0}(C)_y + \frac{X_{R_c}}{Z_{R_c}} \Omega_{R_c/R_0 z} + \Omega_{R_c/R_0 x} - \frac{Y_{R_c}}{Z_{R_c}^2} \left( -V_{R_0}(C)_z - Y_{R_c} \Omega_{R_c/R_0 x} + X_{R_c} \Omega_{R_c/R_0 y} \right) \right)$$

$$\text{eq III 22} \quad \dot{v} = -\frac{F_v}{Z_{R_c}} V_{R_0}(C)_y + \frac{v}{Z_{R_c}} V_{R_0}(C)_z + \left( F_v + \frac{v^2}{F_v} \right) \cdot \Omega_{R_c/R_0 x} - \frac{u \cdot v}{F_u} \Omega_{R_c/R_0 y} - \frac{u \cdot F_v}{F_u} \Omega_{R_c/R_0 z}$$

On peut écrire (eq III 21) et (eq III 22) sous la forme :

$$\dot{\underline{S}} = \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = L_S^T \cdot T$$

$$\text{avec : } T \left( V_{R_0}(C)_x; V_{R_0}(C)_y; V_{R_0}(C)_z; \Omega_{R_c/R_0 x}; \Omega_{R_c/R_0 y}; \Omega_{R_c/R_0 z} \right)^T$$

et :

$$L_S^T = \begin{pmatrix} -\frac{F_u}{Z_{R_c}} & 0 & \frac{u}{Z_{R_c}} & \frac{u \cdot v}{F_v} & -\left( F_u + \frac{u^2}{F_u} \right) & v \frac{F_u}{F_v} \\ 0 & -\frac{F_v}{Z_{R_c}} & \frac{v}{Z_{R_c}} & \left( F_v + \frac{v^2}{F_v} \right) & -\frac{u \cdot v}{F_u} & -u \frac{F_v}{F_u} \end{pmatrix}$$

Pour la suite nous considérerons que  $F_u = F_v = 1$ , nous obtenons alors avec le même raisonnement et sans perte de généralité :

$$\begin{cases} x = \frac{X_{R_c}}{Z_{R_c}} \\ y = \frac{Y_{R_c}}{Z_{R_c}} \end{cases}$$

$$\underline{\dot{S}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = L_S^T \cdot T$$

$$\text{avec : } T \left( V_{R_0}(C)_x; V_{R_0}(C)_y; V_{R_0}(C)_z; \Omega_{R_c/R_0 x}; \Omega_{R_c/R_0 y}; \Omega_{R_c/R_0 z} \right)^T$$

et :

eq III 23

$$L_S^T = \begin{pmatrix} -\frac{1}{Z_{R_c}} & 0 & \frac{x}{Z_{R_c}} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z_{R_c}} & \frac{y}{Z_{R_c}} & (1+y^2) & -xy & -x \end{pmatrix}$$

### 3.2.2 Matrice d'interaction pour une primitive de type droite.

Dans un univers 3D, une droite est représentée par l'intersection de 2 plans non parallèles, et nous est donnée dans le référentiel lié à la caméra sous la forme :

$$h(X, Y, Z, Q) = \begin{cases} a_1 \cdot X + b_1 \cdot Y + c_1 \cdot Z + d_1 = 0 \\ a_2 \cdot X + b_2 \cdot Y + c_2 \cdot Z + d_2 = 0 \end{cases}$$

Projetons ces deux équations dans le plan image grâce aux formules de projections :

eq III 24

$$\begin{cases} x = X/Z \\ y = Y/Z \\ 1/Z = -\frac{(a_1 \cdot x + b_1 \cdot y + c_1)}{d_1} \\ 1/Z = -\frac{(a_2 \cdot x + b_2 \cdot y + c_2)}{d_2} \end{cases}$$

On peut donc par égalité écrire :

$$-\frac{(a_2 \cdot x + b_2 \cdot y + c_2)}{d_2} + \frac{(a_1 \cdot x + b_1 \cdot y + c_1)}{d_1} = 0$$

$$\Leftrightarrow (a_1 \cdot d_2 - a_2 \cdot d_1) \cdot x + (b_1 \cdot d_2 - b_2 \cdot d_1) \cdot y + (c_1 \cdot d_2 - c_2 \cdot d_1) = 0$$

L'équation cherchée est de la forme :  $Ax + By + C = 0$

avec :

$$\begin{cases} A = a_1 \cdot d_2 - a_2 \cdot d_1 \\ B = b_1 \cdot d_2 - b_2 \cdot d_1 \\ C = c_1 \cdot d_2 - c_2 \cdot d_1 \end{cases}$$

Cette représentation n'est pas minimale ; c'est pourquoi nous choisirons une autre représentation n'utilisant que 4 paramètres.

### Représentation polaire

L'équation d'une droite est donnée en utilisant cette représentation par :

*eq III 25*  $g(\underline{X}, \underline{q}) = x \cos \theta + y \sin \theta - \rho = 0$   
avec :  $\underline{q} = (\rho, \theta)$  et  $\underline{X} = (x, y)$

Si nous considérons la représentation de la droite :

$$Ax + By + C = 0$$

on obtient :

$$\left. \begin{matrix} A = \cos \theta \\ B = \sin \theta \end{matrix} \right\} \theta = \arctan\left(\frac{B}{A}\right) \text{ et } \rho = -\frac{C}{\sqrt{A^2 + B^2}}$$

en dérivant  $g(\underline{X}, \underline{q}) = 0$  :

$$\frac{\partial g}{\partial \underline{q}}(\underline{X}, \underline{q}) \cdot \dot{\underline{q}} = - \frac{\partial g}{\partial \underline{X}}(\underline{X}, \underline{q}) \cdot \dot{\underline{X}}$$

*eq III 26*  $\dot{\rho} + \dot{\theta} \cdot (x \cdot \sin \theta - y \cdot \cos \theta) = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$

en utilisant (eq III 25) remplaçons x en fonction de y :

$$x = \frac{\rho - y \sin \theta}{\cos \theta}$$

(eq III 26) devient :

$$\dot{\rho} + \left[ \left( \frac{\rho - y \sin \theta}{\cos \theta} \right) \cdot \sin \theta - y \cdot \cos \theta \right] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\dot{\rho} + [(\rho - y \sin \theta) \cdot \tan \theta - y \cdot \cos \theta] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} - [y \sin \theta \cdot \tan \theta + y \cdot \cos \theta] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} - \left[ y \frac{\sin^2 \theta}{\cos \theta} + y \frac{\cos^2 \theta}{\cos \theta} \right] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\text{eq III 27} \quad -\frac{\dot{\theta}}{\cos \theta} \cdot y + \dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

A partir de l'équation (eq III 25), écrivons  $x$  en fonction de  $y$  si  $\cos \theta \neq 0$  (ou  $y$  en fonction de  $x$  sinon). L'équation (eq III 27) peut alors s'écrire, en utilisant (eq III 23) et (eq III 24) :

$$-\left( \frac{\dot{\theta}}{\cos \theta} \right) \cdot Y + (\dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta}) = Y \cdot K_1 \cdot T + K_2 \cdot T$$

on en déduit aussitôt :

$$\begin{cases} \dot{\theta} = -K_1 \cdot \cos \theta \cdot T \\ \dot{\rho} = (K_2 + K_1 \cdot \sin \theta) \cdot T \end{cases}$$

d'où, en posant  $\underline{S} = (\theta, \rho)^T$  :

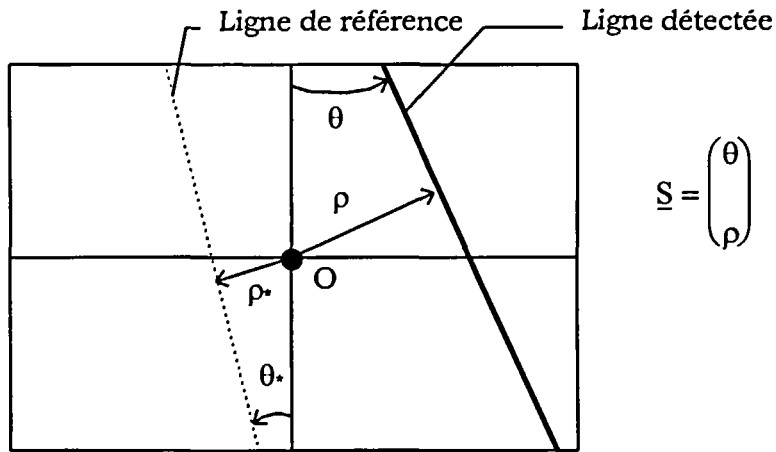
$$\text{eq III 28} \quad \dot{\underline{S}} = \begin{pmatrix} \lambda_{\theta} \cdot \cos \theta & \lambda_{\theta} \cdot \sin \theta & -\lambda_{\theta} \cdot \rho & -\rho \cdot \cos \theta & -\rho \cdot \sin \theta & -1 \\ \lambda_{\rho} \cdot \cos \theta & \lambda_{\rho} \cdot \sin \theta & -\lambda_{\rho} \cdot \rho & (1 + \rho^2) \cdot \sin \theta & -(1 + \rho^2) \cdot \cos \theta & 0 \end{pmatrix} \cdot T = L_{\underline{S}}^T \cdot T$$

$$\text{avec : } \lambda_{\theta} = \frac{1}{d_i} \cdot (a_i \cdot \sin \theta - b_i \cdot \cos \theta) \quad \text{et} \quad \lambda_{\rho} = \frac{1}{d_i} (a_i \cdot \rho \cdot \cos \theta + b_i \cdot \rho \cdot \sin \theta + c_i)$$

Nous venons ainsi de déterminer un lien entre les variations des coordonnées  $(\rho, \theta)$  d'une droite de l'image et les mouvements de notre caméra. Ce lien est la matrice d'interaction semblable à une liaison mécanique virtuelle entre la scène décrite par le mouvement d'une droite et la caméra.

### 3.3 L'approche fonction de tâche

Dans la modélisation de la scène, nous avons considéré une primitive visuelle (une droite) donnée par le vecteur  $\underline{S}$ . A partir de cette information, et par comparaison avec une référence ( $\underline{S}^*$ ) (*Figure III 14*) nous désirons établir une loi de commande pour nos véhicules. Chaumette dans [Chau90] propose une approche par fonction de tâche qui induit un mouvement de la caméra selon les 6 axes en fonction de l'erreur détectée dans les informations visuelles ( $\underline{S}-\underline{S}^*$ ). Cette approche est, en quelque sorte, l'inverse de celle décrite au paragraphe précédent. En effet, on va chercher à exprimer les mouvements de la caméra en fonction de ceux de la droite détectée dans l'image.



**Figure III 14 : détection des droites dans l'image**

Considérons la fonction de tâche (ou d'erreur) donnée par Chaumette pour la commande référencée vision :

eq III 29 
$$e = C \cdot (\underline{S} - \underline{S}^*)$$

> C est une matrice de combinaison permettant d'utiliser un nombre d'informations visuelles supérieur au nombre de degrés de liberté contraints par la fonction de tâche e. Elle traduit l'erreur ( $\underline{S}-\underline{S}^*$ ) exprimée dans l'espace image, dans l'espace cartésien du robot. Cela signifie que l'erreur dans l'image devient, par la matrice C, une erreur sur les axes du robot.

> e est un vecteur à 6 composantes correspondant aux 6 degrés de liberté du robot.  $\underline{S}-\underline{S}^*$  est un vecteur à 2 composantes qui sont les informations visuelles. C est une matrice à 6 lignes et deux colonnes ( $C_1$  et  $C_2$ ). Ecrivons :

$$C = (C_1, C_2)$$



D'après l'équation (eq III 29) nous avons :

$$\text{eq III 30} \quad e = C_1 \cdot (\theta - \theta^*) + C_2 \cdot (\rho - \rho^*) = e_\theta + e_\rho$$

Pour notre application qui vise à asservir un robot évoluant à vitesse moyenne constante par rapport à une droite, nous admettons que l'objet considéré (la droite), ne se déplace pas par rapport à la caméra. Cette hypothèse simplificatrice n'est pas toujours satisfaite pour une trajectoire quelconque. Néanmoins, nous considérons qu'elle est réaliste pour une trajectoire rectiligne ou pour une trajectoire « approximée » par morceaux à une trajectoire rectiligne.

### 3.3.1 Une commande proportionnelle

Classiquement en commande référencée vision nous prenons le torseur cinématique représentant les mouvements de la caméra comme vecteur de commande [Esp90] [Sam91b].

Dans le cas où la primitive reste fixe par rapport au repère caméra, une loi de commande possible est :

$$\text{eq III 31} \quad T = -\lambda \cdot e$$

$\lambda$  étant un réel positif non nul. Ceci correspond à une décroissance exponentielle de la fonction de tâche :

$$\dot{e} = -\lambda \cdot e$$

Nous choisirons une loi de commande similaire donnée par :

$$T = -\lambda \cdot (\beta \cdot e_\theta + e_\rho)$$

$\lambda$  et  $\beta$  étant deux réels positifs non nuls.

Par rapport à la loi donnée par Chaumette (eq III 31), nous avons introduit un gain supplémentaire sur un des deux termes de l'erreur visuelle. La différence entre les deux lois de commande est que, dans notre cas, nous n'imposons pas la même décroissance au cap «  $\theta$  » et à l'erreur latérale «  $\rho$  ». En effet les différents styles de conduite d'un robot mobile imposent que l'on puisse différencier l'importance que l'on donne à l'erreur latérale et au cap. Une conduite précise mais brusque et nécessitant beaucoup d'énergie privilégiera l'erreur latérale, alors qu'une conduite plus « coulée » mais moins précise utilisera davantage le cap.

C'est pourquoi nous imposerons une première décroissance exponentielle à  $e_\theta$  et une autre décroissance exponentielle à  $e_\rho$  en posant :

$$\text{eq III 32} \quad \dot{e} = \dot{e}_\theta + \dot{e}_\rho = -\lambda \cdot (\beta \cdot e_\theta + e_\rho)$$

La fonction de tâche  $e$  est alors bornée par une fonction exponentielle dont la décroissance sera prise égale au plus petit des deux termes  $\lambda \cdot \beta$  et  $\lambda$ . Ce comportement que nous imposons correspond à un système du deuxième ordre dont le temps de réponse et le dépassement seront conditionnés par la donnée de  $\lambda$  et  $\beta$ . (voir simulations chapitre IV). On retrouve le même type de loi de commande appliquant un gain différent sur chaque attribut d'une primitive dans les travaux de Hutchinson [Hut96]. La forme que nous avons retenue est la suivante (les détails du calcul sont donnés en annexe B) :

$$\text{eq III 33} \quad T = -\lambda \cdot L_{\underline{s}=\underline{s}^*}^{T+} \cdot B \cdot (\underline{s} - \underline{s}^*) \quad \text{avec } B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$$

T donne une commande en vitesse contrôlant les mouvements de la caméra en fonction des coordonnées de la droite détectée et de la droite de référence exprimées dans l'espace image. Cette commande ne nécessite aucune reconstruction de la position des droites dans la scène et sera donc très simple à calculer. A la fin de ce chapitre, nous montrerons comment relier le déplacement de la caméra commandé par le vecteur T aux actionneurs de nos véhicules.

### 3.3.2 Une commande proportionnelle intégrale

Pour un processus sans intégration propre, la loi de commande proportionnelle laisse, en général, subsister une erreur résiduelle  $\varepsilon$ . En effet, lorsque l'erreur est nulle, la sortie du correcteur proportionnel est également nulle. Pour diminuer au maximum l'erreur résiduelle, la solution est d'augmenter le gain de la commande. Le gain idéal est infini, mais cette commande de type « tout ou rien » sollicite énormément le système tout en étant instable (en particulier si le système commandé comporte des retards purs...).

Au contraire, la commande proportionnelle intégrale est une commande progressive et persévérante. Tant que subsiste une erreur, la commande croît ou décroît, mais l'annulation de l'erreur n'implique pas forcément une commande nulle. En effet, la sortie du correcteur proportionnel intégral dépend de l'erreur mais aussi du point de fonctionnement considéré. Cette notion peut être illustrée par l'exemple du conducteur automobile qui enfonce progressivement l'accélérateur jusqu'à atteindre la vitesse désirée. Il maintient alors son pied à cette position exacte, assurant une vitesse constante.

Dans ce cas l'équation de commande que nous retiendrons sera de la forme :

$$T = -\lambda \cdot \left( \beta \cdot \left( e_\theta + k_1 \int_0^t e_\theta \cdot dx \right) + e_\rho + k_2 \int_0^t e_\rho \cdot dx \right)$$

$\lambda$ ,  $\beta$ ,  $k_1$  et  $k_2$  étant des réels positifs.

En procédant de la même manière qu'au paragraphe précédent (les calculs sont donnés en annexe B) on obtient :

*eq III 34*

$$T = -\lambda \cdot L_{\underline{s}=\underline{s}^*}^T \cdot B \cdot \left[ (\underline{S} - \underline{S}^*) + K_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot dx \right]$$

avec :

- T torseur cinématique représentant les mouvements de la caméra,

-  $\lambda \in \mathcal{R}^{*+}$  et  $B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$  gains proportionnels,

-  $L_{\underline{s}=\underline{s}^*}^{T+}$  torseur d'interaction calculé à l'équilibre,

-  $K_i = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}$  matrice de gains intégrals.

Nous avons donc calculé une loi de commande utilisant un correcteur proportionnel intégral. Son entrée est dans le cas de la commande référencée vision l'information visuelle ( $\underline{S} - \underline{S}^*$ ) exprimée dans le repère caméra. Cette dernière loi comporte l'avantage d'accroître la robustesse quant aux erreurs de modélisation de notre système ou des perturbations basses fréquences auxquelles il sera soumis (cf chapitre IV).

### Remarque

Pour les deux lois de commande précédemment citées nous utilisons la matrice d'interaction calculée à la position d'équilibre. Ceci représente une approximation forte mais qui ne remet pas en cause la convergence du système même pour des conditions initiales très éloignées de la position d'équilibre (cf chapitre IV). Un apprentissage permet d'acquérir les coordonnées de la position d'équilibre ( $\underline{S}^*$ ). Pour cela le véhicule suit une ligne droite sur une distance de quelques mètres en étant correctement asservi. Une simple moyenne des informations visuelles donne la position d'équilibre. Les erreurs de modélisation seront compensées par la boucle fermée.

### 3.4 Application des deux lois de commande référencée vision à nos véhicules.

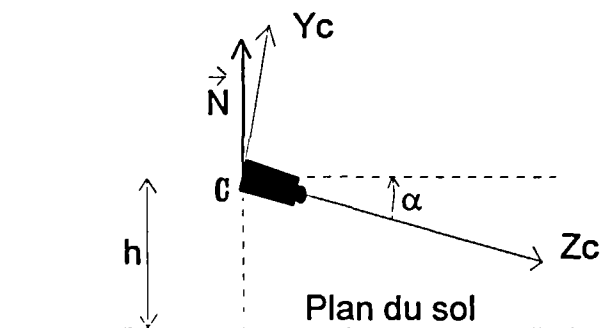
Les deux lois de commande que nous venons d'établir comportent plusieurs paramètres :

- des gains  $(\lambda, \beta, k_1, k_2)$ ,
- $T$  le torseur d'interaction calculé à l'équilibre,
- $\underline{S}$  l'information visuelle courante,
- $\underline{S}^*$  l'information visuelle à l'équilibre.

Les gains seront ajustés lors de nos essais en simulation. Nous allons maintenant calculer la matrice d'interaction au point de fonctionnement  $\underline{S}=\underline{S}^*$ .

#### 3.4.1 Calcul de la matrice d'interaction pour $\underline{S}=\underline{S}^*$

L'algorithme de traitement d'images donne les coordonnées polaires de l'interface détectée, par rapport à une ligne de référence. Celle-ci correspond à la ligne détectée lorsque la machine est correctement asservie. Dans ces conditions, la nouvelle ligne de référence a pour coordonnées  $(\rho^*=0, \theta^*=0)$  dans le repère image. Dans la scène, cette droite de référence appartient au plan du sol et au plan  $x=0$  (exprimés tous les deux dans le repère  $(C X_c Y_c Z_c)$ ) (cf **Figure III 15**).



**Figure III 15 : Situation de la caméra**

Calculons l'équation du plan du sol exprimée dans le repère capteur  $(C, X_c, Y_c, Z_c)$  :

Soit un vecteur normal au plan du sol :  $\vec{N} \begin{pmatrix} 0 \\ \cos\alpha \\ \sin\alpha \end{pmatrix}$

L'équation du plan est de la forme :

$$(\cos\alpha).y + (\sin\alpha).z + cte = 0$$

Soit M, un point du plan :

$$M \begin{pmatrix} 0 \\ -h/\cos\alpha \\ 0 \end{pmatrix}$$

h étant la hauteur de la caméra par rapport au sol (h>0)

$$\implies (\cos\alpha).(-\frac{h}{\cos\alpha}) + 0 + cte = 0 \implies cte = h$$

L'équation du plan du sol est :

$$\underline{(\cos\alpha).y + (\sin\alpha).z + h = 0}$$

Equation de la droite de référence en polaire dans le repère image

L'équation de la droite de référence dans le repère (C, X<sub>c</sub>, Y<sub>c</sub>, Z<sub>c</sub>) est :

$$\begin{cases} (\cos\alpha).y - (\sin\alpha).y + h = 0 \\ x = 0 \end{cases}$$

En utilisant la même méthode que celle développée au paragraphe « Matrice d'interaction pour une primitive de type droite » nous pouvons calculer les coordonnées polaires de la droite de référence exprimées dans le repère image :

$$\begin{cases} \rho^* = 0 \\ \theta^* = 0 \end{cases} \quad \text{d'où : } \underline{S^*} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

D'après ces équations on trouve :

$$\lambda_{\theta^*} = \frac{1}{h}(-(\cos\alpha)) \quad \text{et} \quad \lambda_{\rho^*} = \frac{1}{h}(\sin\alpha)$$

La matrice d'interaction est donc égale à :

$$L_{\underline{s}^*-\underline{s}^*}^T = \begin{pmatrix} -\frac{\cos\alpha}{h} & 0 & 0 & 0 & 0 & -1 \\ \frac{\sin\alpha}{h} & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

### 3.4.2 Expression des lois de commande pour l'essieu rigide

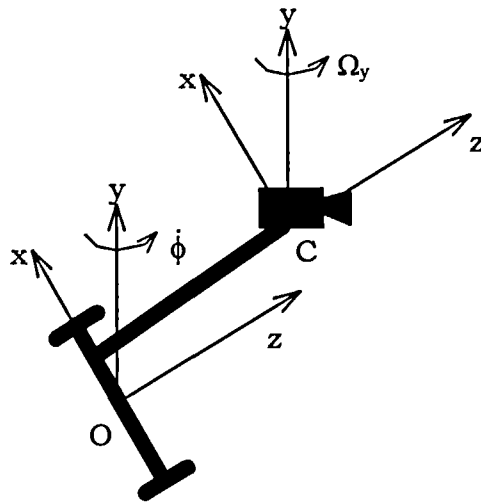
Dans nos lois de commande nous utilisons la pseudo-inverse de  $L_{S=S}^T$  qui est donnée par :

$$L_{S=S}^{T+} = L_{S=S}^T \cdot (L_{S=S}^T \cdot L_{S=S}^T)^{-1} = \begin{pmatrix} -\frac{h \cdot \cos \alpha}{1+h^2} & \frac{h \cdot \sin \alpha}{1+h^2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{(\cos \alpha \cdot \sin \alpha)}{1+h^2} & -\frac{(\cos^2 \alpha + h^2)}{1+h^2} \\ -\frac{(\sin^2 \alpha + h^2)}{1+h^2} & -\frac{(\cos \alpha \cdot \sin \alpha)}{1+h^2} \end{pmatrix}$$

En reportant  $L_{S=S}^{T+}$  dans (eq III 33) nous obtenons :

$$\text{eq III 35} \quad T = \begin{pmatrix} V_x \\ V_y \\ V_z \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} = -\lambda \cdot \begin{pmatrix} -\frac{h \cdot \cos \alpha}{1+h^2} & \frac{h \cdot \sin \alpha}{1+h^2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{(\cos \alpha \cdot \sin \alpha)}{1+h^2} & -\frac{(\cos^2 \alpha + h^2)}{1+h^2} \\ -\frac{(\sin^2 \alpha + h^2)}{1+h^2} & -\frac{(\cos \alpha \cdot \sin \alpha)}{1+h^2} \end{pmatrix} \cdot B \cdot \begin{pmatrix} \theta - 0 \\ \rho - 0 \end{pmatrix}$$

Le torseur cinématique T représente les mouvements de la caméra par rapport au repère  $R_0$ . Dans nos applications, nous considérons une liaison rigide entre le véhicule et la caméra. Les mouvements appliqués à la caméra le seront donc aussi au véhicule (cf **Figure III 16**).



**Figure III 16 : rotation de l'essieu rigide et de la caméra.**

L'orientation de la caméra par rapport à l'axe  $Oy$  est la même que celle de l'essieu rigide et donc la même que celle du véhicule tout entier.

Nous avons donc :

$$\dot{\phi} = \Omega_y$$

Nous avons vu précédemment que l'essieu rigide à deux roues était contrôlé par son orientation lorsqu'il évoluait à vitesse moyenne constante (cf chapitre II). Nous allons donc utiliser la 5<sup>ème</sup> équation du système (eq III 35) afin de contrôler nos engins qui évoluent à vitesse moyenne constante. Cette équation donne la vitesse de rotation de la caméra, donc du véhicule, par rapport à l'axe vertical  $Oy$ .

Nous obtenons à partir de (eq III 35) :

eq III 36

$$\dot{\phi} = \Omega_y = \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right]$$

Cette équation correspond à la loi de commande référencée vision proportionnelle. Pour obtenir la loi de commande référencée vision proportionnelle intégrale, il suffit de remplacer (eq III 33) par (eq III 34) dans les calculs précédents. Nous obtenons alors l'équation de commande proportionnelle intégrale suivante :

eq III 37

$$\dot{\phi} = \Omega_y = \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \left( \theta + k_1 \cdot \int_0^t \theta \cdot du \right) + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \left( \rho + k_2 \cdot \int_0^t \rho \cdot du \right) \right]$$

D'après l'équation 5 du chapitre II nous avons :

$$\begin{cases} \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \\ v_{\text{moy}} = \frac{r \cdot (\dot{q}_1 + \dot{q}_2)}{2} \end{cases}$$

nous obtenons donc :

**eq III 38**

$$\begin{cases} \dot{q}_1 = \frac{2v_{\text{moy}} - b \cdot \dot{\phi}}{2r} \\ \dot{q}_2 = \frac{2v_{\text{moy}} + b \cdot \dot{\phi}}{2r} \end{cases}$$

Les équations (eq III 36), (eq III 38) d'une part, et (eq III 37), (eq III 38) d'autre part, nous donnent une loi de commande proportionnelle et proportionnelle intégrale pour l'essieu rigide à deux roues. Il est maintenant aisé de l'étendre à la tondeuse ou à la moissonneuse-batteuse compte tenu des remarques faites au chapitre II.

#### Exemple d'application : la moissonneuse-batteuse

Le calcul de la loi de commande pour la moissonneuse-batteuse se fait à partir de celle de l'essieu rigide (eq III 38) et de la matrice A calculée au chapitre II.

$$A = \begin{pmatrix} \frac{1}{r} & \frac{b}{2Lr} \\ \frac{1}{r} & \frac{-b}{2Lr} \end{pmatrix} \text{ avec : } \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = A \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

Rappelons que les deux actionneurs de la moissonneuse-batteuse commandent  $v$  et  $\delta$  ; on a alors :

**eq III 39**

$$\begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

De (eq III 38) (eq III 39) nous déduisons :

$$\begin{cases} v = v_{\text{moy}} \\ \tan \delta = -\frac{L \cdot \dot{\phi}}{v_{\text{moy}}} \end{cases}$$



$\dot{\phi}$  est donné par (eq III 36) (eq III 37) suivant que l'on choisisse une loi de commande proportionnelle ou proportionnelle intégrale.

Les deux actionneurs de la moissonneuse-batteuse étant ( $\delta$  et  $v$ ), l'équation précédente permet de les commander en fonction de la variation des coordonnées de la droite détectée dans l'image. Nous rappelons que la vitesse moyenne de l'engin doit rester constante (hypothèse de départ) et que seul l'angle des roues directrices est commandable.

### **Conclusion**

Le formalisme de la commande référencée capteur permet de calculer une relation liant les mouvements de l'interface fauchée/non-fauchée dans l'image à ceux de la caméra. On en déduit des lois de commande par retour proportionnel ou proportionnel intégral pour le déplacement de l'essieu auquel est lié cette caméra. Au moyen de la modélisation décrite au chapitre II, les mouvements de l'essieu rigide de l'engin sont exprimés en fonction du contrôle de ses actionneurs. Aussi avons nous calculé une loi de commande référencée vision pour nos véhicules que nous expérimenterons au chapitre suivant.

Notons que la matrice d'interaction est calculée uniquement pour la position d'équilibre. Aussi le lien entre l'image et la scène n'est réalisé que pour une seule position de la caméra et n'est pas recalculé à chaque itération. Chaumette a montré que cela n'empêchait pas la convergence de la loi de commande même pour une position initiale éloignée de la position d'équilibre. Les équations mises en jeu pour commander nos véhicules démontrent une simplicité évidente quant à leur mise en oeuvre et laisse supposer des temps de calcul relativement faibles.

## CHAPITRE IV

### SIMULATIONS ET ESSAIS REELS

## **1. SIMULATION DES DEUX LOIS DE COMMANDE**

Nous présentons, dans cette première partie, différents résultats de simulation des lois de commande développées au chapitre III. Nous exposerons diverses situations qui tenteront d'intégrer le maximum d'éléments présents dans le processus réel de fonctionnement. Par exemple, le retard pur et la période d'échantillonnage de l'algorithme de traitement d'images seront pris en compte par les simulateurs ainsi que la pente du terrain ou encore la précision des informations visuelles délivrées par l'algorithme de traitement d'images. Nous modéliserons également la caméra mais nous négligerons tous les paramètres dynamiques de nos véhicules (pneumatiques, hydrauliques, adhérence des roues...).

Pour cela nous avons développé deux simulateurs :

- le premier, en langage C, a des possibilités très limitées (trajectoire rectiligne sur terrain plat). Il a été utilisé au début de nos travaux afin de tester notre première loi de commande,
- le second a été programmé sous un langage évolué « MATLAB » et ses possibilités sont beaucoup plus importantes. Il peut simuler des trajectoires rectilignes ou courbes ; il peut également intégrer une pente au terrain ; enfin, il permet de tester les réactions de nos machines dans diverses situations.

## 1.1 La commande par reconstruction de trajectoire

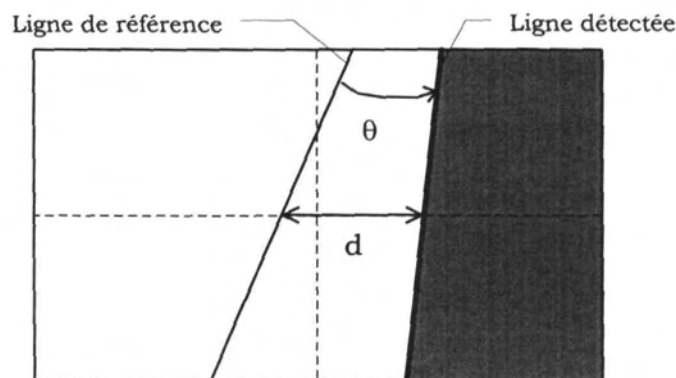
La première loi de commande développée en 1993 et appelée « commande trajectoire » (cf. chapitre III) a été implantée sur un simulateur réalisé en langage C. Celui-ci permet de suivre une trajectoire rectiligne en faisant varier les conditions initiales ainsi que la qualité des informations visuelles. A cette époque, l'engin dont nous disposions était la tondeuse. C'est donc son modèle qui a été retenu dans ce simulateur et pour cette première loi de commande.

Notre système d'aide à la conduite se décompose en deux parties qui sont la perception et le contrôle de l'engin.

### **Détaillons les entrées et les sorties de ces deux systèmes :**

#### - Le traitement d'images

Les informations visuelles de notre système sont données par les coordonnées de la droite modélisant l'interface fauchée/non fauchée détectée dans l'image (**Figure IV 1**)



**Figure IV 1 : données visuelles**

Ces informations visuelles sont :

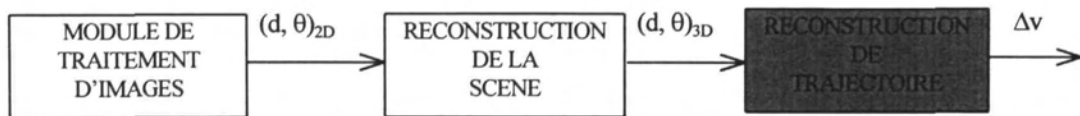
- $d$  la distance au centre de l'image entre la ligne de référence et la ligne détectée,
- $\theta$  l'angle entre la droite détectée et la ligne de référence.

#### Remarque

L'algorithme de traitement d'images fournit directement les coordonnées de la droite détectée en fonction de celle de référence. Aussi, dans tous nos calculs, la ligne de référence aura pour coordonnées ( $d=0$ ,  $\theta=0$ ). Au chapitre III § 1.4, nous avons vu que la période d'échantillonnage de l'algorithme de traitement d'images était en moyenne de 300 ms avec un retard pur d'au moins 600 ms.

### - L'algorithme de commande

Pour asservir le véhicule sur l'interface fauchée/non-fauchée, nous allons reconstruire une trajectoire lui permettant de passer de sa position courante à la position de l'interface. Cette technique suppose de reconstruire la position 3D de l'interface détectée dans l'image puis de calculer un chemin réalisable par la machine pour rejoindre cette position désirée. Dans le simulateur comme dans l'application finale un algorithme de reconstruction 2D 3D permet de faire le lien entre les données visuelles et l'entrée du système de commande. La sortie est la consigne ( $\Delta v$ ) appliquée au correcteur bas niveau qui contrôle la direction du véhicule.

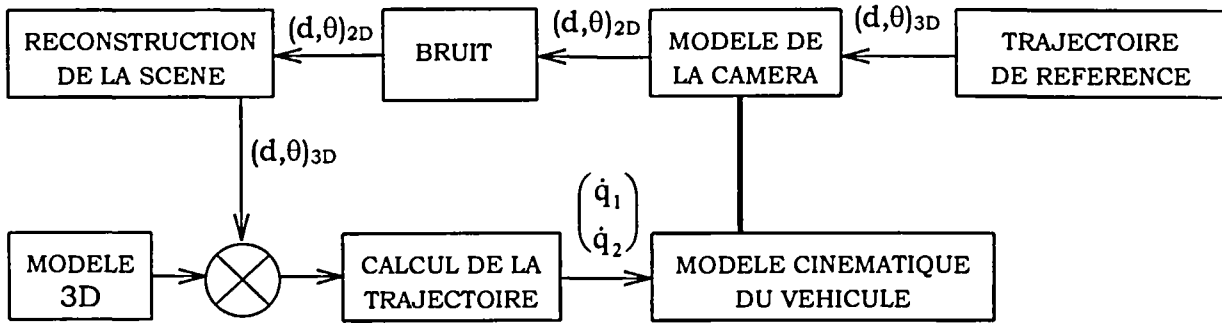


**Figure IV 2 : variables du système**

### 1.1.1 Le simulateur

Le simulateur est composé de sept modules (**Figure IV 3**) :

- un modèle de la caméra calculé à partir de la méthode de Faugeras et Toscani. Ce modèle permet de convertir les données de la scène en données visuelles captées par une caméra,
- un module permettant de bruitez les données extraites de la caméra. Ce module introduit également un retard correspondant au retard engendré par le traitement d'images,
- un module permettant de reconstruire la scène à partir des données 2D (ce module suppose que toutes les données reconstruites appartiennent au plan modélisant le sol),
- un modèle 3D de la scène représentant la position correcte de la machine,
- un module permettant de calculer la trajectoire à suivre,
- un modèle cinématique de la machine,
- un module donnant la trajectoire de référence (pour ce simulateur, la seule possibilité est une droite).



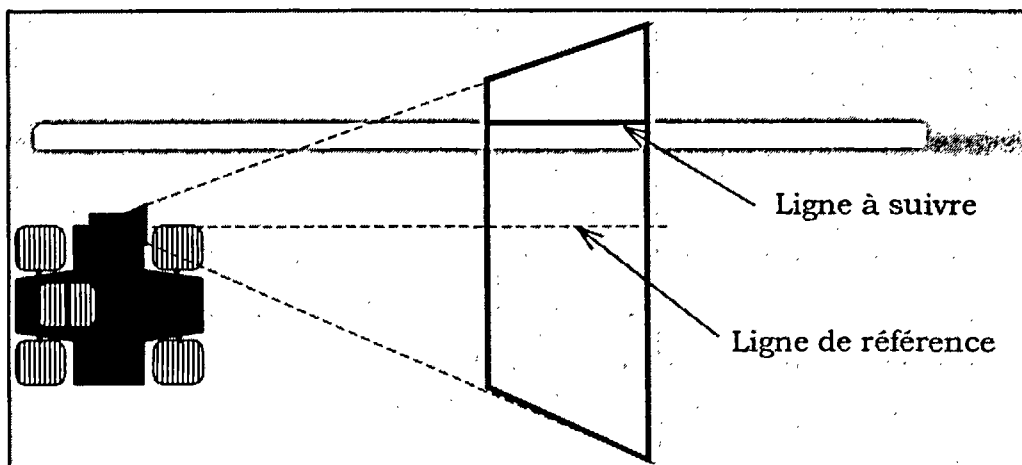
**Figure IV 3 : schéma de fonctionnement du simulateur**

### Remarque

Lorsque nous avons identifié la tondeuse [deb94] nous avons constaté que, malgré la présence d'un correcteur bas niveau qui régule la vitesse des roues, il y a un retard pur de l'ordre de 300 ms. Aussi, dans le simulateur, nous avons modifié le modèle cinématique afin de tenir compte de ce retard.

