

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**

**CENTRE REGIONAL ASSOCIE DE GRENOBLE**

---

**MEMOIRE**

**Présenté en vue d'obtenir**

**le DIPLOME D'INGENIEUR CNAM**

**en**

**AUTOMATISME**

**par**

**Marcel RIESS**

---

**Qualification d'un télémètre à balayage laser  
pour la robotique mobile :  
Intégration et expérimentations**

---

**JURY :     Président : J.L. THOMAS**

**Membres : E. CHAMBEROD  
              B. DESCOTES-GENON  
              P. DIGIACOMO  
              M. NOUGARET**

**Invité :     G. BAILLE**



**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**

**CENTRE REGIONAL ASSOCIE DE GRENOBLE**

---

**MEMOIRE**

**Présenté en vue d'obtenir**

**le DIPLOME D'INGENIEUR CNAM**

**en**

**AUTOMATISME**

**par**

**Marcel RIESS**

---

**Qualification d'un télémètre à balayage laser  
pour la robotique mobile :  
Intégration et expérimentations**

---

les travaux relatifs au présent mémoire ont été effectués à l'Institut National de  
Recherche en Informatique et Automatique (INRIA) au sein du Service Moyens Robotiques



# Remerciements

---

J'exprime toute ma gratitude à Monsieur J.L. THOMAS, Professeur au Conservatoire National des Arts et Métiers d'avoir accepté de présider le jury.

Je remercie également :

Monsieur M. Nougaret, Professeur à l'IUT 1 de GRENOBLE et responsable de la spécialité Automatismes au C.U.E.F.A., pour sa participation au jury et ses conseils concernant la rédaction de ce mémoire ;

Monsieur B. DESCOTES-GENON, Professeur à l'IUT 1 de GRENOBLE, d'avoir accepté de prendre part au jury ;

Monsieur E. CHAMBEROD, Maître de conférences à l'IUT 1 de GRENOBLE, également pour sa participation au jury ;

Monsieur P. DIGIACOMO, Ingénieur de recherche (INPG) au laboratoire GRAVIR, pour m'avoir proposé ce sujet et pour m'avoir apporté toute l'aide nécessaire pour mener ce travail à terme ;

Monsieur G. BAILLE, Ingénieur de recherche INRIA, pour ses précieux conseils et sa disponibilité.

Je remercie aussi J. HERMOSILLO et F. LARGE du projet SHARP pour leur précieuse collaboration.

Enfin, je souhaite remercier les membres de l'équipe des Moyens Robotiques de l'INRIA Rhône-Alpes pour leur accueil et l'ambiance de travail, en particulier S. ARIAS et H. MATHIEU qui ont investi du temps pour la réalisation de ce mémoire.



# Table des matières

---

<b>INTRODUCTION.....</b>	<b>1</b>
<b>CHAPITRE I - TELEMETRE A BALAYAGE LASER.....</b>	<b>3</b>
1. PRESENTATION DU CAPTEUR LASER .....	4
1.1. Principe optique .....	4
1.2. Principe électronique .....	5
1.3. Détection de catadioptrés.....	6
1.4. Caractéristiques optiques .....	6
1.4.1. Portée.....	6
1.4.2. Résolution en distance.....	7
1.4.3. Précision.....	7
1.4.4. Angle de balayage .....	8
1.4.5. Résolution angulaire.....	8
1.5. Caractéristiques électriques .....	9
1.5.1. Temps de réponse .....	9
1.5.2. Caractéristiques .....	9
1.5.3. Interfaces TOR.....	10
1.5.4. Interface série .....	10
1.5.5. Liste de télégrammes.....	10
2. CHAMPS D'APPLICATIONS .....	12
2.1. Applications « tout ou rien » .....	12
2.2. Applications de mesure.....	12
<b>CHAPITRE II - APPLICATIONS.....</b>	<b>15</b>
1. ANTI COLLISION .....	16
1.1. Principe fonctionnel.....	17
1.1.1. Asservissement en vitesse .....	18
1.1.2. Correcteur .....	18
1.2. Problématiques et solutions liées au bruit .....	19
1.2.1. Utilisation sélective des données.....	20
1.2.2. Elimination du bruit .....	22
1.3. Résultats.....	24
2. SUIVI DE VEHICULE.....	25
2.1. Solution technique à base de caméra linéaire .....	26
2.2. Solution technique à base de capteur télémétrique à balayage laser .....	26
3. LOCALISATION.....	27

3.1. Problématique .....	27
3.2. Solutions techniques .....	28
3.3. Réduction des erreurs d'odométrie par recalage .....	28
3.4. Réduction des erreurs d'odométrie par apprentissage et fusion de données .....	30
4. CONCLUSION .....	32
<b>CHAPITRE III - SUIVI DE VEHICULE.....</b>	<b>33</b>
1. OBJECTIFS .....	34
2. LE CYCAB .....	35
2.1. Introduction.....	35
2.2. Présentation du Cycab .....	35
2.2.1. Caractéristiques générales.....	35
2.2.2. Le châssis .....	36
2.2.3. Alimentation générale .....	37
2.2.4. Les moteurs de traction.....	37
2.2.5. Le vérin motorisé de direction .....	37
2.2.6. Les freins de parking .....	38
2.2.7. L'Interface Homme Machine .....	38
2.2.8. La boucle d'arrêt d'urgence.....	38
2.3. Architecture matérielle de contrôle/commande .....	38
2.3.1. Description fonctionnelle des nœuds .....	38
2.3.2. L'alimentation des nœuds .....	40
2.3.3. Le module de calcul .....	40
2.3.4. La carte d'interface .....	41
2.3.5. Le nœud de direction.....	42
2.3.6. Le nœud de traction.....	42
3. LOCALISATION PAR CATADIOPTRES .....	43
3.1. Modèle géométrique .....	43
3.2. Détection et localisation des catadioptres .....	44
3.3. Problématique liée à l'environnement non structuré.....	46
3.4. Dimensionnement des catadioptres .....	47
3.5. Erreur angulaire .....	48
4. COMMANDE .....	51
<b>CHAPITRE IV - DEVELOPPEMENTS LOGICIELS.....</b>	<b>53</b>
1. ARCHITECTURE LOGICIELLE DU CYCAB.....	54
1.1. Rappels sur l'informatique embarquée du CYCAB Rhône-Alpes.....	54
1.2. Choix sur la communication.....	54
1.3. Environnement de Programmation.....	54

1.4. Modèle d'exécution du logiciel sur les nœuds .....	54
1.5. Le Protocole CAN-CY CAB .....	55
1.6. Pilotage du CY CAB via le CAN .....	56
2. INTERFACE CAPTEUR.....	57
2.1. Introduction.....	57
2.2. Présentation de l'interface logicielle .....	57
2.3. Interface de communication série .....	58
2.4. Gestionnaire de télégrammes.....	58
2.5. Echanges de données entre le capteur et l'interface .....	58
2.6. Menu utilisateur .....	63
2.7. Structures et variables globales .....	63
2.8. Interface TCP/IP (Serveur de socket) .....	64
2.9. Architecture Multi-thread .....	65
2.10. Modes de mesures.....	66
2.11. Datation des mesures .....	68
3. ORCCAD .....	70
3.1. Introduction à ORCCAD .....	70
3.2. Les modules .....	71
3.2.1. La Ressource Physique Cycab .....	71
3.2.2. La Ressource physique SickCata .....	72
3.3. La Tâche Robot de suivi .....	73
3.4. Compilation et intégration.....	74
<b>CHAPITRE V - RESULTATS .....</b>	<b>76</b>
1. PERFORMANCES DU SYSTEME DE DETECTION DE CIBLE .....	77
1.1. Distance de détection de la cible .....	77
1.2. Angle de détection de la cible.....	78
1.3. Invariance au parasites.....	79
1.4. Détection verticale .....	79
2. EXPERIMENTATIONS DU SUIVI DE VEHICULE .....	80
<b>CONCLUSION .....</b>	<b>82</b>
<b>ANNEXE.....</b>	<b>84</b>
1. INTERFACE CAPTEUR.....	85
1.1. Liste des télégrammes.....	85
1.1.1. Télégrammes du PC vers le capteur .....	85
1.1.2. Télégrammes du capteur vers le PC .....	86

---

1.2. Fonctionnalités du menu utilisateur .....	87
1.3. Thread serveur de socket .....	88
1.4. Thread de mesure.....	89
<b>2. DONNEES EN VUE D'EXPERIMENTATIONS .....</b>	<b>90</b>
2.1. Cinématique du Cycab.....	90
2.1.1. Notations .....	90
2.1.2. Modélisation.....	90
2.1.3. Détermination des vitesses de rotation des roues.....	92
2.2. Calibration du Cycab .....	95
2.2.1. Calibration de la translation .....	95
2.2.2. Calibration de la direction.....	96
<b>BIBLIOGRAPHIE.....</b>	<b>98</b>
2.3. Articles.....	98
2.4. Rapports techniques .....	99
2.5. Conférences .....	99
2.6. Thèses .....	99
2.7. Rapports de stage .....	100
2.8. Documents constructeur .....	100

# Table des illustrations

---

Figure I.1.1.a : Construction optique .....	4
Figure I.1.1.b : Phases de fonctionnement .....	5
Figure I.1.2 : Synoptique électronique .....	5
Figure I.1.4.1.a : Portée Capteur d'intérieur (Indoor).....	6
Figure I.1.4.1.b : Portée Capteur extérieur (Outdoor).....	7
Figure I.1.4.1.c : Réflectivité de divers matériaux.....	7
Figure I.1.4.5 : Diamètre spot, et distances inter-spot pour le capteur d'intérieur .....	8
Figure I.1.5.2 : Caractéristiques du capteur .....	9
Figure I.1.5.4 : Structure des télégrammes.....	10
Figure I.2.1 : Champs de surveillance programmables .....	12
Figure I.2.2 : Mise en forme des données de mesure .....	13
Figure II.1 : Fonction anti-collision .....	16
Figure II.1.1 : Mesures sur 9 partitions pour l'anti-collision.....	17
Figure II.1.1.2.a : Distances de sécurité et d'arrêt .....	18
Figure II.1.1.2.b : Boucle d'asservissement en vitesse .....	19
Figure II.1.2.a : Données télémétriques avant filtrage .....	19
Figure II.1.2.b : Vitesses avant filtrage .....	20
Figure II.1.2.1 a : Distances maximales de prise en compte des mesures .....	21
Figure II.1.2.1.b : Limitation du champ de surveillance .....	21
Figure II.1.2.1.c : Données télémétriques après limitation du champ de surveillance.....	22
Figure II.1.2.2.a : Vitesses après limitation du champ de surveillance .....	23
Figure II.1.2.2.b : Vitesses après filtrage .....	23
Figure II.2.a : Suivi de véhicules.....	25
Figure II.2.b : Cinématique du système de véhicules .....	25
Figure II.3 : Le modèle idéal du Cycab (gauche) et le modèle actuel (droite).....	28
Figure II.3.3 : Localisation par balises passives.....	30
Figure II.3.4 : Principe d'une commande à base de réseau de neurones .....	31
Figure III.2.2 : Vue éclatée du Cycab .....	35
Figure III.2.2.1 : Synoptique de l'architecture matérielle de Contrôle/Commande.....	36
Figure III.2.2.2 : Architecture matérielle : Implémentation sur le châssis.....	37
Figure III.2.3.1 : Schéma fonctionnel d'un nœud .....	39
Figure III.2.3.3 : Schéma fonctionnel du microcontrôleur MC68332 .....	40
Figure III.3.1: Repérage de la cible .....	43
Figure III.3.2.a : Repérage de la cible .....	44
Figure III. 3.2.b : Niveaux de réflexion.....	45
Figure III.3.3 : Niveaux de réflexion.....	46
Figure III.3.4.a : Angle de déviation verticale .....	47
Figure III.3.4.b : Diamètre des spots en fonction de la distance .....	47
Figure III.3.4.c : Distance interspots en fonction de la distance et de la résolution.....	47
Figure III.3.4.d : Largeur minimale du catadioptré .....	48
Figure III.3.4.e : Largeur minimale de détection en fonction de la distance .....	48
Figure III.3.5.a : Largeur minimale du catadioptré.....	49
Figure III.3.5.a : Erreur de mesure de $dc$ .....	49
Figure III.4.a : Contrôle longitudinal .....	51
Figure III. 4.b : Contrôle latéral .....	51
Figure IV.1.4 : Modèle d'Exécution.....	55
Figure IV.1.5 : Trame CAN .....	56

Figure IV.2.2 : Schéma bloc de l'interface logicielle capteur .....	57
Figure IV.2.5.a : Commande simple .....	59
Figure IV.2.5.b : Commande avec réponse .....	59
Figure IV.2.5.c : Commande avec mauvais CRC .....	59
Figure IV.2.5.d : Mode "sur demande" .....	60
Figure IV.2.5.e : Mode "continu" .....	60
Figure IV.2.5.f : Temps de réponse global.....	61
Figure IV.2.5.g : <i>Tresp</i> en fonction de la résolution angulaire .....	62
Figure IV.2.7 : Partage de données .....	64
Figure IV.2.9a : Architecture multi-threads.....	65
Figure IV.2.9b : Architecture multi-threads.....	66
Figure IV.2.10.a : Modes de mesure .....	67
Figure IV.2.10.b : <i>Trcl</i> suivant les modes de mesure.....	68
Figure IV.2.10.c : <i>Trca</i> suivant les modes de mesure .....	68
Figure IV.3.2.1.a : Ressource physique Cycab .....	71
Figure IV. 3.2.1.b : Ports d'entrée et de sortie de la ressource physique Cycab .....	72
Figure IV.3.2.2 : Ressource physique SickCata.....	73
Figure IV.3.3 : Tâche robot de suivi .....	74
Figure V.1.2 : Angle d'orientation de la cible .....	78
Figure V.2.a : Asservissement longitudinal .....	80
Figure V. 2.b : Asservissement latéral de direction .....	81
Figure A.1.3 : Thread serveur de socket .....	88
Figure A.1.4 : Thread de mesure .....	89
Figure A.2.1.2 : Modélisation du véhicule Cycab.....	90
Figure A.2.1.3.a : Mise en évidence de l'orientation $q$ au centre des roues .....	92
Figure A.2.1.3.b : Mécanisme de direction du Cycab.....	94
Figure A.2.1.3.c : Evolution des braquages en fonction de $f$ .....	94
Figure A.2.1.3.d : Détail du mécanisme de direction sur l'essieu avant .....	95

# Introduction

---

De nombreux travaux de recherche sont réalisés sur la problématique générique des véhicules intelligents et de la robotique mobile. On peut citer le programme européen *Eurêka PROMOTHEUS*<sup>1</sup> qui a eu pour but d'améliorer la sécurité et le trafic sur les routes européennes, le projet *Praxitèle* [18] s'articulant autour d'un système de transport public individuel à base de véhicules électriques et le concept de *la Route automatisée* englobant divers concepts de conduite automatisée et les infrastructures qui y sont associées. Les applications sont très diverses, et très souvent, elles permettent à l'homme de réaliser des actions qu'il lui serait difficile ou même impossible de réaliser seul. Lorsque l'homme veut se déplacer rapidement il peut, par exemple, utiliser un véhicule automobile qu'il pilote lui-même. Le véhicule intelligent a alors pour mission de l'aider dans sa tâche de conduite : il est opéré. Certains environnements sont inaccessibles à l'homme, soit par leur éloignement (une autre planète par exemple), soit par leur dangerosité (les fonds sous marins, les espaces radioactifs). L'opérateur humain contrôle alors le véhicule par télé-opération. Enfin, quand les tâches sont simples, répétitives, le véhicule intelligent n'est plus que supervisé par l'opérateur humain qui lui trace les grandes lignes de son action. Le véhicule se déplace alors de façon autonome. Dans tous les cas, et quel que soit le mode de contrôle du véhicule (opéré, télé-opéré, autonome), il est nécessaire d'avoir une bonne information sur l'état du véhicule et de son environnement et du véhicule dans son environnement. Pour cela, le véhicule doit être équipé d'un système de perception (qui est adapté à la mission qu'il doit réaliser), fournissant toutes les informations nécessaires à l'aboutissement de la mission qui lui est confié.