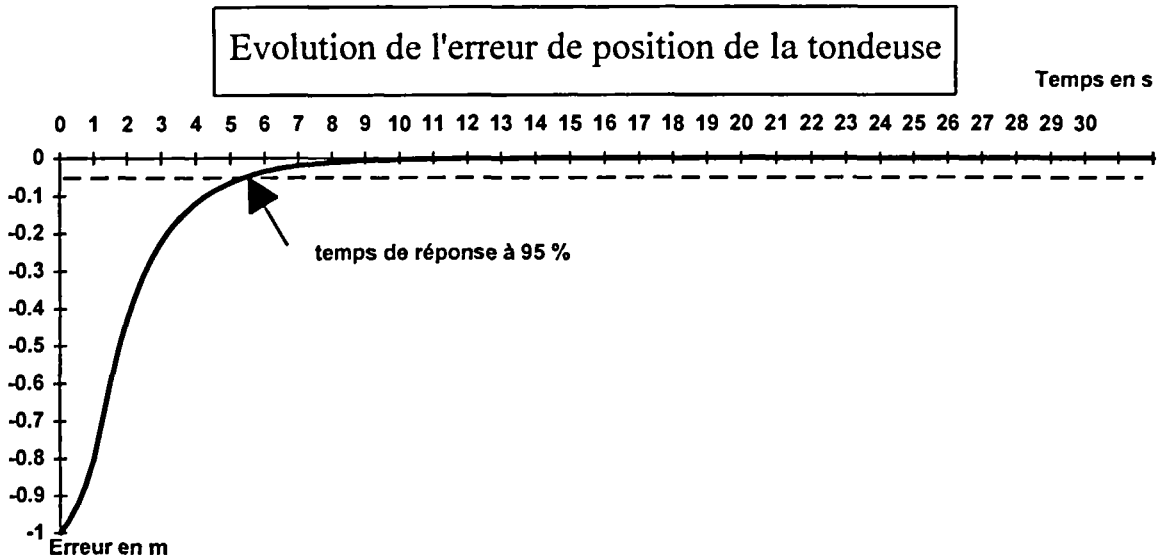
### 1.1.2 Résultats

Les courbes que nous allons présenter ont été réalisées en simulant une erreur initiale d'un mètre entre la machine et l'interface (**Figure IV 4**). Nous n'avons pas introduit d'erreur initiale sur l'orientation de la machine car une erreur en position induit inéluctablement une erreur d'orientation lors de sa correction. Ceci est dû à la non-holonomie de notre machine faisant que tout déplacement latéral de l'engin ne peut être introduit que par un changement de cap (cf. chapitre II). La période d'échantillonnage est fixée à 300 ms et les essais se déroulent sur un intervalle de temps de l'ordre de 30 s. Le terrain est considéré comme plat et nous négligeons toute perturbation.

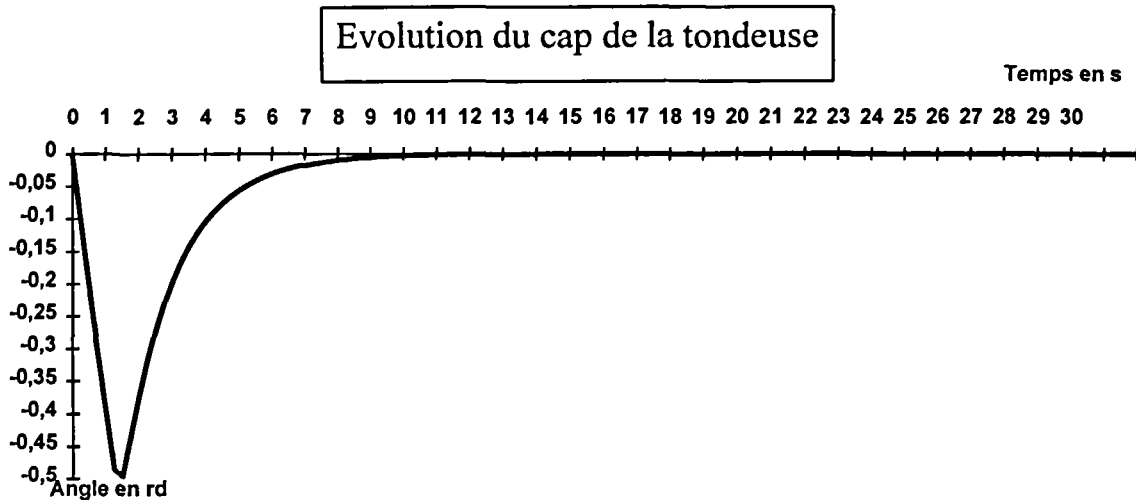


**Figure IV 4 : conditions initiales de simulation**

Les deux premières courbes présentées donnent la réponse de la machine à un échelon d'un mètre lorsque celle-ci avance à une vitesse moyenne de 3 km/h .



**Courbe IV 1 : réponse de la tondeuse à un échelon d'un mètre**



**Courbe IV 2 : orientation de la machine lors de sa réponse à un échelon d'un mètre**

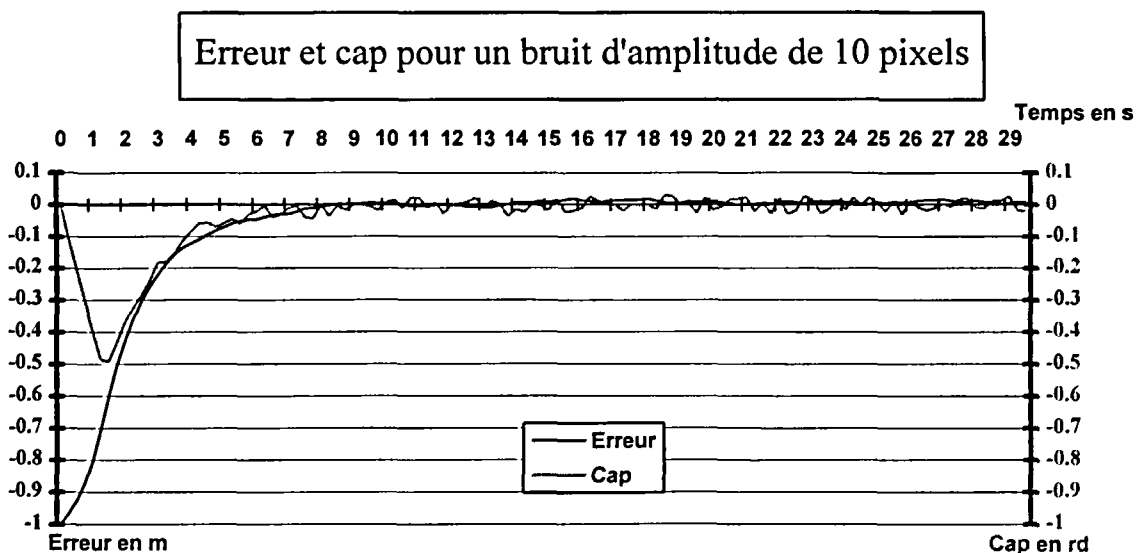
La première courbe (**Courbe IV 1**) donne l'erreur de positionnement de la machine par rapport à la trajectoire à suivre. On peut constater qu'une erreur initiale d'un mètre est compensée en 8 s. La deuxième courbe (**Courbe IV 2**) donne l'erreur d'orientation de la machine par rapport à la tangente de la trajectoire. Au

départ cette erreur est nulle, puisque nous avons positionné la machine parallèlement à la trajectoire. Puis l'erreur croît pendant environ 2 s avant de décroître pour s'annuler en même temps que l'erreur de position. On peut constater la présence d'un petit segment au sommet de la courbe de l'erreur d'orientation. Il est provoqué par la saturation des actionneurs de la machine. En effet, des contraintes mécaniques interdisent une différence de vitesse ( $\Delta v$ ) supérieure à 2 km/h et ceci est pris en compte dans le simulateur.

Ces deux courbes montrent un bon asservissement de la machine qui compense l'erreur initiale en moins de 10 s. Elles montrent également la stabilité de la méthode utilisée puisqu'on ne constate aucun dépassement ni aucune oscillation. Le temps de réponse dépend essentiellement de la distance entre la machine et le centre de l'image. Si cette distance croît, les deux cercles auront un rayon de courbure plus grand et le temps de réponse sera plus important, si au contraire cette distance diminue le temps de réponse aura tendance à décroître.

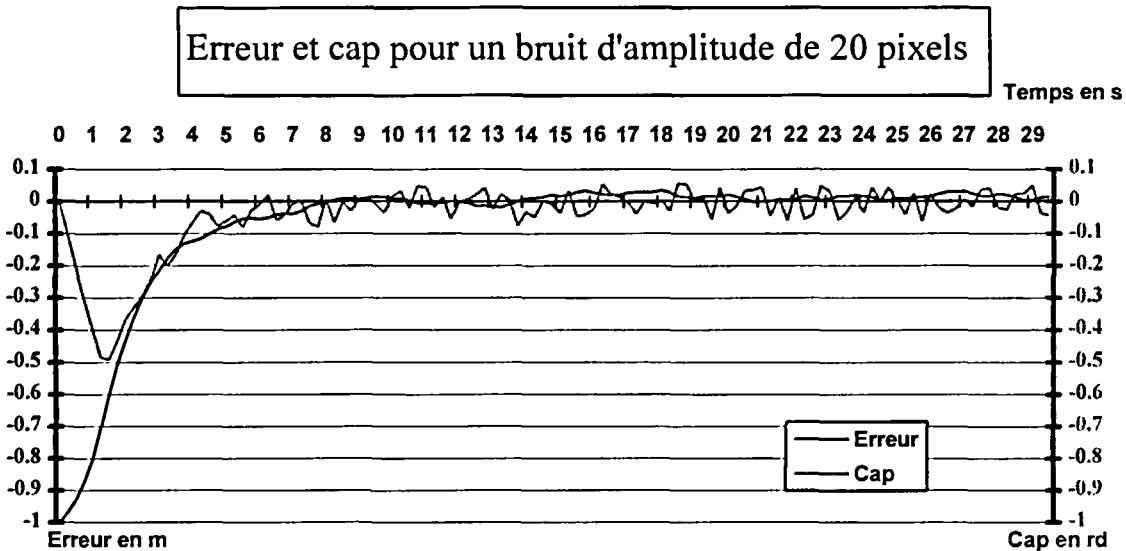
### 1.1.2.1 Influence du bruit sur la commande trajectoire

Notre machine évolue en milieu extérieur ; de ce fait, elle est soumise à de nombreuses perturbations (terrain non uniforme, interface bruitée). Aussi, nous avons décidé de tester notre algorithme en bruitant les mesures visuelles. Pour cela, nous avons rajouté un bruit à moyenne nulle sur la position de l'interface. Les résultats sont présentés sur les courbes suivantes pour différentes amplitudes de bruit.



***Courbe IV 3 : réponse indicelle pour un bruit de 10 pixels***





***Courbe IV 4 : réponse indicielle pour un bruit de 20 pixels***

On constate un effet de filtre de l'algorithme puisque les mesures sur l'erreur montrent une imprécision de la machine de l'ordre de l'amplitude du bruit divisée par 2 (un calibrage grossier de la caméra donne 1 cm pour 1 pixel). On observe, par contre, des variations du cap de la machine relativement importantes. Ceci est très gênant dans le cadre d'un asservissement visuel car des variations importantes de cap entraînent des variations importantes de l'image à traiter. Or, l'algorithme de traitement d'images que nous employons utilise le résultat de l'image  $n$  comme hypothèse de départ pour l'image  $n+1$ . De plus, l'interface risque de sortir de l'image.

Lors des essais de la tondeuse en conditions réelles, nous avons observé que ce comportement « trop brusque » entraînait de nombreuses difficultés. Ceci est, à mon avis, particulièrement important et trop souvent négligé. L'interaction vision commande est un fait réel dont nous reparlerons au chapitre V.

## 1.2 La Commande Référencée Vision (CRV)

Parallèlement au développement de la commande « trajectoire » nous avons étudié un nouveau formalisme modélisant le lien contrôle-perception dans l'espace du capteur. Ce formalisme développé par Claude Samson et Bernard Espiau est appelé « Commande Référencée Capteur » et a été appliqué avec succès aux robots manipulateurs percevant leur environnement par vision [Chau90]. Ce procédé comporte l'avantage de ne pas reconstruire la scène et d'être peu sensible au bruit. Nous avons donc synthétisé un correcteur pour nos robots mobiles utilisant les mêmes principes.

A cette occasion, nous avons développé un nouveau simulateur utilisant le logiciel MATLAB. Ce logiciel intègre tous les procédés de calcul matriciel et permet de programmer très facilement toutes sortes de phénomènes physiques. Avec notre premier simulateur, il nous paraissait difficile et surtout fastidieux de programmer des trajectoires complexes avec des virages, des échelons, des terrains en pente....C'est pourquoi nous avons décidé de le porter sous MATLAB mais en l'enrichissant de nouvelles fonctionnalités.

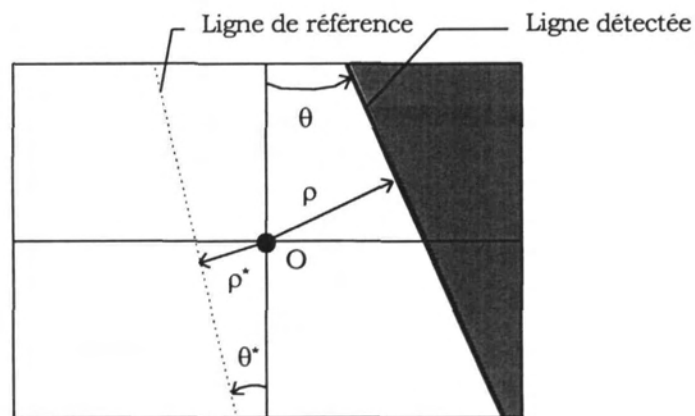
Les résultats que nous présenterons dans la suite de ce chapitre ont été réalisés en employant le modèle cinématique de la moissonneuse-batteuse. Cependant, comme nous l'avons montré au chapitre II, le comportement cinématique des deux engins est semblable. Aussi, les résultats que nous obtiendrons seront facilement généralisables au premier véhicule et à bien d'autres (cf. chapitre V).

Comme pour le premier simulateur, nous avons tenté de tenir compte des particularités de l'algorithme de traitement d'images et de la commande référencée vision.

### Détaillons les entrées et les sorties de ces deux modules :

#### - Le traitement d'images

Pour la seconde loi de commande les informations visuelles de notre système sont données par les coordonnées polaires de la droite modélisant l'interface fauchée/non-fauchée détectée dans l'image. Le centre de l'image se situe 6.5 m devant la machine.



**Figure IV 5 : données visuelles.**

Ces informations visuelles sont :

- $(\rho - \rho^*)$  la distance au centre de l'image entre la ligne de référence et la ligne détectée,
- $(\theta - \theta^*)$  l'angle entre la droite détectée et la ligne de référence.

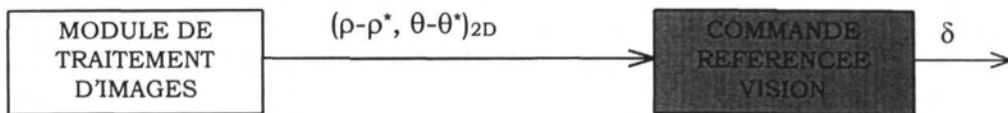
#### Remarque

L'algorithme de traitement d'images fournit directement les coordonnées de la droite détectée en fonction de celle de référence. Au chapitre III nous avons vu que ces coordonnées étaient l'angle avec la verticale et l'ordonnée à l'origine. Nous les avons donc traduites dans un repère polaire pour les exprimer sous la forme  $(\rho, \theta)$ . Dans tous nos calculs, la ligne de référence aura pour coordonnées  $(\rho=0, \theta=0)$ . Ceci introduit un changement de repère dont nous ne tenons pas compte dans le calcul de notre matrice d'interaction. Cette approximation revient à approcher le torseur d'interaction calculé pour n'importe quelle droite de l'image par celui donné pour la droite  $(\rho=0, \theta=0)$ .

Au chapitre III, nous avons montré que la période d'échantillonnage de l'algorithme de traitement d'images était de l'ordre de 300 ms avec un retard pur de 600 ms.

### - L'algorithme de commande

Pour asservir le véhicule sur l'interface fauchée/non-fauchée, on utilise ici une loi de « commande référencée vision ». Celle-ci est composée d'une interaction entre la scène et les mouvements de la caméra et d'un modèle cinématique de l'engin reliant les mouvements de la caméra aux actionneurs de la moissonneuse-batteuse. Les entrées de la loi de commande sont directement les coordonnées polaires de la droite détectée dans l'image en fonction de la droite de référence.



**Figure IV 6 : variables du système**

Pour la moissonneuse-batteuse, l'équation donnée par la commande référencée vision proportionnelle est (chapitre III) :

$$\begin{cases} v = v_{\text{moy}} \\ \tan \delta = -\frac{L \cdot \dot{\phi}}{v_{\text{moy}}} = -\frac{L \cdot \lambda}{v_{\text{moy}}} \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right] \end{cases}$$

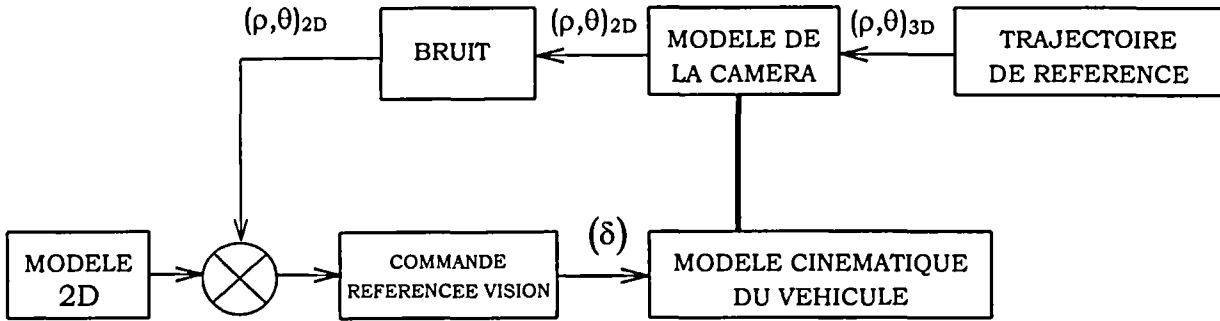
Cette équation donne l'angle des roues du véhicule ( $\delta$ ) en fonction de la position de la caméra ( $\alpha$  et  $h$ ), des paramètres géométriques de l'engin ( $L$ ), de sa vitesse moyenne ( $v_{\text{moy}}$ ) et des informations visuelles ( $\rho$  et  $\theta$ ).  $\lambda$  et  $\beta$  sont deux gains permettant de paramétrer la dynamique du processus de commande.

### 1.2.1 Le simulateur

Le simulateur développé sous MATLAB est composé de 6 modules (**Figure IV 7**) :

- un modèle de la caméra permettant de convertir les données de la scène en données caméra,
- un module permettant de bruitez les données extraites de la caméra. Ce module introduit également un retard correspondant au retard engendré par le traitement d'images (600 ms),
- un modèle 2D de la scène représentant la position correcte de la machine,
- un module de commande référencée vision,

- un modèle cinématique de la machine,
- un module donnant la trajectoire de référence (pour ce simulateur, nous avons de nombreuses possibilités : droites, courbes, terrain avec ou sans pente...).



**Figure IV 7 : schéma de fonctionnement du simulateur**

Remarques

Le correcteur bas niveau implanté sur la moissonneuse-batteuse permet de contrôler l'angle de direction du véhicule avec un retard pur de 200 ms et une période de 20 ms. Le modèle cinématique de la machine sera modifié afin de tenir compte des 200 ms du retard pur.

**1.2.2 Résultats**

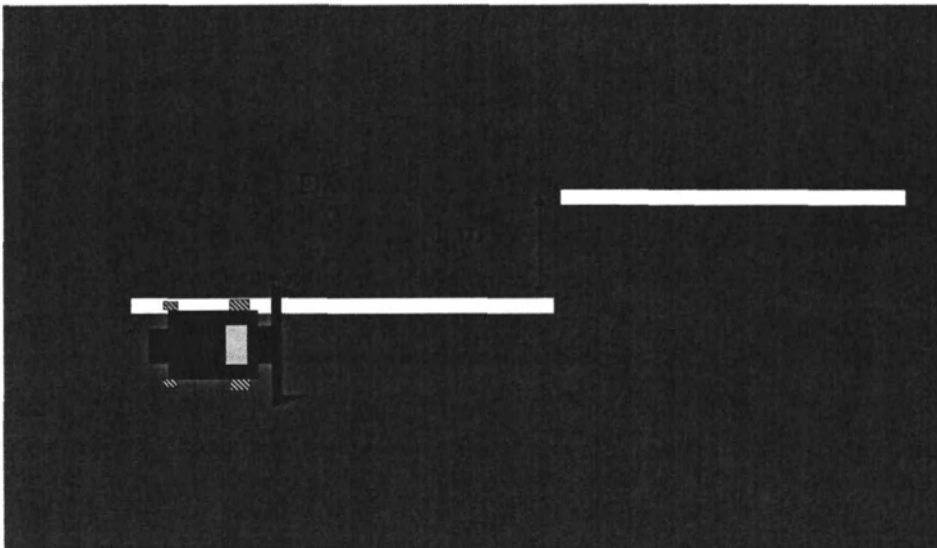
Les conditions de simulations sont les mêmes qu'au paragraphe précédent si ce n'est quelques paramètres liés à la géométrie de la machine (place de la caméra, saturation des actionneurs,...). Dans un premier temps, une trajectoire rectiligne incluant un échelon initial d'un mètre est simulée. Puis nous rajouterons un bruit sur les informations visuelles afin de se rapprocher le plus possible des conditions réelles de travail de la machine agricole. La première loi testée avec la moissonneuse-batteuse est la loi de commande référencée vision calculée au chapitre III. Avec les approximations d'usage aux petits angles, son équation est :

eq IV 1 
$$\begin{cases} v = v_{\text{moy}} \\ \delta = -\frac{L \cdot \dot{\phi}}{v_{\text{moy}}} = -\frac{L \cdot \lambda}{v_{\text{moy}}} \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right] \end{cases}$$

Il faut maintenant déterminer les gains  $\lambda$  et  $\beta$ . Ils sont calculés en deux étapes : la première consiste à comparer la réponse du système en boucle ouverte par rapport à celle d'un conducteur. La deuxième sert à optimiser les gains déterminés précédemment en les faisant varier et en comparant les réponses indicelles obtenues.

### Première étape

Le principe de cette première étape est d'identifier en boucle ouverte la sortie de la loi de commande et les corrections d'un conducteur. Pour cela, un échelon d'un mètre est dessiné sur le sol puis il est demandé au conducteur de diriger la machine afin de suivre au mieux la ligne blanche. Le point que le conducteur doit asservir sur le bord extérieur gauche de la ligne blanche est le point D qui symbolise l'extrémité du diviseur gauche de la moissonneuse batteuse. Les équations de commande données au chapitre III § 3 concernent l'asservissement de la caméra et non pas celui d'un autre point du véhicule. Cette différence est prise en compte par l'intermédiaire de la droite de référence qui est « apprise » lorsque le point D de la machine est superposé au bord gauche de la ligne blanche et que l'engin est parallèle à cette ligne. Nous avons montré que nous ne commandons que l'orientation de nos véhicules, ce qui permet d'assimiler les mouvements du point D à ceux du centre optique de la caméra.

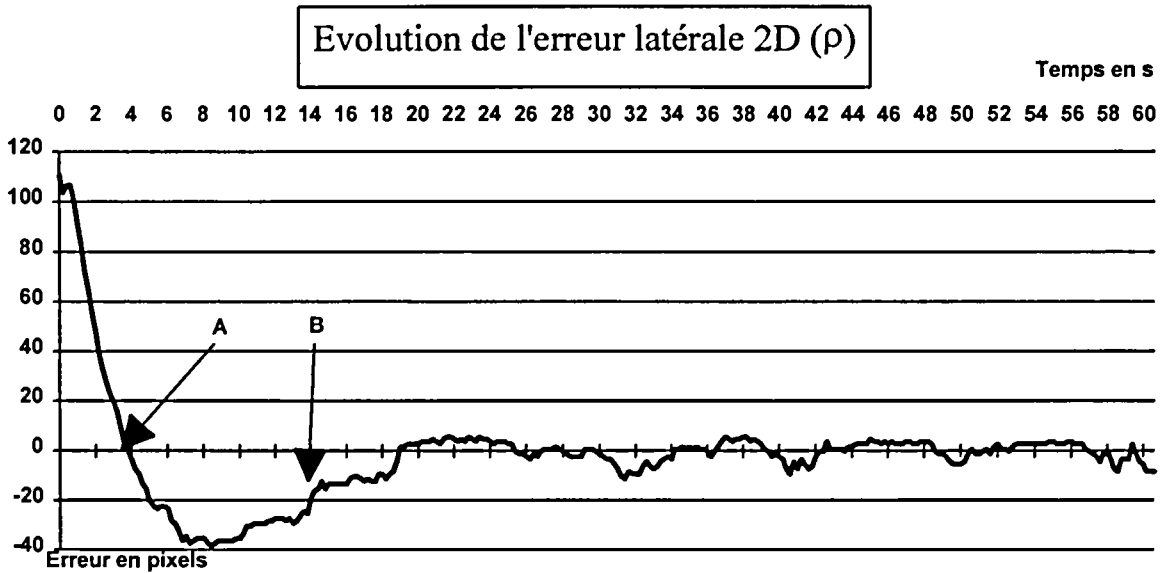


**Figure IV 8 : piste à suivre par le conducteur**

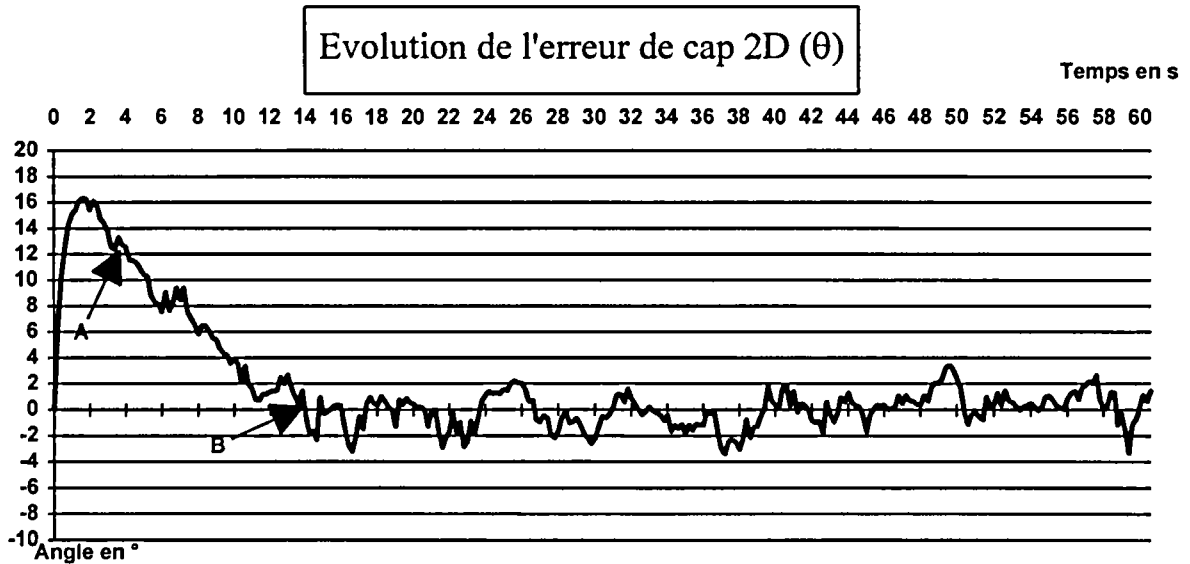
Lors de cette expérience, trois grandeurs sont enregistrées :

- $\delta$  l'angle des roues directrices,
- $\rho$  et  $\theta$  les coordonnées polaires du bord gauche de la ligne blanche détectée dans l'image.

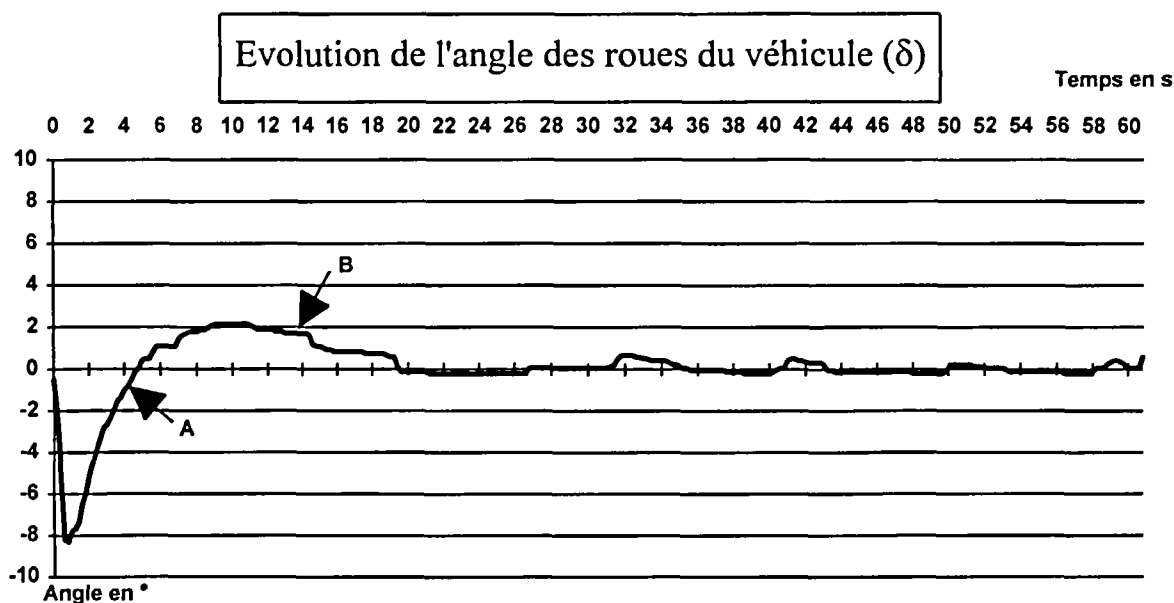
Donnons les courbes représentant ces trois variables :



*Courbe IV 5 : réponse indicielle d'un conducteur*



*Courbe IV 6 : réponse indicielle d'un conducteur*



**Courbe IV 7 : réponse indicielle d'un conducteur**

A partir de ces trois courbes il faut déterminer les gains de l'équation (eq IV 1) en identifiant  $\delta$  à celui mesuré sur la courbe (**Courbe IV 7**). L'équation de commande (eq IV 1) est de la forme :

$$\text{eq IV 2} \quad \delta = P \cdot \rho + Q \cdot \theta$$

P et Q étant deux constantes dépendant de la position de la caméra, de la géométrie de l'engin et de sa vitesse moyenne.

Utilisons les points A et B des courbes précédentes afin de déterminer P et Q. Au point A,  $\rho$  est nul et nous pouvons calculer Q. Au point B,  $\theta$  est nul et nous pouvons calculer P. Il est évident que l'approximation faite en reliant les trois courbes par l'équation (eq IV 2) est très grossière, mais la méthode n'est destinée qu'à donner une idée de la valeur de P et Q, donc de  $\lambda$  et  $\beta$ . Signalons tout de même que les valeurs P et Q pourraient également être approchées par un procédé d'optimisation linéaire (moindre carré...).

### Deuxième étape

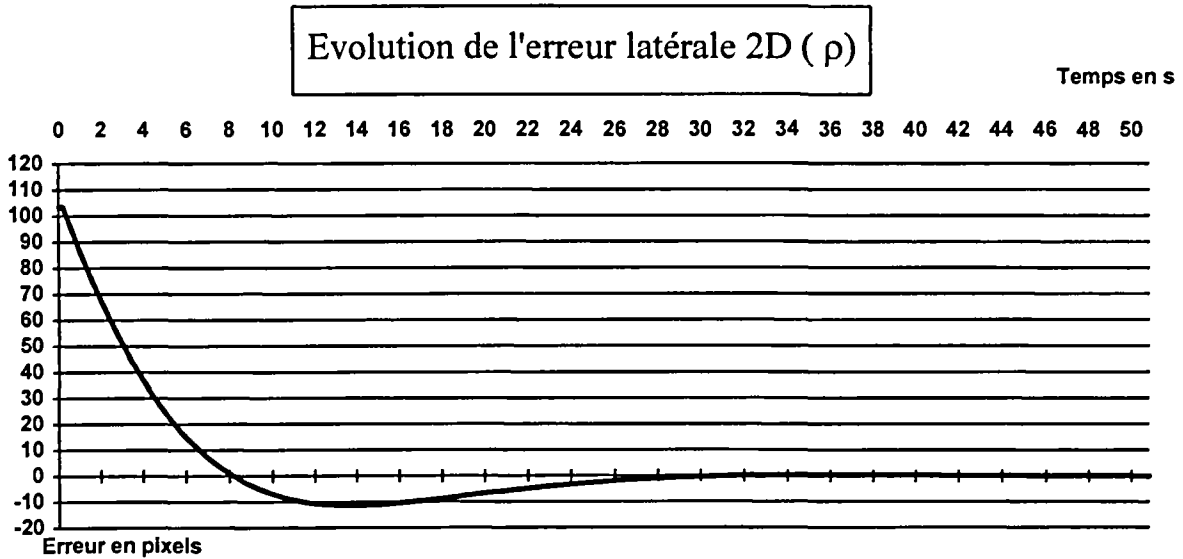
La deuxième étape consiste à faire varier les gains  $\lambda$  et  $\beta$  et à observer en simulation les courbes de réponse indicielle. On retiendra le couple  $\lambda$  et  $\beta$  pour lequel la courbe obtenue est la plus proche de celle désirée. Celle-ci n'est pas forcément la courbe que l'on obtiendrait avec un conducteur car chaque pilote a un style de conduite qui lui est propre. Le couple  $\lambda$  et  $\beta$  qui correspond au style de conduite (temps de réponse et dépassement) que nous désirons pour notre engin sera conservé.