La recherche sur le thème des systèmes de transport intelligents se positionne comme un domaine particulier de la robotique mobile. Les chercheurs travaillant dans le domaine de la robotique mobile sont très vite confrontés aux problèmes de la connaissance de l'environnement et ceux liés à la localisation du robot dans cet environnement.

Le service robotique de l'INRIA Rhône-Alpes est chargé de la mise en œuvre des outils matériels et logiciels pour les expérimentations robotiques des projets de recherche. Dans ce cadre de recherche sur l'autonomie des véhicules, plusieurs expérimentations ont fait apparaître le besoin de données sensorielles télémétriques complémentaires à celles délivrées par les capteurs en place : caméra, ultrasons, odométrie.

Afin d'obtenir plus d'information sur l'environnement du véhicule, le service Robotique de l'INRIA a choisi d'intégrer un capteur télémétrique à balayage laser sur un véhicule électrique de type CYCAB et d'exploiter les données issues du capteur dans le cadre d'expérimentations sur l'autonomie : anti-collision, suivi de véhicule et localisation. L'étude, la qualification, l'intégration du capteur, l'exploitation des données, et le développement des expérimentations sont l'objet du travail que je présente.

---

<sup>1</sup> PROMOTHEUS : PROgram for an European Traffic with Highest Efficiency and Unprecedented Security



# Chapitre I

## TELEMETRE A BALAYAGE LASER

*En combinant un télémètre à temps de vol avec un système de rotation du faisceau de mesure, la société SICK a développé un télémètre de mesure à balayage en 2D. Ce chapitre décrit le principe fonctionnement de ce type de capteur, renseigne sur les caractéristiques techniques et introduit un certain nombre d'applications relatives à la conduite de véhicules autonomes.*



## 1. PRESENTATION DU CAPTEUR LASER

### 1.1. Principe optique

Le télémètre à balayage laser<sup>2</sup> est issu du PLS (Programmable Laser Scanner) développé par la société allemande SICK pour des applications de sécurité en milieu industriel et commercialisé depuis 1994. Ce capteur fait partie de la famille des LIDAR.<sup>3</sup>

Une impulsion laser très courte (un tir) est émise par une diode laser. Simultanément une horloge est démarrée. La distance entre le capteur et l'objet cible est directement proportionnelle au temps, dit "temps de vol", mis par l'impulsion lumineuse à parcourir la distance capteur-objet puis objet-capteur.

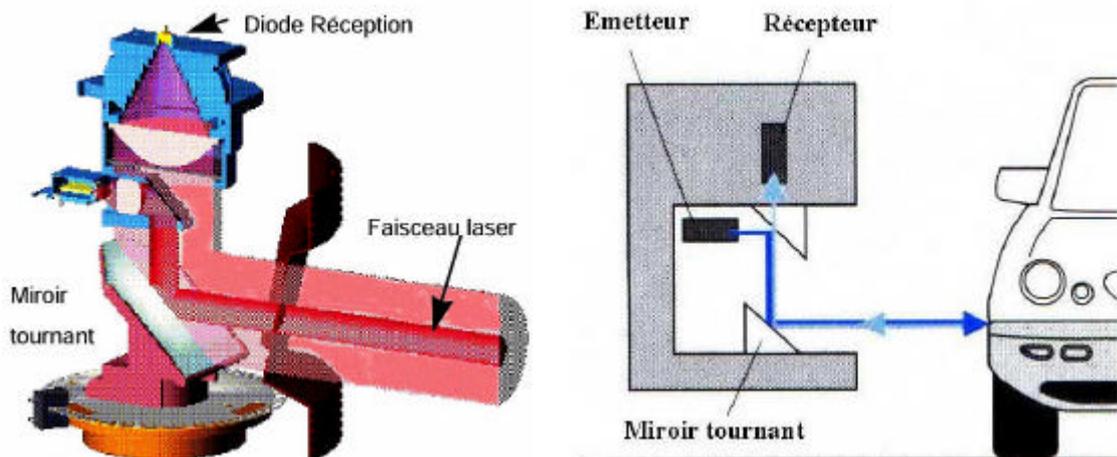
$$D = \text{distance capteur-objet} = C/2 * t$$

avec :  $C @ 300\,000 \text{ km/s}$

$t = \text{temps s'écoulant entre le départ de l'impulsion lumineuse et le retour de l'écho lumineux}$

Exemple : pour  $D=1\text{m}$  ;  $t @ 6 \text{ nS}$

Le faisceau laser émis est dévié par un miroir tournant (Figure I.1.1.a), permettant ainsi de balayer une surface de 360°.



**Figure I.1.1.a : Construction optique**

Pour des raisons mécaniques, le faisceau sortant a une course limitée à 100° ou 180° suivant le modèle de capteur. Le faisceau réfléchi par un obstacle quelconque se trouvant dans l'axe du tir revient vers le capteur et tombe sur un récepteur. Le signal de réception arrête l'horloge. A partir de la mesure du temps séparant l'émission de la réception, on déduit la distance capteur-obstacle ; connaissant la position angulaire du "tir" on peut en déduire la position exacte de l'objet dans un plan (coordonnées polaires). Le miroir tourne en permanence.

<sup>2</sup> LASER : acronyme de Light Amplification by Stimulated Emission of Radiation

<sup>3</sup> LIDAR : acronyme de Light Detecting and Ranging. Appareil de détection qui émet un faisceau laser et en reçoit l'écho, permettant de déterminer la distance d'un objet.

Lors d'une rotation, trois phases sont à considérer (figure I.1.1.b):

1. acquisition de la mesure
2. auto-tests et tir sur cible interne calibrée
3. communication avec l'interface (envoi des coordonnées polaires)

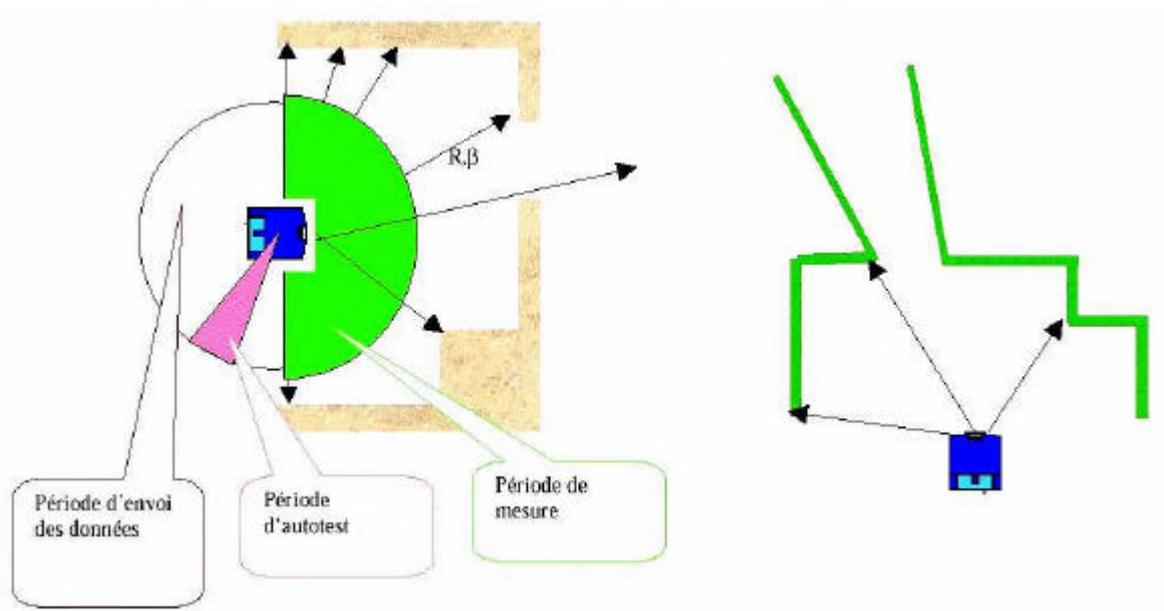


Figure I.1.1.b : Phases de fonctionnement

## 1.2. Principe électronique

A chaque incrément du moteur pas-à-pas, un "tir" est effectué par l'émetteur laser ; le compteur de temps est démarré simultanément et arrêté par l'impulsion de réception détectant le retour du laser (Figure I.1.2). Un détecteur de crête permet de compenser la mesure en fonction de la forme de l'impulsion reçue, celle-ci dépendant de la distance de mesure.

Dans les versions d'extérieur "Outdoor" permettant de faire des mesures par tout temps, une compensation de la réflexion directe sur le brouillard est prévue au niveau du gain de la partie analogique. Les données (distance cible-capteur) sont envoyées en temps réel au système maître (PC) via une interface série à haut débit (RS232).

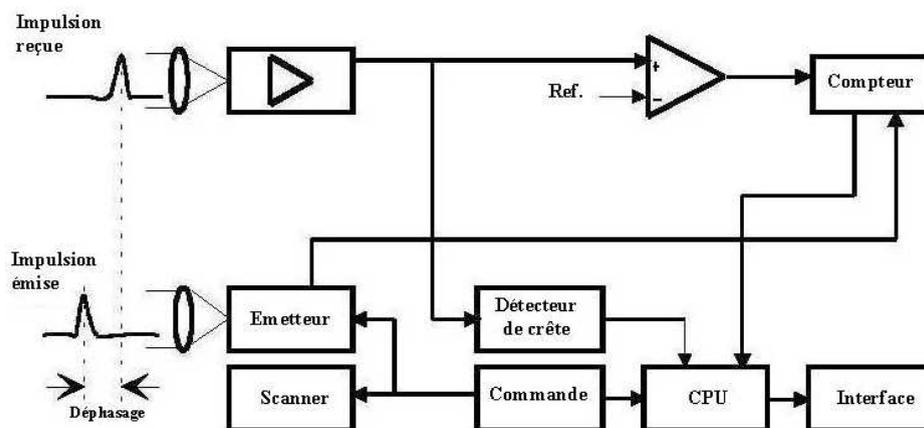


Figure I.1.2 : Synoptique électronique

### 1.3. Détection de catadioptrés

Le capteur SICK mesure le temps de vol, aller et retour, du faisceau laser. Avec une vitesse de propagation de la lumière constante, on obtient ainsi la distance de la cible. L'intensité du faisceau réfléchi, celui qui est détecté par le capteur, est bien plus faible que le faisceau émis. En effet, la cible va absorber une partie du faisceau incident.

Il est possible, sur le capteur SICK, de mesurer l'amplitude du faisceau réfléchi. Cette amplitude peut être quantifiée sur trois bits (maximum) en sortie du capteur, soit 8 niveaux. Cette mesure présente peu d'intérêt pour les matériaux standards, par contre, elle est d'un grand intérêt lorsque la cible est un catadioptré. Dans ce cas, le signal image du faisceau réfléchi est de grande amplitude; il peut ainsi être quantifié.

Un balisage de l'environnement avec des catadioptrés est ainsi possible. Cette information supplémentaire est reconnaissable par le capteur. Ceci permet un élargissement de l'éventail des applications, mais aussi une exploitation plus rapide des données. Nous utilisons les catadioptrés pour le suivi de véhicule présenté au chapitre III. L'installation de catadioptrés sur le véhicule suivi évite d'avoir à le reconnaître par sa forme arrière – ce qui serait réalisable sans problème mais nécessiterait un temps de traitement bien plus long.