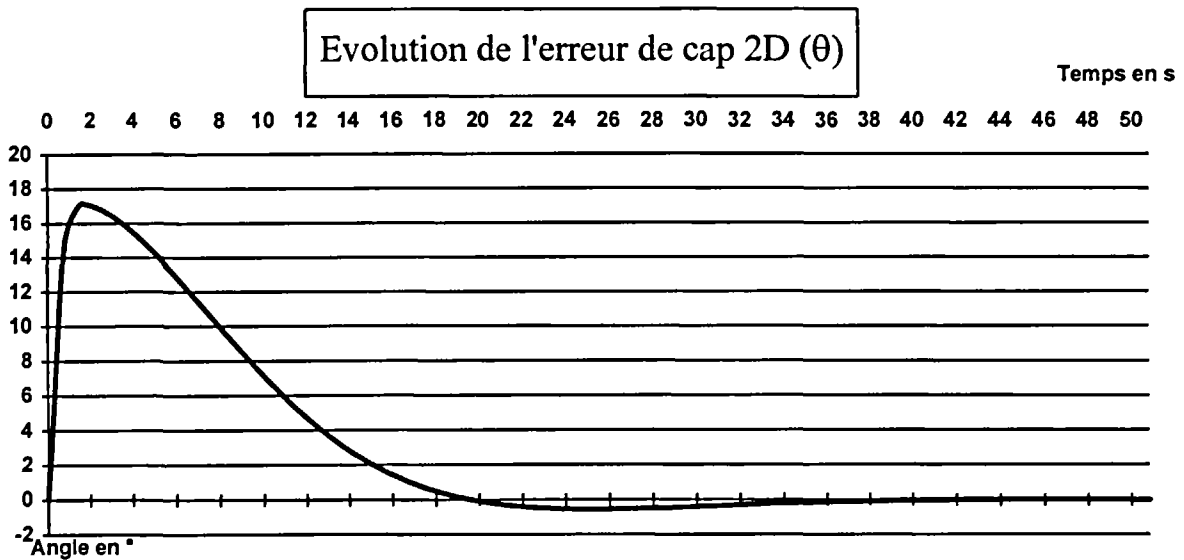
Ainsi, les deux valeurs que nous avons choisies pour  $\lambda$  et  $\beta$  sont :

$$\begin{cases} \lambda = 0.3 \\ \beta = 2.6 \end{cases}$$

Pour ces gains, nous obtenons les courbes de réponses indicielles suivantes :

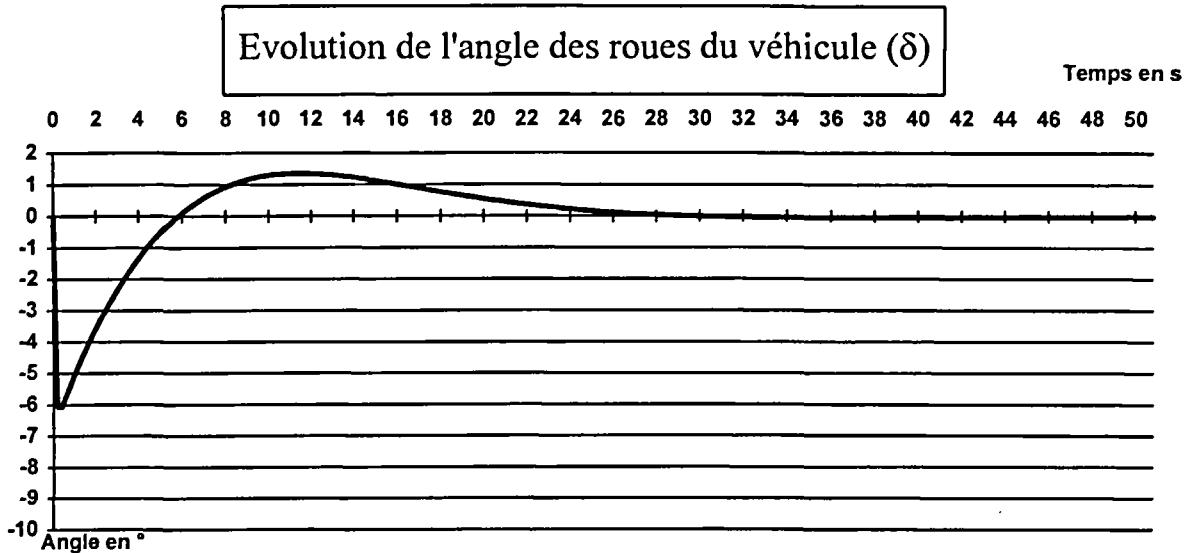


***Courbe IV 8 : réponse indicielle du système***



***Courbe IV 9 : réponse indicielle du système***

- les deux premières courbes (Courbe IV 8 Courbe IV 9 ) représentent la sortie du module de traitement d'images,
- la troisième courbe (Courbe IV 10) représente la sortie du module de commande :



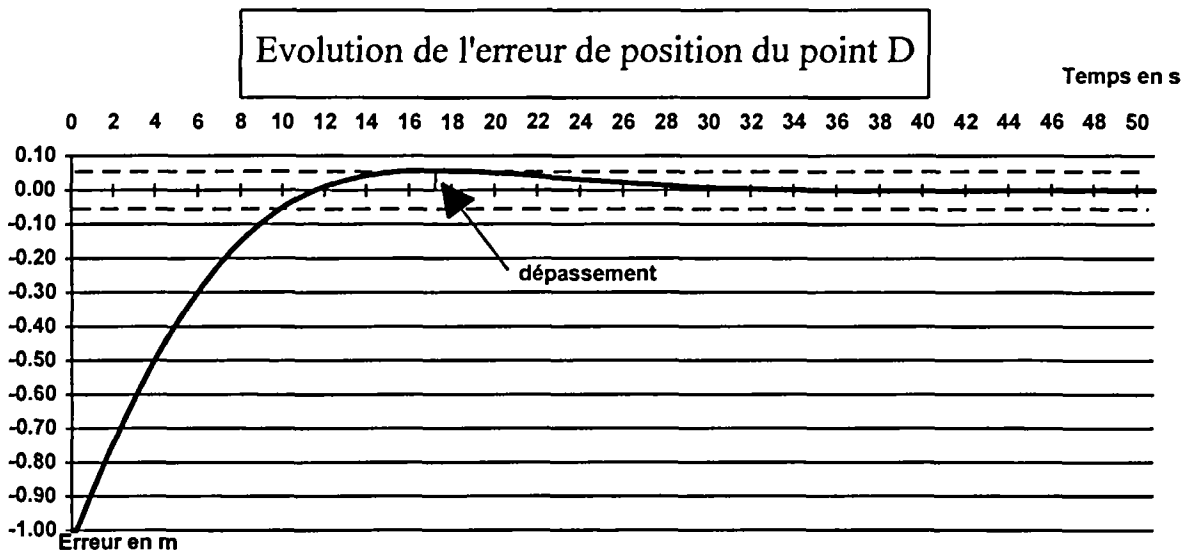
**Courbe IV 10 : réponse indicielle du système**

Ces trois courbes ont été obtenues par simulation en utilisant la loi de commande référencée vision proportionnelle. Elles représentent l'évolution des trois paramètres  $\theta$ ,  $\rho$ , et  $\delta$  qui sont les entrées et la sortie du système de commande. Nous observons que la forme des courbes est très proche de celles enregistrées avec l'opérateur humain. La principale différence concerne le temps de réponse qui est beaucoup plus important qu'avec le conducteur. Cela est normal car nous avons choisi un style de conduite avec une dynamique plus lente et un dépassement moins important.

Les trois variables représentées sont les entrées et la sortie du module de commande. Si elles sont très importantes pour évaluer les caractéristiques et en particulier les gains des équations de commande, elles ne représentent pas la qualité de l'asservissement. En effet, rien ne prouve que lorsque la machine est asservie dans l'image (droite détectée et droite de référence superposées), qu'elle soit correctement asservie par rapport à l'interface fauchée/non-fauchée. Nous n'avons pas démontré que la primitive visuelle (une droite) permettait d'asservir correctement la machine. Nous ne ferons pas cette démonstration mais nous montrerons, par nos simulations et par nos essais en conditions réelles, que notre primitive visuelle (une seule droite) est suffisante pour caractériser la liaison de la machine avec l'interface fauchée/non-fauchée.

Pour cela, mesurons la distance séparant le point D (extrémité du diviseur gauche) de l'interface fauchée/non-fauchée (*Figure IV 8*). Le but est de montrer que notre primitive visuelle est suffisante pour régler à zéro cette distance.

La courbe suivante est la représentation de cette distance pour une réponse indicielle du système.



**Courbe IV 11 : erreur au point D pour une réponse indicielle du système**

Nous constatons sur cette courbe (*Courbe IV 11*) que l'erreur au point D s'annule en 12 s et que le dépassement est de l'ordre de 5 cm. De plus, la convergence ne présente pas d'oscillation. Au vu des courbes précédentes nous pouvons conclure que l'équation (eq IV 1) permet de commander la machine avec de bonnes performances et que notre primitive visuelle semble assurer une liaison satisfaisante entre le point D et la trajectoire rectiligne.

### 1.2.2.1 Influence de $\lambda$ et $\beta$

Ce paragraphe est destiné à montrer l'influence des gains  $\lambda$  et  $\beta$  sur les performances de l'asservissement mesurées au point D. Pour cela, faisons varier ces gains autour de leur valeur d'origine et comparons les réponses indicielles obtenues.

Nous rappelons que l'équation de commande est de la forme :

$$\begin{cases} v = v_{\text{moy}} \\ \delta = -\frac{L \cdot \dot{\phi}}{v_{\text{moy}}} = -\frac{L \cdot \lambda}{v_{\text{moy}}} \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right] \end{cases}$$

Ce qui peut également s'écrire sous la forme :

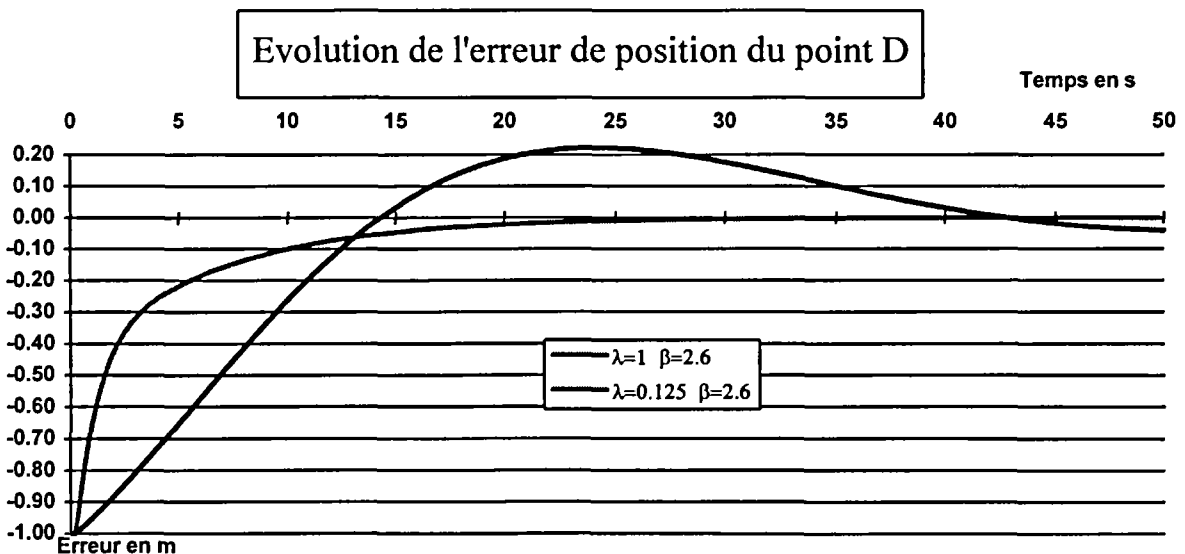
$$\delta = a \cdot \lambda \cdot \rho + b \cdot \lambda \cdot \beta \cdot \theta = P \cdot \rho + Q \cdot \theta$$

avec  $\lambda$ ,  $\beta$ , et  $b$  réels positifs et  $a$  réel négatif.

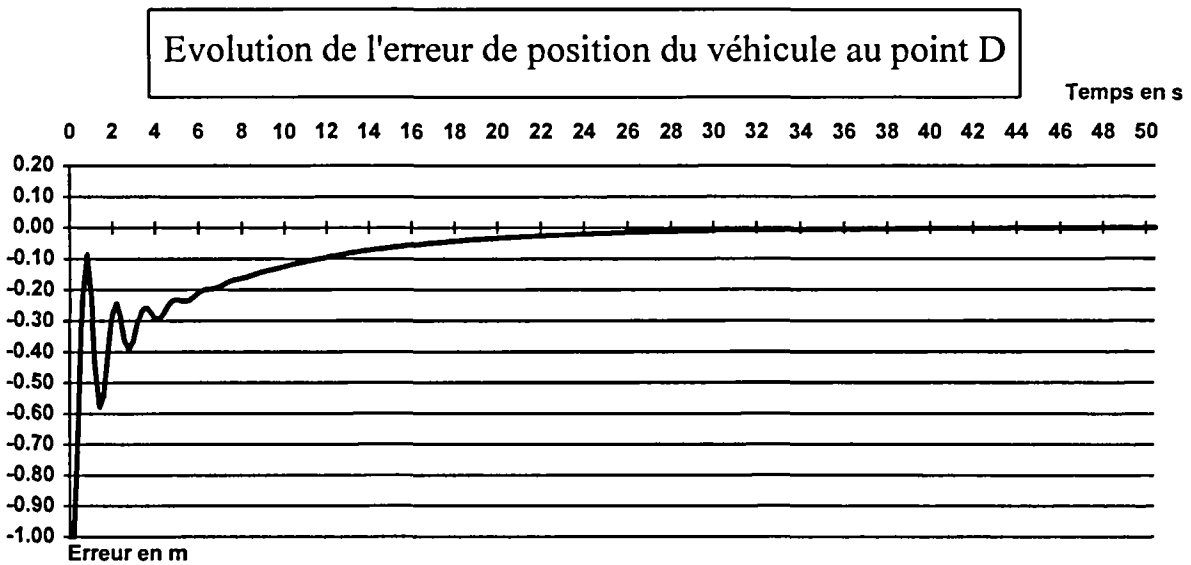
D'après l'équation précédente, nous constatons que les deux gains  $\lambda$  et  $\beta$  agissent différemment sur la loi de commande. Le gain  $\lambda$  peut être factorisé et concernera la dynamique générale de  $\delta$ . Le gain  $\beta$  n'agit que sur une entrée du correcteur. En privilégiant une entrée par rapport à l'autre, il détermine si la commande sera très fluide en utilisant plus le cap, ou plus « brutale » en favorisant l'erreur latérale.

**Influence de  $\lambda$  sur l'erreur au point D**

$\lambda$  est un gain global qui agit aussi bien sur l'erreur latérale ( $\rho$ ) que sur l'erreur de cap ( $\theta$ ). Il est donc un paramètre global de l'asservissement et la sortie du correcteur ( $\delta$ ) lui est proportionnelle.  $\lambda$  est donc directement lié à la dynamique du système. Donnons trois exemples où nous avons fait varier  $\lambda$ .

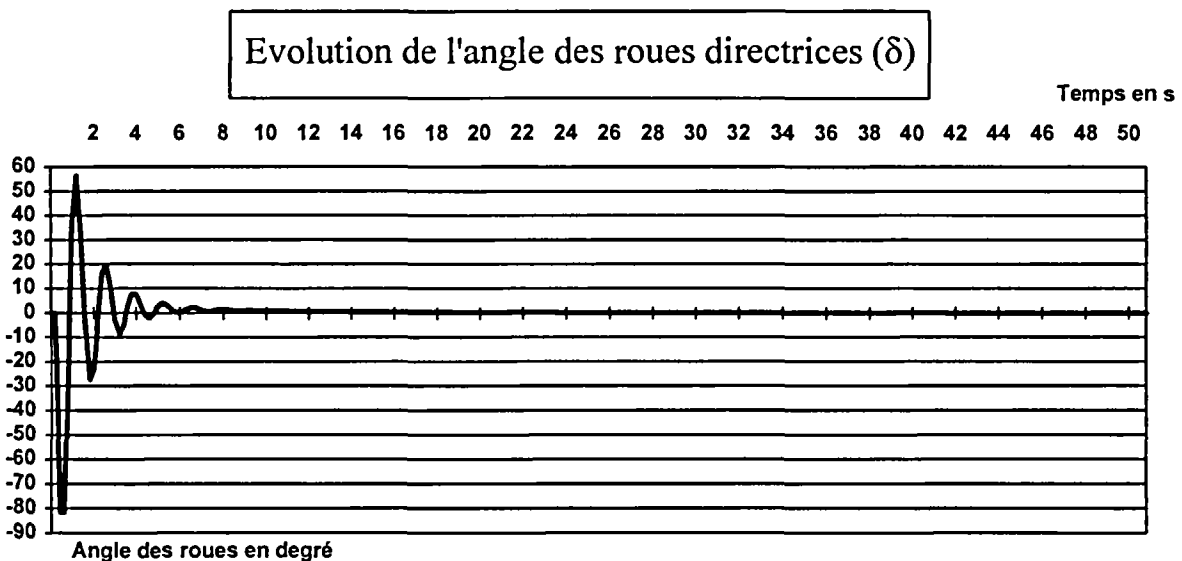


**Courbe IV 12 : réponse indicielle du système pour deux valeurs de  $\lambda$**



***Courbe IV 13 : réponse indicielle du système pour  $\lambda=4$  et  $\beta=2.6$***

Ces trois courbes montrent que, lorsqu'on augmente la valeur de  $\lambda$ , la dynamique du système s'accroît jusqu'à l'apparition d'oscillations amorties. **Rappelons que ces courbes ne représentent pas la sortie du correcteur mais l'erreur de position du point D qui représente une avance de correction due au montage.** Afin de se rapprocher des courbes plus classiques de l'automatique, donnons celle de la sortie du correcteur ( $\delta$ ) :



***Courbe IV 14 : sortie du système lors d'une réponse indicielle pour  $\lambda=4$  et  $\beta=2.6$***

On distingue bien ici un phénomène oscillatoire amorti autour de 0, ce qui est normal puisque le gain global du correcteur ( $\lambda$ ) est élevé.

### Conclusion

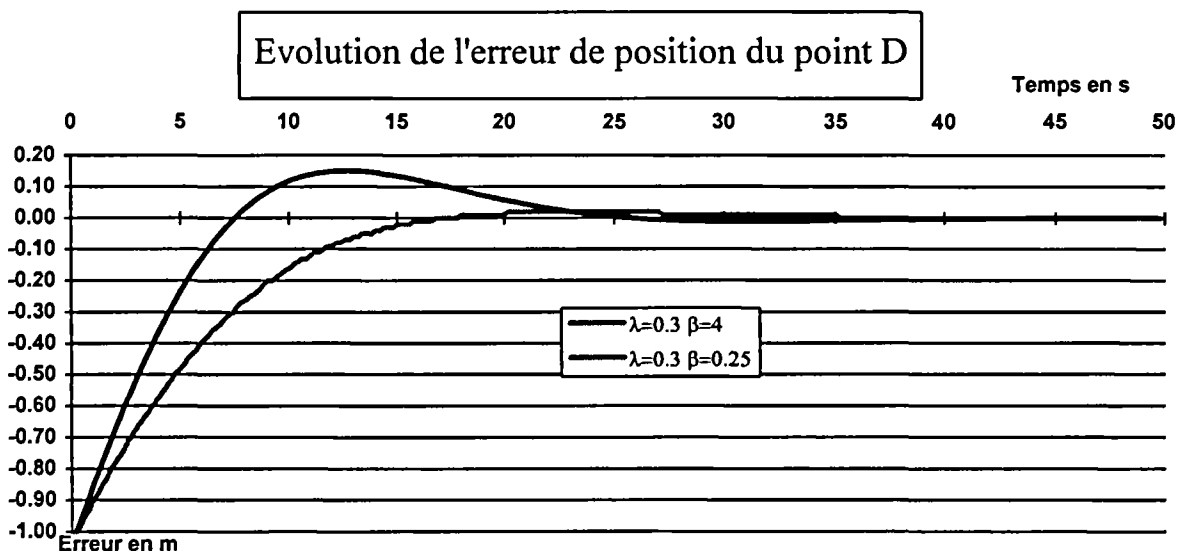
Le gain  $\lambda$  a bien le comportement attendu. Il conditionne la dynamique globale du système et en particulier le temps de réponse. Comme nous pouvions le prévoir, des valeurs trop élevées de  $\lambda$  provoquent un régime oscillatoire qui pourra même aller jusqu'à la divergence. Ce n'est pas le cas sur nos courbes car le simulateur intègre la saturation qui existe sur la commande des véhicules. Nous rappelons que la mesure de l'erreur représentant l'entrée du correcteur se fait à une distance de 6.5 m de la machine. Cela représente une avance importante que nous n'avons pas modélisée car elle ne peut qu'accroître la stabilité de notre système.

### Remarque

Sur la première courbe ( $\lambda=0.125$ ) nous constatons un dépassement important qui est dû à une dynamique trop faible du correcteur. Il subit alors le « comportement intégrateur » du véhicule.

### Influence de $\beta$ sur l'erreur au point D

$\beta$  est un gain qui agit uniquement sur l'erreur de cap ( $\theta$ ). Ce paramètre introduit donc une importance relative d'une entrée du correcteur par rapport à l'autre. Donnons deux exemples avec des valeurs très différentes de  $\beta$ .

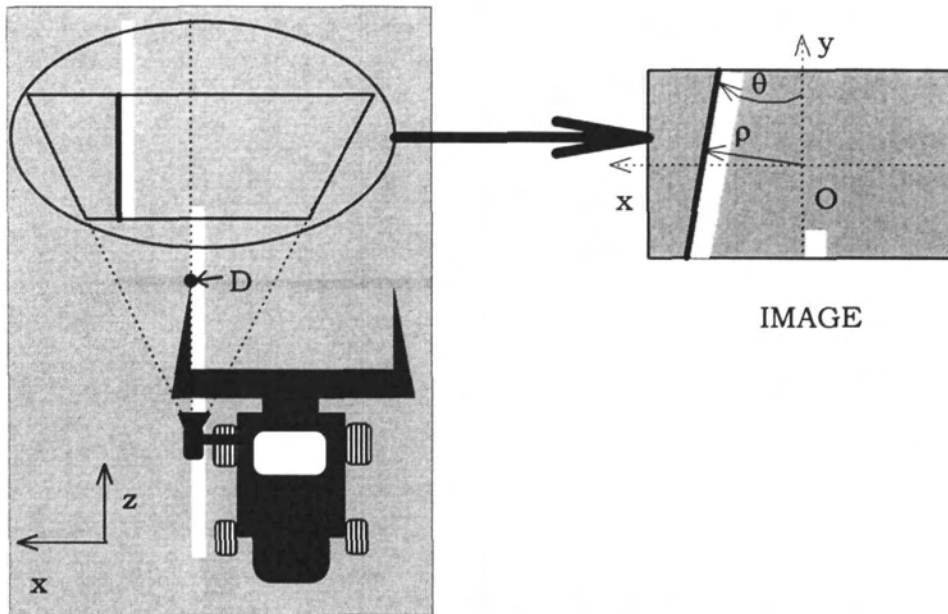


**Courbe IV 15 : réponse indicielle du système pour deux valeurs de  $\beta$**

Ces deux courbes montrent que si  $\beta$  a une influence sur la dynamique de notre système, elle est beaucoup moins importante que celle du paramètre  $\lambda$ . L'influence du paramètre  $\beta$  concerne surtout le dépassement et la stabilité du système.

### Explications

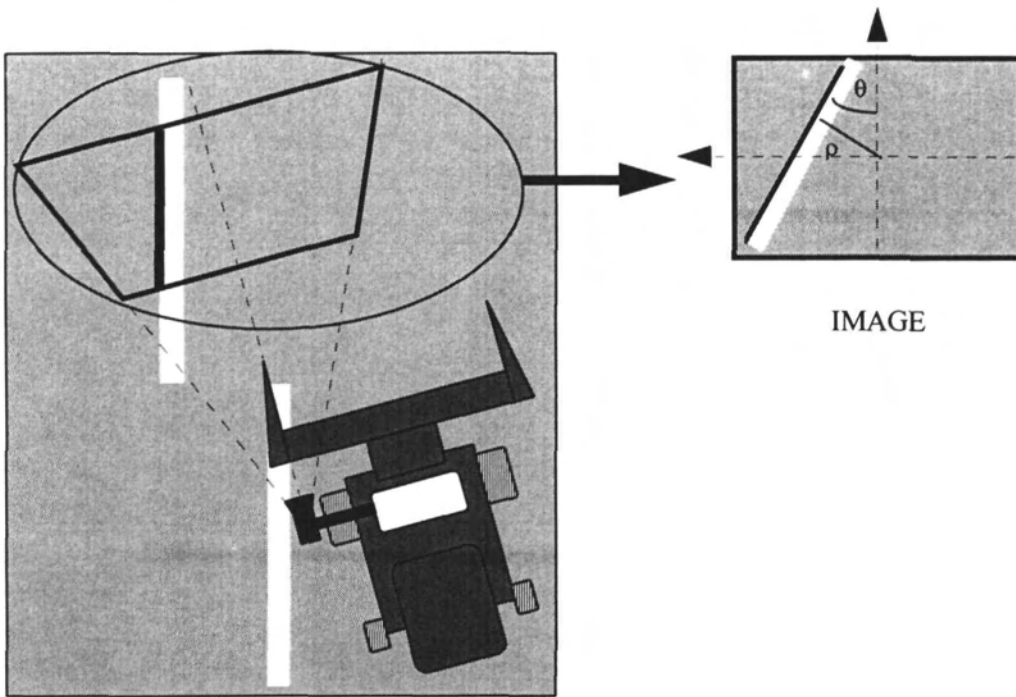
Pour comprendre l'importance du paramètre  $\beta$ , explicitons les entrées et la sortie de notre correcteur lors de la réponse du système (correcteur + machine + caméra) à un échelon.



**Figure IV 9 : informations visuelles lors de la réponse à un échelon**

Sur la figure précédente la machine est parallèle à la ligne à suivre, mais décalée d'un mètre en position latérale. Si nous considérons les entrées du correcteur, c'est-à-dire les informations visuelles, lors de la réponse du système à un échelon on constate deux phénomènes :

- ⇒ alors que la machine est rigoureusement parallèle à l'interface, l'angle ( $\theta$ ) de la droite détectée n'est pas nul. Ce phénomène est provoqué par la projection perspective qui ne respecte pas les angles. De plus, la valeur détectée en orientation implique un éloignement de la machine par rapport à l'interface. En effet, plus l'échelon sera important et plus l'orientation de la droite sera grande. Cependant, ceci est compensée par l'erreur latérale qui permet de ramener la machine vers l'échelon. A cet instant un deuxième phénomène apparaît ;
- ⇒ lorsque la machine change d'orientation par rapport à la ligne à suivre, elle forme un angle avec cette ligne qui croît dans les premiers instants de la réponse. Celui-ci se retrouve dans l'image et tend à éloigner le véhicule de la ligne blanche (**Figure IV 10**).



**Figure IV 10 : variation des informations visuelles lors de la réponse à un échelon**

En ce qui concerne la sortie de la loi de commande ( $\delta$ ) nous avons :

$$\delta = P \cdot \rho + Q \cdot \theta = \delta_{\rho} + \delta_{\theta} \text{ avec } P < 0 \text{ et } Q > 0$$

De par le modèle de la moissonneuse batteuse, lorsque ( $\rho$ ) est positif ( $\delta_{\rho}$ ) doit être négatif et lorsque ( $\theta$ ) est positif ( $\delta_{\theta}$ ) doit être négatif. Dans l'exemple de la réponse à une échelon ceci s'exprime sous la forme :

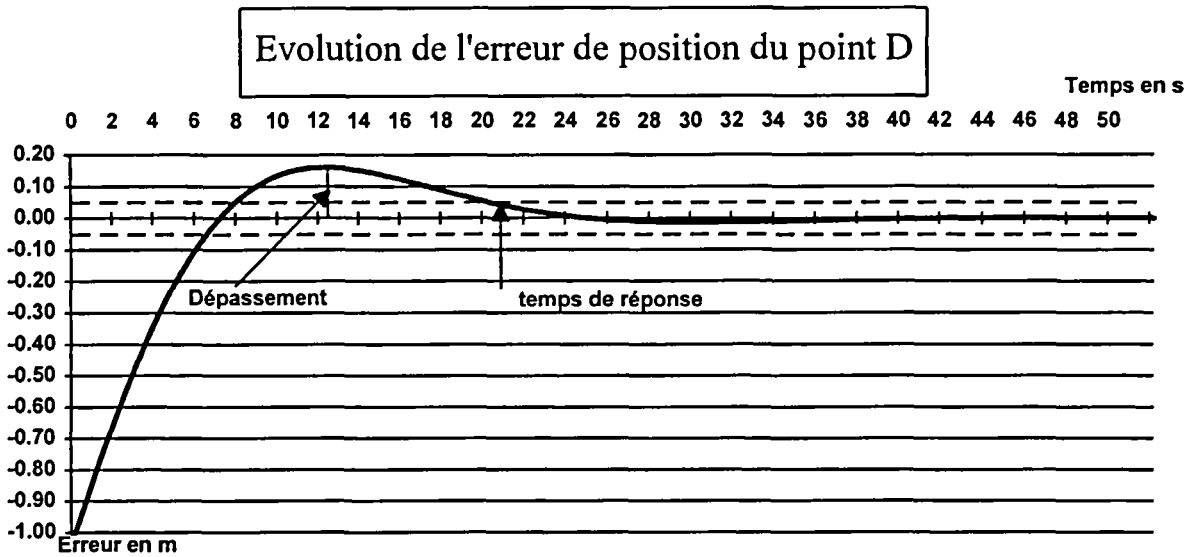
$$(\delta_{\rho} = P \cdot \rho) < 0 \quad \text{et} \quad (\delta_{\theta} = Q \cdot \theta) > 0$$

On a donc bien dans le cas de la réponse à un échelon (**Figure IV 9**) un terme ( $\delta_{\rho}$ ) qui tend à annuler l'erreur et un terme ( $\delta_{\theta}$ ) qui tend à l'augmenter. Ceci n'est pas une contradiction car le terme qui est fonction de l'orientation permet d'atténuer l'angle de la machine avec l'interface. De par la non-holonomie du véhicule ce phénomène contribue à limiter le dépassement enregistré lors de la réponse indicielle. Le terme ( $\delta_{\theta}$ ) lisse la trajectoire en limitant l'angle de la machine avec la ligne à suivre.

Nous avons vu que  $P$  était de la forme  $a \cdot \lambda$  et que  $Q$  était de la forme  $b \cdot \lambda \cdot \beta$ ,  $b$  étant un réel strictement positif et  $a$  un réel négatif. Le gain  $\beta$  va donc permettre au deuxième terme  $\delta_{\theta}$  de compenser en partie la dynamique de  $\delta_{\rho}$ . Si  $\beta$  est trop grand, on aura même une commande ( $\delta$ ) de signe inverse à celle désirée. Dans ce cas, on constate une divergence de l'algorithme.

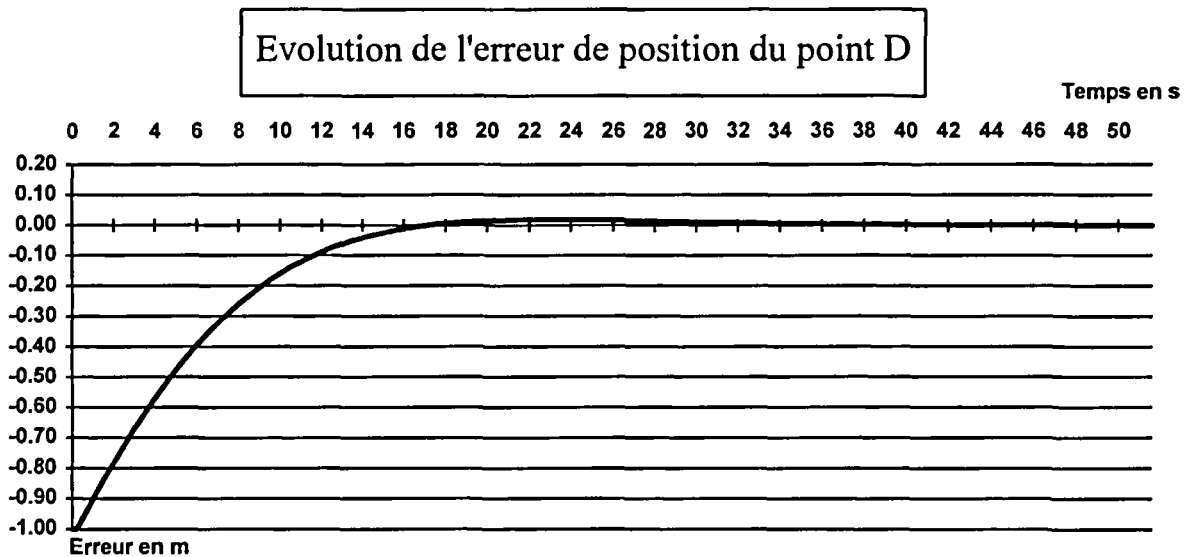


Prenons l'exemple de la courbe suivante réalisée avec  $\beta=0$  :



**Courbe IV 16 : réponse indicielle du système pour  $\lambda=0.3$  et  $\beta=0$**

Puis l'exemple suivant avec  $\beta=4$  :



**Courbe IV 17 : réponse indicielle du système pour  $\lambda=0.3$  et  $\beta=4$**

On constate sur la première courbe un dépassement très important de l'ordre de 18 cm et un temps de réponse de l'ordre de 20 s. La deuxième courbe montre un dépassement voisin de zéro et un temps de réponse de l'ordre de 14 s.

Les nombreuses simulations que nous avons faites montrent que le gain  $\beta$  donne de la stabilité et de la robustesse au système vis-à-vis des perturbations tel que le bruit présent dans les informations visuelles. En fait, la valeur de  $\beta$  doit être choisie afin de régler la dynamique du système avec un dépassement de l'ordre de 5 %.

### Conclusion

Au vu de ces deux courbes, il est clair que le rôle du terme  $\delta_0$  est primordial pour notre application puisqu'il donne une grande stabilité au comportement du système. En agissant comme un filtre temporel, il permet de diminuer le dépassement constaté lors de la réponse indicielle du système et de lisser la trajectoire. Pour nos applications où le passage  $n$  de l'engin construit la trajectoire du passage  $n+1$ , ceci évitera la propagation et l'amplification des perturbations. En effet, l'interface fauchée/non-fauchée étant très perturbée, une commande qui ne filtrerait pas ces perturbations aboutirait rapidement à une trajectoire très déformée.

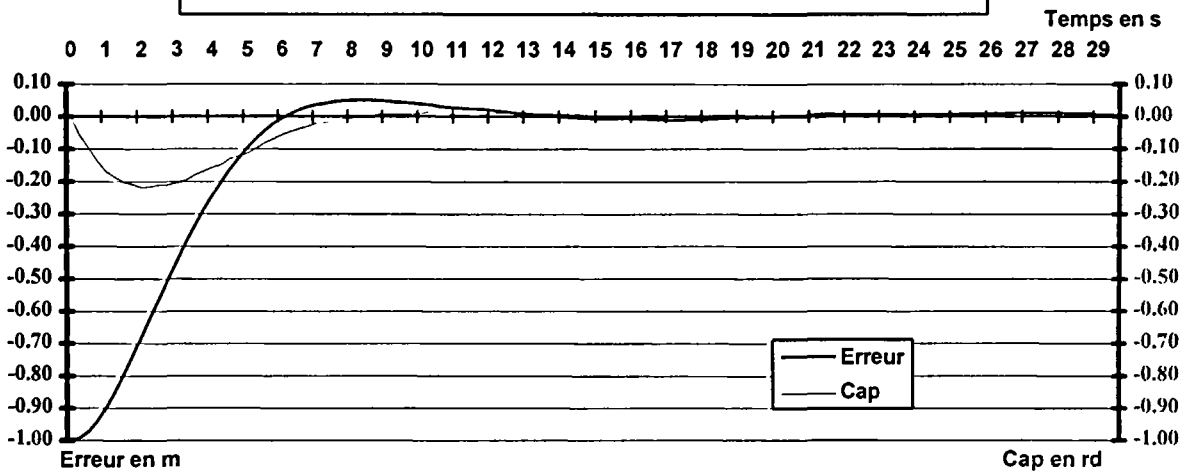
#### 1.2.2.2 Influence du bruit sur la loi de commande référencée vision (CRV)

Dans ce paragraphe nous avons modifié les paramètres du deuxième simulateur afin de l'adapter à la tondeuse. Nous avons montré au paragraphe (1.1.2.1) que le bruit des mesures visuelles avait une influence très pénalisante sur la commande « trajectoire ». Aussi, afin de pouvoir comparer la robustesse des deux lois de commande par rapport au bruit des informations visuelles, nous avons implanté en simulation la CRV sur la tondeuse. Rappelons que l'engin considéré en simulation n'a pas d'importance puisque les deux véhicules ont le même comportement cinématique. Cependant, dans un souci de clarté, nous avons considéré le cas de la tondeuse.

Lors des différentes campagnes de moissons où nous avons fauché du blé et de l'orge, nous avons quantifié un bruit sur les informations visuelles d'une amplitude de l'ordre de 16 pixels. Du reste, ce chiffre correspond à la précision de l'algorithme de traitement d'images.

Afin de comparer nos lois de commande nous avons rajouté un bruit de 20 pixels sur la position de l'interface dans l'image. Examinons la réaction de notre engin à des mesures bruitées lorsqu'il est commandé par la CRV proportionnelle :

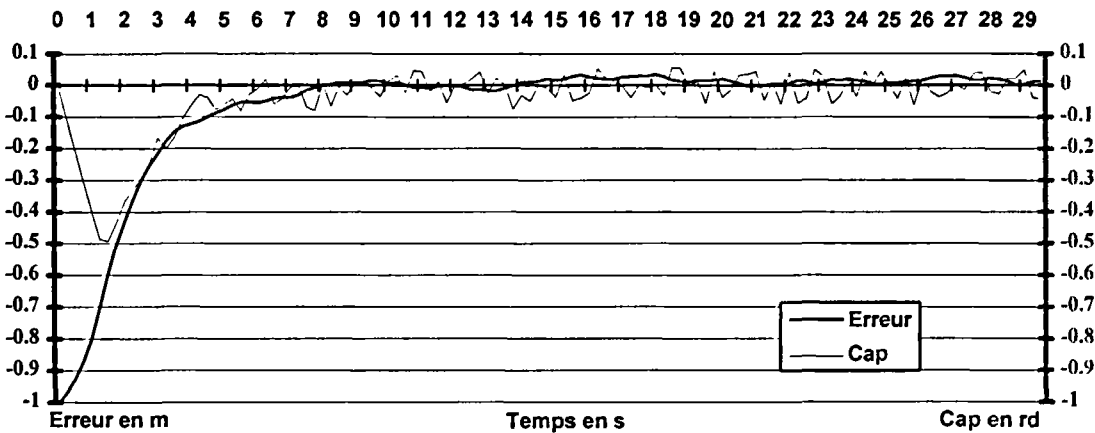
Erreur et cap pour un bruit d'amplitude de 20 pixels



***Courbe IV 18 : réponse indicielle du système avec la CRV***

Rappelons pour mémoire la courbe faite dans les mêmes conditions avec la commande trajectoire :

Erreur et cap pour un bruit d'amplitude de 20 pixels



***Courbe IV 19 : réponse indicielle du système avec la commande trajectoire***

Les deux lois de commande présentent un bon comportement au niveau de la position de l'engin. La CRV semble tout de même plus précise mais, dans les deux cas, l'erreur reste en dessous de 5 cm. Par contre, lorsqu'on considère l'orientation de la machine, nous constatons une différence importante entre les deux lois. Avec la commande trajectoire, l'orientation de la machine peut varier de

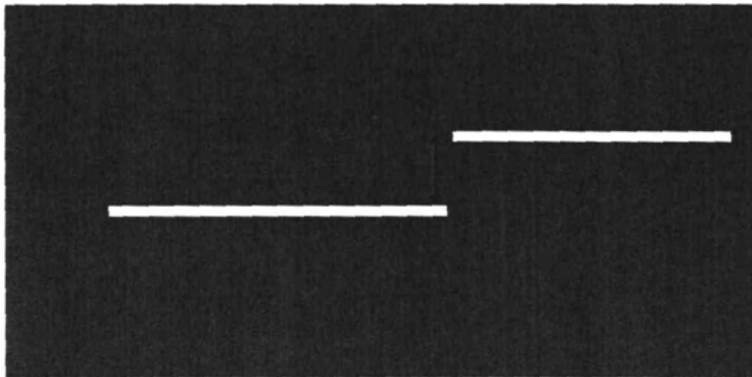
plus de 5 degrés, alors qu'avec la CRV cette variation n'atteint pas 1 degré. Cela s'explique facilement : si on compare les réponses indicielles des deux lois de commande, on s'aperçoit que la dynamique de la commande trajectoire est beaucoup plus importante que la dynamique de la CRV dans les 2 premières secondes de la réponse à l'échelon. Pendant cet intervalle de temps c'est l'erreur latérale 2D qui est prépondérante. La commande « trajectoire » réagit donc beaucoup plus fortement que la CRV aux variations de l'erreur latérale 2D ( $\rho$  ou  $d$  avec  $\rho=d.\cos\theta$ ). Or, c'est justement cette variable qui est bruitée, il est donc logique que la commande « trajectoire » soit la plus sensible à ce bruit.

Nous avons vu au chapitre III que l'algorithme de traitement d'images utilisait le résultat de l'image  $n$  comme point de départ du traitement de l'image  $n+1$ . Or la commande trajectoire provoque des changements importants d'orientation de la machine donc des variations conséquentes de l'image qui perturbent le fonctionnement de l'algorithme Markovien. De ce fait, nous pouvons conclure que la CRV semble beaucoup mieux adaptée à nos applications d'aide au guidage de machines agricoles.

## 2. ESSAIS REELS

Les essais en simulation des deux lois de commande ont donné des résultats satisfaisants et ont montré la convergence des deux algorithmes. Bien que notre souci de nous rapprocher le plus possible de la réalité fut constant dans nos simulations, nous avons fait de nombreux tests réels en implantant nos lois de commande sur les deux véhicules.

Pour effectuer ces tests nous avons choisi de suivre une interface plus « pratique » et plus précise que celle formée par de la végétation fauchée. Une ligne blanche peinte sur le sol (*Figure IV 11*) offre l'avantage de pouvoir s'utiliser en toute saison et de n'introduire que très peu de bruit dans notre système. Pour des raisons de temps de calcul, l'algorithme Markovien de traitement d'images a une précision de 16 pixels puisqu'il découpe l'image en sites de 16\*16 pixels. Afin d'évaluer au mieux nos lois de commandes, nous avons choisi de développer un algorithme de seuillage adaptatif amplement suffisant pour détecter une ligne blanche sur le sol en bonnes conditions et ayant des temps de calculs très réduits sur notre architecture et au regard de la dynamique de notre système. Sa précision est de 4 pixels et la période de rafraîchissement des informations visuelles est de 200 ms.

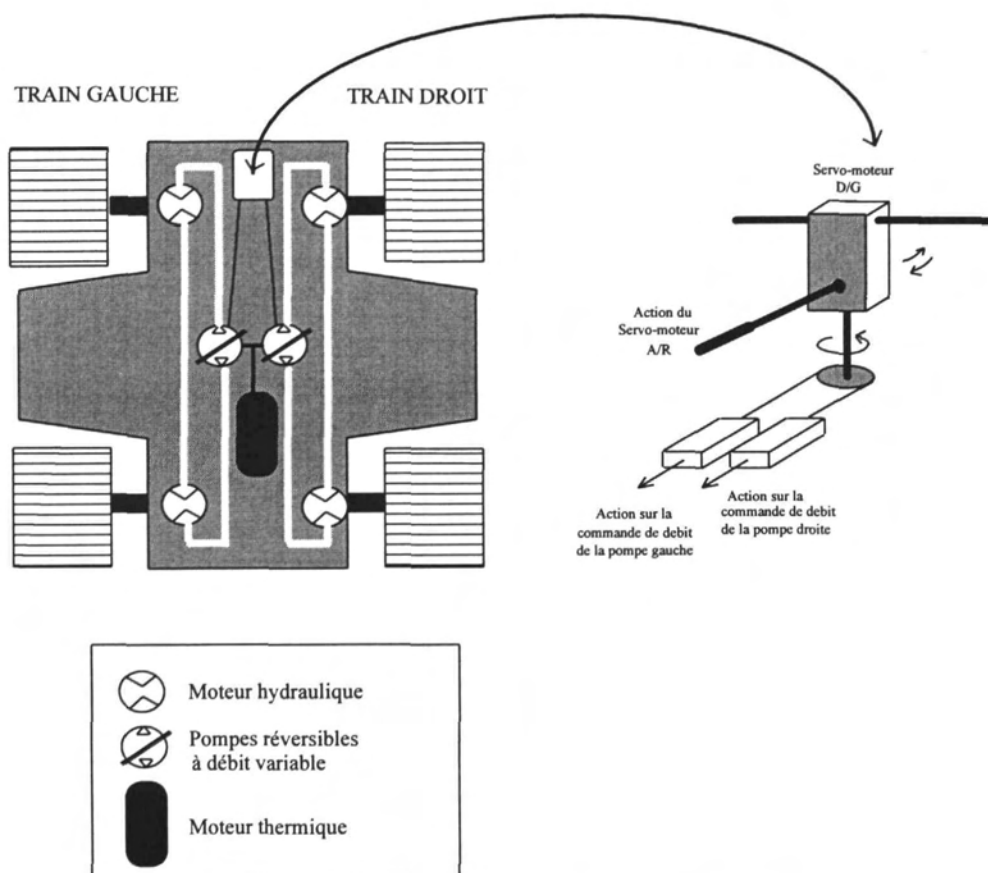


*Figure IV 11 : Piste d'essais avec un échelon*

## 2.1 Implantation sur la tondeuse

En Août 1993 notre premier prototype, une tondeuse professionnelle, a été instrumenté. Cette machine est contrôlée grâce à un système de radio-commande et à une électronique embarquée qui actionne deux circuits hydrauliques.

Deux servomoteurs contrôlent l'avancement et la direction de la machine (**Figure IV 12**). Le premier actionne les deux circuits hydrauliques (gauche et droit) de manière identique. Il contrôle donc la vitesse moyenne de l'engin. Le deuxième introduit une différence d'ouverture entre les deux circuits impliquant une différence de vitesse entre les roues gauche et droite.



**Figure IV 12 : partie mécanique de la tondeuse**

Les deux circuits hydrauliques sont contrôlés par les tensions que l'on applique aux servomoteurs. Des tests rapides [Deb94] ont montré qu'il était très difficile de trouver une relation simple entre la tension appliquée aux servomoteurs et la vitesse des roues (même en régime permanent). En effet, de nombreux phénomènes de dérives apparaissent régulièrement ou aléatoirement. De plus, la moindre perturbation modifie considérablement ses caractéristiques.

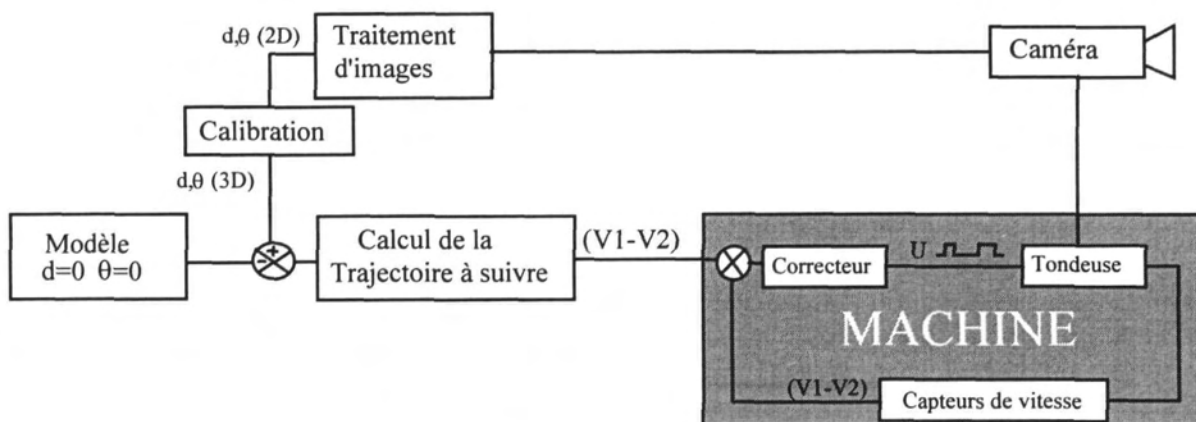
Aussi, un contrôleur « bas niveau », capable de donner une relation simple, fiable et robuste entre les tensions appliquées aux servomoteurs et la vitesse de rotation des roues, a été développé.

Les caractéristiques de ce contrôleur bas niveau sont :

- une période d'échantillonnage de 80 ms,
- un retard pur de 300 ms,
- un temps de réponse de 600 ms.

### 2.1.1 Implantation de l'ensemble contrôle-commande

Le module de commande se présente sous la forme de deux boucles d'asservissement imbriquées (**Figure IV 13**). La première boucle calcule, à partir des informations visuelles traduites dans l'espace 3D, la trajectoire que la machine doit suivre et en déduit la différence de vitesse à appliquer aux roues. La seconde (en gris) doit assurer le bon respect de cette consigne, et ce, quelles que soient les conditions dans lesquelles évolue la tondeuse.



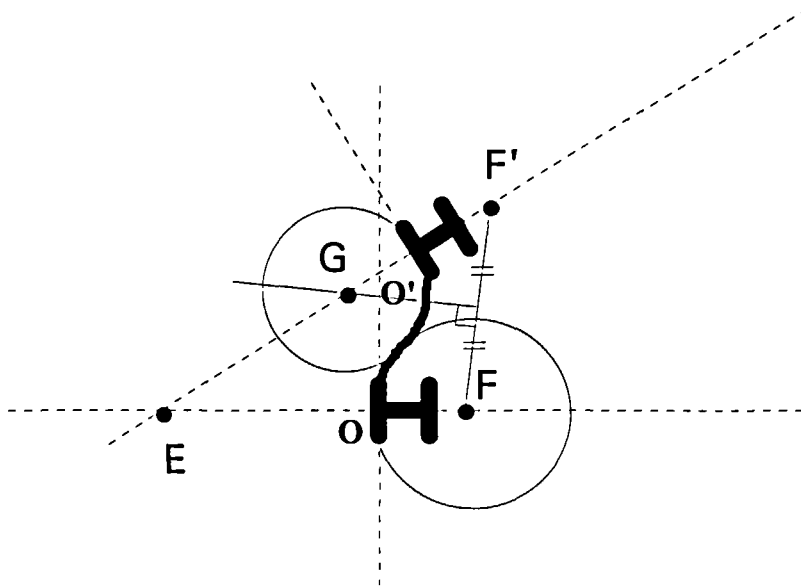
**Figure IV 13 : Asservissement final.**

### 2.1.1.1 Traitement des données visuelles

L'algorithme de traitement d'images utilisé le bord de la ligne blanche par une droite dont les coordonnées ( $d$  et  $\theta$ ) sont transmises au système de commande.  $d$  est la distance au centre de l'image entre la droite détectée et la droite de référence ;  $\theta$  est l'angle formé par les deux droites. La droite de référence est la droite détectée lorsque la machine est correctement asservie.

Ces deux informations ( $d, \theta$ ) sont des informations 2D. Avant de les comparer au modèle de la scène (droite passant par le centre des roues gauches de la machine), il faut les interpréter dans l'espace 3D. Grâce à la matrice de calibrage inverse déterminée au chapitre précédent, nous pouvons calculer les valeurs de  $d$  et  $\theta$  dans l'espace 3D.

Munis de ces deux informations tridimensionnelles nous sommes capables de calculer la trajectoire permettant à l'engin de passer de sa position initiale à la position finale exprimée dans l'espace de la scène (cf. chapitre III).



**Figure IV 14 : trajectoire de la tondeuse**

Un dernier problème subsiste. En effet, il y a un décalage entre le moment où l'image est saisie et celui où les informations arrivent au programme de commande. Ce décalage correspond au temps de calcul nécessaire au programme de traitement d'images pour extraire les informations visuelles.



Il faut intégrer ce décalage au programme de commande afin de se rapprocher le plus possible de la situation réelle. Pour cela, nous allons faire une approximation et supposer que, durant ce décalage, la machine évolue uniquement en ligne droite, c'est-à-dire que la différence de vitesse entre les deux trains de roues est nulle. Les temps de calculs étant de l'ordre de 200 ms, la machine va parcourir environ 25 cm. Sur une telle distance, il est tout à fait concevable d'approximer la trajectoire de l'engin à une droite.

Pour tenir compte de ce décalage, il suffit de réduire la distance séparant la position initiale de la machine de sa position finale du chemin parcouru durant le temps de calcul. C'est pourquoi il est indispensable que celui-ci soit transmis au programme de commande en même temps que les données visuelles.

Trois données sont donc utiles pour calculer la loi de commande de la tondeuse : l'écart, le cap et le temps mis par le programme pour effectuer ces calculs.

#### 2.1.1.2 Présentation de l'architecture

L'architecture retenue se compose de trois cartes VME de la société ELTEC. Les deux premières cartes à base de microprocesseurs 68040 sont dédiées au traitement d'images. La troisième (SAC30) à base de microprocesseur 68030 est réservée à la commande de la machine. Ces trois cartes fonctionnent selon le modèle du pipeline. Elles échangent des données et sont synchronisées sur la carte de traitement d'images qui a le temps de calcul le plus long.

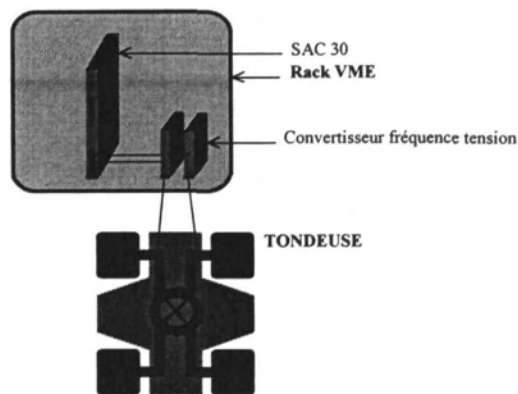
Il a fallu implanter sur celle-ci l'algorithme de commande qui fournit la consigne de différence de vitesse, ainsi que les éléments nécessaires à l'asservissement bas niveau. La difficulté consiste à faire exécuter en parallèle ces deux algorithmes sur la même carte. Nous avons donc décidé de partager le temps CPU entre les deux programmes. Cependant, il est nécessaire, pour assurer une bonne régulation de la différence de vitesse, de ne pas changer la période d'échantillonnage du correcteur bas niveau. Nous avons donc étudié les temps de calcul des deux algorithmes.

Le premier qui génère la consigne est un enchaînement de calculs simples dont la durée est négligeable. Le deuxième doit opérer en permanence puisqu'il assure le bon respect de la précédente consigne. Cependant, il est possible de l'arrêter à chaque nouvelle arrivée d'informations visuelles pour permettre au calcul de trajectoire de générer la consigne ( $V_1 - V_2$ ). Ceci se produit environ toutes les 200 ms et ne nuit pas au bon fonctionnement du correcteur bas niveau.

### 2.1.1.3 Implantation du correcteur bas niveau

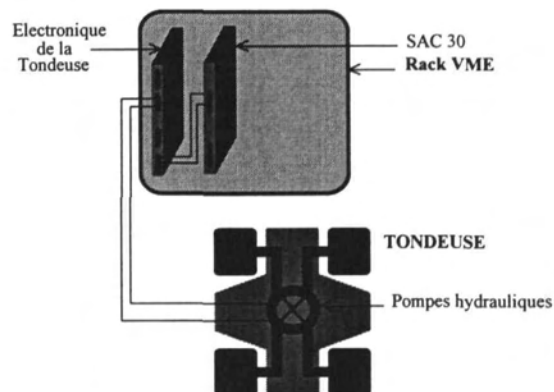
Pour implanter ce correcteur, il a fallu, dans un premier temps, mesurer la vitesse des roues. Pour cela, nous avons utilisé des capteurs inductifs qui fournissent une information fréquentielle. Grâce à deux cartes de conversion fréquence/tension, nous transformons cette information en une tension reliée directement à une entrée de la carte SAC30 (*Figure IV 15*).

Pour avoir une information suffisamment précise nous effectuons 200 acquisitions en 20 ms.



***Figure IV 15 : Saisie des vitesses des roues et conversion pour la SAC30.***

La sortie du correcteur bas niveau (un proportionnel intégral à gains adaptatifs) est directement introduite dans l'électronique de commande de la tondeuse sous forme d'une longueur de créneau variable (*Figure IV 16*).



***Figure IV 16 : Action du correcteur sur la machine.***

### 2.1.1.4 Mise en oeuvre informatique de l'asservissement

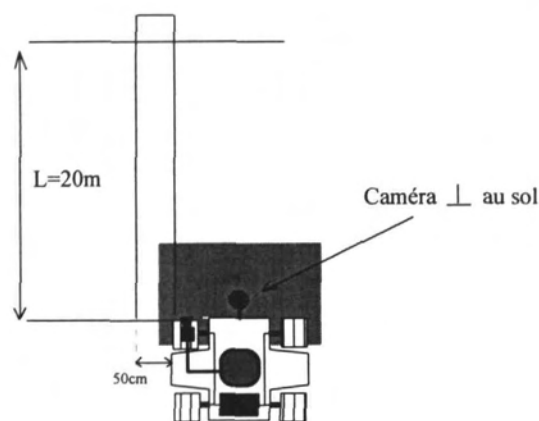
Nous avons expliqué précédemment qu'il fallait exécuter l'algorithme d'asservissement et le correcteur bas niveau en concurrence. Le système d'exploitation que nous avons choisi sur nos cartes est le système OS9. Il permet d'utiliser différents protocoles multitâches avec une très grande facilité.

La première tâche qui calcule la trajectoire à suivre, est « réveillée » lorsque survient une information visuelle, puis se « rendort » aussitôt pour laisser un maximum de temps CPU à la deuxième tâche qui gère le correcteur bas niveau. Cette dernière est exécutée en permanence puisqu'elle permet le bon fonctionnement de la direction de la machine.

Cependant, ces deux tâches doivent communiquer puisque la consigne de direction est calculée par la première tâche mais est exécutée par la seconde. Ceci est réalisé grâce à l'utilisation d'une mémoire partagée par les deux algorithmes. En effet, le premier écrit sa consigne dans un module, signale au deuxième qu'il vient d'écrire pour que celui-ci vienne lire les données et les exécuter. Il a donc fallu mettre en place tout un protocole de lecture/écriture afin d'éviter les conflits tout en exécutant les consignes de direction le plus rapidement possible. Ceci est détaillé dans l'annexe C.

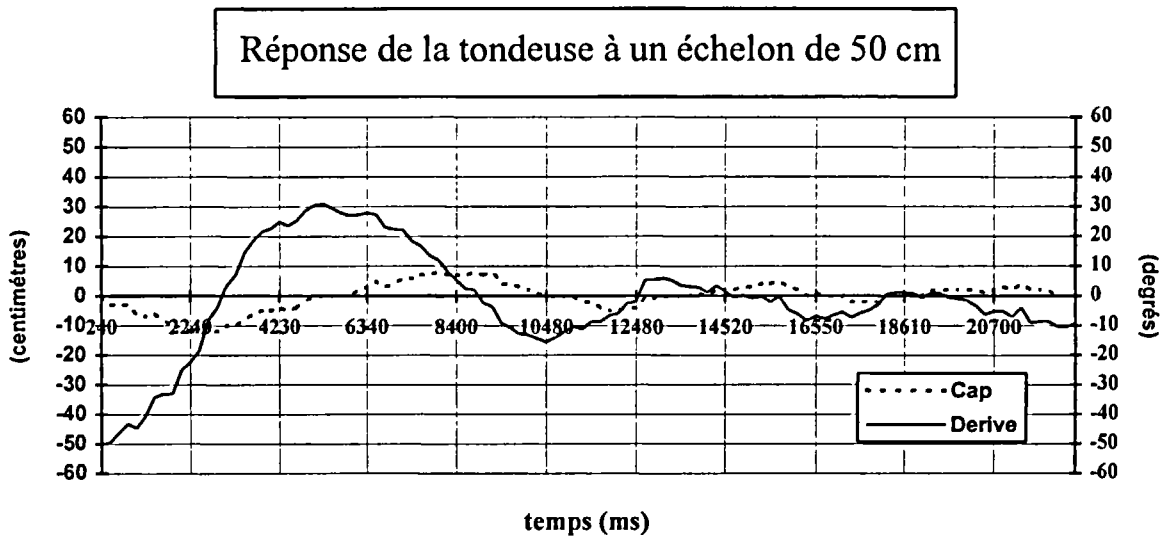
### 2.1.2 Essai réel

Afin de mesurer la qualité de notre asservissement, une deuxième caméra a été placée à l'avant de la tondeuse perpendiculairement au sol (**Figure IV 17**). Celle-ci nous donne à chaque instant une mesure de la distance entre la ligne blanche et le point de la tondeuse à asservir. L'algorithme qui nous permettra de dépouiller les images enregistrées par cette caméra a posteriori est le même que celui qui sert à guider la machine en temps réel. Sa précision est de l'ordre de  $\pm 5$  cm.



**Figure IV 17 : piste d'essais**

Les résultats présentés ci-après ont été enregistrés lors de la réponse de la machine à un échelon de 50 cm (*Figure IV 17*). La vitesse moyenne de la machine a été fixée à 3 km/h.



***Courbe IV 20 : réponse à un échelon de 50 cm avec la commande trajectoire***

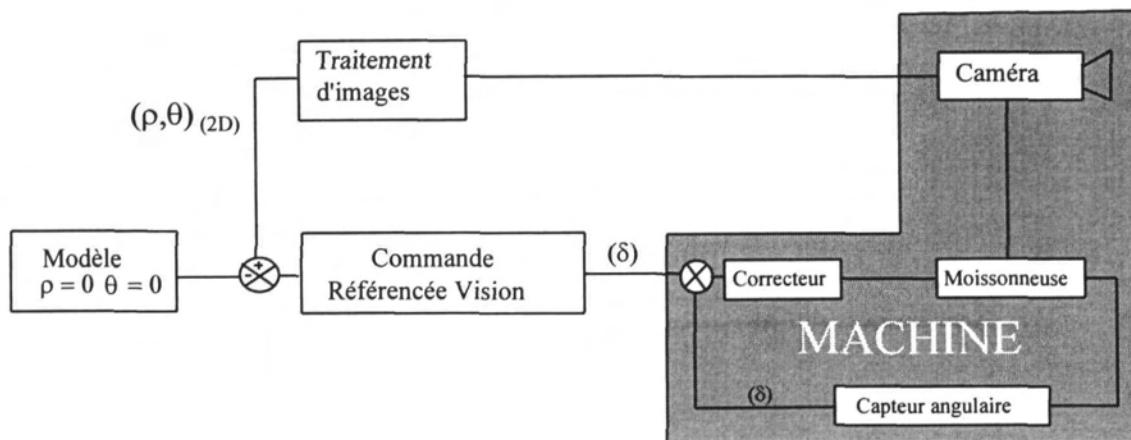
La réponse à un échelon de 50 cm de la tondeuse montre un temps de réponse de l'ordre de 8 s et un fort dépassement de 30 cm. Cependant, de nombreux problèmes mécaniques et un essoufflement du « système hydraulique » ont rendu la machine difficilement commandable. Il était, par exemple, évident que la machine tournait plus facilement d'un côté que de l'autre, mais ceci pouvait varier au cours du temps. Aussi est-il difficile d'émettre un jugement objectif sur la qualité de cette première loi de commande dans des conditions réelles de fonctionnement. Pour cela il faudrait l'implanter sur le deuxième prototype, ce que nous ferons au cours de l'année 1996.

## 2.2 Implantation de la CRV sur la moissonneuse-batteuse

Les résultats de simulations obtenus avec la Commande Référencée Vision étant très encourageants, nous avons implanté cette loi de commande sur un véhicule agricole : une moissonneuse-batteuse. Ses caractéristiques de fonctionnement sont relativement semblables à celles de la tondeuse :

- la vitesse moyenne est de l'ordre de 4 km/h,
- la machine doit s'asservir sur une interface fauchée/non-fauchée de végétation,
- cette interface est détectée grâce à un algorithme de segmentation en région d'images issues d'une caméra CCD,
- la période du traitement d'images est de l'ordre de 300 ms avec un retard pur de 600 ms.

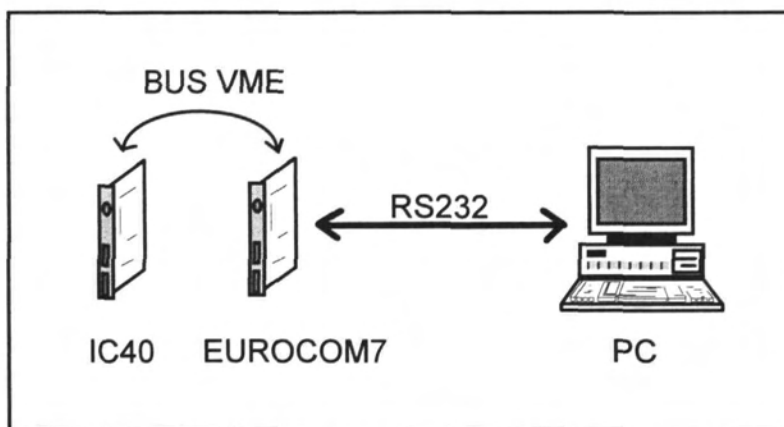
Ci-dessous le schéma général de fonctionnement du système est présenté. Contrairement à la tondeuse, nous n'avons pas identifié la moissonneuse-batteuse ni développé le correcteur bas niveau. En effet, la machine comprenait déjà un tel équipement réalisé par l'équipe ELAN du Cemagref d'Antony et implanté sur une carte électronique contrôlée par un PC.



**Figure IV 18 : schéma général de l'asservissement**

### 2.2.1 Présentation de l'architecture

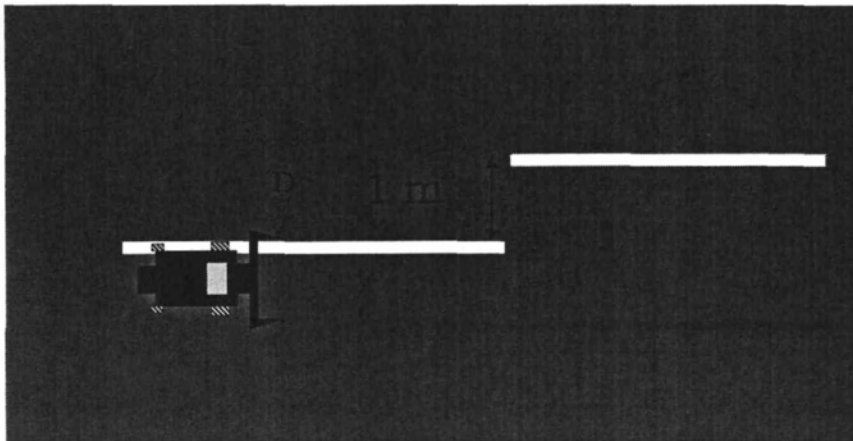
L'architecture retenue se compose de 2 cartes VME et d'un PC. Les deux cartes VME à base de microprocesseur 68040 sont dédiées au traitement d'images. Les données visuelles extraites par l'algorithme Markovien de segmentation en région de l'image sont communiquées grâce à une liaison série à un PC. Celui-ci comprend la Commande Référencée Vision qui calcule l'angle des roues directrices ainsi que l'asservissement bas niveau qui contrôle l'angle de direction. Ce dernier est implanté sur une carte électronique indépendante ne sollicitant pas la puissance de calcul du PC qui est entièrement dédiée à l'algorithme de commande.



**Figure IV 19 : schéma de l'architecture**

### 2.2.2 Essais réels

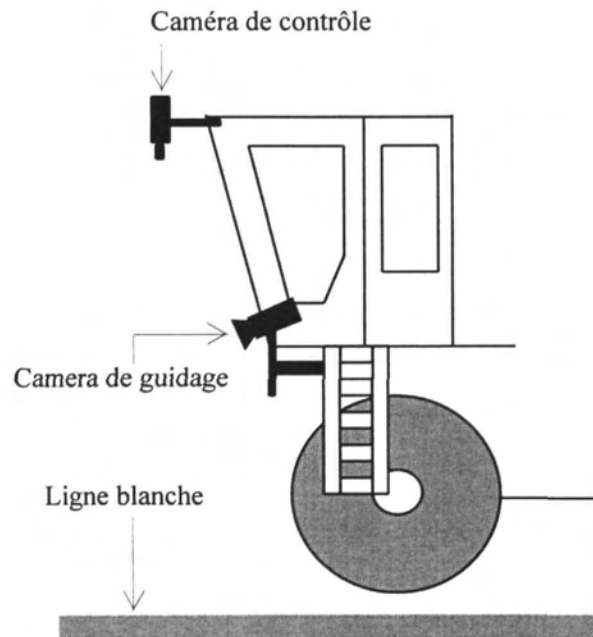
Comme pour la commande « trajectoire », la loi commande référencée vision a été testée sur un échelon de 1 mètre matérialisé par une ligne blanche.



**Figure IV 20 : trajectoire suivie par la moissonneuse-batteuse**

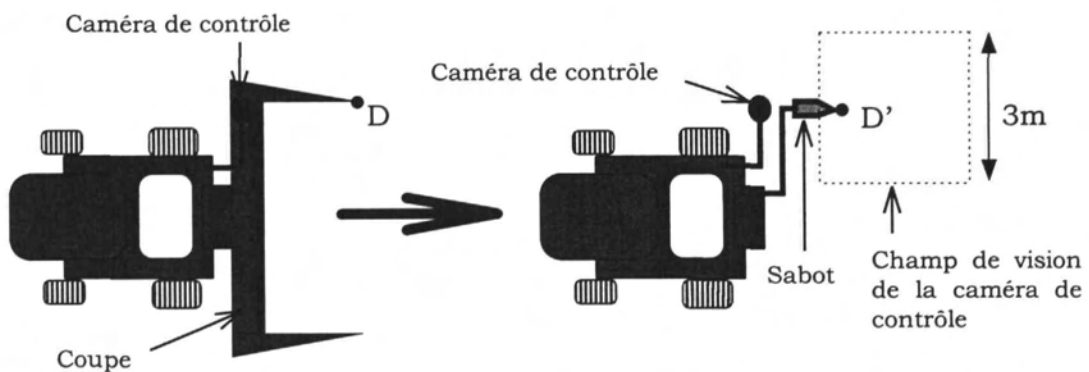
Le but de l'application est d'asservir l'extrémité du diviseur gauche (point D) sur l'interface fauchée/non-fauchée. Pour cette étude, nous chercherons à l'asservir sur le bord gauche de la ligne blanche. Comme pour les essais réalisés en simulation, la mesure de la qualité de l'asservissement est donnée par la distance du point D à la ligne blanche.

Afin de mesurer cette distance, une seconde caméra a été placée en haut de la cabine du véhicule (**Figure IV 21**).



**Figure IV 21 : Vue des 2 caméras permettant le guidage et le contrôle de la qualité de l'asservissement.**

Il est difficile de mesurer avec une caméra la qualité de l'asservissement au point D (extrémité du diviseur gauche) car « la coupe » se trouve dans le champ de la caméra. Aussi, celle-ci a été remplacée par un sabot traînant sur le sol.

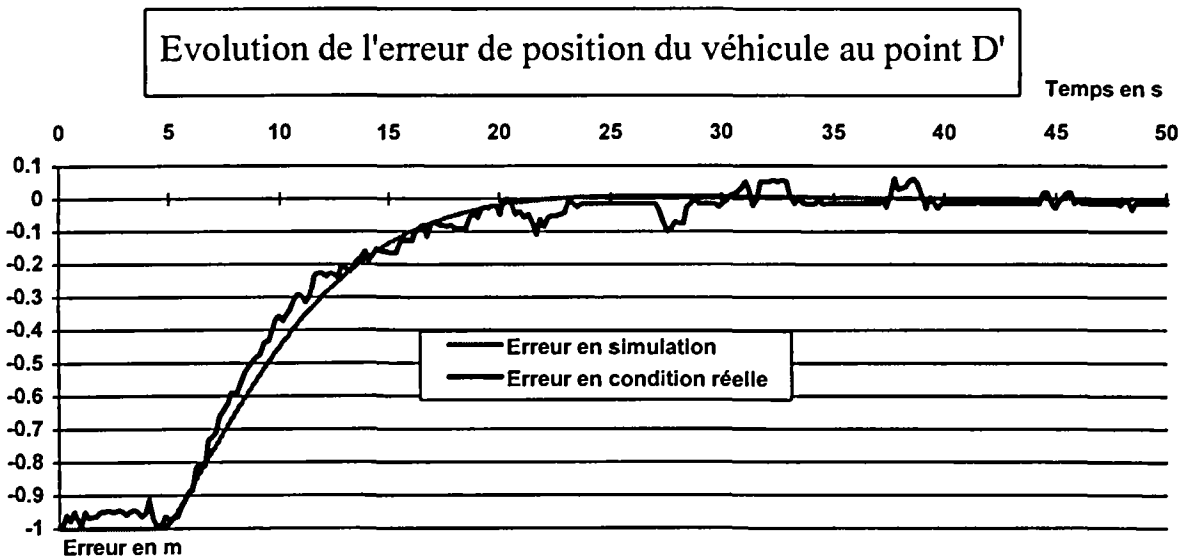


**Figure IV 22 : nouveau point de contrôle pour la moissonneuse-batteuse**

La qualité de l'asservissement sera donc mesurée au point D' qui est moins éloigné de l'essieu de la machine que le point D. Néanmoins, de rapides tests de simulation ont montré que la qualité de l'asservissement était sensiblement la même pour les deux points.

Le calcul des lois de commande référencée vision proportionnelle et proportionnelle intégrale a été fait en utilisant les équations du chapitre III pour une vitesse moyenne de 4 km/h. Bien que nous ayons fait des essais à des vitesses moyennes différentes, nous ne modifierons pas nos lois de commande afin de montrer leur robustesse vis-à-vis de la vitesse.

La première courbe a été réalisée pour une vitesse moyenne d'avancement de 4 km/h :

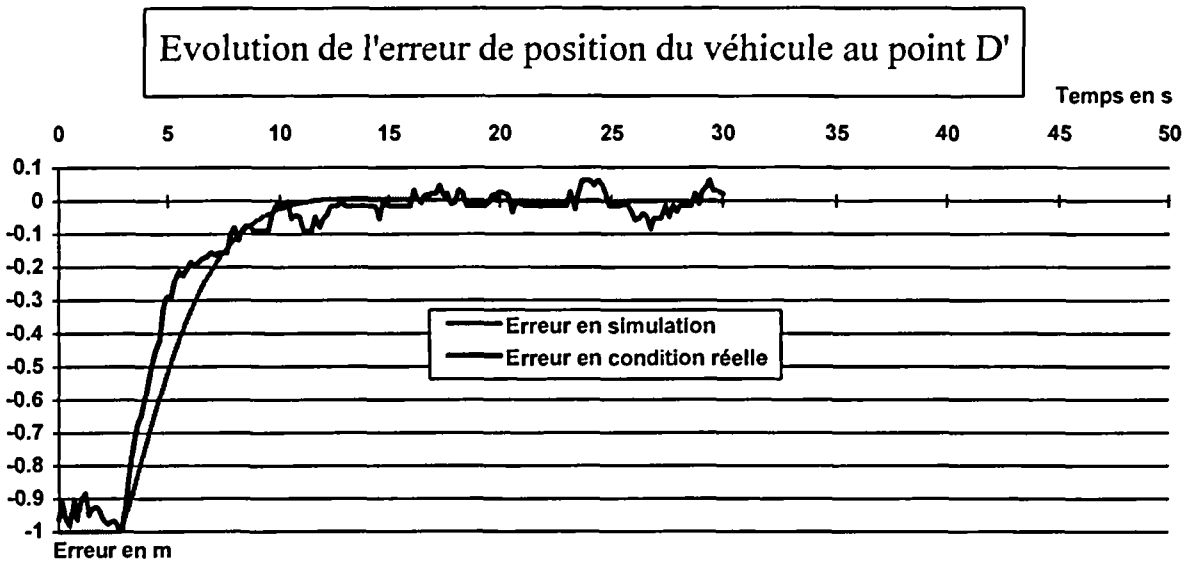


**Courbe IV 21 : réponse indicielle de la moissonneuse-batteuse pour  $v=4$  km/h**

La courbe (**Courbe IV 21**) montre une convergence rapide de la machine avec un temps de réponse de l'ordre de 15 s. Nous constatons que la courbe réelle et la courbe simulée sont superposées. Le bruit que l'on observe sur la courbe réelle est dû à la précision de  $\pm 5$  cm de l'algorithme utilisé pour dépouiller les images donnant la qualité de l'asservissement.