### 1.4. Caractéristiques optiques

#### 1.4.1. Portée

La portée de ce type de télémètre dépend du pouvoir de réémission des objets cible. Cette propriété à absorber la lumière est la réflectivité<sup>4</sup>.(Cf. Figure I.1.4.1.c). Les courbes ci dessous indiquent les portées maximales obtenues en fonction de la réflectivité de l'objet :

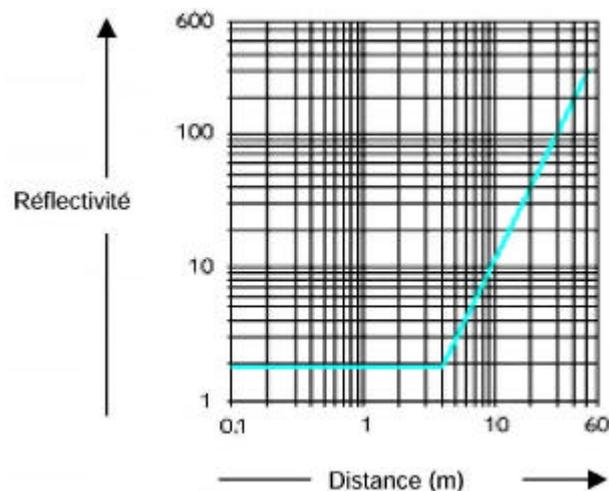
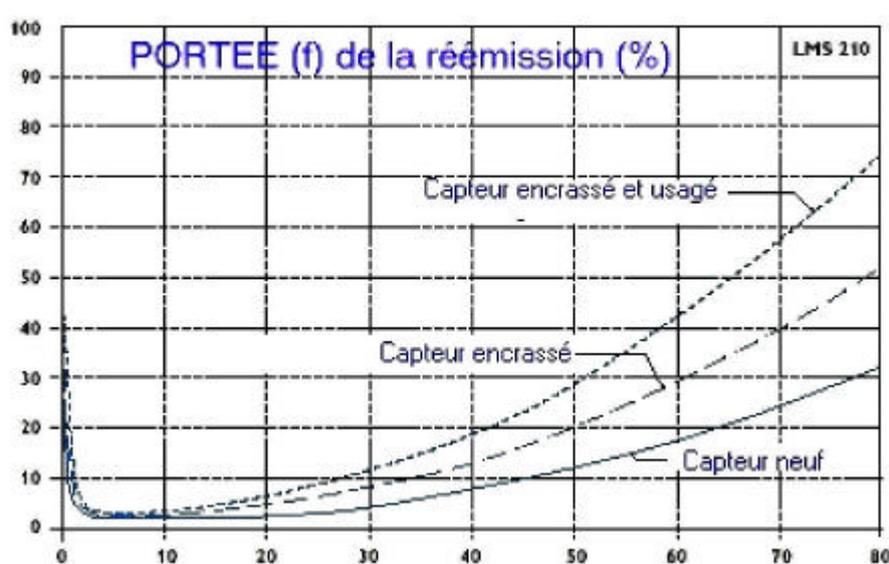


Figure I.1.4.1.a : Portée Capteur d'intérieur (Indoor)

<sup>4</sup> Réflectivité : Rapport de l'énergie réfléchiée à l'énergie incidente totale.



**Figure I.1.4.1.b : Portée Capteur extérieur (Outdoor)**

- De 0 à 10 mètres, les capteurs sont capables de faire des mesures sur pratiquement n'importe quel corps se trouvant dans la nature (excepté les corps à réflexion totale comme les miroirs);
- jusqu'à 30 mètres : mesure possible sur tout corps de réflectivité >10% (carton noir mat);
- à 50 mètres : mesure correcte sur corps ayant au moins 30% de réflectivité.

Matériaux	Reflectivité
Mousse caoutchouc noire	2%
Chaussure noire brillante	3%
Carton noir mat	10%
Carton gris	20%
Bois (écorce rugueuse, sale)	40%
PVC, gris	50%
Papier, blanc mat 80g/m <sup>2</sup>	80%
Réflecteurs	>2000%

**Figure I.1.4.1.c : Réflectivité de divers matériaux**

#### 1.4.2. Résolution en distance

La résolution des capteurs est de 10 mm, ce qui correspond à un temps de vol de  $60 \cdot 10^{-12}$  s. Le compteur de distance permet de lire des distances comprises entre 0 et 80 m en version extérieure et 160 m en version intérieure.

#### 1.4.3. Précision

La précision sur la mesure de distance est indépendante de la distance et des conditions d'environnement. Elle comprend une erreur systématique (pour les gammes totales de réflectivité, de portée et de température) de +/- 15 mm (pour les Indoor) et de +/- 30 mm (pour les Outdoor) plus une erreur de 10 mm due à la résolution.

Cette dernière peut être améliorée en augmentant le nombre de mesures et en calculant la valeur moyenne statistique. D'après les données du constructeur, le facteur d'amélioration de la précision est égal à  $n$ , ( $n$  est le nombre de balayages = nombre de mesures du même point).

Par ailleurs, la précision globale peut être augmentée en utilisant des techniques d'interpolation. On peut ainsi approcher des précisions de mesure de l'ordre du millimètre.

Enfin, on compense les erreurs induites par les variations de température en disposant d'une cible de référence : distance et angle connus précisément. Cette cible de référence permet de mesurer les variations dues à la température et donc de corriger la mesure. La cible de référence est intégrée au capteur et fait partie du système d'auto-calibration.

#### 1.4.4. Angle de balayage

Suivant les versions, l'angle de balayage peut être physiquement de  $100^\circ$  ou de  $180^\circ$ . La transmission des informations vers l'interface de communication série peut être sélective afin de ne traiter que les données utiles.

#### 1.4.5. Résolution angulaire

La résolution angulaire peut être programmée à  $0,25^\circ$ ,  $0,5^\circ$  ou  $1^\circ$ . Elle dépend directement de la résolution du moteur pas-à-pas d'entraînement du miroir. A chaque pas un "tir" est exécuté. Par conséquent, le nombre de tirs est obtenu en fonction de la résolution angulaire.

Dans le cas où le capteur a une ouverture de  $180^\circ$ , on a :

- 181 tirs pour une résolution de  $1^\circ$
- 361 tirs pour une résolution de  $0,5^\circ$
- 721 tirs pour une résolution de  $0,25^\circ$

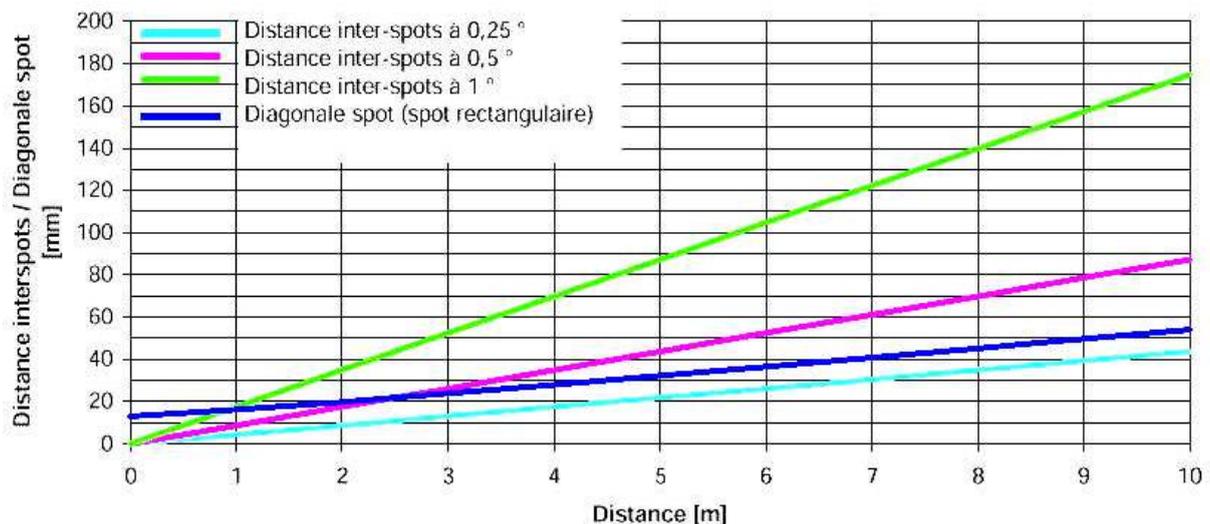


Figure I.1.4.5 : Diamètre spot, et distances inter-spot pour le capteur d'intérieur

## 1.5. Caractéristiques électriques

### 1.5.1. Temps de réponse

Le temps de réponse (temps entre le début d'un balayage et la mise à disposition des données)  $tr$  est lié à la vitesse de rotation du moteur ( constante = 4500 tr/mn) et à la résolution angulaire  $ra$  :

- $ra = 1^\circ$ ,  $tr = 13$  ms
- $ra = 0.5^\circ$ ,  $tr = 26$ ms
- $ra = 0.25^\circ$ ,  $tr = 52$  ms

Dans tous les cas, la vitesse de transfert des données sur l'interface série permet d'obtenir les mesures complètes d'un balayage en temps réel.

### 1.5.2. Caractéristiques

Le tableau de la figure I.1.5.2 fournit les caractéristiques du capteur utilisé dans le cadre des travaux présentés.

Version	LMS291- S05
Ouverture	180°
Portée sur noir 10 %	35 m
Portée sur gris 18%	50 m
Portée sur réflecteur	80 m (1)
Résolution angulaire (2)	0,25° 0,5° 1°
Résolution en distance	10 mm
Précision absolue pour 1 tir	+/- 30 mm
Diamètre spot à 10m	250 mm
Temps de réponse (3)	13 ms 26 ms 52 ms
Sorties PNP	3
Sorties série (4)	RS232/422

**Figure I.1.5.2 : Caractéristiques du capteur**

Notes :

(1) : due à limitation interne du compteur

(2) : paramétrable

(3) : fonction de la résolution angulaire choisie

(4) : commutable

### 1.5.3. Interfaces TOR

Trois interfaces de sortie TOR (Tout Ou Rien) peuvent être affectées à 3 champs de détection, indiquant la pénétration d'un objet quelconque dans un champ pré-programmé. Deux jeux de 3 champs peuvent être programmés dans le scanner.

### 1.5.4. Interface série

L'interface série est du type RS422 (4 fils) ou RS232<sup>5</sup>. La vitesse est programmable de 9600 Bds à 500 KBds

Les données sont échangées à l'aide de télégrammes. Un télégramme est une suite de caractères composée d'une entête, de données et d'une fin.



Figure I.1.5.4 : Structure des télégrammes

Désignation	Taille des données [Bits]	Description
STX	8	Bit de Start (02H)
ADR	8	Adresse du capteur (dans le cas de plusieurs capteurs). L'adresse d'un capteur s'obtient en additionnant 80H au numéro du capteur (0 à 4).
Len	16	Nombre de caractères de données (CMD + Data + Status).
CMD	8	Caractère de commande lors de l'envoi d'un télégramme vers le capteur. En réponse, le capteur renvoie un télégramme avec la commande initiale + 80H.
Data	N x 8	Optionnel, dépend de la commande.
Status	8	Optionnel, le capteur ne transmet de statut que quand il y a envoi de données. Aucun statut n'est envoyé au capteur.
CRC	16	Caractère de contrôle des données.

### 1.5.5. Liste de télégrammes

Une liste de télégrammes permet d'échanger des commandes, ou codes de commande et données entre le capteur et l'interface série du système de commande (PC).

<sup>5</sup> RS232 : Recommended Standard –232C, une liaison série bidirectionnelle asynchrone. RS422 est une version plus récente de la norme.

Ces télégrammes donnent accès aux fonctionnalités suivantes :

- Lecture et modification de la configuration du capteur ;
- Mesure télémétrique sur requête ou en continu ;
- Commande d'apprentissage d'un champ ;
- Configuration des champs ;
- Lecture et modification de la configuration du capteur.

Une liste des télégrammes est présentée dans le tableau 1.1 en Annexe.

La totalité des fonctionnalités sont détaillées dans le tableau 1.2 en Annexe.

## 2. CHAMPS D'APPLICATIONS

Les champs d'applications sont très vastes, on peut cependant les classer en deux grandes catégories :

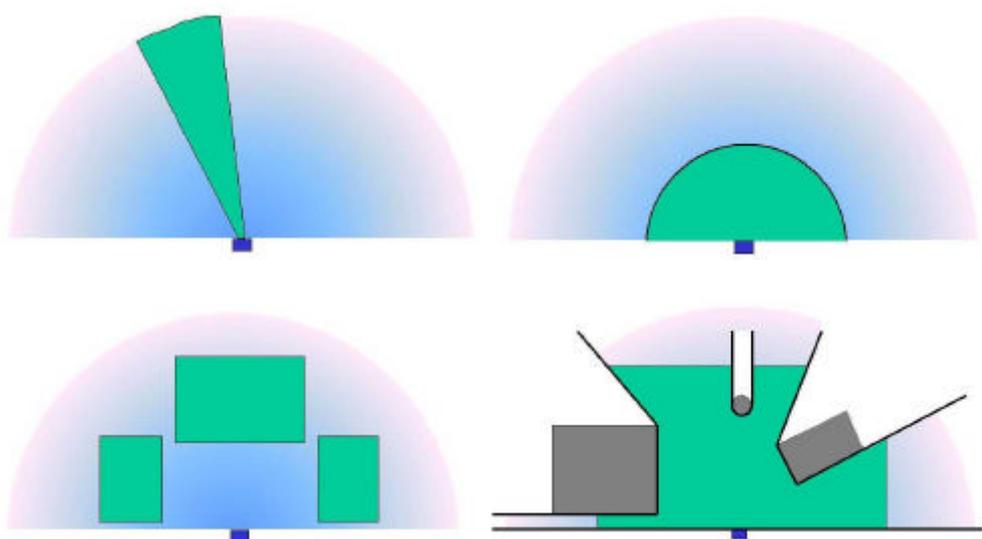
- Applications « tout ou rien »;
- Applications de mesure.

### 2.1. Applications « tout ou rien »

Le capteur permet de définir des zones (champs) à surveiller. L'état logique d'un champ varie lorsque un objet externe modifie les mesures télémétriques dans ce champ.

Trois champs peuvent être actifs simultanément. Les champs peuvent être de type segment, demi disque ou rectangle.

Une fonction "d'apprentissage" permet de définir un champ spécifique à partir d'une mesure de l'environnement à surveiller.



**Figure I.2.1 : Champs de surveillance programmables**

La mise en œuvre de ce type d'application est simple et ne nécessite pas de traitement particulier de données.

Cette fonctionnalité de surveillance est principalement utilisée dans des applications d'anti-collision ou d'anti-intrusion

### 2.2. Applications de mesure

Des applications plus sophistiquées nécessitent une récupération et un traitement des données de mesure. Le capteur est capable de délivrer le résultat de mesures de nature différentes sous diverses formes (tels que décrites ci-après).

Toutes les données de mesure issues du capteur proviennent des trois informations suivantes :

- Angle de tir;
- Distance;
- Niveau de réflectivité.

Un assortiment de traitements des données par le capteur permet d'obtenir les informations suivantes :

- Valeurs d'un balayage complet en coordonnées polaires et les niveau de réflectivité;
- Valeurs moyennées (moyenne des mesures de plusieurs balayages);
- Valeurs partitionnées (les mesures sont réalisées entre deux angles au choix);
- Valeurs moyennées et partitionnées;
- Plus petite distance perpendiculaire au rayon de balayage;
- Plus petite valeur d'un segment;
- Mesure en coordonnées cartésiennes.



**Figure I.2.2 : Mise en forme des données de mesure**

Le chapitre suivant illustre des applications qui exploitent les données télémétriques de ce type de capteur.



# Chapitre II

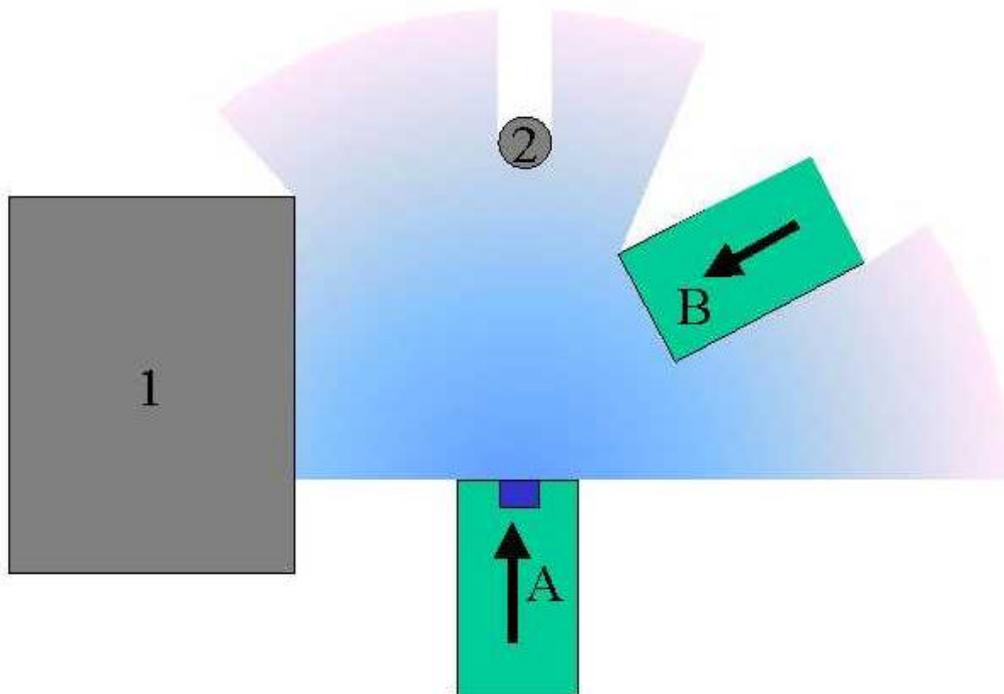
## APPLICATIONS

*Ce chapitre décrit les expérimentations, - principe, problématiques, résultats - ,que nous avons développé dans le cadre des recherches sur la conduite automatique exploitant des données télémétriques issues d'un capteur laser à balayage.*

## 1. ANTI COLLISION

Une des premières missions que doit remplir un véhicule autonome, est de garantir l'anti-collision avec d'autres objets inertes ou en mouvement. Dans la figure II.1, le véhicule A (équipé du télémètre laser) doit éviter les obstacles fixes 1 et 2 et le véhicule B qui est en mouvement. La fonction d'anti-collision est utilisée principalement dans deux cas :

- En mode de conduite manuelle en tant qu'aide à la conduite ;
- En mode de conduite autonome.



**Figure II.1 : Fonction anti-collision**

L'anticollision va consister à réduire de façon automatique la vitesse, qui est sous le contrôle du conducteur ou d'un système automatique., lorsqu'un obstacle est détecté dans une certaine zone.

Les principaux avantages du capteur dans ce type de missions sont :

- La portée des mesures : avec une portée de 80 mètres, le capteur télémétrique à balayage laser permet une anticipation de commande bien supérieure à celle autorisée par les systèmes à ultrasons. Il est donc possible de mettre en place des algorithmes de commande d'un degré d'autonomie supérieur aux simples actions réflexes, par exemple pour la planification des mouvements ;
- La précision des mesures : la précision centimétrique autorise des manœuvres plus fines et couvre en fait toutes les classes de manœuvres possibles pour ce type de véhicule ;
- La simplicité de mise en œuvre de la fonctionnalité : un seul capteur couvre l'espace (plan) sur 180°.

### 1.1. Principe fonctionnel

La fonction anti-collision peut être implémentée de plusieurs manières :

- L'utilisation des fonctionnalités associées à la surveillance de champs pré-programmés (Cf. Chapitre I.2.1);
- Exploitation des données télémétriques.

Dans le cas de l'utilisation de champs de surveillance, seules trois zones peuvent être délimitées. Dans les cas où l'application requiert une analyse et un traitement plus fin des données, nous préférons utiliser les données télémétriques brutes associées à des pré-traitements adaptés réalisés par le capteur.

Nous avons effectués plusieurs expérimentations pour déterminer le type de pré-traitement à effectuer : nombre de trames, nombre de secteurs, nature de la donnée dans le secteur. Nous avons retenu le découpage en 9 secteurs avec distance minimale dans chacun.

Le schéma de la figure II.1.1 montre l'implémentation d'une fonction d'anti-collision basée sur les principes suivants :

- L'espace à surveiller est découpé en 9 partitions égales ;
- Le capteur fournit la mesure minimale pour chaque secteur ;
- Si un secteur ne contient pas d'obstacle, la mesure obtenue est la valeur maximale liée à la portée du capteur ;
- On ne s'intéresse qu'à l'obstacle le plus proche.

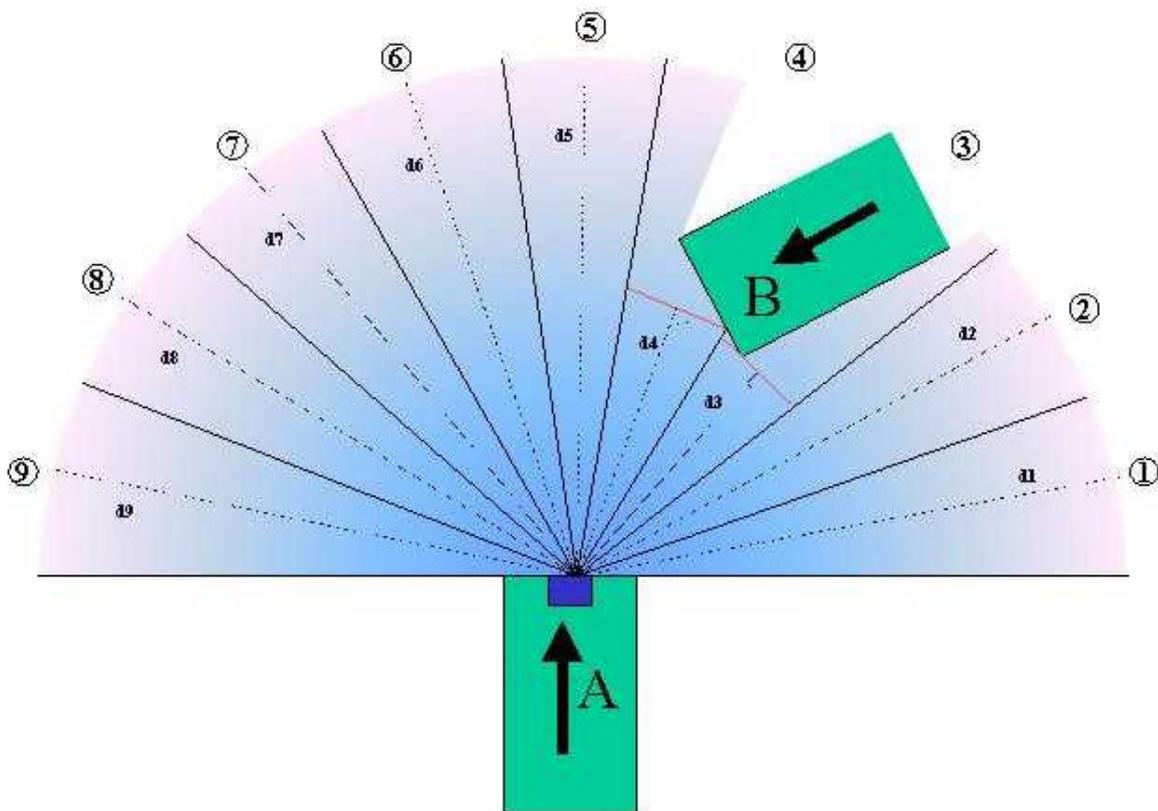


Figure II.1.1 : Mesures sur 9 partitions pour l'anti-collision

Dans cette illustration, l'obstacle représenté par le véhicule **B** est le plus proche du véhicule **A** qui doit éviter la collision. La mesure  $d$  sera prise en compte en tant que distance  $d$  minimale entre le capteur et l'objet le plus proche.

### 1.1.1. Asservissement en vitesse

De la distance  $d$  et de sa variation dans le temps, on obtient une vitesse  $V_r$ . Cette vitesse est une vitesse relative entre le véhicule **A** et l'obstacle **B**. Un asservissement de vitesse du véhicule **A** en fonction de la vitesse relative entre **A** et **B** permet de prendre en compte les cas suivants :

- **B** est immobile : la vitesse relative est la vitesse propre de **A**;
- **A** suit **B** à vitesse égale : la vitesse relative est nulle;
- **A** est immobile : la vitesse relative est la vitesse propre de l'obstacle;
- **A** et **B** sont en mouvement : la vitesse relative peut être négative ou positive.

### 1.1.2. Correcteur

Le véhicule possède un asservissement en vitesse commandé par une manette. Tant que le véhicule ne se trouve pas à une distance minimum de l'obstacle le plus proche, aucune correction n'est faite sur la consigne en vitesse. Si il y a franchissement de cette distance, un correcteur permet d'intervenir sur la consigne de vitesse.

On définit une distance de sécurité  $d_0$  ainsi qu'une distance d'arrêt  $da$ , comme identifiées sur la figure II.1.1.2.a.

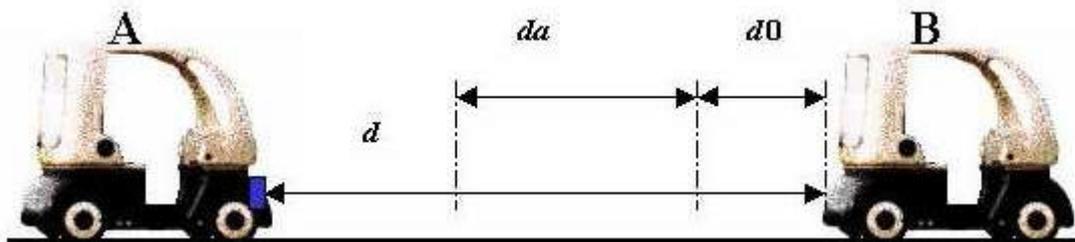


Figure II.1.1.2.a : Distances de sécurité et d'arrêt

La distance de sécurité  $d_0$  est fixe. La distance d'arrêt dépend de la vitesse  $V_A$  de **A** :

$$da = \frac{V_A^2}{2gf} \text{ avec}$$

$da$  : distance d'arrêt

$V_A$  : vitesse propre du véhicule **A**

$g$  : constante de gravité

$f$  : coefficient de frottement

Les lois de commande s'écrivent alors :

- Si  $d > (da + d_0)$  : la vitesse de commande  $V_e$  de **A** est la consigne en vitesse  $V_c$

- Si  $d \leq (da + d0)$  : la vitesse de commande  $V_e$  de **A** devient :  $V_e = V_c - kV_r$  avec

$$V_r : \text{vitesse relative entre A et B et } k = 1 - \left( \frac{da}{d - d0} \right)$$

Le coefficient variable k s'annule si  $d = da + d0$

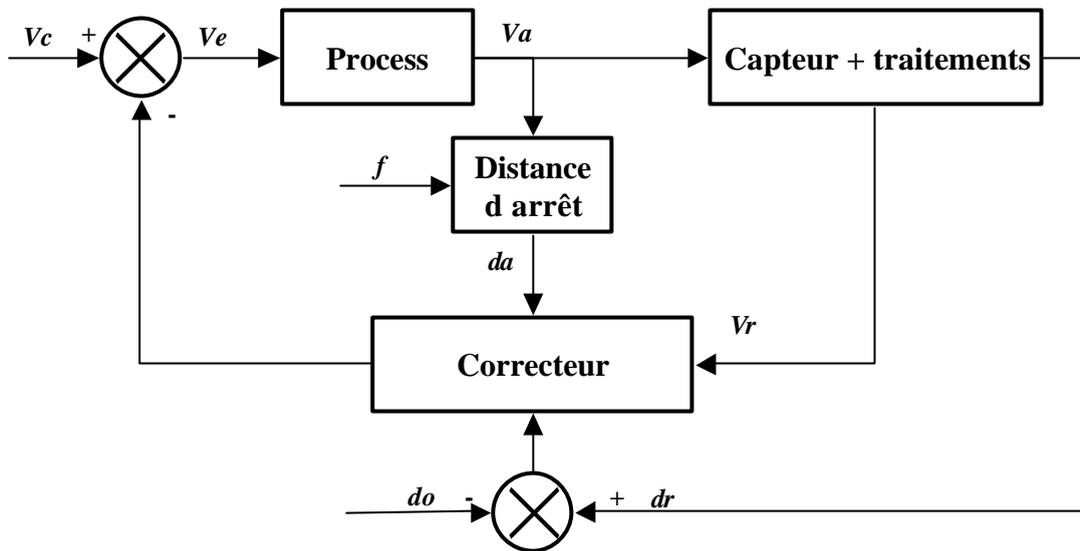


Figure II.1.1.2.b : Boucle d'asservissement en vitesse

## 1.2. Problématiques et solutions liées au bruit

La courbe de la figure II.1.2.a représente les mesures télémétriques issues du capteur, pour le cas ci-après.