La courbe suivante a été réalisée à une vitesse moyenne de 8 km/h. La loi de commande est exactement la même que précédemment, le paramètre  $v_{\text{moy}}$  n'a pas été modifié afin de tester la robustesse de l'asservissement vis-à-vis de la vitesse moyenne de l'engin.



***Courbe IV 22 : réponse indicielle de la moissonneuse-batteuse pour  $v=8$  km/h***

La courbe (*Courbe IV 22*) montre un bon asservissement de la machine bien que la vitesse ait doublé. La courbe réelle est toujours très proche de la courbe simulée. Le temps de réponse est deux fois plus court, ce qui est normal puisque la vitesse a doublé et que la loi de commande est inchangée.

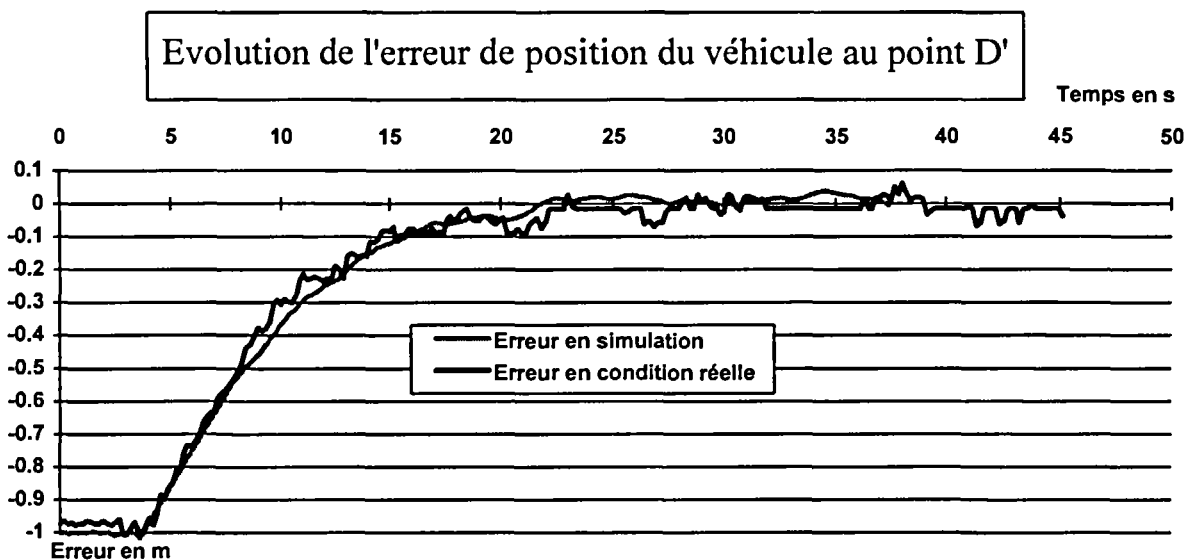
De ces courbes nous pouvons déduire que :

- Notre loi de commande référencée vision proportionnelle permet d'asservir correctement notre véhicule agricole sur une droite,
- elle est robuste vis-à-vis de la vitesse d'avancement (des variations de 100 % ne posent pas de problème),
- notre simulateur est fidèle à la réalité pour de faibles vitesses mais nous sommes convaincus que ce ne serait pas le cas pour des vitesses supérieures où la déformation des pneus ainsi que le comportement des structures hydrauliques et mécaniques de l'engin ne seraient plus négligeables.

### 2.2.2.1 Influence du bruit sur les mesures visuelles

La détection de la ligne blanche est beaucoup plus simple que celle de l'interface fauchée/non-fauchée de blé et surtout beaucoup plus précise. Afin de nous rapprocher des conditions réelles de travail de la machine (fauche de blé) nous allons rajouter un bruit blanc de  $\pm 16$  pixels sur la position de la droite détectée.

Le résultat de ce test est donné par la courbe suivante qui a été réalisée à une vitesse de 4 km/h.



***Courbe IV 23 : réponse indicielle pour un bruit blanc d'amplitude 32 pixels***

L'influence du bruit est inexistante sur cette courbe. En simulation, nous avons montré que, si l'influence du bruit était faible, elle existait quand même. Malheureusement, la précision de l'algorithme donnant la qualité de notre asservissement ne nous permet pas de mesurer l'influence d'un tel bruit car celle-ci demeure inférieure à cette précision.

### Conclusion

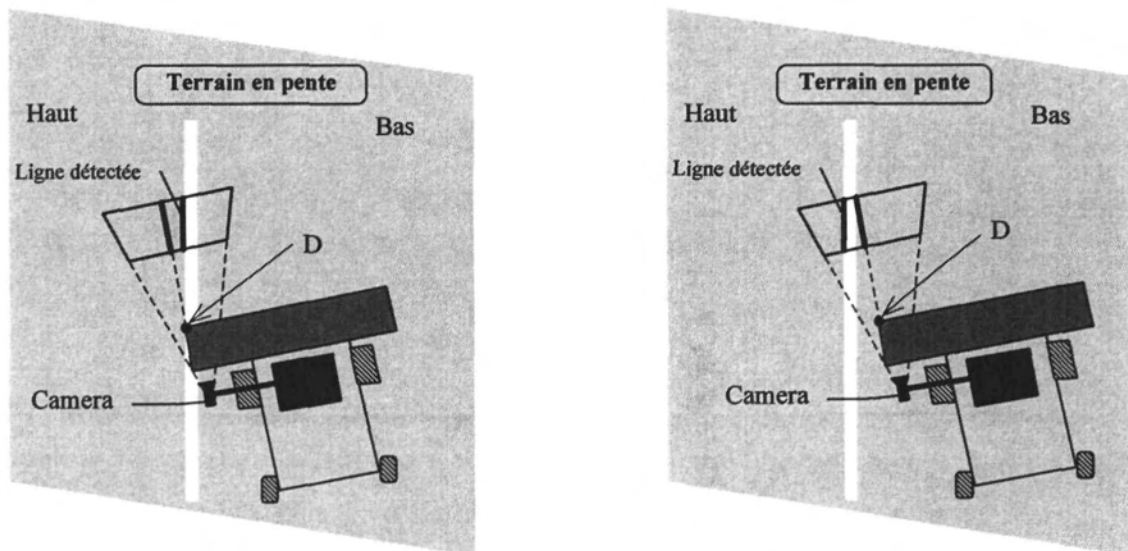
Les essais effectués avec la moissonneuse-batteuse ont montré la convergence et la stabilité de la commande référencée vision appliquée à un véhicule agricole non holonome. Cette loi de commande s'est montrée robuste par rapport aux variations de vitesse. Par contre, nous n'avons pas pu conclure quant à sa robustesse par rapport au bruit ajouté aux informations visuelles. Cependant, tous les essais effectués en simulation montrent un très bon comportement de la CRV par rapport à l'ajout d'un bruit à moyenne nulle aux mesures visuelles. Ceci nous permet d'être optimiste quant à l'emploi de la CRV en conditions réelles de travail (fauche de blé).

## 2.3 Contrôle de la moissonneuse-batteuse dans des conditions difficiles

Tous les essais que nous avons présentés ont été réalisés sur un terrain plat. Or, nos véhicules doivent régulièrement travailler sur des terrains en pente et avec des conditions d'adhérence précaires. Aussi, en 1995, nos efforts ont porté sur l'asservissement de nos prototypes dans des conditions « difficiles ».

### 2.3.1 Terrain en pente

Dans ce paragraphe, nous allons nous intéresser au cas où le terrain comporte une pente perpendiculaire à l'interface. Pour cela, nous avons tracé une ligne blanche sur un terrain en pente puis nous l'avons fait suivre par la moissonneuse-batteuse. Les deux schémas suivants présentent le suivi de cette ligne lorsqu'un opérateur pilote la machine puis lorsque nous utilisons la CRV proportionnelle.

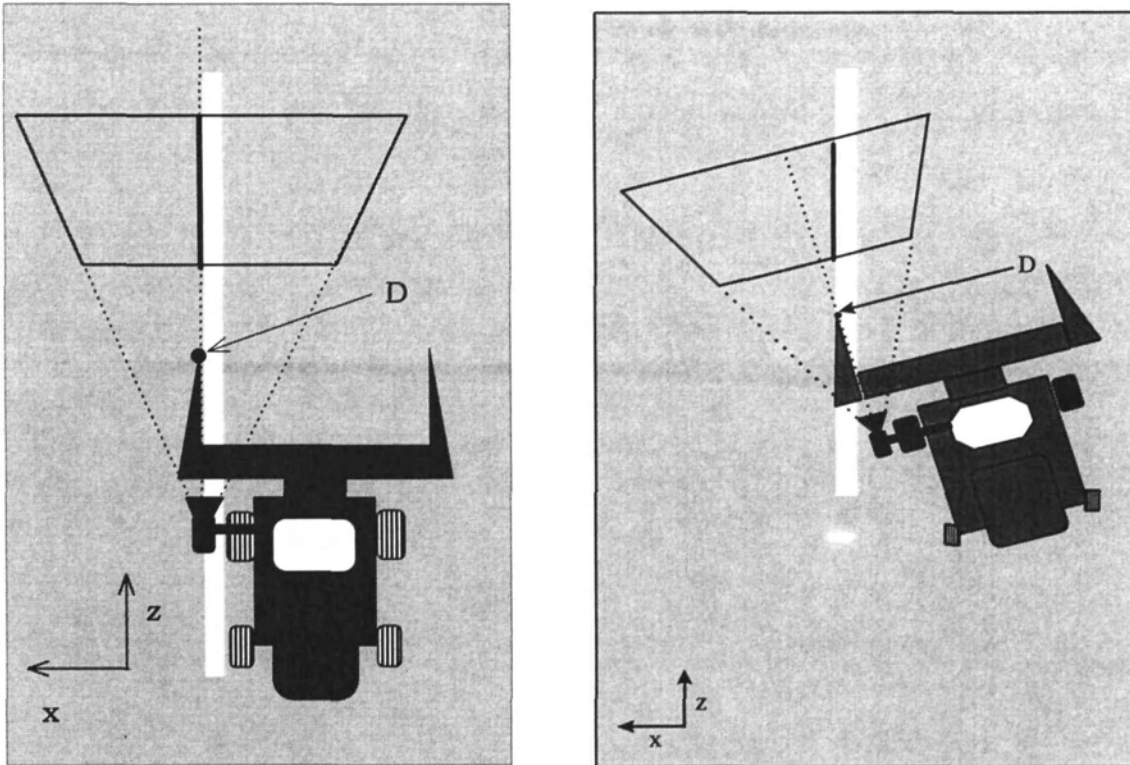


**Figure IV 23 : évolution de la machine sur un terrain en pente**

Dans ces deux exemples nous constatons que :

- dans les deux cas la machine fait un angle avec l'interface et la consigne de braquage est non nulle,
- avec la CRV il subsiste une erreur statique importante,
- lorsque la machine est correctement asservie, la ligne détectée n'est pas superposée à la ligne de référence.

Examinons les entrées et la sortie de notre système de commande lorsque le terrain est plat et lorsqu'il présente une pente transversale :

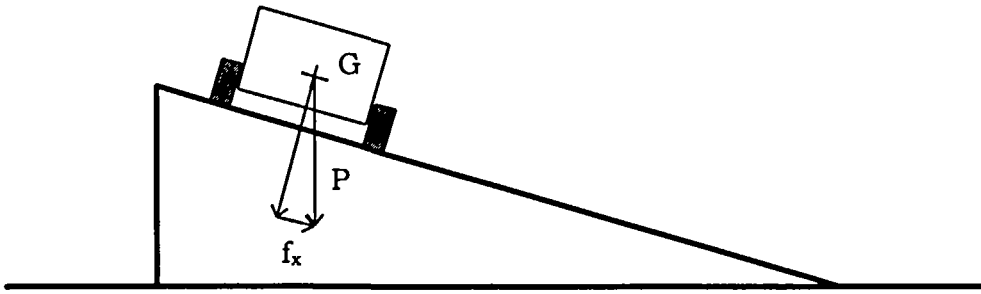


**Figure IV 24 : deux droites de références pour deux situations différentes**

Comme on peut le voir sur les figures précédentes, la droite de référence n'est pas la même dans les deux cas. Ceci est dû à l'orientation de la machine qui est différente. En effet, lorsque le terrain est en pente, la machine a tendance à glisser vers le bas. Pour compenser ce phénomène, l'opérateur braque les roues directrices et change donc l'orientation de la machine. Afin de modéliser les différents paramètres intervenant dans cette situation nous allons écrire l'équation fondamentale de la dynamique pour un corps solide sur un plan incliné.

### Equations dynamiques

Sur un terrain en pente, le poids de la machine exerce une force dont une des composantes est parallèle à l'axe  $Ox$  (**Figure IV 25**). Cette composante que nous appellerons  $f_x$  a tendance à faire glisser la machine vers le bas de la pente. Elle peut être modélisée par une accélération latérale.



**Figure IV 25 : bilan des forces lorsque le véhicule évolue sur une pente transversale**

Ecrivons la relation fondamentale de la dynamique avec glissement :

$$\sum \vec{f} = m \cdot \vec{a}$$

Projetons cette relation sur l'axe  $Ox$  perpendiculaire à la ligne à suivre. A l'équilibre, nous pouvons écrire pour le centre de gravité de notre machine :

eq IV 3 
$$m \cdot \ddot{x} = f_x$$

Cette relation n'est qu'une représentation très grossière et très partielle des phénomènes dynamiques qui interviennent lorsque la machine évolue sur ce type de terrain. Nous l'avons retenue car nous n'en ferons **qu'une exploitation qualitative**. Ce que nous allons montrer, c'est que les perturbations dues à la pente peuvent être approchées par ce type de modélisation. En fait, ce n'est pas la pente du terrain que nous souhaitons modéliser mais son action sur notre système de commande et sur la position latérale de la machine.

Avec les approximations d'usage aux petits angles, les équations cinématiques de la moissonneuse-batteuse deviennent (cf. chapitre II) :

eq IV 4 
$$\begin{cases} \dot{x} = v_{\text{moy}} \cdot \phi \\ \dot{\phi} = -\frac{v_{\text{moy}}}{L} \cdot \delta \end{cases}$$

Nous obtenons pour l'accélération latérale de l'essieu rigide le modèle :

eq IV 5 
$$\ddot{x} = -\frac{v_{\text{moy}}^2}{L} \cdot \delta$$

En considérant que le centre de gravité de la machine se trouve au milieu de l'essieu rigide, les équations (eq IV 3) et (eq IV 4) nous donnent à l'équilibre :

$$\text{eq IV 6} \quad \ddot{x} = v_{\text{moy}} \cdot \dot{\phi} = -\frac{v_{\text{moy}}^2}{L} \cdot \delta = \frac{f_x}{m}$$

L'équation (eq IV 6) montre que, pour compenser l'action de la pente sur la position latérale de notre machine, il faut que :

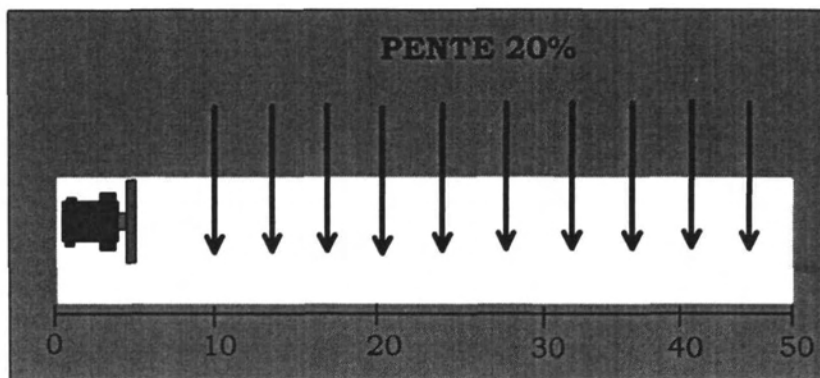
- l'angle des roues directrices ( $\delta$ ) ne soit pas nul (même si la machine est correctement asservie),
- ( $\dot{\phi}$ ) la vitesse de rotation de l'essieu rigide donc de la machine soit constante.

Afin de simuler l'action de la pente sur notre machine, nous avons modifié les équations cinématiques de la moissonneuse-batteuse en rajoutant les termes :

$$\begin{cases} \dot{x}_{\text{pente}} = \frac{f_x}{m} \cdot t \\ \dot{\phi} = \frac{f_x}{v_{\text{moy}} \cdot m} \end{cases}$$

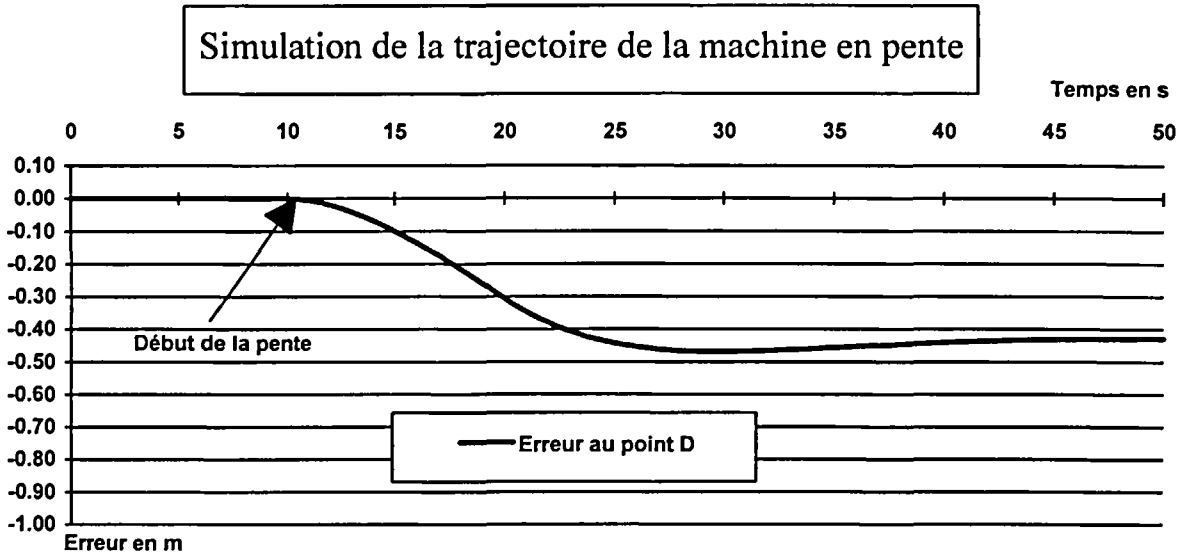
avec des conditions initiales nulles.

Simulons maintenant une trajectoire rectiligne avec l'apparition d'une pente à  $t=10$  s. Pour calculer le coefficient  $f_x$  nous avons identifié l'erreur statique mesurée en condition réelle avec celle constatée en simulation pour la même valeur de pente.

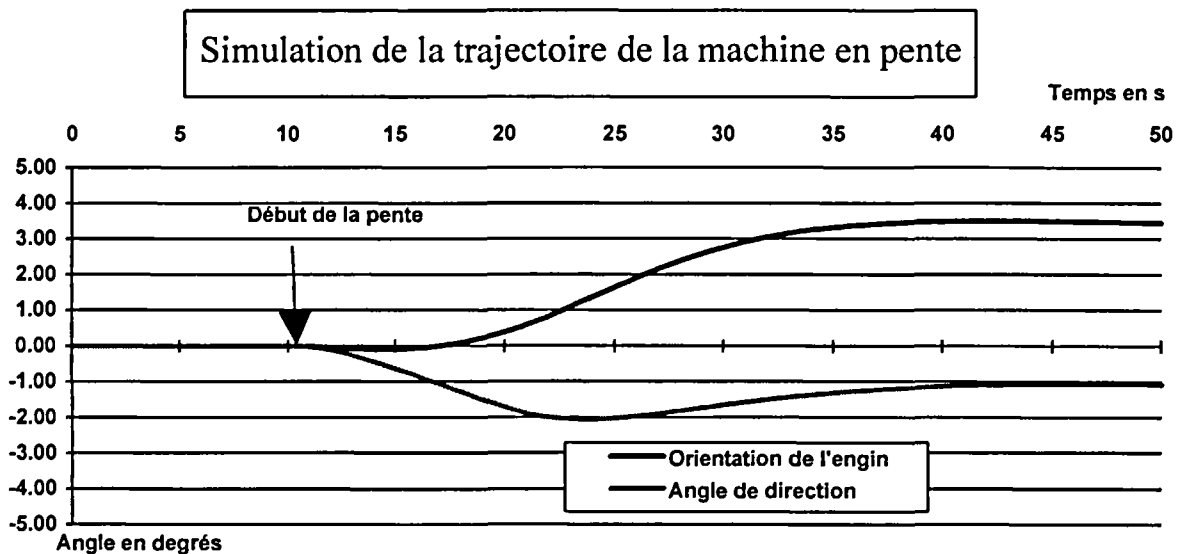


**Figure IV 26 : conditions de simulation**

Les courbes suivantes représentent une simulation de la trajectoire de notre machine lorsqu'elle évolue sur un terrain avec une pente perpendiculaire à sa trajectoire à partir de  $t=10$  s.



**Courbe IV 24 : suivi d'une trajectoire rectiligne sur un terrain en pente**



**Courbe IV 25 : suivi d'une trajectoire rectiligne sur un terrain en pente**

La première courbe (**Courbe IV 24**) donne l'erreur au point D. Bien que cette erreur soit importante (45 cm), la trajectoire de la machine se stabilise parallèlement à l'interface. La deuxième courbe (**Courbe IV 25**) donnant l'angle de direction de l'engin montre bien qu'il y a réaction du système de commande afin de compenser la dérive due à la pente mais, manifestement, cette réaction n'est pas

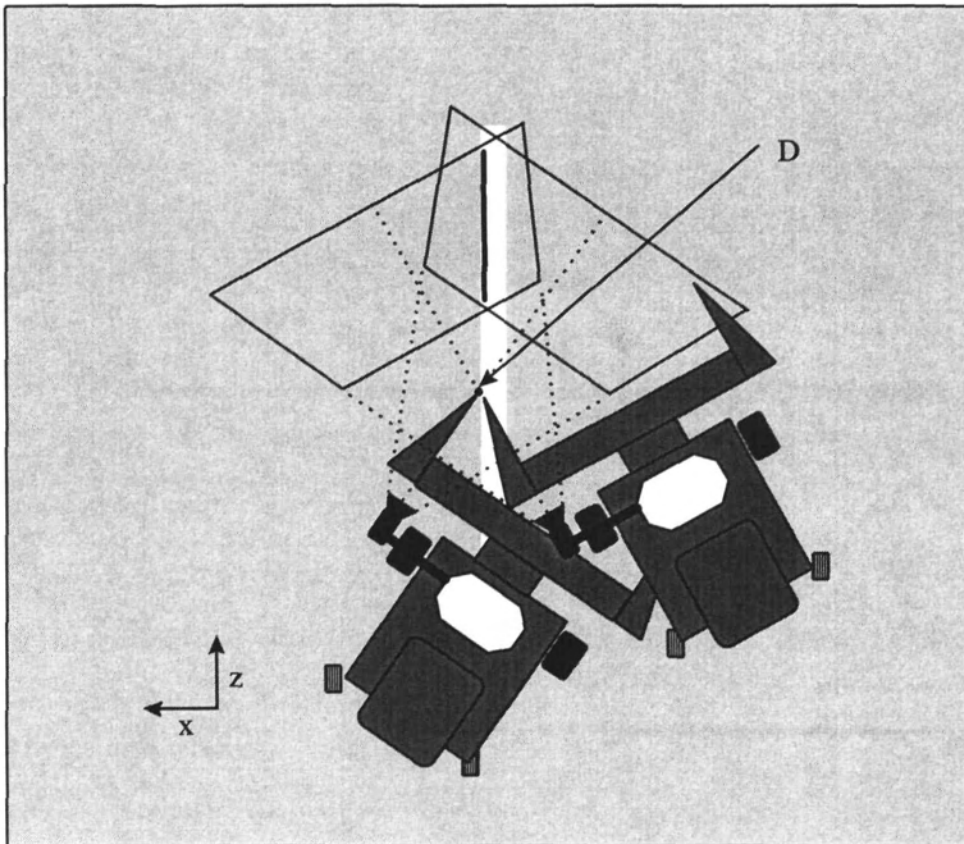
suffisante. Ce cas de convergence, avec une erreur statique non nulle, est caractéristique d'un correcteur proportionnel et est généralement résolu par l'ajout d'un intégrateur au système.

L'asservissement de notre machine évoluant sur un terrain en pente se décomposera en deux phases. La première consistera à analyser l'action de la pente sur l'engin. La deuxième devra corriger cette action en ajoutant un pôle au système.

#### Analyse de l'erreur statique en pente

Dans le cadre de nos applications nous ne désirons pas utiliser d'autre capteur que la caméra ; nous chercherons donc à détecter la pente dans l'image. En fait, ce n'est pas réellement la pente du terrain qui nous intéresse mais son influence sur la position de la machine et sur les paramètres visuels  $\rho$  et  $\theta$  extraits de l'image.

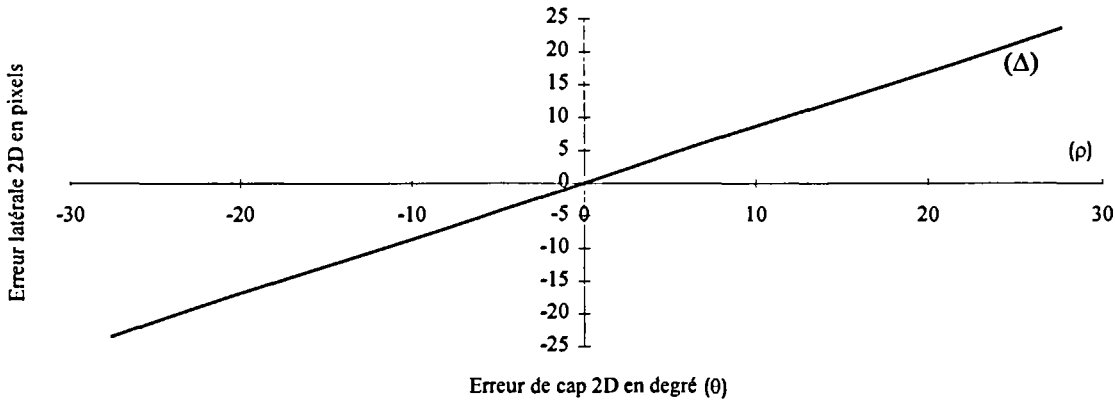
Pour cela, nous allons considérer de multiples cas de figure où la machine évolue en pente. Entre les différentes situations, le point D étant correctement asservi, la seule différence est l'orientation de la machine par rapport à l'interface (**Figure IV 27**).



**Figure IV 27 : rotation de la machine autour du point D**



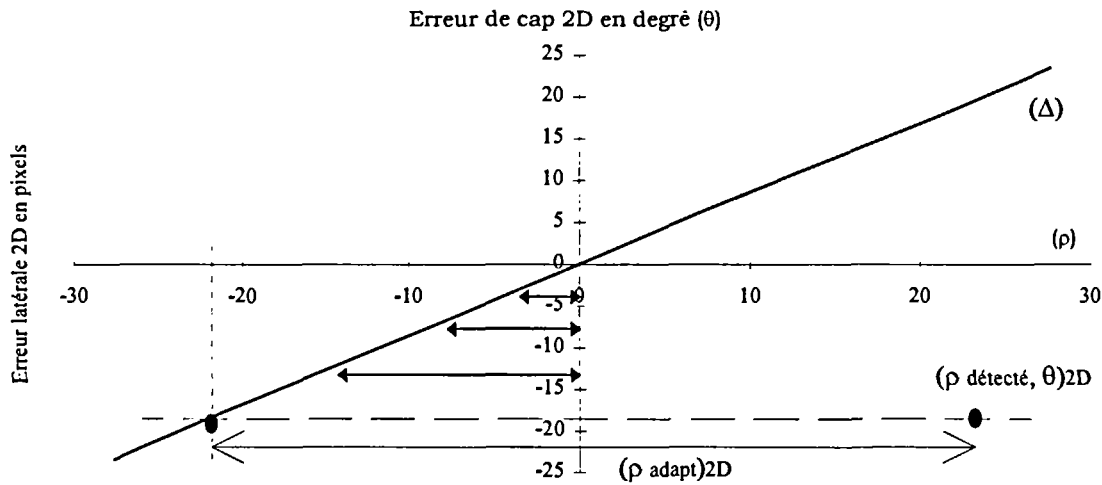
Pour chaque valeur considérée de la pente nous allons noter les paramètres  $\rho$  et  $\theta$  de la droite détectée dans l'image, puis nous allons les représenter :



**Courbe IV 26 : ensemble des points de fonctionnement**

Pour des pentes comprises entre -40% et +40% nous obtenons une droite passant par l'origine (pente nulle). De cette droite, nous déduisons que tout asservissement correct de la machine correspond à un point de la droite  $(\Delta)$ . La droite  $(\Delta)$  représente donc l'ensemble des points de fonctionnement de la machine.

Lorsque la machine évolue sur un terrain en pente, la CRV proportionnelle induit un changement d'orientation fonction de l'action de la pente sur la machine. Cette orientation permet à l'engin de se stabiliser avec une erreur statique constante. Le couple  $(\rho, \theta)_{2D}$  alors détecté dans l'image n'est pas un couple de  $(\Delta)$ . L'angle  $\theta_{2D}$  détecté est, cependant, très proche de celui que l'on aurait si on était parfaitement asservi. En effet, lorsque la machine est asservie avec une erreur statique non nulle, son orientation est la même que lorsqu'elle est asservie avec une erreur statique nulle. Cette orientation n'est fonction que de la vitesse et de l'action de la pente sur la machine. Dans l'image, la différence d'orientation de l'interface entre les deux situations existe car la projection faite par la caméra ne conserve pas les angles. Cependant, on considérera que la machine sur une période (300 ms) ne s'est pas suffisamment décalée pour que la valeur de  $\theta_{2D}$  détectée soit différente de celle du point de fonctionnement appartenant à la droite  $(\Delta)$ . Nous mesurons alors la différence entre la valeur détectée  $\rho_{\text{détectée}}$  et la valeur de  $\rho$  correspondant à  $\theta$  sur la droite  $(\Delta)$ . Nous appelons cette différence  $\rho_{\text{adapt}}$ .



**Courbe IV 27 : détermination du point de fonctionnement**

Les entrées de la loi de commande ne seront plus alors  $\rho$  et  $\theta$  mais  $\rho_{\text{adapt}}$  et  $\theta$ . Cela revient à décaler petit à petit les coordonnées de notre point de fonctionnement jusqu'à atteindre celui correspondant à l'action de la pente sur le véhicule.

### Loi de commande

Nous avons montré que le point de fonctionnement de la machine n'était pas le même lorsqu'elle évoluait sur un terrain en pente ou sur un terrain plat. Le paragraphe précédent expose le moyen de calculer de proche en proche ce nouveau point de fonctionnement correspondant à l'asservissement correct de la machine.

La première loi de commande proportionnelle fera correspondre à un asservissement correct de la machine une consigne  $\delta$  nulle. Or, nous avons montré que la pente induisait une vitesse de rotation de la machine non nulle et donc un angle  $\delta$  non nul. Il paraît donc utile de rajouter un intégrateur au système. La deuxième loi de commande référencée vision, présentée au chapitre III, utilise un intégrateur qui rajoute un pôle au système :

$$\dot{\phi} = \Omega_y = \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \left( \theta + k_1 \cdot \int_0^t \theta \cdot du \right) + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \left( \rho + k_2 \cdot \int_0^t \rho \cdot du \right) \right]$$

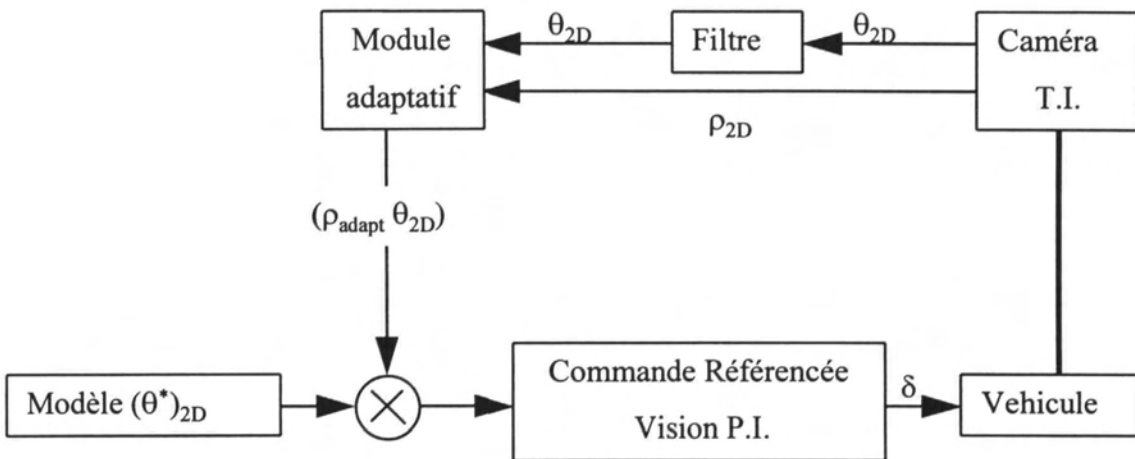
La correction de la trajectoire de la machine en pente ne se fait que par l'intermédiaire du paramètre  $\rho$ , donc nous prendrons :

$$\begin{cases} k_1 = 0 \\ k_2 \neq 0 \end{cases}$$

Notre nouvelle loi de commande sera donc :

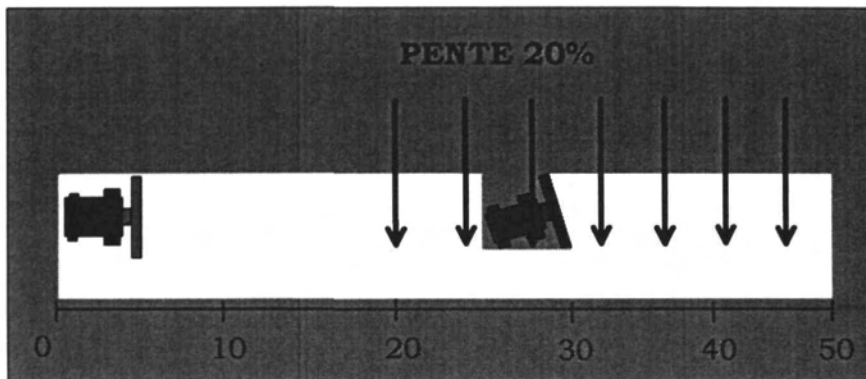
$$\dot{\phi} = \Omega_y = \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1+h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1+h^2} \right) \cdot \left( \rho_{\text{adapt}} + k_2 \cdot \int_0^t \rho_{\text{adapt}} \cdot du \right) \right]$$

En intégrant la droite ( $\Delta$ ), qui donne l'ensemble des points de fonctionnement et qui définit l'action de la pente sur la trajectoire de la machine, sous forme d'un élément appelé module adaptatif nous obtenons le schéma d'asservissement suivant :



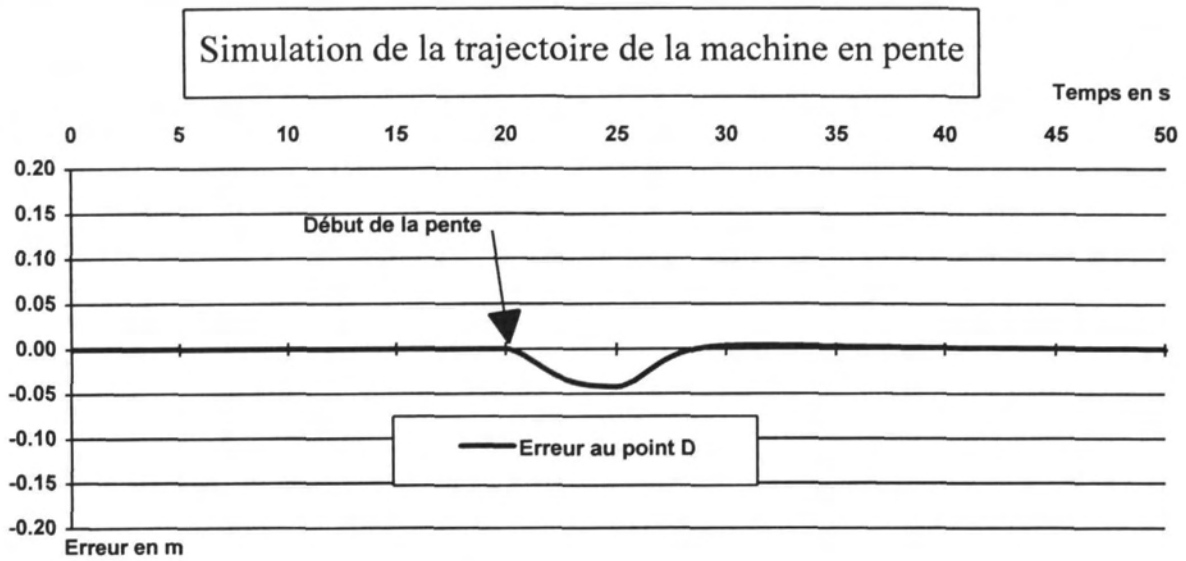
**Figure IV 28 : implantation d'un coorrecteur PI dans la boucle d'asservissement**

Avec cette dernière loi de commande que nous appellerons CRV intégrale, et en utilisant le processus « adaptatif » précédemment décrit, nous avons fait de nouvelles simulations. Pour cela, nous simulons une pente à partir de  $t=20s$  (**Figure IV 29**) sur une trajectoire rectiligne.