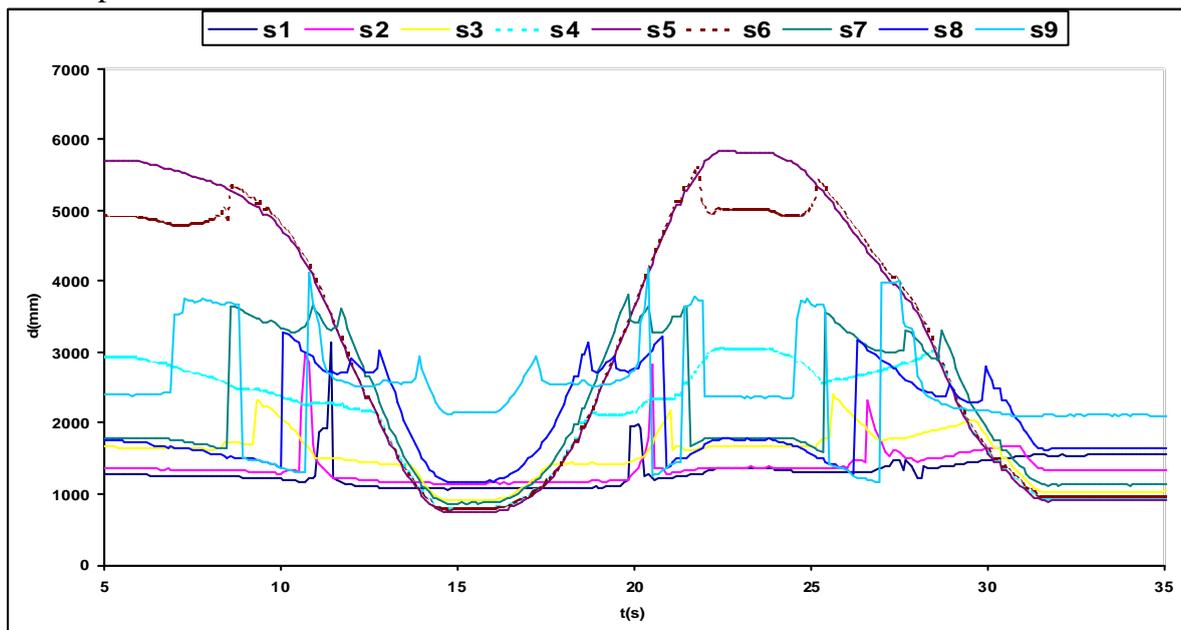
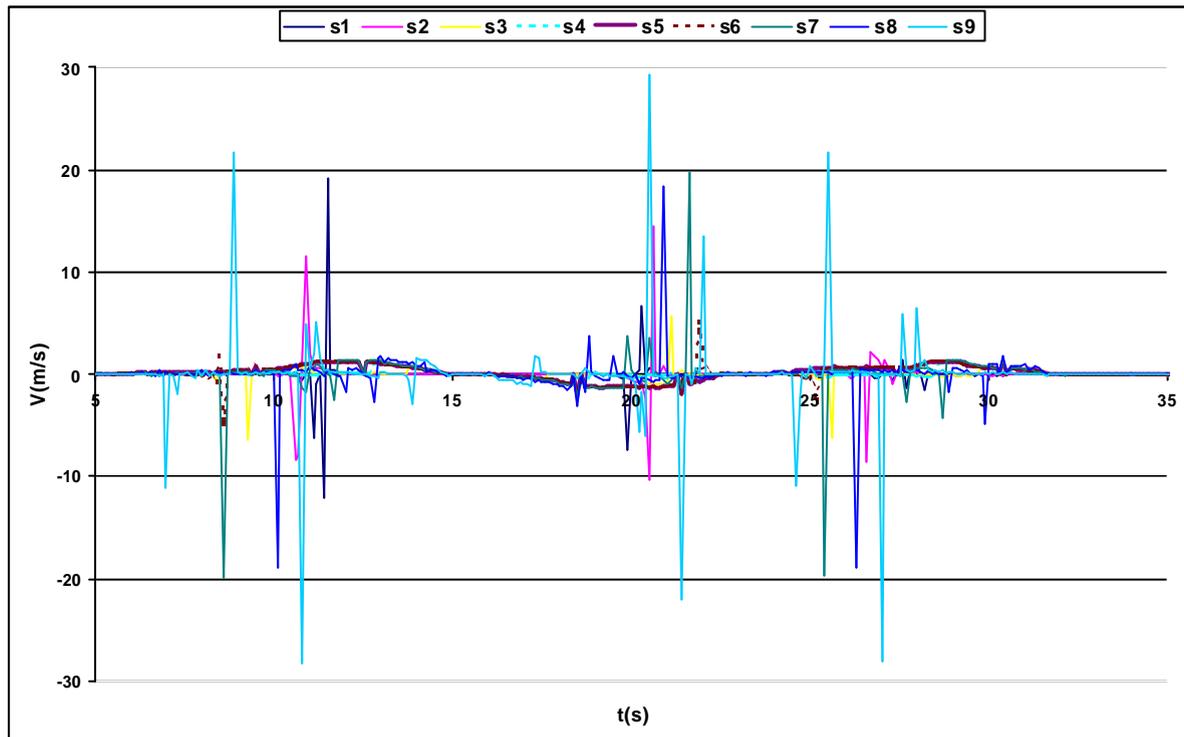


Figure II.1.2.a : Données télémétriques avant filtrage

Ces données ont été obtenues lors d'un déplacement longitudinal (arrière – avant – arrière) du véhicule. La mesure de distance dans chaque secteur est représentée par un tracé. On peut constater que les données issues du segment frontal (s5) sont quasi linéaires alors que les données des autres segments présentent des discontinuités importantes.

Les discontinuités sont dues à la détection d'obstacles ou de trous (portes, cloisons) dans les parties latérales au véhicule.

La figure II.1.2.b montre l'influence de ces discontinuités lorsque l'on considère les vitesses issues des données télémétriques.



**Figure II.1.2.b : Vélocities avant filtrage**

Pour permettre une exploitation efficace des données, il est nécessaire de procéder à une mise en forme des données. Il faut isoler les signaux utiles puis procéder à un filtrage afin d'adoucir la consigne.

Le filtrage est réalisé en deux étapes :

- Elimination des mesures inutiles ;
- Elimination du bruit : un lissage des données est obtenu par adjonction d'un filtre passe bas.

### 1.2.1. Utilisation sélective des données

L'utilisation sélective des données par limitation géométrique du champ de surveillance permet de ne considérer que l'espace utile pour la fonction d'anti-collision.

Une solution simple consiste à fixer une distance maximum de prise en compte des mesures pour chaque secteur :

N° de secteur	Distance maximale de prise en compte des mesures	Commentaires
1 et 9	70 cm	Prise en compte d'une zone de sécurité de 10 cm de chaque coté du véhicule
2 et 8	75 cm	Prise en compte d'une zone de sécurité de 10 cm de chaque coté du véhicule
3 et 7	157 cm	Prise en compte d'une zone de surveillance de 70 cm de chaque coté du véhicule. Cette zone couvre les obstacles se trouvant dans la trajectoire du rayon de braquage du véhicule
4 et 6	240 cm	Continuité de la zone surveillée par les segments 3 et 7
5	Pas de limitation	La limitation dépendra de la distance d'arrêt $da$ du véhicule

Figure II.1.2.1 a : Distances maximales de prise en compte des mesures

La figure II.1.2.1.b schématise l'implémentation des zones de limitation **Z**, la zone cible lors d'une trajectoire à rayon de courbure maximal **T** et l'impact de la présence d'un obstacle de type mur **M**.

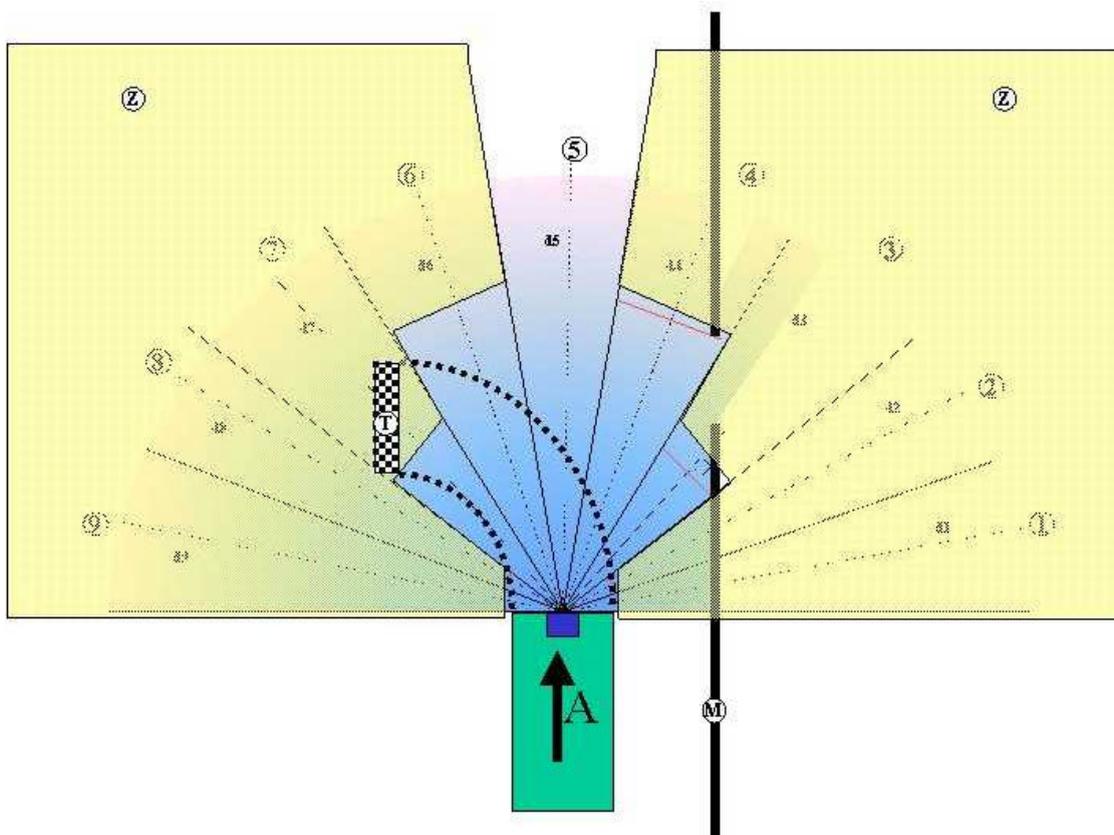
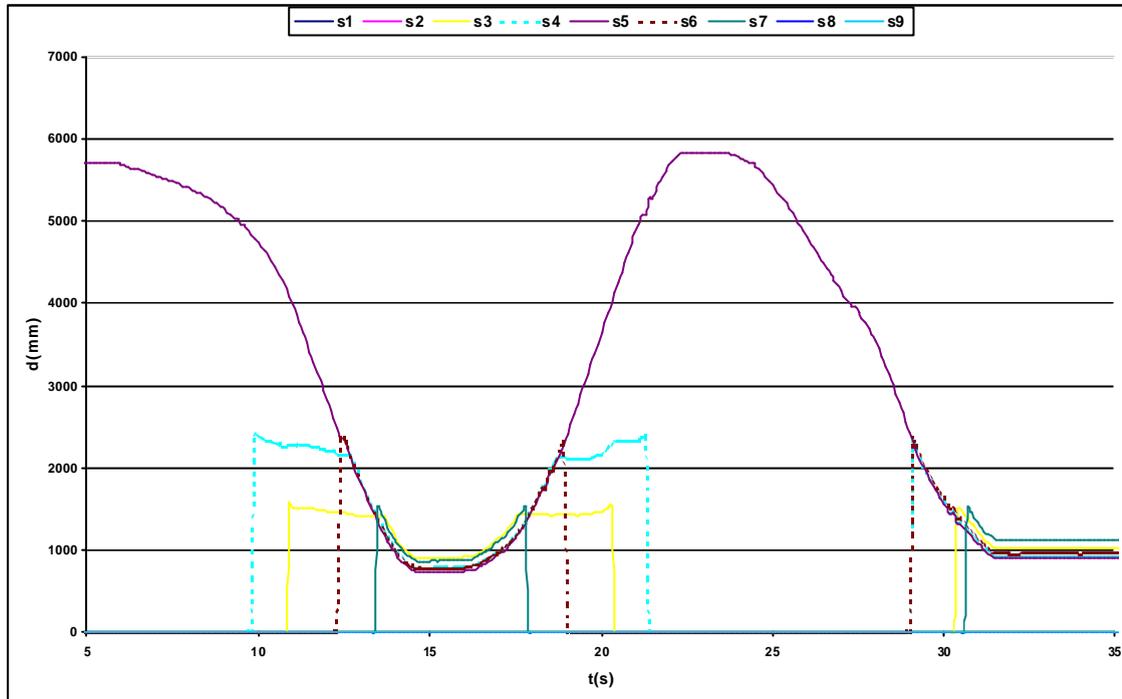


Figure II.1.2.1.b : Limitation du champ de surveillance

La prise en compte sélective des valeurs de mesure est réalisée par un algorithme de traitement des données. Le traitement des données est réalisé soit par l'interface capteur, soit par l'application principale.

Le résultat de cette limitation est visible sur la figure II.1.2.1.c.



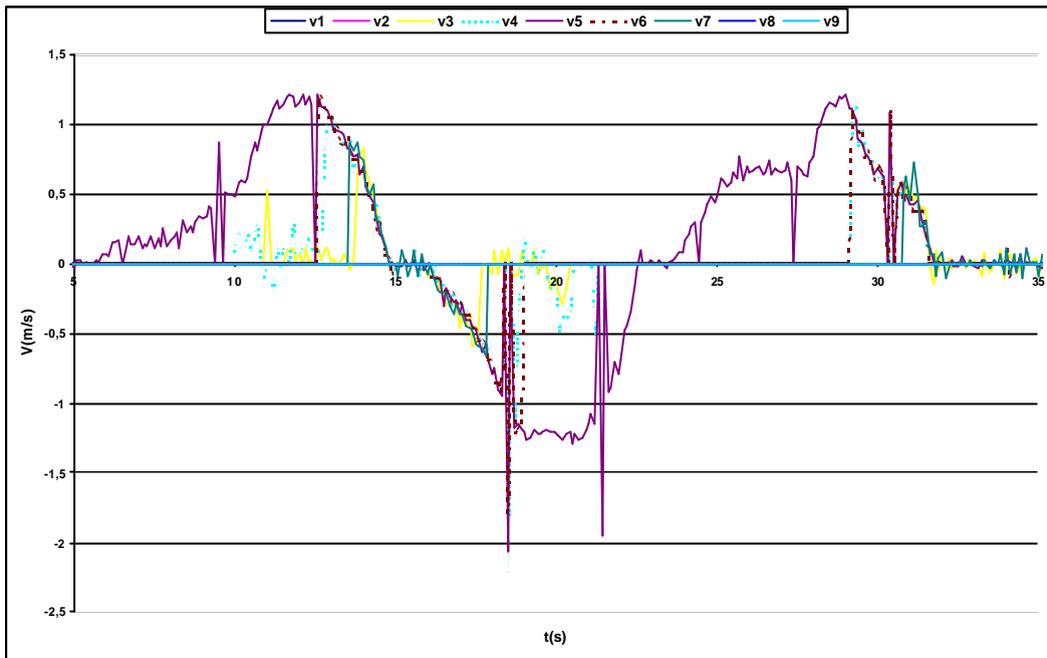
**Figure II.1.2.1.c : Données télémétriques après limitation du champ de surveillance**

Remarques :

- Une mesure est considérée comme égale à zéro lorsqu'elle dépasse la zone de surveillance ;
- Etant donné que les valeurs de mesure issues du capteur sont exprimées en millimètre et que la résolution du capteur est de l'ordre du cm, un pré-filtrage du bruit de mesure a été réalisé en utilisant la formule suivante : nouvelle valeur =  $10 * \text{abs}(\text{mesure} / 10)$ .

### 1.2.2. Elimination du bruit

Comme on peut le constater sur la figure II.1.2.2.a, les informations de vitesse ont besoin d'être filtrées avant exploitation.

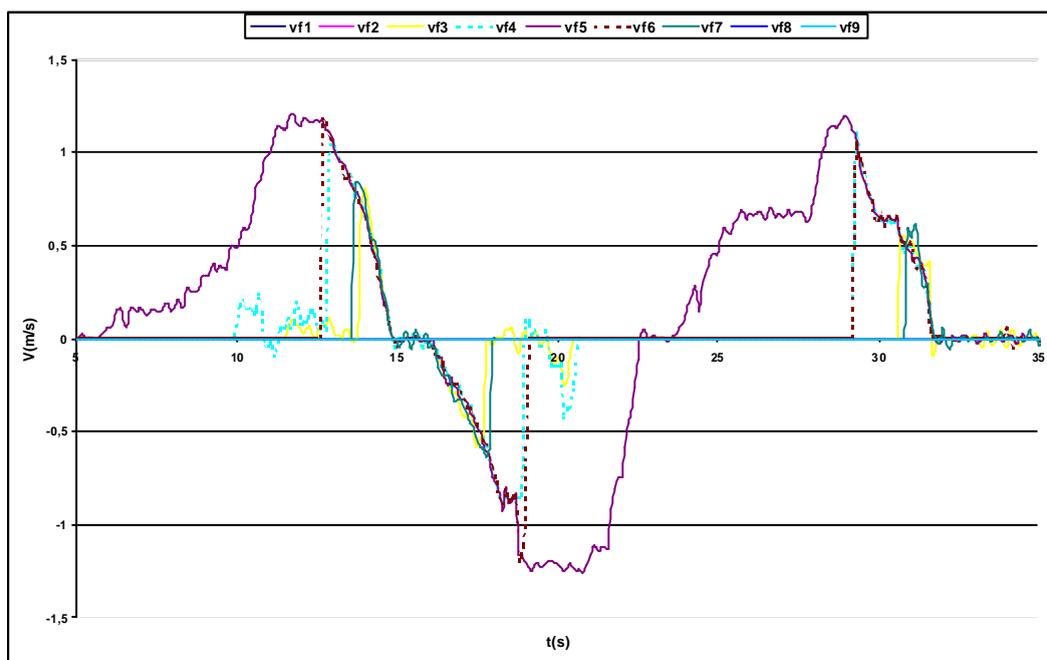


**Figure II.1.2.2.a : Vélocities après limitation du champ de surveillance**

Un lissage des données est obtenu par adjonction d'un filtre de moyenne. Les équations du filtre s'écrivent :

$$\text{Si } |e_k - e_{k-1}| < T \quad \text{alors } s_k = \frac{(e_{k-1} + e_k)}{2}$$

$$\text{sinon } s_k = \frac{(s_{k-3} + s_{k-2})}{2} \quad \text{avec } T = 0.2$$



**Figure II.1.2.2.b : Vélocities après filtrage**

### 1.3. Résultats

Des expérimentations ont été menées sur le Cycab dans l'environnement de la halle robotique et dans un environnement de parking.

Lors de la phase de mise au point de l'anti-collision sur le Cycab, nous n'avons eu qu'une seule collision (sans aucune gravité) due à un problème de signe. Cette fonction d'anti-collision, ou ce comportement, peut être adjoint à n'importe quelle mission confiée au Cycab.

Il faut toutefois déjà noter que l'anti-collision fait appel à une configuration spéciale (9 secteurs) du télémètre laser et que si la mission a besoin du télémètre pour une autre application (suivi, localisation) ; il sera alors nécessaire de reconfigurer les paramètres internes du capteur.

Nous disposons maintenant d'une tâche robot d'anti-collision. Nous n'avons pas pu mettre en défaut l'anti-collision lors de l'expérimentation de conduite dans l'environnement de la halle robotique et du parking extérieur.

## 2. SUIVI DE VEHICULE

Lors d'une utilisation en libre service d'un parc de véhicules individuels, il se pose le problème du rapatriement de ces véhicules vers les zones de plus forte demande. Un système de récupération par train de véhicules a été imaginé. Le projet californien PATH ainsi que le projet national PRAXITELE ont exploités ce concept.

Le concept de train de véhicules à accrochage immatériel reprend le principe du train classique, à savoir le fait que le véhicule de tête dirige, tandis que les autres sont asservis sur la trajectoire de leur prédécesseur.



**Figure II.2.a : Suivi de véhicules**

Pour qu'une telle mission soit réalisable, il faut pouvoir maîtriser un certain nombre d'informations. Il faut connaître la position relative du véhicule à suivre, son orientation et les vitesses relatives entre les deux véhicules.

$x$  et  $y$  : Coordonnées cartésiennes du véhicule suivi, relatives au véhicule commandé

$d$  : Distance entre les deux véhicules

$q$  : Angle entre les axes longitudinaux de chaque véhicule

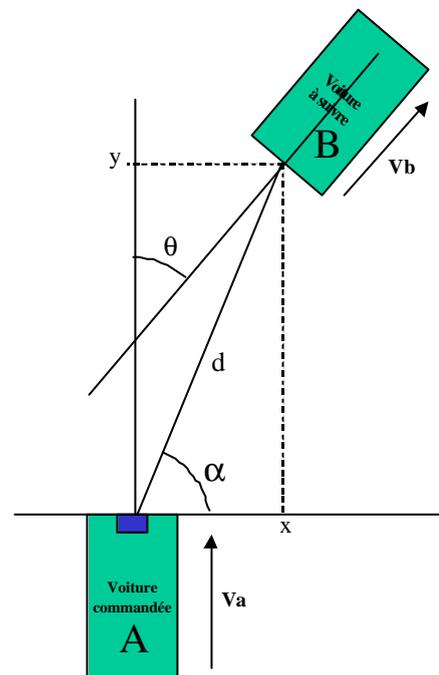
$\mathbf{a}$  : Position angulaire du véhicule suivi par rapport au véhicule commandé

$$x = d \cos(\mathbf{a})$$

$$y = d \sin(\mathbf{a})$$

$V_a$  : vitesse propre du véhicule A

$V_b$  : vitesse propre du véhicule B



**Figure II.2.b : Cinématique du système de véhicules**

### 2.1. Solution technique à base de caméra linéaire

Une solution intégrant une caméra linéaire a été mise en œuvre et expérimentée par le projet SHARP<sup>6</sup> [26] avec l'assistance technique du service Robotique de l'INRIA Rhône Alpes. Une caméra linéaire positionnée sur le véhicule à commander visionne un système de trois points lumineux sur le véhicule à suivre. Les points lumineux émettent une lumière infra rouge pulsée et sont disposés de manière à récupérer (par triangulation) la distance, le décalage latéral et l'angle entre les deux véhicules.

En pratique, cette solution a permis d'obtenir des résultats concluants. La commande permet un asservissement robuste du véhicule suiveur à des vitesses nominales. Néanmoins, les problèmes suivants tendent à limiter la solution à des conditions opérationnelles restrictives :

- Le système cible de détection est un système actif. Il est donc nécessaire d'équiper les véhicules avec ce système qui a besoin d'énergie pour les diodes électroluminescentes. D'autre part, ce système présente l'inconvénient d'être influencé par l'ensoleillement, principalement lorsqu'il est à l'horizon, dans le plan de la caméra. Cet effet provoque la perte de la cible par la caméra ;
- Le système de détection à base de caméra linéaire est limité au niveau de l'angle de détection  $\alpha$  de la cible (de  $-20^\circ$  à  $+20^\circ$ ) et de l'angle d'orientation  $\theta$  de la cible (maximum  $45^\circ$ ) ;
- La portée de ce système est de 8m.

### 2.2. Solution technique à base de capteur télémétrique à balayage laser

L'utilisation d'un capteur télémétrique à balayage laser introduit des possibilités plus larges en terme de :

- Angle de détection d'une cible ( $180^\circ$ );
- Portée du capteur;
- Insensibilité aux éclairages parasites, en particulier le soleil;
- Passivité de la cible (à base de catadioptrés);
- Possibilité d'utiliser les données du capteur pour des applications parallèles (Anti-collision, cartographie, positionnement).

Le suivi de véhicule à base de capteur télémétrique à balayage laser est exposé au Chapitre III.

---

<sup>6</sup> SHARP : Projet mené au sein de l'UMR GRAVIR (CNRS, INPG, INRIA et UJF) traitant de programmation automatique et systèmes décisionnels en robotique.

### 3. LOCALISATION

On entend par "système de localisation" l'ensemble constitué par les capteurs et les techniques, qui permettent au véhicule de naviguer de manière autonome ou semi-autonome dans son environnement. Classiquement, on distingue deux types de solutions :

- La localisation relative qui permet au véhicule de naviguer à l'estime ("dead reckoning"), en utilisant uniquement les mesures des mouvements propres du robot fournies par ses capteurs proprioceptifs<sup>7</sup> ;
- La localisation absolue qui fait appel aux mesures des capteurs extéroceptifs<sup>8</sup> pour estimer la situation du véhicule dans un repère lié à l'environnement.

#### 3.1. Problématique

L'odométrie<sup>9</sup> reste la solution la plus courante et la plus facile à mettre en œuvre pour mesurer les variations de position et de cap d'un robot terrestre navigant sur un sol plan. On utilise généralement des codeurs optiques incrémentaux montés sur l'axe de rotation des roues. Ceux-ci sont plus faciles à interfacer et fournissent des mesures dont la résolution est équivalente à celle des codeurs absolus, pour un coût beaucoup plus faible. Les codeurs absolus sont préférables pour mesurer des mouvements de rotation effectués à faible vitesse ou peu fréquemment, comme par exemple la rotation d'un volant ou le changement d'orientation d'une roue. L'inconvénient majeur de l'odométrie est la dérive importante des mesures liée au processus d'intégration des vitesses des roues qui est mis en œuvre pour estimer les déplacements du véhicule dans un repère cartésien. Cette accumulation d'erreurs trouve son origine principalement dans les phénomènes de glissement, dans celle des paramètres du modèle géométrique du robot, et dans les phénomènes de dérapage qui ne sont pas pris en compte dans l'estimation.

Les erreurs sont classées en deux groupes :

##### Erreurs systématiques

- Diamètre inégal des roues ;
- Variation du diamètres des roues dans le temps ;
- Non-alignement des roues ;
- Résolution des encodeurs.

##### Erreurs non systématiques

- Déplacement sur surfaces non plane ;
- Glissement des roues ;
- Sol glissant ;
- Sur-accélération ;
- Rotation rapide (ensabotage) ;
- Forces externes (interaction avec des objets externes) ;
- Contact roue-sol non ponctuel ;

---

<sup>7</sup> Capteur proprioceptif : capteur qui mesure des signaux internes au robot.

<sup>8</sup> Capteur extéroceptif : Organe sensoriel qui mesure des grandeurs relatives caractérisant la relation d'un système, ici le véhicule avec son environnement.

<sup>9</sup> Odométrie : Instrument constitué par une roue et un compte-tours dans le but de mesurer la distance parcourue.

La figure II.3 présente le modèle géométrique idéal d'un véhicule à quatre roues directrices en comparaison avec la géométrie réelle du véhicule Cycab. Cette géométrie introduit des effets de glissement car l'axe de rotation des trains avant et arrière ne sont pas concourants en un même point.

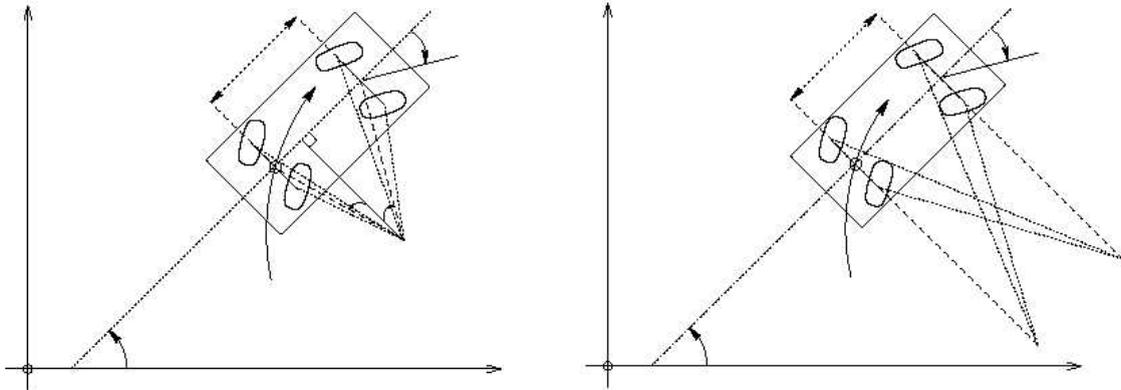


Figure II.3 : Le modèle idéal du Cycab (gauche) et le modèle actuel (droite)

### 3.2. Solutions techniques

Une solution mécanique à ce problème consiste à monter les codeurs sur des roues "odométriques" situées sur l'axe des roues motrices, et moins sujettes aux perturbations mécaniques que celles-ci.