**Figure IV 29 : simulation d'une pente**

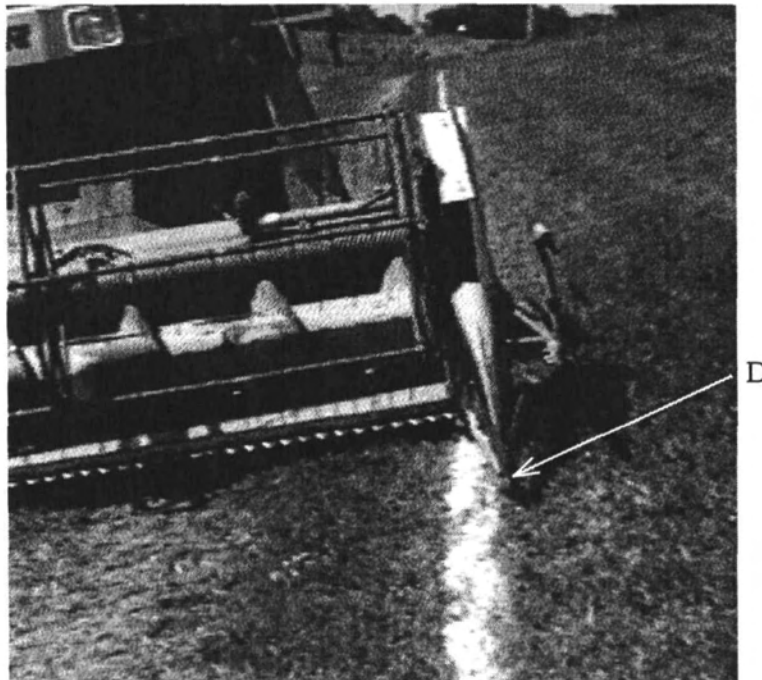
La courbe suivante donne l'erreur au point D le long de la trajectoire précédente.



***Courbe IV 28 : suivi de trajectoire sur terrain en pente avec la CRV intégrale***

La courbe précédente montre que l'effet de la pente est corrigé en 10 s et que l'erreur qu'elle provoque n'excède pas 5 cm.

Enfin, nous avons testé cette nouvelle loi en grandeur réelle sur un terrain ayant une pente de 20 %. Malheureusement, nous n'avons pas pu utiliser une deuxième caméra pour mesurer la qualité de notre asservissement. Cependant, le résultat est excellent comme le montre la photo prise lors de ces essais :



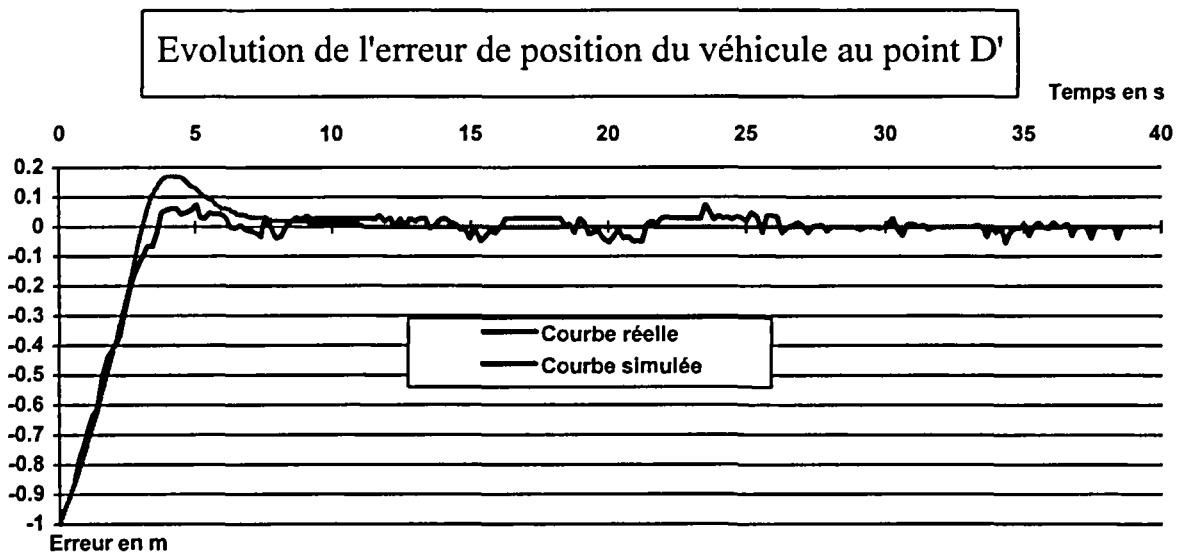
***Photo IV 1 : asservissement de la machine en conditions difficiles***

## Conclusion

Grâce à la droite ( $\Delta$ ) qui donne l'ensemble des points de fonctionnement et au procédé exposé précédemment qui permet de déterminer la valeur du point de fonctionnement, nous avons pu analyser l'action de la pente dans l'image. L'erreur calculée est alors la nouvelle entrée d'une loi de commande référencée vision intégrale qui permet de corriger l'action de la pente transversale sur la position du diviseur gauche de la machine.

### 2.3.2 Autres résultats avec la CRV intégrale

Dans ce paragraphe, nous avons fait quelques tests sur la nouvelle loi de commande afin de déterminer si elle ne perturbait pas le fonctionnement de nos véhicules sur terrain plat. Les courbes suivantes donnent la réponse de la machine à un échelon en simulation et en conditions réelles (suivi de ligne blanche) avec la CRV intégrale. Pour cet exemple, la vitesse de la machine est de 6 km/h.

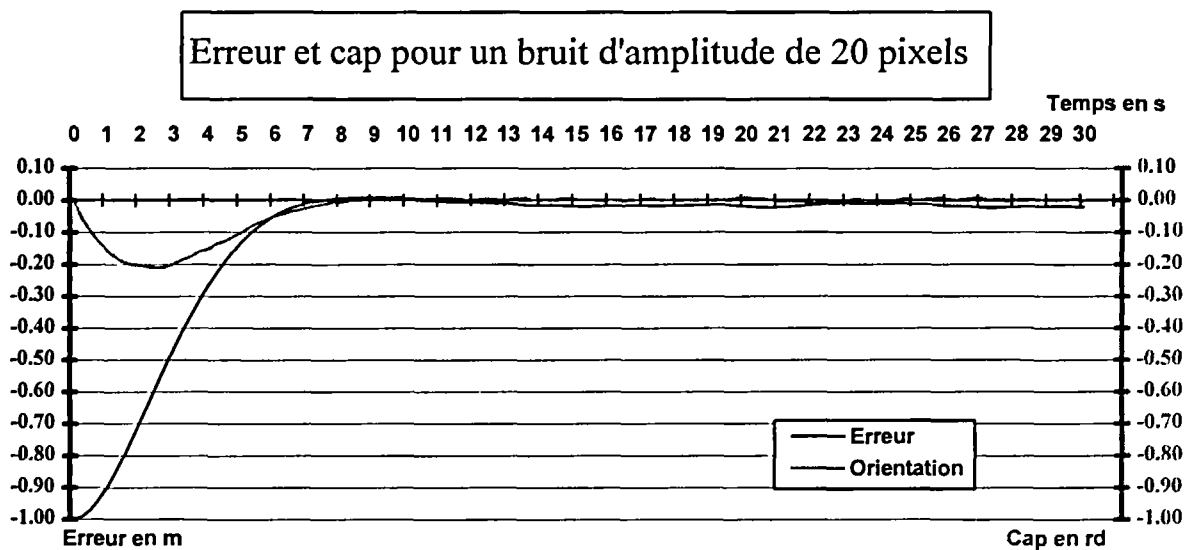


**Courbe IV 29 : réponse indicielle de la moissonneuse-batteuse avec la CRV intégrale**

Les courbes précédentes montrent un bon asservissement de la machine lors de la réponse à un échelon d'un mètre. Cependant, le temps de réponse de 4 secondes est relativement faible. Ce n'est pas un avantage car cela impose une dynamique trop forte à notre véhicule. Le principal inconvénient de notre loi de commande référencée vision intégrale est d'imposer une dynamique très forte au système si on désire compenser efficacement l'action de la pente sur la position de

la machine. Cet inconvénient est dû au « temps de montée » de l'intégrale que l'on compense en partie par des gains élevés. Ceux-ci risquent de rendre le système beaucoup plus sensible au bruit et aux perturbations de toutes sortes. Si on diminue ces gains, l'action de la pente est toujours compensée, mais avec une période transitoire plus longue.

Donnons, à titre de comparaison, les courbes de réponses indicielles pour la CRV intégrale incluant un bruit d'amplitude 20 pixels sur les informations visuelles. Ces courbes ont été réalisées en utilisant le modèle cinématique de la tondeuse et à une vitesse de 3 km/h afin de pouvoir les comparer avec celles déjà faites dans les mêmes conditions (paragraphe 1.1.2.1).



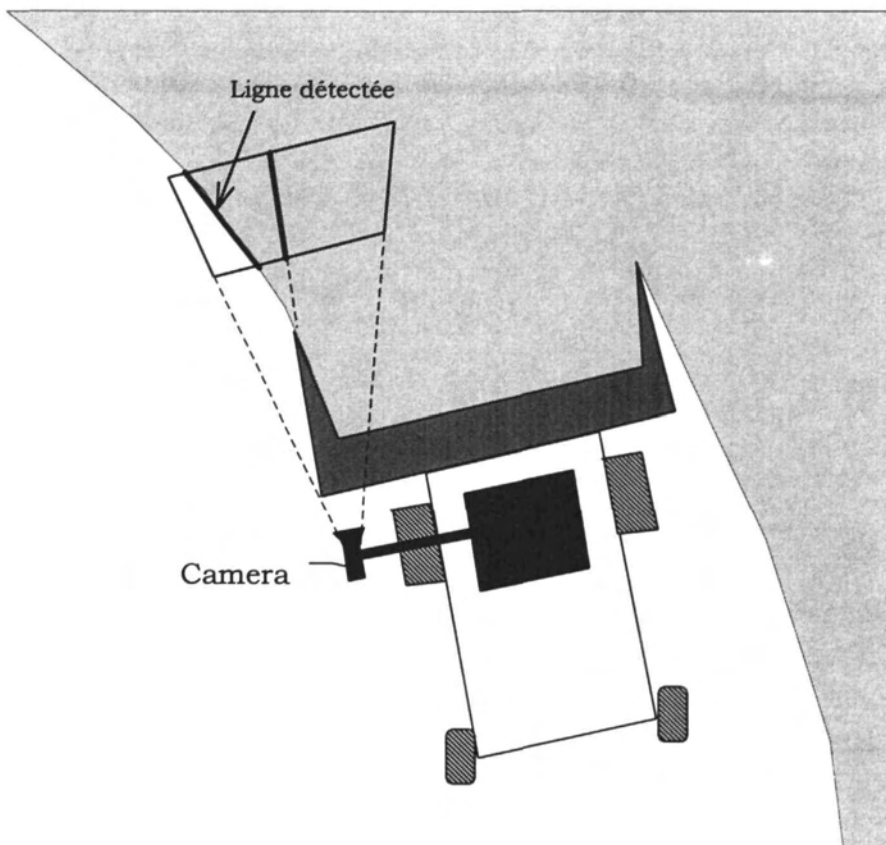
**Courbe IV 30 : réponse indicielle de la tondeuse avec la CRV intégrale pour un bruit de 20 pixels**

Si on compare ces résultats avec ceux obtenus pour les autres lois de commande il est clair que la CRV intégrale est un peu moins robuste que la CRV proportionnelle, mais beaucoup plus que la commande « trajectoire », surtout en ce qui concerne l'orientation de l'engin qui ne varie pas de plus de 1.5 degré.

## **Remarques**

### **① Et pour un virage ?**

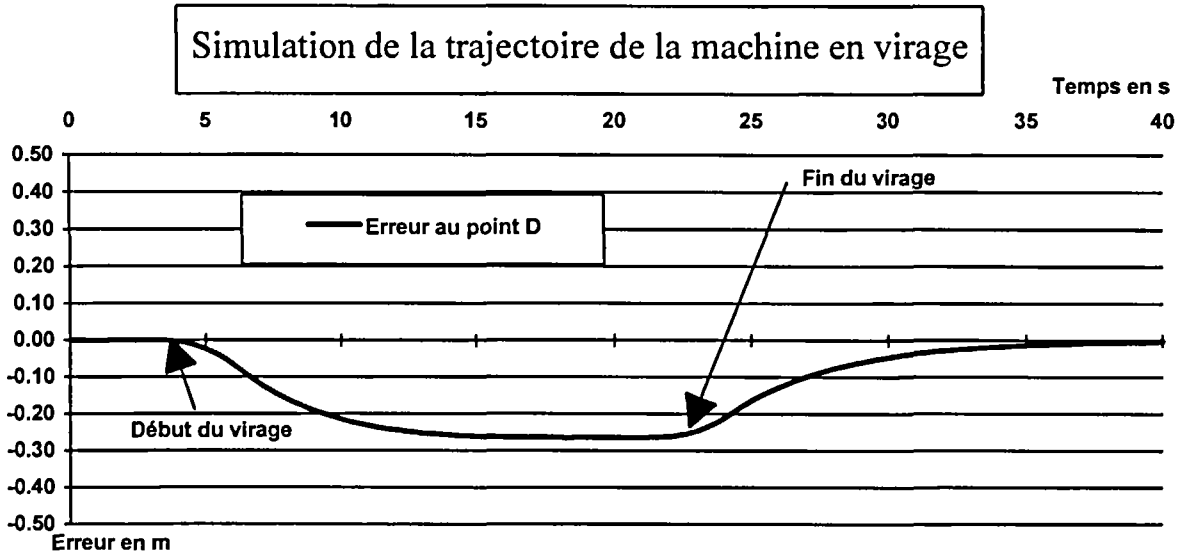
Il est évident que la primitive de type droite n'est pas adaptée à la modélisation des virages. En effet, elle suppose que la droite détectée est parallèle à la direction de la machine. Il est clair que ce n'est pas le cas lorsque la trajectoire est courbe (**Figure IV 30**).



**Figure IV 30 : trajectoire à suivre avec courbure**

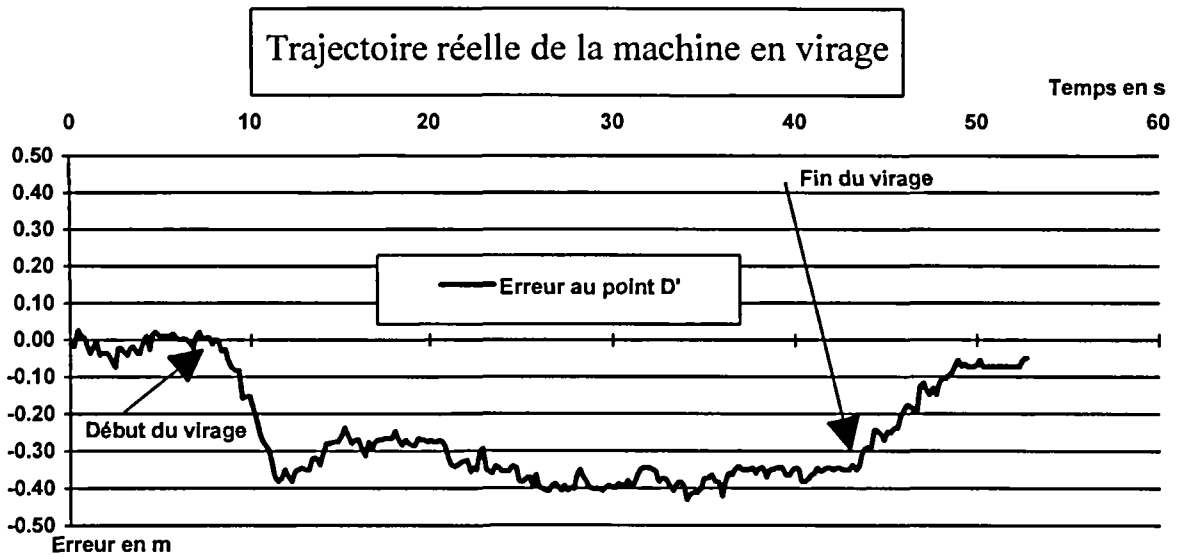
Comme dans le cas de la pente, la droite de référence n'est plus la même. Elle dépend de la caméra et de sa position mais aussi du rayon de courbure de la trajectoire. Avec la primitive de type droite, et en considérant seulement le point de fonctionnement ( $\rho=0$ ,  $\theta=0$ ), l'engin va avoir tendance à « couper » le virage.

Donnons un exemple de simulation du suivi d'une courbe ayant un rayon de 30 m avec la CRV proportionnelle.



***Courbe IV 31 : simulation d'un virage avec la CRV***

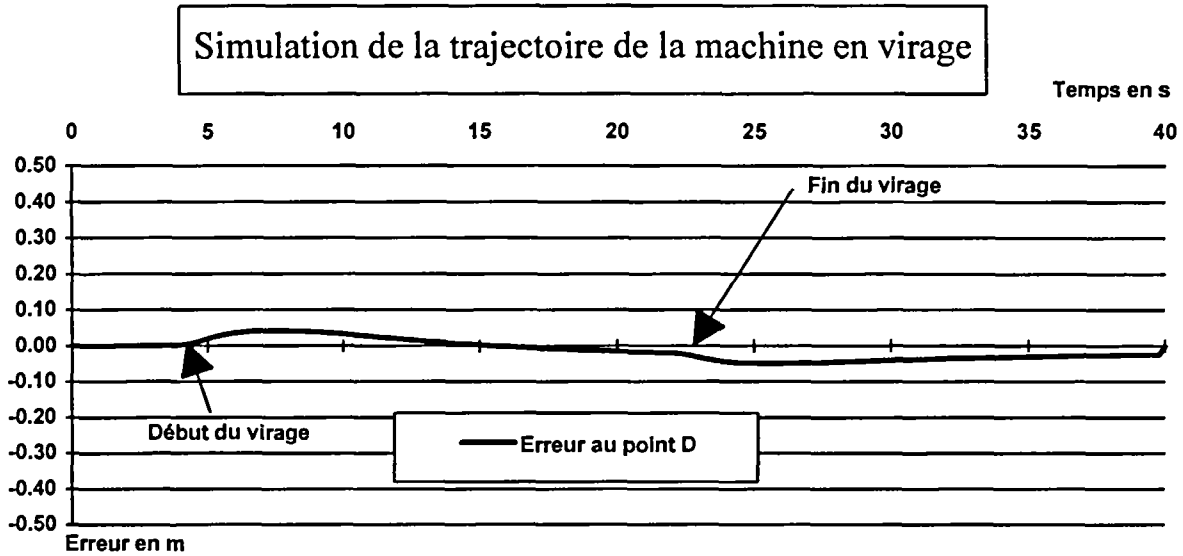
Avec la première loi de commande référencée vision (proportionnelle) on constate une erreur importante que l'on retrouve dans nos tests réels :



***Courbe IV 32 : suivi d'un virage en conditions réelles avec la CRV***

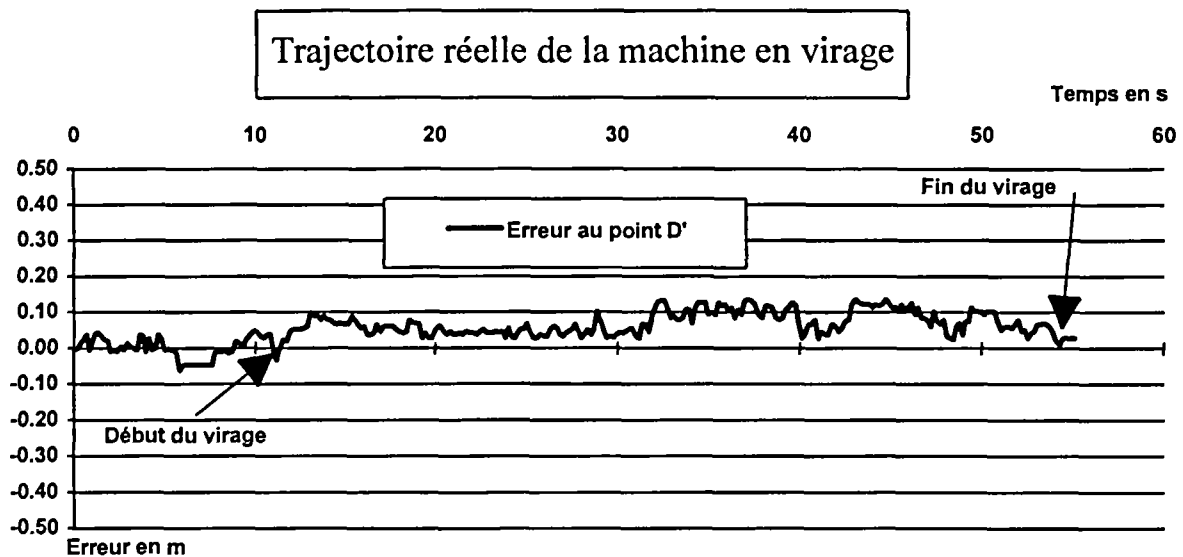


Nous avons refait les mêmes tests avec la deuxième loi de commande (CRV intégrale) et l'ensemble des points de fonctionnement ( $\Delta$ ) :



***Courbe IV 33 : simulation d'un virage avec la CRV intégrale***

Les résultats pour cette dernière loi de commande sont excellents sauf pour le début et la fin du virage qui correspondent au temps de montée de l'intégrale. De même, des tests en conditions réelles montrent un bon asservissement de la machine en virage :

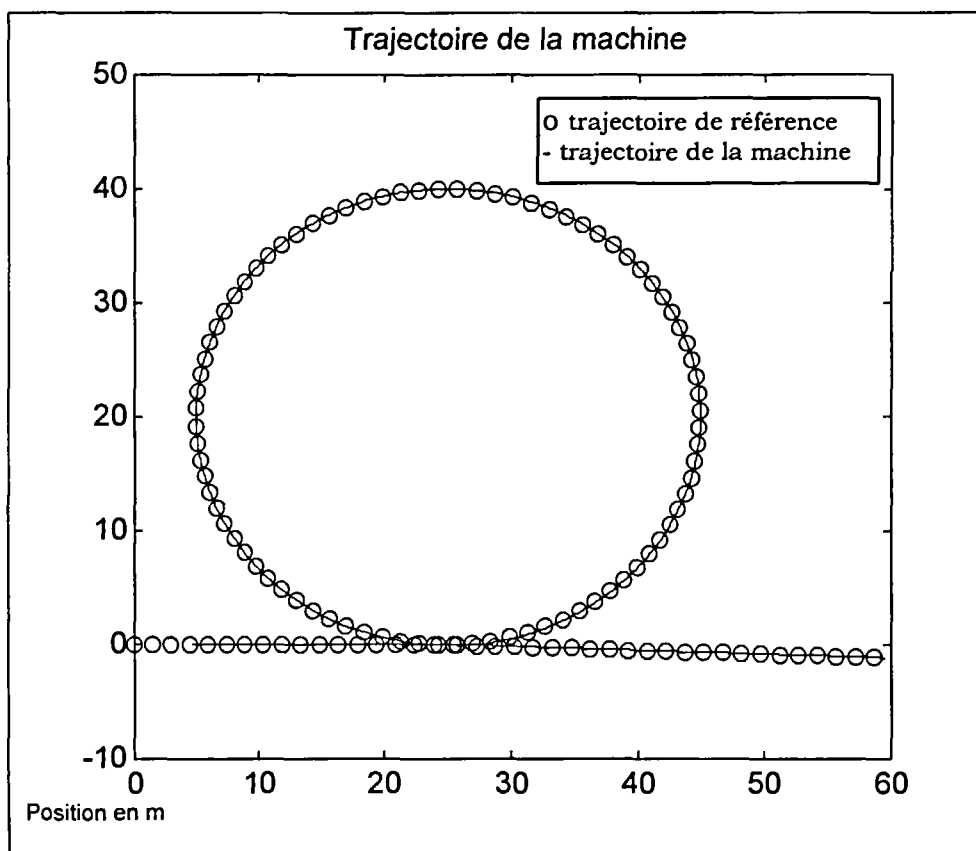


***Courbe IV 34 : suivi d'un virage en conditions réelles avec la CRV intégrale***

### Explications

La deuxième loi de commande référencée vision a été développée pour des situations de travail en pente. Nous avons vu que l'action de la pente pouvait être approchée par une accélération latérale constante. La trajectoire qui résulterait de cette seule accélération est une parabole. Cette loi de commande permet donc d'asservir notre robot mobile sur une trajectoire en forme de parabole. Notre période d'échantillonnage étant de l'ordre de 300 ms, nous pouvons approximer par morceau une courbe par des parties de parabole de même longueur.

Donnons un exemple supplémentaire, celui d'une trajectoire en forme de cercle :



**Courbe IV 35 : suivi d'une trajectoire en forme de cercle**

### ② : travail en condition difficile

Les véhicules agricoles évoluent souvent sur des terrains en pente et les conditions d'adhérence des machines peuvent être très variables au cours du temps. Nos lois de commande doivent être capables de s'adapter à toutes ces conditions. Donnons, comme exemple, le patinage des roues qui peut provoquer une rotation et une dérive de la machine ; la CRV intégrale permet de compenser ces perturbations qui induisent une erreur statique avec une loi de type proportionnelle.

## **Conclusion**

Nos applications ne sont pas très exigeantes au niveau des performances des lois de commande. Cependant, elles nécessitent une grande robustesse par rapport aux bruits et aux variations de modèles. En effet, le passage  $n$  de l'engin définira la trajectoire du passage  $n+1$  et ainsi de suite. L'algorithme de commande devra donc avoir un effet filtrant par rapport aux variations de modèles et aux bruits introduits par la précision et l'instabilité des algorithmes de traitement d'images. Ceci évitera qu'une perturbation ne se propage en s'amplifiant d'un passage de la machine sur l'autre.

Les différents essais de simulations ont montré la convergence de nos lois de commande avec des performances suffisantes pour nos applications. Ces tests ont également montré que la CRV était beaucoup plus robuste par rapport aux bruits que la commande « trajectoire ».

Cependant, les essais que nous avons faits, en conditions réelles, ont mis en évidence les limites de la primitive visuelle de type droite utilisée pour la CRV. Cette primitive ne relate pas forcément la liaison que l'on désire entre la trajectoire à suivre et la position du véhicule agricole. Par exemple, lorsque le terrain présente une pente transversale par rapport au sens d'avancement ou lorsque la trajectoire présente une courbure faible, le point de fonctionnement varie. Nous avons mis au point un système permettant de modéliser l'effet de la pente sur la position de la machine ou la présence d'une courbure constante dans une trajectoire. Cette modélisation utilise le cap de la droite détectée dans l'image afin de déterminer le nouveau point de fonctionnement du système. Les résultats obtenus avec cette modélisation sont tout à fait satisfaisants aussi bien pour le cas du terrain en pente que pour une trajectoire comportant une courbure constante. Cette modélisation ne détermine pas une seule droite de référence mais un ensemble de droites de références correspondant à l'asservissement de l'extrémité du diviseur gauche (pour la moissonneuse-batteuse) et à une orientation variable. Le correcteur proportionnel intégral conserve une vitesse de rotation non nulle à la machine lui permettant de suivre une courbe ou une trajectoire rectiligne sur terrain en pente. Les inconvénients de ce type de correcteur sont connus : temps de montée de l'intégrale, retard... Il serait donc intéressant de chercher un autre correcteur n'ayant pas les défauts de l'intégrale mais ayant une sortie persistante lorsque l'erreur s'annule (par exemple une commande avec modèle interne).

# Chapitre V

## Perspectives

## INTRODUCTION

Le domaine de la robotique mobile agricole ne se limite pas naturellement aux deux engins que nous avons présentés. Il est donc intéressant de montrer comment on peut utiliser la modélisation décrite au chapitre II pour calculer les équations cinématiques simplifiées d'autres véhicules.

Au cours des essais du système d'aide à la conduite dans des conditions réelles de travail, nous avons constaté des problèmes de détection de l'interface fauchée/non-fauchée qui m'ont conduit à utiliser une interface plus simple à détecter (une ligne blanche) pour valider nos lois de commande. Aussi, des études sont réalisées pour ajouter d'autres capteurs extéroceptifs afin d'accroître la robustesse du module de perception. Il est alors envisagé d'effectuer une fusion des données perçues multi-capteurs ou multi-algorithmiques et d'en transmettre le résultat au module de commande du système d'aide à la conduite.

Une autre voie peut également être proposée. Elle consisterait à fusionner les données sur l'environnement au niveau de la synthèse du correcteur, c'est-à-dire à rapprocher l'étape de fusion de la prise de décision et de l'action. On peut également envisager un « noyau fédérateur » possédant une forme « d'intelligence » et permettant l'interaction des différentes données de perception et de contrôle. De ce fait, la connaissance de chaque sous-système serait enrichie par les hypothèses émises à partir des données calculées dans les autres modules de l'ensemble.

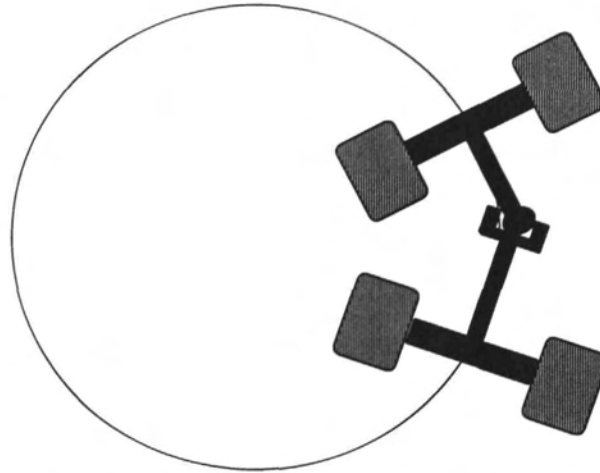
## 1. PLATE-FORMES MOBILES ARTICULEES

Les machines que nous avons présentées regroupent une grande partie de l'ensemble des véhicules agricoles quant à leur mobilité. Par exemple, le modèle à deux roues fixes et deux roues directrices concerne tous les tracteurs agricoles, les moissonneuses-batteuses, les ensileuses, etc... Cependant, au moins une catégorie de matériel n'a pas été envisagée. Elle est représentée par les engins ayant 2 essieux à roues fixes et une articulation entre les deux essieux (*Photo V 1*).



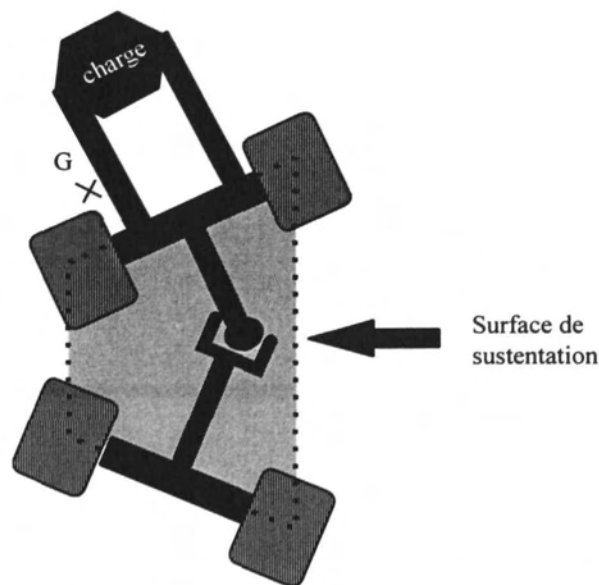
*Photo V 1 : un engin agricole articulé*

Ces machines présentent l'avantage d'avoir une grande maniabilité et un rayon de courbure très faible sans provoquer des glissements très importants au niveau des roues comme le font les véhicules de type « chenillard ». Un autre avantage est que la symétrie du véhicule, lorsqu'elle existe, permet aux deux essieux de suivre la même trajectoire (*Figure V 1*).



***Figure V 1 : trajectoire d'un engin articulé***

Cependant, ces véhicules comportent un inconvénient majeur pour le domaine agricole. De par le type de direction retenu, il est possible que le centre de gravité sorte de la surface de sustentation lors d'une manoeuvre. C'est le cas lorsque l'engin évolue sur terrain en pente ou lorsqu'il est utilisé pour la manipulation de charges importantes (***Figure V 2***).

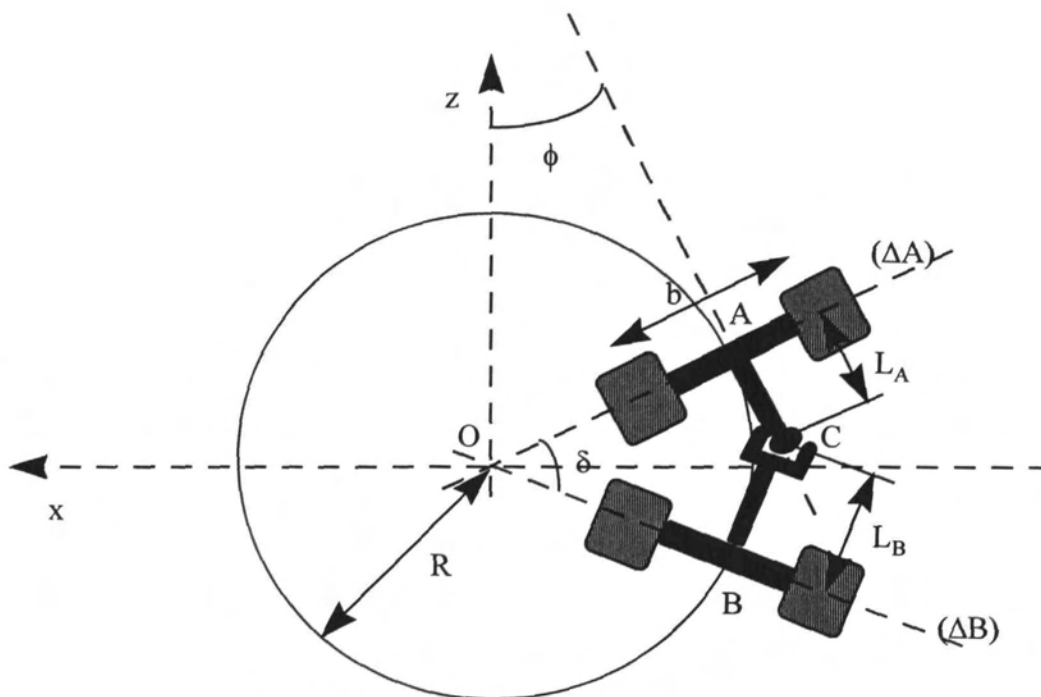


***Figure V 2 : instabilité du véhicule articulé dans certaines configurations***

Les risques de renversement sont alors très importants et l'on déplore de nombreux accidents. Néanmoins, ces véhicules articulés sont de plus en plus répandus dans l'agriculture à cause de leur grande maniabilité. Nous proposons dans la suite un modèle cinématique de cet engin ainsi que le calcul de la commande référencée vision proportionnelle qui lui est associée dans le but d'adapter notre système d'aide à la conduite à son fonctionnement.

## 1.1 Modèle cinématique

Afin de contrôler un véhicule articulé, il faut déterminer une relation entre ses actionneurs et son déplacement. Pour cela, nous utiliserons la méthode décrite au chapitre II consistant à déterminer la matrice « A » reliant le contrôle de l'engin aux mouvements d'un de ses essieux rigides. Nous prendrons comme essieu principal celui placé à l'avant du véhicule (essieu A). Puis nous déterminerons une relation entre la position et l'orientation de l'essieu A et les paramètres géométriques du véhicule.



**Figure V 3 : cinématique du véhicule articulé**

Pour une vitesse moyenne ( $V_{moy}$ ) et un angle de direction ( $\delta$ ) constants, le centre de l'essieu A se déplace sur un cercle de rayon  $R$  et de centre  $O$ . Tous les points du véhicule ont une trajectoire portée par un cercle de centre  $O$  et en particulier l'essieu arrière B. Le point  $O$  est donc donné par l'intersection des deux droites  $(\Delta A)$  et  $(\Delta B)$  définies par les deux essieux rigides A et B. Le point C symbolise l'articulation du véhicule. La distance de A à C est donnée par  $L_A$  et celle de B à C par  $L_B$ . Calculons le rayon du cercle décrit par l'essieu A.

Une relation trigonométrique simple entre  $R$  et  $\delta$  est donnée par :

$$R = -\frac{L_A + d}{\tan \delta} \quad \text{avec : } d = \frac{L_B}{\cos \delta}$$



On obtient alors :

$$R = -\frac{L_A + \frac{L_B}{\cos \delta}}{\tan \delta}$$

Pour nos applications concernant l'aide à la conduite, les angles de direction sont, en général, très faibles (inférieurs à 5 degrés). On fera donc les approximations suivantes : ( $\cos \delta = 1$  et  $\sin \delta = \delta$ ).

On a donc :

$$\text{eq V 1} \quad R = -\frac{L_A + L_B}{\delta}$$

Au point A on a toujours :

$$\text{eq V 2} \quad \dot{\phi} = \frac{V(A)}{R}$$

En considérant  $q_1$  et  $q_2$  les angles de rotation des roues gauche et droite de l'essieu A et  $r$  leur rayon on a :

$$\text{eq V 3} \quad V(A) = \frac{r}{2}(\dot{q}_1 + \dot{q}_2)$$

Au chapitre II, nous avons retenu pour un essieu fixe les équations cinématiques suivantes :

$$\text{eq V 4} \quad \begin{cases} \dot{z} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \cos \phi \\ \dot{x} = \frac{r}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \sin \phi \\ \dot{\phi} = \frac{r \cdot (\dot{q}_2 - \dot{q}_1)}{b} \end{cases}$$

$x$  et  $z$  représentent la position du centre de l'essieu A et  $\phi$  son orientation.

De (eq V 1) (eq V 2) (eq V 3) (eq V 4) nous déduisons un modèle cinématique de l'engin articulé donné sous forme matricielle par :

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \frac{r}{2} \cdot \sin \phi & \frac{r}{2} \cdot \sin \phi \\ \frac{r}{2} \cdot \cos \phi & \frac{r}{2} \cdot \cos \phi \\ -\frac{r}{b} & \frac{r}{b} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{r} & \frac{b}{2r(L_A + L_B)} \\ \frac{1}{r} & \frac{-b}{2r(L_A + L_B)} \end{pmatrix} \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

avec :

$$\text{eq V 5} \quad \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & \frac{b}{2r(L_A + L_B)} \\ \frac{1}{r} & \frac{-b}{2r(L_A + L_B)} \end{pmatrix} \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix} = A \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

On a donc déterminé un modèle cinématique approché du véhicule articulé par l'intermédiaire du calcul de sa matrice Jacobienne « A ». Cette matrice nous permettra de lui appliquer nos lois de commandes développées pour l'essieu rigide au chapitre III.

## 1.2 Application de la CRV à une plate-forme articulée

Dans un avenir proche, nous souhaitons porter l'application « aide à la conduite » sur un engin articulé de petite taille. L'algorithme de perception sera inchangé mais les différentes lois de commande devront tenir compte de la cinématique de ce type d'engin. Nous ne le ferons pas mais il serait souhaitable qu'elles garantissent également tout risque de renversement. Dans ce paragraphe, nous présenterons une application de la commande référencée vision proportionnelle à un engin de type articulé.

Rappelons les équations de commande obtenues pour l'essieu rigide au chapitre III :

$$\text{eq V 6} \quad \begin{cases} \dot{q}_1 = \frac{2v_{\text{moy}} - b \cdot \dot{\phi}}{2r} \\ \dot{q}_2 = \frac{2v_{\text{moy}} + b \cdot \dot{\phi}}{2r} \end{cases}$$

avec :

- b, largeur de l'essieu,

- $v_{\text{moy}}$ , vitesse du centre de l'essieu,
- $r$ , rayon des roues,
- $\dot{\phi}$ , vitesse de rotation de l'essieu,
- $q_1$  et  $q_2$ , angle de rotation des roues gauche et droite.

La commande référencée vision donne une équation liant l'orientation  $\phi$  de l'essieu aux informations visuelles issues de la caméra. Dans le cas où la primitive détectée est une droite représentée par ses coordonnées polaires ( $\rho$  et  $\theta$ ), l'orientation de la caméra, donc du véhicule, est donnée par :

$$\text{eq V 7} \quad \dot{\phi} = \Omega_y = \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right]$$

Les équations (eq V 6) et (eq V 7) donnent une loi de commande pour l'essieu rigide du véhicule articulé. Pour exprimer cette loi en fonction des actionneurs de la machine nous utilisons la matrice A calculée au paragraphe précédent.

D'après l'équation (eq V 5) on a :

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = A \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & \frac{b}{2r(L_A + L_B)} \\ \frac{1}{r} & \frac{-b}{2r(L_A + L_B)} \end{pmatrix} \cdot \begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix}$$

en inversant A :

$$\begin{pmatrix} v \\ v \cdot \tan \delta \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r \cdot (L_A + L_B)}{b} & -\frac{r \cdot (L_A + L_B)}{b} \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

En considérant les actionneurs ( $v$  et  $\delta$ ) du véhicule articulé, nous obtenons l'équation de commande suivante :

$$\text{eq V 8} \quad \begin{cases} v = v_{\text{moy}} \\ \delta = -\frac{(L_A + L_B)}{v_{\text{moy}}} \cdot \lambda \cdot \left[ \left( \frac{\cos \alpha \cdot \sin \alpha}{1 + h^2} \right) \cdot \beta \cdot \theta + \left( \frac{\cos^2 \alpha + h^2}{1 + h^2} \right) \cdot \rho \right] \end{cases}$$

L'équation (eq V 8) donne une loi de commande pour les actionneurs du véhicule articulé en fonction des coordonnées polaires d'une droite détectée dans l'image. L'hypothèse ( $\cos \delta = 1$ ) la rend indépendante de l'emplacement de l'articulation, ce qui représente une approximation importante. De ce fait, la loi de

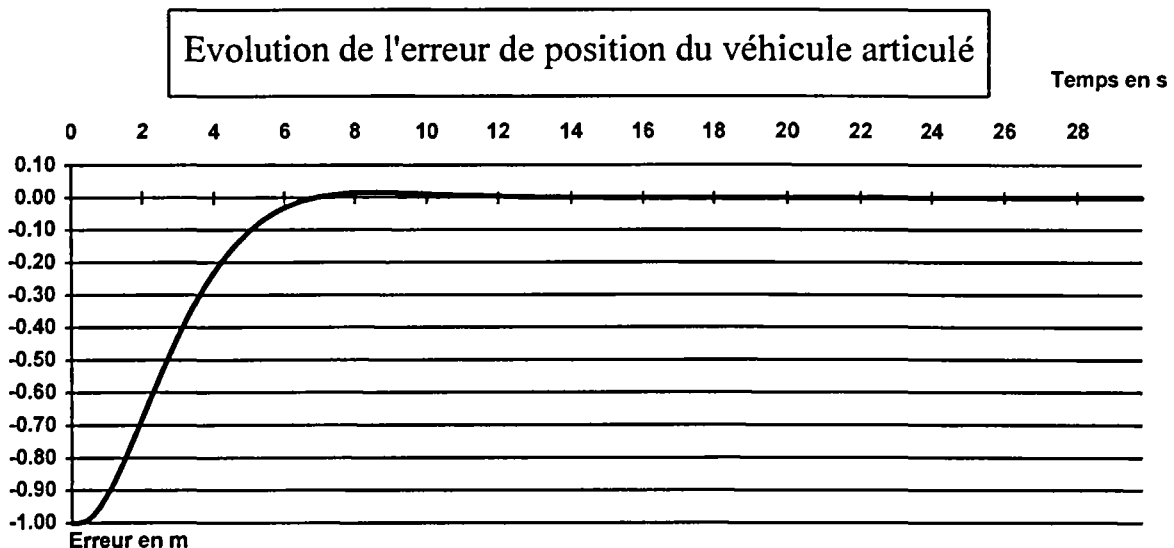
commande est exactement la même que pour la moissonneuse-batteuse, ce qui est logique puisque les équations cinématiques sont les mêmes.

### 1.3 Simulation

Nous allons maintenant simuler la réponse à un échelon d'un mètre d'un engin articulé utilisant un capteur caméra et une loi de commande référencée vision proportionnelle. Cette simulation représente l'asservissement de la machine par rapport à une droite avec une erreur initiale de position d'un mètre. L'engin utilisé a les dimensions suivantes :

- $L_A = L_B = 0.75$  m
- $b = 1$  m
- $r = 0.5$  m

Ci-dessous, nous représentons une réponse indicielle de ce véhicule :



***Courbe V 1 : réponse indicielle du véhicule articulé pour une vitesse de 4 km/h***

Sur la courbe (***Courbe V 1***) on constate un asservissement de la machine avec un temps de réponse de l'ordre de 6 s et un dépassement quasiment nul. La courbe ne présente pas d'oscillation. Bien que les équations de commande soient identiques la courbe est différente de celle de la moissonneuse-batteuse. Ceci est normal car les dimensions des deux véhicules ne sont pas les mêmes ainsi que la valeur de saturation des actionneurs. De plus, le point D (point à asservir) a été

pris sur la roue avant gauche du véhicule articulé, ce qui n'est pas le cas pour la moissonneuse-batteuse (cf. chapitre IV).

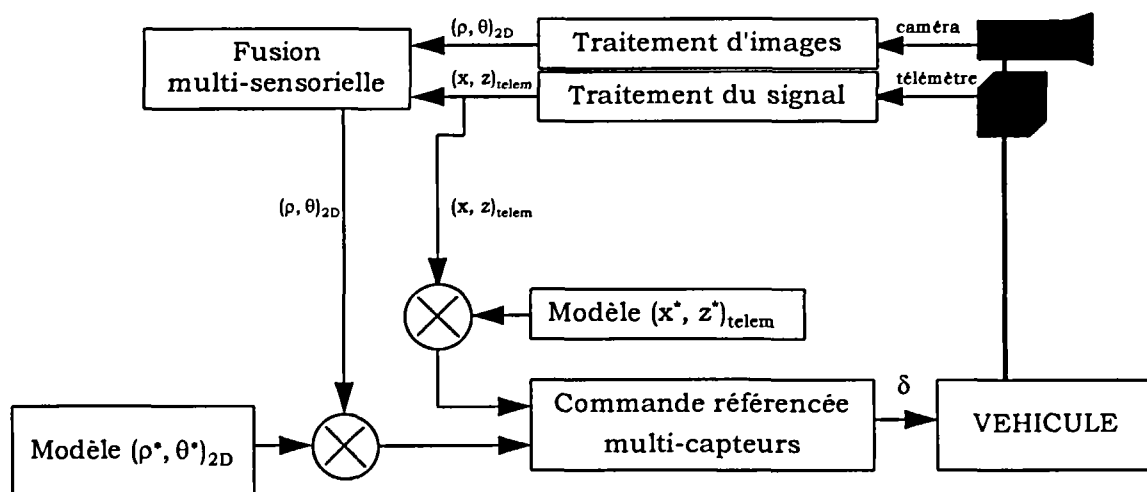
### **Conclusion**

La modélisation que nous avons retenue pour nos véhicules mobiles nous permet d'appliquer rapidement nos lois de commande à de nombreux engins agricoles ou non. Elle suppose, cependant, que le véhicule mobile possède un essieu rigide réel ou que son comportement soit assimilable à celui d'un essieu rigide virtuel. Par exemple, de nombreux engins à 4 roues directrices ne répondent pas à cette condition et nous ne pourrions pas leur appliquer directement nos lois de commande.

## 2. COMMANDE MULTI-CAPTEURS

La plupart des systèmes robotiques utilisent plusieurs capteurs de nature différentes. Pour beaucoup d'applications [Jur94] [Deb94] des capteurs extéroceptifs (caméra, capteur ultrason, télémètre laser) et des capteurs proprioceptifs (odomètres, gyromètres, accéléromètres) sont associés pour combiner des informations sur le déplacement du robot et sur son action dans son environnement. En dehors de la stéréovision, on rencontre assez peu de systèmes utilisant en même temps et pour les mêmes fonctions plusieurs capteurs extéroceptifs [Far93]. Cependant, les objectifs d'autonomie, particulièrement en robotique mobile, ont mis en évidence la nécessité d'utiliser des moyens de perception différents (vision, télémètre laser, radar etc.). Il s'agit alors de fusionner les différentes informations issues de tous les capteurs afin d'améliorer la précision et la qualité des données par l'utilisation de la redondance d'informations dans le temps ou dans l'espace.

On peut également proposer un second type de fusion opérant non pas sur les données perçues mais sur les variables de commande issues de ces données. Cette opération nécessite une étape préalable de calibrage intercapteur. Pour cela, on pourra, par exemple, utiliser la notion de grille d'occupation présentée par Moravec dans [Mor88]. Une fois ce calibrage réalisé, il faut chercher un moyen de combiner les différentes valeurs de commande calculées à partir des informations issues des différents capteurs. Pour cela, nous proposons une direction de recherche utilisant le formalisme de la CRC [Sam91b] que nous avons appliqué à la vision au chapitre III et que nous emploierons pour des données issues d'un télémètre laser. Les deux méthodes de fusion ne sont pas antagonistes et peuvent être intégrées d'une manière complémentaire. Donnons un exemple de schéma du système ainsi considéré :

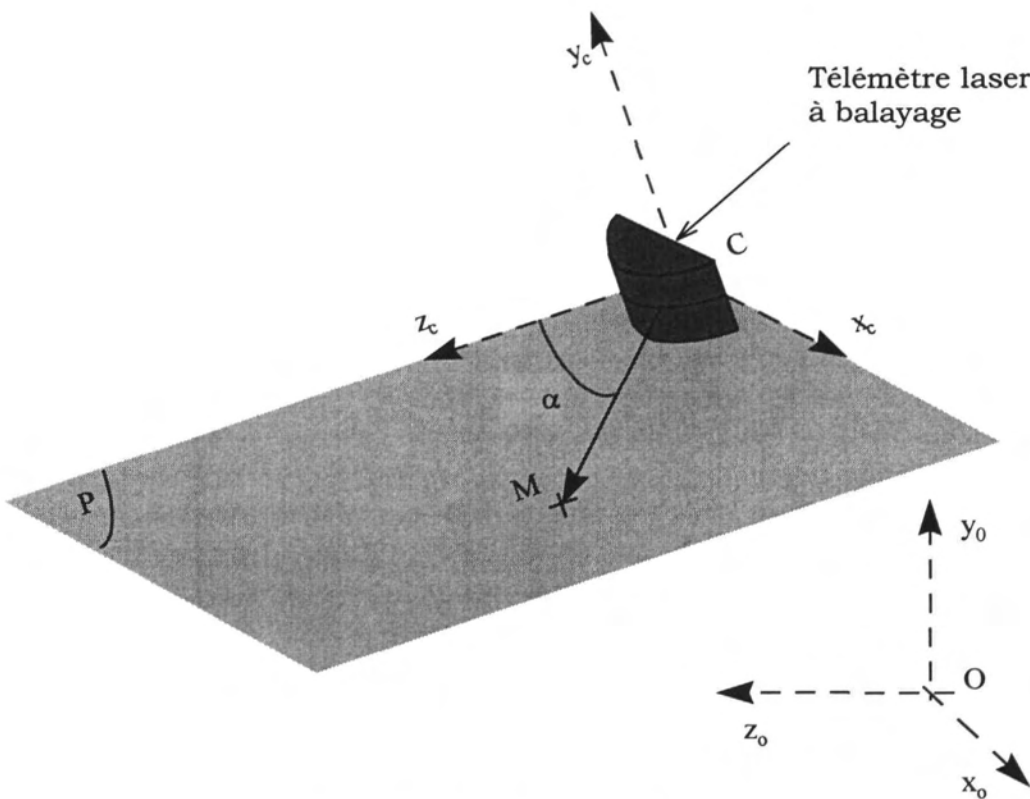


**Figure V 4 : schéma d'un système perception-commande multi-capteurs**

## 2.1 Torseur d'interaction pour un télémètre laser

Le torseur d'interaction permet de lier les vitesses d'une primitive de la scène (point, droite, cercle, sphère, etc...) au déplacement du capteur dans l'espace. Au chapitre III nous avons donné un exemple du calcul de ce torseur pour un capteur caméra et pour des primitives de type point et droite.

Donnons un second exemple du calcul de ce torseur pour un télémètre laser à balayage et pour une primitive de type point.



**Figure V 5 : Modèle du télémètre laser**

Soit une scène dans un repère cartésien  $(O, x_0, y_0, z_0)$ . Soit un télémètre laser à balayage dont le faisceau est centré en  $C$  et appartient au plan  $P$ . Les points balayés par ce capteur appartiennent donc tous au plan  $P$  ( $C, x_c, z_c$ ). Soit  $M$  un point de  $P$  de coordonnées  $(x_{R0}, y_{R0}, z_{R0})$  dans  $(O, x_0, y_0, z_0)$  et  $(x_{Rc}, 0, z_{Rc})$  dans  $(C, x_c, y_c, z_c)$ .

Soit la relation cinématique liant la vitesse de  $M$  dans  $(C, x_c, y_c, z_c)$  à ses vitesses exprimées dans  $(O, x_0, y_0, z_0)$  :

$$V_{Rc}(M) = -V_{R0}(C) - \Omega_{Rc/R0} \wedge \vec{CM}$$

On obtient alors :

$$\begin{pmatrix} \dot{x}_{R_c} \\ \dot{y}_{R_c} \\ \dot{z}_{R_c} \end{pmatrix} = \begin{pmatrix} -V_{R_0}(C)_x - z_{R_c} \cdot \Omega_{R_c/R_0y} + y_{R_c} \cdot \Omega_{R_c/R_0z} \\ -V_{R_0}(C)_y - x_{R_c} \cdot \Omega_{R_c/R_0z} + z_{R_c} \cdot \Omega_{R_c/R_0x} \\ -V_{R_0}(C)_z - y_{R_c} \cdot \Omega_{R_c/R_0x} + x_{R_c} \cdot \Omega_{R_c/R_0y} \end{pmatrix}$$

En considérant que le point M appartient au plan P on a :

eq V 9

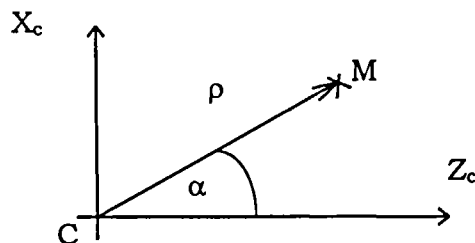
$$\begin{pmatrix} \dot{x}_{R_c} \\ \dot{z}_{R_c} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 & -z_{R_c} & 0 \\ 0 & 0 & -1 & 0 & x_{R_c} & 0 \end{pmatrix} \cdot \begin{pmatrix} V_{R_0}(C)_x \\ V_{R_0}(C)_y \\ V_{R_0}(C)_z \\ \Omega_{R_c/R_0x} \\ \Omega_{R_c/R_0y} \\ \Omega_{R_c/R_0z} \end{pmatrix} = L_{telem}^T \cdot T$$

- T représente les vitesses de déplacement et de rotation du capteur (ici le télémètre laser),

-  $\begin{pmatrix} \dot{x}_{R_c} \\ \dot{z}_{R_c} \end{pmatrix}$  est la vitesse du point M exprimée dans (C,  $x_c$ ,  $y_c$ ,  $z_c$ ),

-  $L_{telem}^T$  est le torseur d'interaction recherché.

On a donc exprimé une relation entre le déplacement du télémètre laser dans la scène et la vitesse d'un point M dans le repère capteur. Cette relation est beaucoup plus simple que pour le capteur caméra car elle n'inclut pas de projection perspective. Cependant, le capteur télémétrique ne donne pas les coordonnées du point M dans le repère cartésien (C,  $x_c$ ,  $y_c$ ,  $z_c$ ) mais dans un repère polaire lié au point C et au plan P (**Figure V 6**).



**Figure V 6 : repère polaire du télémètre laser**



Exprimons la matrice d'interaction à partir des coordonnées polaire  $(\rho, \alpha)$  de M. Les formules de changement de repère sont :

$$\text{eq V 10} \quad \begin{cases} x = \rho \cdot \sin \alpha \\ z = \rho \cdot \cos \alpha \end{cases}$$

Dérivons (eq V 10) :

$$\text{eq V 11} \quad \begin{cases} \dot{x} = \dot{\rho} \cdot \sin \alpha + \dot{\alpha} \cdot \rho \cdot \cos \alpha \\ \dot{z} = \dot{\rho} \cdot \cos \alpha - \dot{\alpha} \cdot \rho \cdot \sin \alpha \end{cases}$$

Inversons le système donné par (eq V 11) :

$$\text{eq V 12} \quad \begin{cases} \dot{\alpha} = \frac{\dot{x} \cdot \cos \alpha - \dot{z} \cdot \sin \alpha}{\rho} \\ \dot{\rho} = \dot{x} \cdot \sin \alpha + \dot{z} \cdot \cos \alpha \end{cases}$$

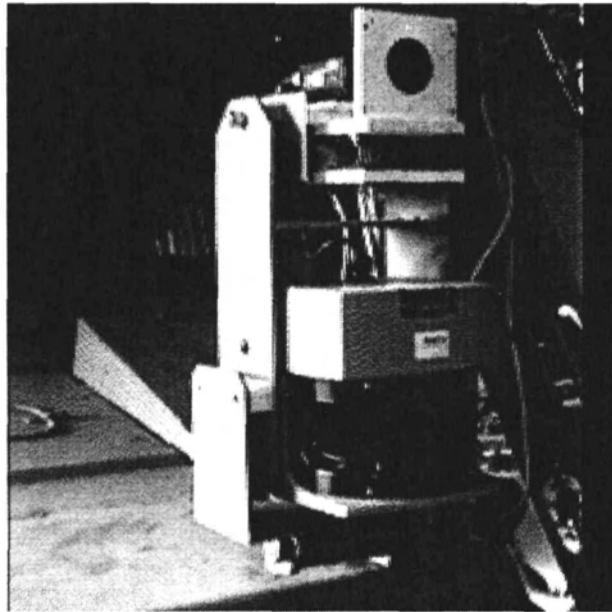
En reportant (eq V 9) et (eq V 10) dans (eq V 12) nous obtenons :

$$\text{eq V 13} \quad \begin{pmatrix} \dot{\alpha} \\ \dot{\rho} \end{pmatrix} = L_{\text{telem}}^T \cdot T = \begin{pmatrix} -\cos \alpha & 0 & \frac{\sin \alpha}{\rho} & 0 & -1 & 0 \\ \sin \alpha & 0 & -\cos \alpha & 0 & 0 & 0 \end{pmatrix} \cdot T$$

L'équation (eq V 13) exprime une liaison virtuelle entre les mouvements du capteur télémétrique et les vitesses d'un point détecté par ce capteur. Contrairement à l'équation (eq V 9) il n'est pas nécessaire de reconstruire les coordonnées cartésiennes du point détecté. Pour commander nos véhicules, nous utiliserons une loi de commande référencée capteur proportionnelle au moyen de la pseudo-inverse de  $L_{\text{telem}}^T$ .

## 2.2 Couplage caméra télémètre laser

Afin d'augmenter la robustesse des informations perçues par la caméra, nous étudions actuellement son couplage avec un télémètre laser.



*Photo V 2 : couplage caméra télémètre laser*

Aujourd'hui, un processus de fusion des informations visuelles et télémétriques est envisagé au niveau du module de perception afin d'accroître la qualité de la position de la droite détectée dans l'image, mais nous pensons également qu'il pourrait être intéressant de fusionner les données sur l'environnement au niveau du module de commande. Dans la suite, nous détaillons le principe de cette dernière fusion.

Pour une droite de l'image, nous avons :

$$\text{eq V 14} \quad \begin{pmatrix} \dot{\theta}_{\text{image}} \\ \dot{\rho}_{\text{image}} \end{pmatrix} = L_{\text{image}}^T \cdot T$$

- $T$  représente les vitesses de déplacement et de rotation du capteur (ici le couple caméra télémètre laser),
- $L_{\text{image}}^T$  est la matrice d'interaction associé au vecteur vitesse de la droite dans l'image  $(\dot{\theta}, \dot{\rho})^t$  et aux mouvements du capteur  $T$ .

L'équation (eq V 14) donne une relation entre T et la dérivée des coordonnées d'une droite de la scène projeté dans l'image. L'équation (eq V 13) donne une relation entre T et les vitesse d'un point M du plan P. Donnons un nouveau torseur  $L^T$  combinant les deux relations en nous inspirant de la méthode donnée par Martinet dans [Mar96]. La relation obtenue est de la forme :

$$\begin{pmatrix} \dot{\theta}_{\text{image}} \\ \dot{\rho}_{\text{image}} \\ \dot{\alpha}_{\text{telem}} \\ \dot{\rho}_{\text{telem}} \end{pmatrix} = L_{\text{couple}}^T \cdot T \quad \text{avec : } L_{\text{couple}}^T = \begin{pmatrix} L_{\text{image}}^T \\ L_{\text{telem}}^T \end{pmatrix}$$

On a donc relié nos informations capteur aux mouvements du couple caméra télémètre laser. A cet instant, on conçoit bien l'importance de l'étape de calibrage intercapteur puisque les mouvements des capteurs donnés par le torseur cinématique T résulteront de la somme des variations des informations capteurs. Il serait également souhaitable que les deux capteurs soient synchronisés afin que toutes les données aient la même origine des temps.

Remarque

La méthode proposée est plutôt une combinaison linéaire des informations qu'une réelle fusion. En effet, bien que l'on puisse privilégier une information par rapport à une autre en introduisant des gains sur les différents torseurs, on ne peut pas prendre de réelle décision sans ajouter une étape exprimant la fiabilité des diverses informations.

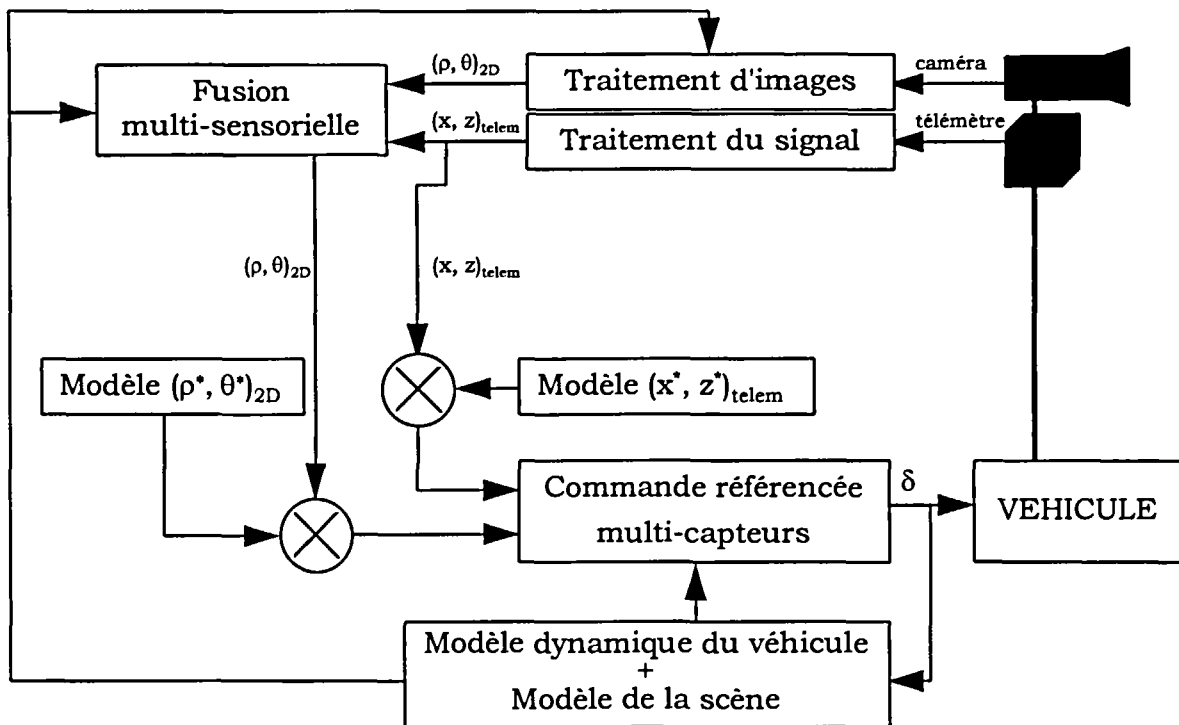
Conclusion

Le guidage automatique des robots mobiles nécessite une grande qualité de la perception de l'environnement. Aussi, l'utilisation de plusieurs capteurs extéroceptifs de nature différente paraît presque incontournable, et ceci d'autant plus que le robot évolue en milieu extérieur. Il se pose alors le problème de l'analyse et de la comparaison des données issues des différents capteurs. S'il est possible de fusionner ces données pour en extraire une information sur l'environnement, il serait peut-être tout aussi intéressant d'élaborer des lois de commande intégrant des informations multisensorielles. Le formalisme de la CRC [Sam91b] nous paraît bien adapté à cette démarche grâce à la matrice d'interaction et à la matrice de combinaison qui permettent d'intégrer un nombre variable d'informations. Cependant, quelle que soit la technique utilisée, il apparaît qu'un calibrage soigné de chaque capteur et inter-capteur est nécessaire (mise en correspondance temporelle et spatiale).

### 3. INTERACTION VISION-COMMANDE

S'il est clair que la précision des algorithmes de vision conditionne la qualité des lois de commande calculées, la réciproque n'apparaît pas directement. Pourtant, lors de nos essais, nous avons constaté qu'une loi de commande trop « brusque » pénalisait les résultats de l'algorithme Markovien en induisant des changements trop importants dans l'image. Ceci est dû au fait qu'il utilise le résultat de l'image  $n$  comme hypothèse dans le traitement de l'image  $n+1$ . De plus, la loi de commande doit tenir compte du champ de perception du capteur afin d'éviter que les informations recherchées deviennent inaccessibles. Enfin, un modèle de la scène devrait permettre d'introduire des hypothèses sur le résultat de la perception en fonction du déplacement du robot.

Pour la suite de nos travaux, nous étudions une première approche intégrant un échange de données et une interaction des modules de commande et de perception (*Figure V 7*). Nous n'avons pas encore envisagé de réaliser un « noyau fédérateur » mais plutôt des règles simples permettant à chaque module d'intégrer sous forme d'hypothèses les informations issues des autres éléments du système.

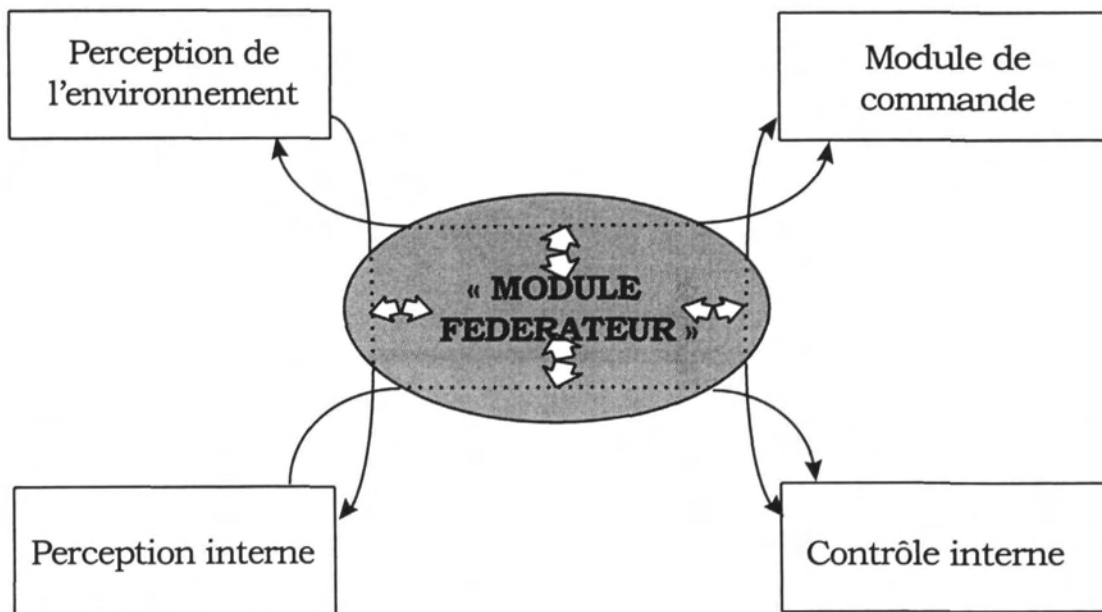


*Figure V 7 : un exemple d'interaction perception-commande*

Toute la difficulté d'un tel système est de déterminer la qualité des informations issues de chaque module afin de calculer une probabilité de chaque hypothèse. La prédiction des informations perçues à partir des hypothèses provenant du modèle dynamique du véhicule pourra être réalisée par un filtre de Kalman donnant un poids aux informations prédites par rapport à celles mesurées.

### Conclusion

Je pense que la mobilité des robots ne doit pas être étudiée en plusieurs thèmes de recherche bien distincts qui ont tendance à oublier l'intégrité des systèmes que l'on cherche à développer. Au contraire, il serait souhaitable de développer un niveau d'intelligence intégrant les différents éléments constituant les fonctions mises en jeu ainsi qu'un modèle, même très partiel, de l'environnement du robot. Cela devrait permettre à l'ensemble des modules des systèmes envisagés d'échanger des informations et des hypothèses sur la sortie et l'entrée de chaque élément. Actuellement, les applications développées au Cemagref permettent une circulation des informations dans un seul sens (perception vers commande), je pense qu'il serait intéressant de développer un « superviseur (**Figure V 8**) » capable de fusionner les données propres ou extérieures au robot et d'en assurer la cohérence.



**Figure V 8 : schéma d'un superviseur « perception action »**

L'une des difficultés d'une telle approche est de réunir les compétences présentes dans les différents thèmes scientifiques et de les étudier non plus en elles-mêmes mais par rapport à l'application visée [Coi93].

---

## CONCLUSION

Dans cet ouvrage, nous avons essayé de montrer comment nous pouvions utiliser la vision pour commander un robot mobile agricole évoluant à vitesse lente ; pour cela nous avons étudié et implanté deux approches très différentes.

La première est basée sur une reconstruction de la scène à partir de la méthode de Faugeras et Toscani. De ce fait nous obtenons la position 3D de la machine par rapport à celle de l'interface détectée dans l'image. Puis une trajectoire formée de deux arcs de cercles tangents est décrite par le véhicule afin de compenser les erreurs de position et d'orientation. Les essais en simulation, ainsi qu'en conditions réelles attestent de la convergence de cette loi de commande avec un dépassement quasiment nul et un temps de réponse compatible avec nos applications. Cependant les hypothèses retenues (sol plan et points détectés au niveau du sol) ne sont pas toujours vérifiées. En effet nos véhicules évoluent sur des terrains accidentés, ce qui introduit inévitablement des erreurs dans le calcul de la position 3D de l'interface fauchée/non fauchée. De plus pour certaines positions de la caméra, la ligne détectée ne correspond plus au pied de la végétation mais à une hauteur variable rendant notre hypothèse utopique. Enfin nous observons lors d'essais en simulation qu'en bruitant les mesures visuelles, l'orientation du véhicule subissait des variations importantes perturbant ainsi le fonctionnement de l'algorithme de traitement d'images ou provoquant la disparition de l'interface du champ de vision de la caméra.

Afin d'éviter l'étape de reconstruction, nous avons étudié et implanté une deuxième loi de commande dont la théorie est inspirée du formalisme de la commande référencée capteur développée à L'INRIA dans la fin des années 80. Cette loi de commande appelée Commande Référencée Vision permet d'asservir le robot dans l'espace image au moyen d'une primitive visuelle choisie de manière à caractériser la liaison du véhicule avec la trajectoire à suivre. Cette loi n'est applicable que dans le cas d'un déplacement du véhicule à vitesse moyenne constante et non nulle. En effet, nous ne prétendons pas asservir un véhicule non-holonyme en position et orientation à vitesse finale nulle. Les problèmes provoqués par la non holonomie n'ont pas été abordés

mais contournés par la modélisation d'un déplacement latéral. Celui-ci est décomposé en un changement d'orientation conjugué à un déplacement suivant l'axe perpendiculaire à l'essieu rigide du robot. Cela suppose que la vitesse de l'essieu ne s'annule jamais, ce qui est toujours le cas puisqu'elle est fixée par l'opérateur. En effet la plupart des véhicules agricoles travaillent à vitesse moyenne constante, la condition précédente n'est donc pas un obstacle aux lois de commande que nous venons de présenter.

Nous avons ensuite analysé le déplacement d'un véhicule agricole sur un terrain en pente. Il est alors clairement apparu que la modélisation utilisant une droite de référence n'était pas suffisante. Le cahier des charges du système d'aide à la conduite préconisant uniquement l'utilisation du capteur caméra, nous avons cherché à **modéliser la pente à partir des informations données par l'image**. Grâce à l'orientation de la droite détectée nous avons **construit un ensemble de points de fonctionnement, permettant de recalculer la droite de référence à chaque instant**. En ajoutant un module intégrateur, nous avons établi qu'il était alors possible de contrôler efficacement nos machines sur un terrain glissant comportant une pente transversale. Des expérimentations, en conditions réelles ont également montré que cette nouvelle loi de commande rendait la machine insensible aux réglages du correcteur bas niveau. En effet un offset sur l'angle des roues n'a aucune incidence sur la trajectoire du véhicule, ce qui n'était pas le cas avec la loi de commande référencée vision proportionnelle.