La réduction des erreurs non systématiques est beaucoup plus délicate. Ces erreurs fluctuent en fonction de l'environnement et des interactions non connues avec celui-ci.

La navigation inertielle utilise des gyromètres<sup>10</sup>, des gyroscopes<sup>11</sup> ou parfois des accéléromètres pour mesurer les vitesses de rotation ou les accélérations linéaires du véhicule. Ces mesures sont ensuite intégrées une ou deux fois pour obtenir les variations de position ou d'attitude. Ici encore, le processus d'intégration favorise la croissance des erreurs au cours du temps et il est nécessaire d'effectuer des recalages absolus périodiques. Par ailleurs, il faut aussi souligner le coût élevé de ces systèmes lorsque l'on veut obtenir une bonne précision.

Il n'existe pas, à l'heure actuelle, de méthode simple et suffisamment performante pour fournir une estimation précise et fiable de la position et du cap du robot pendant toute la durée de l'exécution d'une tâche. C'est la raison pour laquelle on fait généralement appel à la coopération de ces deux types de solutions :

- Réduction des erreurs d'odométrie par recalage ;
- Réduction des erreurs d'odométrie par modélisation et fusion de données.

### 3.3. Réduction des erreurs d'odométrie par recalage

Une solution consiste à effectuer des recalages périodiques absolus à l'aide de mesures extéroceptives.

<sup>10</sup> Gyromètre : Appareil servant à mesurer une vitesse de rotation.

<sup>11</sup> Gyroscopie : Appareil, qui animé d'un mouvement de rotation autour d'un de ses axes, peut être placé d'une manière quelconque sans que la direction de son axe soit modifiée.

On s'intéresse aux capteurs mis en œuvre pour la localisation absolue sur l'environnement, à partir de différentes méthodes que l'on peut classer de la manière suivante :

- localisation sur balises artificielles ;
- localisation par reconnaissance de repères naturels ;
- mise en correspondance de modèles.

Pour toutes ces méthodes, la connaissance d'un modèle a priori décrivant la situation des balises ou la topologie de l'environnement, ou même des modèles d'objet que l'on peut rencontrer dans cet environnement, est indispensable. Bien entendu, les performances de la localisation sont étroitement liées à la précision de ce modèle. Les systèmes extéroceptifs utilisés pour la localisation absolue peuvent également être sollicités en ligne pour mettre à jour le modèle de l'environnement lorsque celui-ci est imprécis, évolutif, ou partiellement connu.

Dans le cas d'une localisation 2D, les deux premières solutions exploitent généralement la mesure du gisement et/ou de la distance de  $N$  balises reconnues dans l'environnement. En utilisant uniquement la mesure d'angle relatif entre balises, on doit avoir  $N=3$ . Si les balises sont repérées par leur position angulaire et leur distance, on prend  $N=2$ . L'estimation de la position et du cap est basée soit sur une simple triangulation géométrique, soit sur la minimisation d'un critère pour estimer la configuration optimale du robot par rapport aux balises. Les balises artificielles, qui peuvent être passives ou actives, doivent être placées en des points connus de l'environnement. Lorsqu'elles sont actives, elles peuvent émettre un signal lumineux (lumière visible, laser), ou sonore (ultrasons, signal radio).

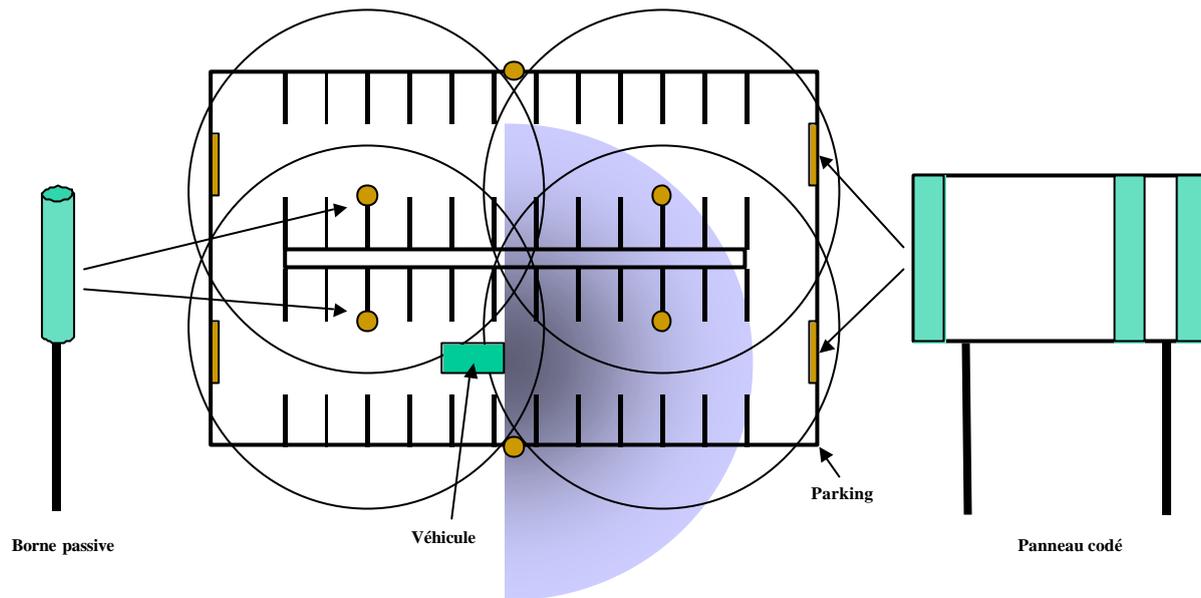
L'inconvénient majeur des techniques de localisation sur balises naturelles est qu'elles nécessitent un traitement préalable plus ou moins long des données capteurs pour les extraire et pour les mettre en correspondance avec un modèle connu a priori. Ainsi, l'exploitation des images vidéo ou des cartes obtenues par télémétrie laser ne permet pas à l'heure actuelle un recalage en temps réel du véhicule. Par ailleurs les temps d'acquisition ou de traitement des données soulèvent le problème de l'asynchronisme des mesures pour un véhicule en mouvement. Il n'existe pas de système autorisant la mesure simultanée et l'estimation en temps réel de la position de plusieurs balises.

Le GPS (Global Positioning System) est de plus en plus utilisé pour la navigation 3D en environnement extérieur. Il utilise la mesure du temps de propagation d'ondes radio émises par  $N$  satellites dont la position orbitale est connue. Ces signaux permettent d'estimer par triangulation la configuration (latitude, longitude et altitude) du mobile équipé d'un récepteur. Le GPS différentiel fournit ces mesures avec une précision de quelques mètres pour les systèmes les plus courants. On trouve ce système de localisation sur la moitié des véhicules d'expérimentation terrestre d'extérieur.

Les télémètres laser à balayage fournissent des cartes beaucoup plus précises que les ceintures de capteurs ultrasonores.

Le schéma de la figure II.3.3 illustre un système de localisation par balises passives, installé dans un parking.

Nous avons étudié un système de localisation par balises passives pour l'installer dans un parking. Ce système est décrit sur la figure II.3.3.



**Figure II.3.3 : Localisation par balises passives**

Le capteur utilisé est un télémètre à balayage laser. Les balises passives sont constituées de cylindres recouverts de réflecteurs de type catadioptre. La portée de détection des balises est de 20 mètres. Des panneaux codés permettent de définir une localisation initiale. La précision de localisation est fonction de la période de rafraîchissement des données. Un moyennage de plusieurs mesures permet d'augmenter la précision au détriment de la fréquence d'échantillonnage des données.

*Nombre de mesures pour une localisation :  $n$*

*Période (en s) :  $T=0,05n$*

*Précision (en m) :  $P=0,02/n$*

La précision est  $P$  dans le cas d'un véhicule immobile. La précision réelle en exploitation dynamique sera variable en fonction de la vitesse propre du véhicule.

### 3.4. Réduction des erreurs d'odométrie par apprentissage et fusion de données

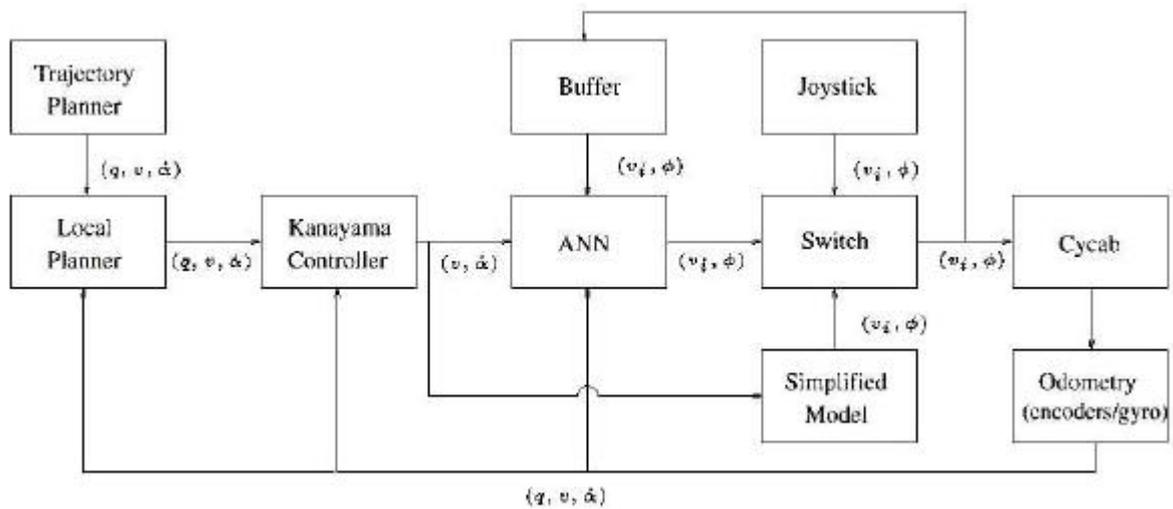
Une autre solution consiste à approcher l'erreur par un modèle, ce qui permettra de prédire son influence sur le système. La modélisation est complexe du fait du nombre important de paramètres impliqués et de leur variations aléatoires.

Avec un filtre de Kalman (estimateur récursif) on obtient une erreur de approximativement 10%.

Un système à base de réseau de neurones artificiels [19] développé par le projet SHARP de l'INRIA Rhône Alpes permet actuellement de réduire cette erreur à 5%.

Le principe est basé sur l'apprentissage des comportements erratiques lorsque le véhicule évolue dans l'environnement d'exploitation. La mise au point de cette méthode nécessite un grand nombre de manipulations pendant lesquelles le réseau de neurones fait l'acquisition des

informations des capteurs proprioceptifs ainsi que des capteurs extéroceptifs de localisation réelle.



**Figure II.3.4 : Principe d'une commande à base de réseau de neurones**

Les résultats obtenus peuvent théoriquement être améliorés par la prise en compte de capteurs inertiels. Les informations de positionnement réel obtenues à l'aide du capteur télémétrique à balayage laser, permettent d'améliorer la précision et les temps de manipulation lors des phases d'apprentissage du modèle.

#### 4. CONCLUSION

Disposer de données sur l'environnement à partir de capteurs extéroceptifs apparaît comme un besoin incontournable dans le développement et la mise au point d'applications multiples. Le capteur télémétrique à balayage laser se positionne comme un outil très adapté.

Les mesures réalisées par le capteur offrent des possibilités d'application multiples. On peut concevoir une exploitation simultanée des données par plusieurs applications. Le chapitre suivant propose une description détaillée d'une application de suivi de véhicule.

# Chapitre III

## SUIVI DE VEHICULE

*L'objectif des travaux décrits dans ce chapitre est d'utiliser le capteur à balayage laser dans une application de suivi de véhicule. Le capteur est intégré en remplacement d'une caméra linéaire précédemment utilisée pour effectuer ce suivi. Après un descriptif de la plateforme expérimentale, nous détaillons le système de commande pour aboutir ensuite à des considérations qualitatives.*

## 1. OBJECTIFS

Une solution intégrant une caméra linéaire a été mise en œuvre et expérimentée par le projet SHARP<sup>12</sup> [26] avec l'assistance technique du service Robotique de l'INRIA Rhône-Alpes. Une caméra linéaire positionnée sur le véhicule à commander visionne un système de trois points lumineux sur le véhicule à suivre. Les points lumineux émettent une lumière infrarouge pulsée et sont disposés de manière à récupérer (par triangulation) la distance, le décalage latéral et l'angle entre les deux véhicules.

En pratique, cette solution a permis d'obtenir des résultats concluants. La commande permet un asservissement robuste du véhicule suiveur à des vitesses nominales. Ce système comporte cependant quelques faiblesses au niveau portée, résistance aux rayons solaires et limites des angles de détection de la cible.

Le remplacement de la caméra linéaire par un capteur télémétrique à balayage laser devrait permettre de réduire les limites fonctionnelles imposées par la caméra linéaire. La réalisation de l'application de suivi avec le capteur télémétrique est basée sur l'utilisation des mêmes lois de commande que l'application de suivi à base de caméra, ce qui permettra un comparatif expérimental.

---

<sup>12</sup> SHARP : Projet mené au sein de l'UMR GRAVIR (CNRS, INPG, INRIA et UJF) traitant de programmation automatique et systèmes décisionnels en robotique.

## 2. LE CYCAB

### 2.1. Introduction

L'INRIA a imaginé un système de transport original de véhicules en libre-service pour la ville de demain. Ce système de transport public est basé sur une flotte de petits véhicules électriques spécifiquement conçus pour les zones où la circulation automobile doit être fortement restreinte.

Pour tester et illustrer ce système, deux prototypes, nommés Cycab, ont été réalisés. Le premier est localisé à l'INRIA de Rocquencourt et le second est localisé à l'INRIA Rhône-Alpes.

Depuis janvier 1998, le Service Robotique est en charge du Cycab de l'INRIA Rhône-Alpes. Ce véhicule a servi de plate-forme expérimentale pour les travaux de ce mémoire. Ce chapitre présente une description de l'architecture matérielle et logicielle du véhicule.

### 2.2. Présentation du Cycab

Le Cycab est un prototype de petit véhicule électrique devant servir de plate-forme expérimentale pour les équipes de robotique et vision de l'Unité de Recherche Rhône-Alpes. Il a été construit sur la base d'un châssis tubulaire Andruet raccourci qui supporte une coque conçue par Sylvain Fauconnier et dont les moules appartiennent à l'Inria.