Dans des conditions réelles d'utilisation, par exemple pour la récolte des céréales, nos algorithmes de perception sont encore perfectibles. De ce fait, nous nous orientons aujourd'hui, vers une solution multi-algorithmique, et/ou multi-capteurs. A ce titre, nous développons actuellement un module de fusion permettant d'améliorer la robustesse de la détection de la droite modélisant l'interface fauchée/non fauchée. Nous espérons également intégrer au sein de l'algorithme de commande les différents éléments perçus par le biais d'un « torseur d'interaction multi-capteurs ». Celui-ci effectuera une combinaison linéaire des différentes informations et donnera, après normalisation, une loi de commande pour nos engins rendant plus robuste le lien entre les primitives choisies et l'interface végétale.

Depuis le début de ce projet, ses instigateurs : Michel Berducat, Pierre Bonton, Jean Gallice, Philippe Zwaenepoel, ont considéré les différents thèmes scientifiques nécessaires au guidage d'un robot mobile

dans la perspective de réaliser un prototype. Ceci a conditionné divers choix qui donnent une première réponse satisfaisante au problème d'aide à la conduite de machines agricoles.

Je pense que cette démarche doit être poursuivie et approfondie en utilisant au mieux les résultats de l'ensemble des éléments du système. Par exemple, les hypothèses de traitement de chaque module pourraient être enrichies des résultats émis par ses modules voisins. Il serait possible, dans un premier temps d'éviter les incohérences entre les informations, et de concevoir ensuite un noyau fédérateur. Ce dernier serait susceptible d'apprendre l'enchaînement et la logique des réactions de toutes les parties du système.



---

## BIBLIOGRAPHIE

**[Zad65]**

L.A. Zadeh : *Fuzzy sets*, Information and control, pp. 338-353, 1965.

**[Jul71]**

A. P. Julian : *Design and Performance of a steering Control System for Agricultural Tractors*, J. agric. Engng Res. 16 (3), pp.324-336, 1971.

**[Har73]**

R.M. Haralick, K. Shanmugam, I. Dinstein : *Textural features for image classification*, IEEE Transactions on Systems, Man and Cybernetics, vol. 3, n°6, pp. 610-621, 1973.

**[Chen79]**

P.C. Chen, T. Pavlidis : *Segmentation by texture using a cooccurrence matrix and a split and merge algorithm*, Computer Graphics and Image Processing, vol. 10, pp. 172-182, 1979.

**[Har79]**

R.M. Haralick, may *Statistical and structural approaches to texture*. Proceeding of the IEEE, vol. 67, n°5, pp. 786-804, 1979.

**[Tri84]**

M.M. Trivedi, C.A. Harlow, R.W. Connors : *Object detection based on gray level cooccurrence*, Computer Vision, Graphics, and Image Processing, vol. 28, n°2, pp. 199-219, 1984.

**[Dick85]**

E. D. Dickmanns : *Guiding land vehicles along roadways by Computer Vision*, Proc. Congres Automatique 1985, AFCET, pp. 233-234, Toulouse, Octobre 1985.

**[Wax85]**

A. M. Waxman, J. Le Moigne, B. Srinivasan : *Visual navigation of roadways*, Proceeding of the IEEE International Conference on Robotics and Automation, pp. 862-867, April 1985.

**[Der87]**

H. Derin, H. Elliot : *Modeling and segmentation of noisy and textured image using Gibbs random fields*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, n°1, pp. 39-55, January 1987.

**[Dick87]**

E. D. Dickmanns, A. Zapp : *Autonomous high speed road vehicle guidance by computer Vision*, 10<sup>th</sup> IFAC World Congress, München, July 1987.

**[Esp87]**

B. Espiau, C. Samson : *Sensory based control. Robustness issues and modelling technics. Application to proximity sensing*, NATO Workshop on Kinematic and dynamic Issues in Sensor Based Control, Pisa, 1987.

**[Fau87]**

R.Y. Faugeras, G. Toscani : *Camera calibration for 3D computer vision, International workshop on machine vision and machine intelligence*, Tokyo, Japan, February 1987.

**[Sta87]**

R.Y. STAI : *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shell TV Cameras and Lenses*, IEEE journal of robotics and automation, Vol. 3, N°4, pp. 323-344, 1987.

**[Ioo88]**

E. Ios, M. Boulle, P. Tournassoud : *Etude d'un sous système de navigation pour un robot mobile*, Rapport de Recherche n°783, INRIA Rocquencourt, France, Fevrier 1988.

**[Mor88]**

H. P. Moravec : *Sensor Fusion in Certainty Grids for Mobile Robots*, AI Magazine, 61, 74, summer 1988.

**[Tou88a]**

P. Tornassoud and O. Jehl : *Motion Planning for a Mobile Robot with Kinematic Constraint*, IEEE International conference on robotics and automation, vol. 2, pp. 1785-1790, 1988.

**[Tou88b]**

P. Tournassoud : *Géométrie et intelligence artificielle pour les robots*, 312 pages, Hermès, Paris, ISBN 2-86601-143-0, 1988.

**[Turk88]**

M. Turk, D. Morenthaler, K. Gremban, M. Marra : *VITS - A Vision System for Autonomous Land Vehicle Navigation*, IEEE Transactions on pattern analysis and machine intelligence. Vol.10 N° 3, pp. 342-361, May 1988.

**[Cha89]**

F. Chaumette, P. Rives : *Réalisation et calibration d'un système expérimentale de vision composé d'une caméra mobile embarquée sur un robot manipulateur*. Rapport de recherche INRIA n°994, Rennes, 32 p., Mars 1989.

**[Che89]**

Chen, Chaur Chin, R.C. Dubes : *Experiments in fitting discrete Markov random fields to texture*. CVPR (Computer Vision and Pattern Recognition) 89, San-Diego, pp. 248-303, June 1989.

**[Riv89]**

P. Rives, F. Chaumette, B. Espiau : *Visual Servoing Based on a Task Function Approach*, First International Symposium on Experimental Robotics, Montréal, Canada, June 1989.

**[Sta89]**

R.Y. STAI : *Synopsis of Recent Progress on Camera Calibration for 3D Machine Vision*. O Khatib, J J Craig et T Lozano-Perez, éditeurs, The robotic review, pp. 147-159, 1989.

**[Chau90]**

F. Chaumette : *La commande référencée vision : une approche aux problèmes d'asservissements visuels en robotique*, Thèse d'informatique, Université de Rennes I, Juillet 1990.

**[Cox90]**

I. J.Cox, G. T. Wilfong : *Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot*, Autonomous Robot Vehicles, 462 p., ISBN 0-387-97240-4, 1990.

**[Esp90]**

B. Espiau, F. Chaumette, P. Rives : *Une nouvelle approche de la relation vision-commande en robotique*, Rapport de Recherche INRIA n°1172, Mars 1990.

**[Gen90]**

D. Genam, S. Genam, C. Graffigne, Ping Dong : *Boundary detection by constrained optimization*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 12, n°7, pp. 609-628, July 1990.

**[Sam90]**

C. Samson, K. Ait-Abderrahim : *Mobile robot control, part 1 : Feedback control of a nonholonomic wheeled cart in cartesian space*, Rapport de Recherche INRIA n°1288, Décembre 1991.

**[Esp90]**

B. Espiau, F. Chaumette, P. Rives : *Une nouvelle approche de la relation vision-commande en robotique*, Rapport de Recherche INRIA n°1172, Mars 1990.

**[Tan90]**

K. Taniwaki, H. Kamiyama, T. Soga : *Automated steering control of robotic mower*, Proceeding of IEEE International Conference on Intelligence Robots AND Systems, vol. 1, pp. 280-286, IEEE-NJ 08854 USA, 1990.

**[Sul90]**

J.R. Sullins : *Distributed learning of texture classification*, Computer Vision, ECCV 90, Antibes, pp. 349-358, 1990.

**[Chap91]**

R. Chapuis : *Suivi de primitive image, application à la conduite automatique sur route*, thèse d'électronique, Université Blaise Pascal de Clermont-Ferrand, Janvier 1991.

**[Chau91]**

F. Chaumette, P. Rives, B. Espiau : *Positioning of a Robot with respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing*, proceeding of the IEEE, International Conference on Robotics and Automation, pp. 2248-2252, Sacramento, California, April 1991.

**[Keh91]**

N. Kehtarnavaz, J. S. Lee : *Visual Control of an Autonomous Vehicule (BART) -The Vehicle-Following Problem*, IEEE Transaction on vehicular technology, vol. 40, n°3, August 1991.

**[Pis91]**

R. Pissard-Gibollet, P. Rives : *Asservissement visuel appliqué à un robot mobile : état de l'art et modélisation cinématique*, Rapport de Recherche INRIA n°1577, Décembre 1991.

**[Sam91a]**

C. Samson, K. Ait-Abderrahim : *Feedback Control of a Nonholonomic Wheeled Cart in cartesian Space*, proceedings of the 1991 IEEE International Conference on Robotics and Automation, pp. 1136-1141, Sacramento, California, April 1991.

**[Sam91b]**

C. Samson, B. Espiau, M. Le Borgne : *Robot Control : the Task Function Approach*, Oxford Science Publications, ISBN 0-19-8538057, 1991.

**[Tho91]**

R. H. Thomas : *Sensory Feed-back Requirements in Battlefield Teleoperation*, NATO DRG Seminar on battlefield robotics, Paris 1991.

**[Bot92]**

C. Bois : *Recherche d'une architecture dédiée permettant le guidage automatique par traitement d'images d'un engin d'entretien d'espaces naturels*, mémoire en vue de l'obtention du titre d'ingénieur, filière génie électrique, CUST Clermont-Ferrand, Juin 1992.

**[Bou92]**

G. Bourdon : *Application d'un contrôleur flou à la commande bas niveau d'un robot mobile*, rapport de DEA, Laboratoire de Robotique de Paris, Juillet 1992.

**[Cha92]**

R. Chatila, R. Alami, R. Prajoux : *An architecture integrating task planning and reactive execution control*, Workshop On Architecture For Intelligent Control Systems, IEEE International Conference on Robotics and Automation, Nice, France, Mai 1992.

**[Esp92]**

B. Espiau, F. Chaumette, P. Rives : *A new approach to Visual Servoing in Robotics*, in Trans. on Robotics and Automation, vol. 8, n. 3, 1992.

**[Jur92]**

F. Jurie, P. Rives, J. Gallice, J. L. Brame : *A vision based control approach to high speed automatic vehicle guidance*, in MVA 92, IAPR Workshop on Machine Vision Application, pp. 329-333, December 7-9 1992.

**[Mec92]**

D. Mechineau : *La problématique de la conduite autonome*, colloque sur la conduite autonome des engins agricoles, Paris, pp. 19-35, Mars 1992.

**[Miy92]**

H. Miyata, M. Ohkita : *Control of an Autonomous Mobile Robot*, Robotics and flexible Manufacturing Systems, J.C. Gentina and S.G. Tzafestas (editors), Elsevier Science Publisher B.V. (North-Holland), pp. 151-160, IMACS 1992.

**[Mot92]**

G. Motyl : *Couplage d'une caméra et d'un faisceau laser en commande référencée vision*, Phd Thesis, Clermont-Ferrand, France, September 1992.

**[Ait93]**

K. Ait-Abderrahim : *Commande de robots mobiles*, thèse de Robotique-Automatique, Ecole des Mines de Paris, Janvier 1993.

**[Bet93]**

A. Betourne, A. Fournier : *Kinematics and Control of a Conventional Wheeled Omnidirectional Mobile Robot*, 24<sup>th</sup> International Symposium on Industrial Robots, Tokyo, Japan, November 4-6 1993.

**[Coi93]**

P. Coiffet : *Robot habilis robot sapiens, histoire, développements et futurs de la robotique*, Edition Hermès, 425 p. , ISBN 2-86601-381-6, 1993.

**[Far93]**

C. Fargeon, J. P. Quin : *Robotique mobile*, DGA 1993, Teknea 1993, 358 pages, ISBN 2-87717-036-5.

**[Lee93]**

S.S. Lee and J.H. Williams : *A fast tracking error control method for an autonomous mobile robot*, Robotica, volume 11, pp. 209-215, 1993.

**[Sam93]**

C. Samson : *Commande de véhicules non-holonomes pour le suivi de trajectoire et la stabilisation à une posture désirée*, Proceeding du colloque concernant l'Automatique pour les Véhicules Terrestres, Amiens-France, 22 Septembre 1993.

**[Deb94]**

C. Debain : *Contrôle-Commande d'un engin d'entretien des espaces naturels*, rapport intermédiaire, Cemagref de Clermont-Ferrand, Décembre 1993.

**[Der94]**

M. Derras : *Segmentation non supervisée d'images texturées par champs de Markov : Application à l'automatisation de l'entretien des espaces naturels*, thèse d'électronique, Université Blaise Pascal de Clermont-Ferrand, Décembre 1994.

**[Guld94]**

J. Guldner and V.I. Utkin : *Stabilization of non-holonomic mobile robots using lyapunov functions for navigation and sliding mode control*, Control theory and advanced technology, vol. 10, No 4, part 1, pp. 635-647, December 1994.

**[Hum94]**

J. W. Hummel, K. E. Von Qualen, J. F. Reid : *Machine vision based guidance system for improved swath matching*, XII World congress on Agricultural Engineering, paper n°94-D-136, Milano, August 29 September 1 1994.

**[Jur94]**

F. Jurie, P. Rives, J. Gallice, J. L. Brame : *High-Speed Vehicle Guidance Based on Vision*, Control Eng. Practice, Vol. 2, N°2, pp. 289-297, 1994.

**[Kla94]**

N. D. Klassen, R. J. Wilson, J. N. Wilson : *Guidance systems for agricultural vehicles*, XII World congress on Agricultural Engineering, proceedings vol. 2, pp 1136-1141, Milano, August 29 September 1 1994.

**[Jur94]**

F. Jurie, P. Rives, J. Gallice, J. L. Brame : *High-Speed Vehicle Guidance Based on Vision*, Control Eng. Practice, Vol. 2, N°2, pp. 289-297, 1994.

**[Rou94]**

R. Rouveure : *Driving assistance for agricultural vehicle*, Conferences Proceedings of Neural Networks and their Applications, pp. 132-138, Marseille, Décembre 1994.

**[Vaz94]**

B. Vazquez-Gonzalez, J. H. Sossa-Azuela and J. L. Diaz-de-Léon S. : *Auto Guided Vehicle Control using Expanded Time B-Splines*, IEEE International conference on systems, Man and Cybernetics, San Antonio, Texas, October 2-5 1994.

**[Deb95a]**

C. Debain, D. Khadraoui, M. Derras, P. Martinet, P. Bonton : *Asservissement visuel d'engins agricoles par commande référencée vision*, QCAV 1995, International Conference on Quality Control by Artificial Vision, pp. 250-260, Le Creusot, France, 1995.



**[Deb95b]**

C. Debain, D. Khadraoui, M. Berducat, P. Martinet, P. Bonton : *A Visual Servoing approach to Control Agricultural Mobile Machines*, International Advanced Robotics Programme, the fourth Workshop on Robotics in Agriculture and Food-Industry, pp. 117-124, Toulouse, France, October 30-31 1995.

**[Der95]**

M. Derras, L. Verrat, M. Berducat, P. Bonton : *Image Processing and Algorithms Merging : Real time Control of an Upkeep Machine for Natural Spaces*, International Advanced Robotics Programme, the fourth Workshop on Robotics in Agriculture and Food-Industry, pp. 93-98, Toulouse, France, October 30-31, 1995.

**[Hof95]**

C. Hofner, G. Schmidt : *Path planning and guidance techniques for an autonomous mobile cleaning robot*, Robotics and Autonomous Systems 14, pp. 199-212, 1995.

**[Kha95a]**

D. Khadraoui, C. Debain, P. Martinet, M. Berducat and J. Gallice : *Vision Based Control Law for Agricultural Machines*, International Advanced Robotics Programme, the fourth Workshop on Robotics in Agriculture and Food-Industry, pp. 109-116, Toulouse, France, October 30-31 1995.

**[Kha95b]**

D. Khadraoui, P. Martinet, J. Gallice : *Linear Control of high speed Vehicle in Image Space*, 21<sup>th</sup> international conference on industrial automation, vol. 2, p. 517-522, Nancy, 7-9 June 1995.

**[Mäk95]**

H. Mäkela, H. Lehtinen, K. Rintanen, K. Koskinen : *Navigation system for LHD machines*, 2<sup>nd</sup> IFAC conference on Intelligent Autonomous Vehicles, pp. 314-319, Helsinki University of Technology, Espoo, Finland, 12-14 June 1995.

**[Mas95]**

I. Masaki : *Vision-based mobile robots on highways*, Advanced Robotics, Vol. 9. N° 4, pp. 417-427, 1995.

**[Nap95]**

M. R. Napolitano, M. Kincheloe : *On-line Learning Neural-Network Controllers for Autopilot Systems*, journal of Guidance, Control, and Dynamics, vol. 33, n°6, November-December 1995.

**[Rin95]**

K. Rintanen, H. Mäkela, K. Koskinen, J. Puputti, M. Sampo, M. Ojala : *Development of an autonomous navigation system for an outdoor vehicle*, 2<sup>nd</sup> IFAC conference on Intelligent Autonomous Vehicles, pp. 220-225, Helsinki University of Technology, Espoo, Finland, 12-14 June 1995.

**[Glr96]**

G. Giralt, L. Boissier, L. Marechal-Cotton : *The IARES project : rovers for the human conquest of the moon and mars*, Rapport LAAS N96067, 20 p., Février 1996.

**[Hut96]**

S. Hutchinson, G. Hager, P. Corke : *A Tutorial on Visual Servo Control*, 1996 IEEE International conference on robotics and automation, Mineapolis, April 22-28 1996.

**[Mar96]**

P. Martinet, J. Gallice, D. Khadraoui : *Vision based control law using 3D visual features*, WAC 96, Mai 1996.

## ANNEXE A : Calibrage de la caméra

### A.1. La méthode des moindres carrés

La méthode des moindres carrés fournit une solution à la recherche des paramètres en minimisant un critère d'optimalité.

Soient  $y_i$  différentes mesures effectuées. Soit  $y^*$  une combinaison linéaire de ces mesures pondérées par des paramètres  $a_i$  que nous cherchons à déterminer :

$$y^*(k) = a_1 \cdot y_1(k) + a_2 \cdot y_2(k) + \dots + a_i \cdot y_i(k) + \dots + a_p \cdot y_p(k)$$

$$\text{Soit } H = (y_1 \ y_2 \ \dots \ y_p)^T \quad \text{et} \quad \theta = (a_1 \ a_2 \ \dots \ a_p)^T$$

On peut écrire :

$$y(k) = H^T(k) \cdot \theta$$

$y(k)$  est une mesure entachée de bruit donc

$$y(k) = H^T(k) \cdot \theta + v(k) \quad \text{où } v(k) \text{ représente le bruit.}$$

En appliquant cette équation à  $n$  mesures, on obtient :

$$Y(k) = H^T(k) \cdot \theta + V(k)$$

$$\text{où } Y(k) = (y^{*1(k)} \ y^{*2(k)} \ \dots \ y^{*n(k)})^T$$

$$h(k) = \begin{pmatrix} y_{11(k)} & \dots & y_{1p(k)} \\ \vdots & y_{ji(k)} & \vdots \\ y_{n1(k)} & \dots & y_{np(k)} \end{pmatrix}$$

La méthode des moindres carrés fournit une solution à la recherche des paramètres en minimisant un critère d'optimalité. Ce dernier est l'erreur quadratique entre la solution désirée et la solution estimée.

$$J(\Theta) = [Y - \hat{Y}] \cdot [Y - \hat{Y}]^T = [Y - H\hat{\Theta}] \cdot [Y - H\hat{\Theta}]^T$$

où  $\hat{Y}$  désigne le vecteur de mesures estimées .

Miniser le critère J revient à écrire que sa dérivée par rapport à  $\Theta$  est nulle.

$$\frac{\partial J}{\partial \hat{\Theta}} = 0 \Leftrightarrow \frac{\partial (Y - H\hat{\Theta})^T}{\partial \hat{\Theta}} [Y - H\hat{\Theta}]^T + \frac{\partial (Y - H\hat{\Theta})^T}{\partial \hat{\Theta}} [Y - H\hat{\Theta}]^T = 0$$

$$\Leftrightarrow 2 \frac{\partial (Y - H\hat{\Theta})^T}{\partial \hat{\Theta}} [Y - H\hat{\Theta}]^T = 0$$

$$\Leftrightarrow 2 \frac{\partial (Y^T - \hat{\Theta}^T H^T)}{\partial \hat{\Theta}} [Y - H\hat{\Theta}]^T = 0$$

$$\Leftrightarrow -2H^T [Y - H\hat{\Theta}]^T = 0$$

$$\Leftrightarrow \hat{\Theta} = [H^T \cdot H]^{-1} H^T \cdot Y$$

### **A.2. Calibrage de la caméra (mesures expérimentales utilisant huit points)**

Ci-dessous nous présentons les relevés expérimentaux des coordonnées des huit points utilisés.

N° des points	Coordonnées 3D en mètre			Coordonnées 2D en pixels	
	x	y	z	u	v
1	0.2	3.31	0	81	26
2	-0.1	3.81	0	-16	79
3	0	3.51	0	13	49
4	0	2.41	0.5	10	231
5	-0.3	2.51	0.1	-138	-57
6	0	5.31	0.2	12	225
7	0.3	4.61	0.3	81	235
8	0.2	2.91	0.4	97	199

Matrice de calibration obtenue (3D vers 2D)			
-1454.910546	38.566679	11.908724	-21.916180
-4.277169	-516.119534	-2055.073637	1707.009697
0.137259	-1.603682	0.277712	1.0

### Vérification du calibrage

Nous avons réalisé une vérification du calibrage avec plusieurs points, en comparant les coordonnées de ces points mesurées dans le repère absolu et celles calculées à partir des données fournies par la caméra.

Le tableau ci-dessous représente les résultats obtenus pour trois points.

	x	y	x	y	x	y
Coordonnées mesurées en (m)	0.000	2.710	0.000	3.510	0.600	6.110
Coordonnées estimées en (m)	0.001	2.720	0.001	3.504	0.591	6.108

## ANNEXE B : Calcul des lois de commande proportionnelles et proportionnelles intégrales

### B.1. Calcul de la commande proportionnelle.

Considérons la fonction de tâche (ou d'erreur) donnée par Chaumette pour la commande référencée vision :

$$\text{eq B 1} \quad e = C \cdot (\underline{S} - \underline{S}^*)$$

- C est une matrice de combinaison permettant d'utiliser un nombre d'informations visuelles supérieur au nombre de degrés de liberté contraints par la fonction de tâche e. Elle traduit l'erreur ( $\underline{S} - \underline{S}^*$ ) exprimée dans l'espace image, dans l'espace cartésien du robot. Cela signifie que l'erreur dans l'image devient, par la matrice C, une erreur sur les axes du robot.
- e est un vecteur à 6 composantes correspondant aux 6 degrés de liberté du robot.  $\underline{S} - \underline{S}^*$  est un vecteur à 2 composantes qui sont les informations visuelles. C est une matrice à 6 lignes et deux colonnes ( $C_1$  et  $C_2$ ). Ecrivons :

$$C = (C_1, C_2)$$

D'après l'équation (eq B 1) nous avons :

$$\text{eq B 2} \quad e = C_1 \cdot (\theta - \theta^*) + C_2 \cdot (\rho - \rho^*) = e_\theta + e_\rho$$

Pour calculer notre loi de commande proportionnelle nous imposerons une décroissance exponentielle à  $e_\theta$  et à  $e_\rho$ . Nous poserons :

$$\text{eq B 3} \quad \begin{cases} \dot{e}_\theta = -\lambda \cdot \beta \cdot e_\theta \\ \dot{e}_\rho = -\lambda \cdot e_\rho \end{cases}$$

$\lambda$  et  $\beta$  étant deux réels strictement positifs.

donc :

$$\text{eq B 4} \quad \dot{e} = \dot{e}_\theta + \dot{e}_\rho = -\lambda \cdot (\beta \cdot e_\theta + e_\rho)$$

En utilisant les méthodes classiques du formalisme de la commande référencée vision nous pouvons calculer notre matrice de commande. Pour cela nous dérivons l'équation (eq B 1) :

$$\text{eq B 5} \quad \dot{e} = C \cdot \dot{S} = C_1 \cdot \dot{\theta} + C_2 \cdot \dot{\rho} = \dot{e}_\theta + \dot{e}_\rho$$

avec (eq B 4) nous avons :

$$\begin{aligned}\dot{e} &= -\lambda \cdot (\beta \cdot C_1 \cdot (\theta - \theta^*) + C_2 \cdot (\rho - \rho^*)) \\ \dot{e} &= -C_1 \cdot \lambda \cdot \beta \cdot (\theta - \theta^*) - \lambda \cdot C_2 \cdot (\rho - \rho^*)\end{aligned}$$

car  $\lambda$  et  $\beta$  sont des scalaires appartenant à  $\mathbb{R}^+$ .

nous obtenons :

$$\text{eq B 6} \quad \dot{e} = -\lambda \cdot C \cdot \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \theta - \theta^* \\ \rho - \rho^* \end{pmatrix}$$

avec (eq B 5) et (eq B 6) nous avons :

$$\text{eq B 7} \quad C \cdot \dot{\underline{S}} = -\lambda \cdot C \cdot B \cdot (\underline{S} - \underline{S}^*) \text{ avec } B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$$

Comme le montre Chaumette dans [Chau91], une valeur possible de  $C$  pour satisfaire les conditions de convergence au voisinage de la position désirée est :

$$\text{eq B 8} \quad C = L_{\underline{S}=\underline{S}^*}^{T+}$$

$L^{T+}$  pour  $\underline{S}=\underline{S}^*$  est la pseudo inverse de la matrice d'interaction calculée pour la position d'équilibre.

$$L_{\underline{S}=\underline{S}^*}^{T+} \cdot L_{\underline{S}=\underline{S}^*}^T = I_6$$

La modélisation de la scène nous donne au voisinage de la position d'équilibre :

$$\text{eq B 9} \quad \underline{S} = L_{\underline{S}=\underline{S}^*}^T \cdot T_c$$

$T$  est le torseur cinématique représentant les 6 degrés de liberté du robot.

Des équations (eq B 7), (eq B 8) et (eq B 9) nous déduisons :

$$\text{eq B 10} \quad T = -\lambda \cdot L_{\underline{S}=\underline{S}^*}^{T+} \cdot B \cdot (\underline{S} - \underline{S}^*)$$

avec :

- $T$  torseur cinématique représentant les mouvements de la caméra,
- $\lambda \in \mathbb{R}^+$  et  $B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$  gains proportionnels,
- $L_{\underline{S}=\underline{S}^*}^{T+}$  torseur d'interaction calculé à l'équilibre,

T donne une commande en vitesse de notre caméra en fonction des coordonnées de la droite détectée et de la droite de référence exprimées dans l'espace image.

### **B.2.Calcul de la commande proportionnelle intégrale.**

Nous considérons maintenant une fonction d'erreur incluant l'intégrale de l'erreur.

$$\text{eq B 11} \quad e = C \cdot \left[ (\underline{S} - \underline{S}^*) + K_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot du \right]$$

$$\text{avec } K_i = \begin{pmatrix} k_1 & \\ & k_2 \end{pmatrix}$$

Nous allons appliquer à cette nouvelle fonction d'erreur les mêmes hypothèses que celles utilisées pour la commande proportionnelle au paragraphe précédent. Nous obtenons alors :

$$\text{eq B 12} \quad \dot{e} = -\lambda \cdot C \cdot B \cdot \left[ (\underline{S} - \underline{S}^*) + K_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot du \right] \text{ avec } B = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$$

En dérivant (eq B 12) on obtient :

$$\text{eq B 13} \quad \dot{e} = C \cdot \left( \dot{\underline{S}} + K_i \cdot (\underline{S} - \underline{S}^*) \right)$$

en linéarisant  $\underline{S}$  autour de  $\underline{S}^*$  on posera :

$$\text{eq B 14} \quad \dot{\underline{S}} = \frac{\underline{S} - \underline{S}^*}{T_e} \quad T_e \text{ étant la période d'échantillonnage du système.}$$

L'équation (eq B 14) suppose que l'on soit proche du point de fonctionnement. Cependant nous montrerons au chapitre IV que la loi de commande issue de ces équations converge même pour des conditions initiales éloignées de  $\underline{S}^*$ .

L'équation (eq B 13) devient alors :

$$\text{eq B 15} \quad \dot{e} = C \cdot \left( \dot{\underline{S}} + K_i \cdot T_e \cdot \dot{\underline{S}} \right)$$



A partir de (eq B 12) et de (eq B 15) nous écrivons :

$$\text{eq B 16} \quad \dot{\underline{e}} = \underline{C} \cdot [\underline{I}_2 + \underline{K}_i \cdot \underline{T}_e] \cdot \dot{\underline{S}} = -\lambda \cdot \underline{C} \cdot \underline{B} \cdot \left[ (\underline{S} - \underline{S}^*) + \underline{K}_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot du \right]$$

En supposant que les termes de la matrice  $\underline{K}_i$  soient très petits devant 1, nous approximerons l'équation (eq B 16) par :

$$\text{eq B 17} \quad \dot{\underline{e}} = \underline{C} \cdot \dot{\underline{S}} = -\lambda \cdot \underline{C} \cdot \underline{B} \cdot \left[ (\underline{S} - \underline{S}^*) + \underline{K}_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot du \right]$$

Cette approximation est justifiée car les coefficients de la matrice  $\underline{K}_i$  retenus pour la loi de commande proportionnelle intégrale sont  $k_1 = 0$  et  $k_2 = 0.008$ . La période d'échantillonnage est fixée à 200 ms. La matrice  $\underline{K}_i$  est donc négligeable devant  $\underline{I}_2$ .

En utilisant le même raisonnement qu'au paragraphe précédent nous obtenons :

$$\underline{T} = -\lambda \cdot \underline{L}_{\underline{S}=\underline{S}^*}^{T+} \cdot \underline{B} \cdot \left[ (\underline{S} - \underline{S}^*) + \underline{K}_i \cdot \int_0^t (\underline{S} - \underline{S}^*) \cdot du \right]$$

avec :

-  $\underline{T}$  torseur cinématique représentant les mouvements de la caméra,

-  $\lambda \in \mathfrak{R}^{*+}$  et  $\underline{B} = \begin{pmatrix} \beta & 0 \\ 0 & 1 \end{pmatrix}$  gains proportionnels,

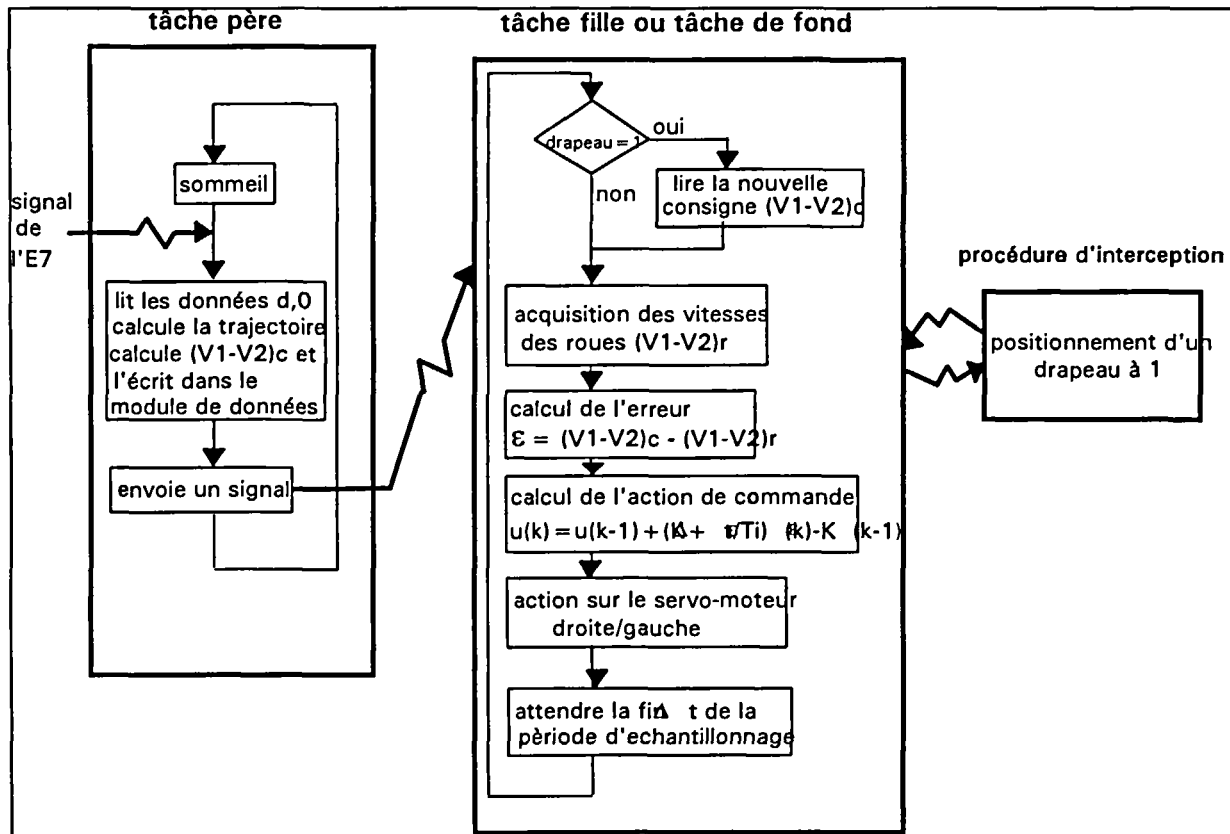
-  $\underline{L}_{\underline{S}=\underline{S}^*}^{T+}$  torseur d'interaction calculé à l'équilibre,

-  $\underline{K}_i = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}$  matrice de gains intégrals.

$\underline{T}$  est le vecteur de commande de notre caméra par retour proportionnel intégral sur l'erreur mesurée dans l'image.

## ANNEXE C : Exécution en parallèle des programmes de commande et d'asservissement bas niveau sur la carte SAC30

Le programme de commande et celui de l'asservissement bas niveau se partagent le temps CPU de la carte SAC30 (mécanisme du multitâche). La tâche de commande (tâche père) est synchronisée par la structure pipeline alors que l'asservissement bas niveau s'exécute sans arrêt en tâche de fond. Ce dernier communique avec la tâche père pour recevoir la consigne  $(V1-V2)c$ .



***Organigramme du processus de contrôle/commande***

Pour gérer ces deux processus, nous avons utilisés les mécanismes du multitâche. Pendant la phase d'initialisation, la tâche père :

- lance la tâche fille (**fork**)
- crée un **module de données** pour transmettre la consigne.

Une fois "forkée" la tâche fille crée **un lien avec le module de données**.

Pendant la phase de traitement, la tâche père écrit la consigne  $(V1-V2)c$  dans le module de données et en avertit la tâche de fond en lui envoyant un **signal**. Cette dernière exécute en permanence l'asservissement. lorsqu'elle reçoit le signal, elle interrompt momentanément son traitement pour exécuter la **procédure d'interception** qui lui signale qu'elle doit lire la nouvelle consigne. Après passage dans la procédure d'interception, la tâche fille reprend son exécution à l'endroit où elle l'avait arrêtée à la réception du signal.

**Imp Sciences 24, Avenue des Landais - 63177 AUBIERE Cedex - Dépôt légal  
4<sup>e</sup> Trimestre 1996**

## RESUME

Ce mémoire vise à exposer les méthodes utilisées pour asservir deux véhicules agricoles sur une ligne symbolisant une interface de végétation. Dans ce cadre, nous décrivons un système d'aide à la conduite composé de deux modules principaux. Le premier concerne la perception de l'environnement qui se fait au moyen d'une caméra CCD et d'une segmentation Markovienne non supervisée d'images texturées. Le second fait l'objet des développements théoriques et pratiques de ce mémoire concernant les différents modèles et lois de commande appliqués aux véhicules.

Le premier chapitre traite de l'environnement des robots mobiles et plus particulièrement des conséquences de celui-ci sur quelques lois de commande couramment utilisées.

Le second chapitre propose une modélisation cinématique des engins qui permettra de leur appliquer facilement des lois de commandes calculées pour l'essieu rigide à deux roues.

Le troisième chapitre expose les développements théoriques de deux lois de commande. La première permet de générer une trajectoire à partir de deux positions d'un véhicule reconstruites dans l'espace de la scène. La seconde est déduite du formalisme de la commande référencée vision et propose un contrôle de véhicules non holonomes dans l'espace image.

Le quatrième chapitre est consacré aux résultats obtenus avec le système d'aide à la conduite et propose une approche originale de la modélisation et de la correction dans l'espace image du glissement d'un robot mobile sur un terrain comportant une pente transversale.

Enfin, le cinquième chapitre donne les différents travaux futurs concernant la robustesse du système d'aide à la conduite et de son application à d'autres robots mobiles.

MOTS CLES : Robots Mobiles, Guidage, Asservissement Visuel, Commande, Segmentation en régions, Champs de Markov, Texture, Agriculture.