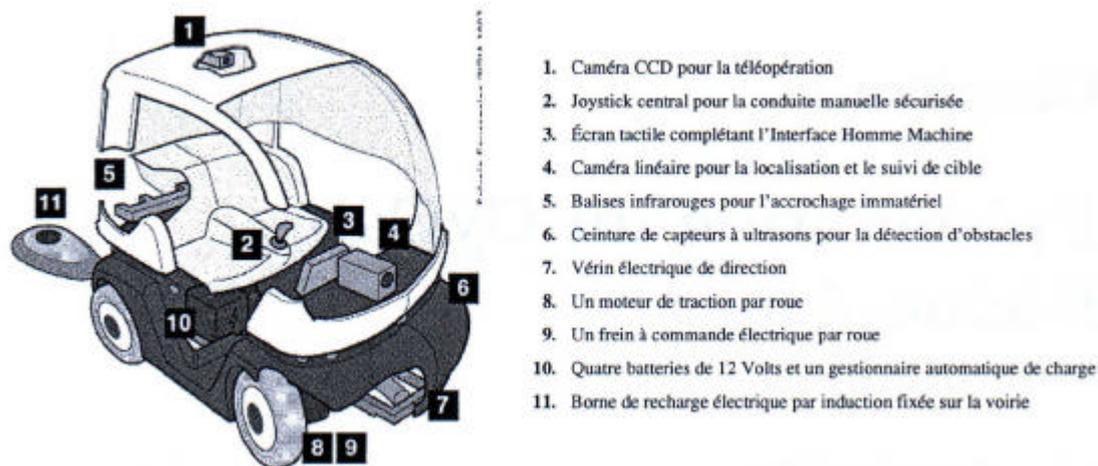


Figure III.2.2 : Vue éclatée du Cycab

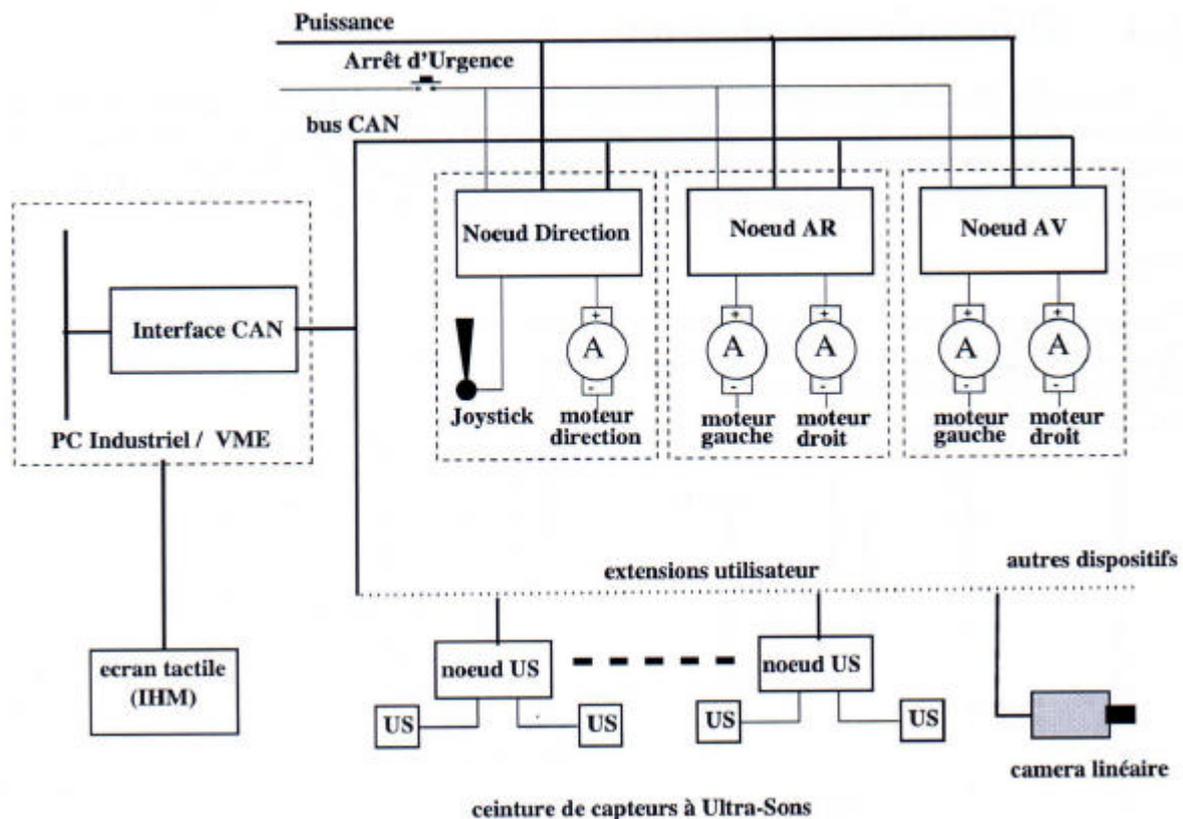
#### 2.2.1. Caractéristiques générales

Les caractéristiques générales du véhicule sont les suivantes :

- Longueur hors tout : 1,90m ;
- Largeur hors tout : 1,20 m ;
- Poids total avec batteries : 300 kg ;
- motorisation : 4 moteurs électriques de 1 kW ;
- 4 roues motrices et directrices ;

- vitesse théorique maximale : 20 km/h ;
- autonomie : 2 heures d'utilisation continue ;
- capacité d'accueil : 2 personnes avec bagages ;
- conduite automatique ou manuelle ;
- recharge automatique par induction.

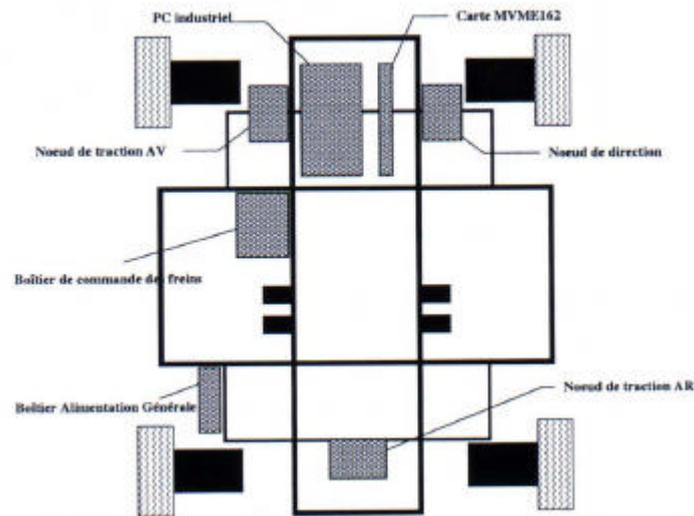
Pour contrôler et commander ces 9 moteurs (4 de traction, 1 de direction et 4 de frein), une architecture matérielle distribuée a été choisie (figure III.2.2.1). Elle est constituée de 3 nœuds intelligents pouvant gérer chacun 2 moteurs. Le premier, appelé nœud de direction, gère le joystick et contrôle le vérin motorisé de direction. Deux nœuds de traction, un par train, contrôlent deux moteurs de traction. La commande de frein de parking est unique pour les quatre moteurs. Un PC industriel et une carte Motorola MVME162 sous VxWorks complètent cette architecture matérielle. Les trois nœuds, le PC et la carte Motorola communiquent entre eux via le bus de terrain CAN (Controller Area Network). Sur ce même bus CAN seront connectés les capteurs ultrasonores ainsi que la caméra linéaire.



**Figure III.2.2.1 : Synoptique de l'architecture matérielle de Contrôle/Commande**

### 2.2.2. Le châssis

Le châssis du Cycab est dérivé du châssis tubulaire équipant les voiturettes de golf commercialisées par la société ANDRUET SA. Ce châssis supporte les 4 batteries, les 4 blocs roue (roue + moteur + frein mécanique à tambour), les 4 moteurs de frein de parking, les biellettes de direction les reliant mécaniquement entre elles, le vérin motorisé et les boîtiers électroniques.



**Figure III.2.2.2 : Architecture matérielle : Implémentation sur le châssis**

### 2.2.3. Alimentation générale

Toutes les alimentations nécessaires au fonctionnement du Cycab (puissance pour les moteurs, alimentation des nœuds, du PC, de la carte MVME162, des capteurs) sont fournies par le boîtier d'alimentation générale à partir des batteries embarquées.

Le véhicule embarque 4 batteries de 12 Volts qui lui donnent en théorie une autonomie de 2 heures à la vitesse maximale de 20 km/h sur des pentes n'excédant pas 10%. Un relais statique sur la boucle d'arrêt d'urgence met le véhicule dans l'état freiné s'il n'est pas alimenté. En série avec le secondaire de ce relais statique est monté un interrupteur de commande manuelle des freins de parking.

### 2.2.4. Les moteurs de traction

La chaîne de traction se compose de quatre "moteur-roues" contrôlables séparément. Les moteurs de traction de type MP 8165/Caddy sont fournis par Andruet. Ils sont modifiés par rapport à ceux équipant les voiturettes de golf :

- Le rotor a été bobiné de façon à fonctionner en 48 Volts pour limiter l'intensité du courant dans les transistors de puissance (à puissance égale) ;
- L'axe du rotor a été prolongé de façon à pouvoir monter un codeur optique incrémental pour les calculs d'odométrie ;
- Une thermistance a été rajoutée pour mesurer la température à l'intérieur des moteurs.

### 2.2.5. Le vérin motorisé de direction

Pour actionner les biellettes de direction, la crémaillère et le volant ont été remplacés par un vérin électrique. Celui-ci est aussi équipé d'un codeur optique incrémental pour le contrôle du braquage. Il lui est associé un capteur de déplacement rectiligne pour pouvoir lire la position initiale du vérin et deux capteurs inductifs pour les fin de course.

### 2.2.6. Les freins de parking

Chaque bloc roue du Cycab est pourvu d'un frein à tambour mécanique commandable électriquement : un moteur de type essuie-glace de voiture tire ou relâche le câble. Les 4 moteurs de frein sont situés sous le plancher ainsi que le boîtier contenant les relais de commande.

Les moteurs ont tous la même commande venant soit d'une sortie Tout ou Rien du nœud de direction, soit d'un bouton poussoir situé sur l'armoire d'alimentation générale, soit de l'arrêt d'urgence et à travers une logique à relais.

### 2.2.7. L'Interface Homme Machine

Le conducteur communique avec le véhicule à l'aide d'un joystick de commande centralisée situé entre les deux sièges ce qui permet une prise en main facile à la fois pour les droitiers et les gauchers, et d'un écran tactile intégré dans la console. Sur le sommet de cette console se trouve un bouton poussoir (type "coup de poing") d'arrêt d'urgence et trois diodes électroluminescentes renseignant sur l'état du véhicule.

Le joystick de commande : En mode de conduite manuelle, le conducteur pilote le Cycab au joystick qui est vu par le système matériel de contrôle/commande comme deux potentiomètres dont la lecture analogique par le nœud de direction simule l'accélérateur et le volant d'un véhicule normal. Il n'y a aucun lien mécanique entre l'organe de commande manuelle (le joystick) et les actionneurs (moteurs de traction et vérin de direction).

L'écran tactile : L'écran LCD tactile sert d'écran de contrôle au PC embarqué. Des touches de fonction programmables permettent de charger, démarrer les applications.

### 2.2.8. La boucle d'arrêt d'urgence

En cas de problème grave ou de danger survenant pendant l'utilisation du Cycab, il est impératif de pouvoir agir rapidement pour bloquer le véhicule en inhibant les commandes des moteurs de traction et de direction et en actionnant les freins de parking. Pour ce faire le Cycab dispose d'un mécanisme d'arrêt d'urgence pouvant être actionné soit par l'interrupteur coup de poing situé au dessus de la console, soit par une télécommande radio (série 400) dont le récepteur est monté dans le rack VME et qui actionne un relais normalement fermé. Cette information ARRÊT d'URGENCE doit être transmise à tous les nœuds de façon à inhiber les commandes des moteurs. Elle doit donc être chaînée, c'est le rôle de la boucle d'arrêt d'urgence dont l'état ouvert est détectée par tous les nœuds.

## 2.3. Architecture matérielle de contrôle/commande

### 2.3.1. Description fonctionnelle des nœuds

Pour commander et contrôler les différents moteurs du Cycab, la solution adaptée est une architecture répartie autour d'un bus de terrain, le bus CAN (Controller Area Network) très répandu dans le monde de l'automobile. L'INRIA a développé des nœuds "intelligents" (Amplificateurs de Puissance Intelligents). Chaque API pilote, pour l'application qui nous intéresse, deux moteurs à courant continu (par exemple les moteurs de traction des roues avant).

Le rôle des nœuds est d'asservir les moteurs en fonction des consignes de vitesse et de braquage qui transitent sur le bus CAN soit en provenance de l'Interface Homme Machine (la position du joystick), soit calculées par un programme de planification de trajectoires. Il doit donc non seulement être capable de fournir la puissance nécessaire au moteur, mais aussi exécuter les boucles d'asservissement de vitesse ou de position. Pour ce faire il doit prendre en compte un certain nombre d'informations en provenance des capteurs proprioceptifs : état, odométrie, fins de course, mesures de température et de courant.

Cette description succincte fait apparaître trois entités composant les nœuds :

- Un module de puissance avec les transistors Mos-Fet de puissance et leur commande de Gate pour piloter deux moteurs ;
- Un module d'interface et de communication dont le rôle est essentiellement de mettre en forme les signaux Tout ou Rien ou de convertir les signaux analogiques en provenance des capteurs, de l'interface homme/machine (IHM) ou allant vers les indicateurs d'état pour qu'ils soient exploitables par le microcontrôleur. Ce module gère aussi les communications sur le bus CAN ;
- Un module de calcul qui, à partir des consignes et des données proprioceptives, calcule les courants à envoyer aux moteurs.

Cette décomposition se retrouve dans le matériel et le nœud se présente sous la forme de trois cartes images de ces modules fonctionnels et logées dans un même boîtier (Cf. Figure III.2.3.1). Cette conception modulaire donne de la souplesse et permet d'adapter ce type de nœud à d'autres types de moteurs ou à d'autres gammes de puissance en ne changeant par exemple que l'étage de puissance.

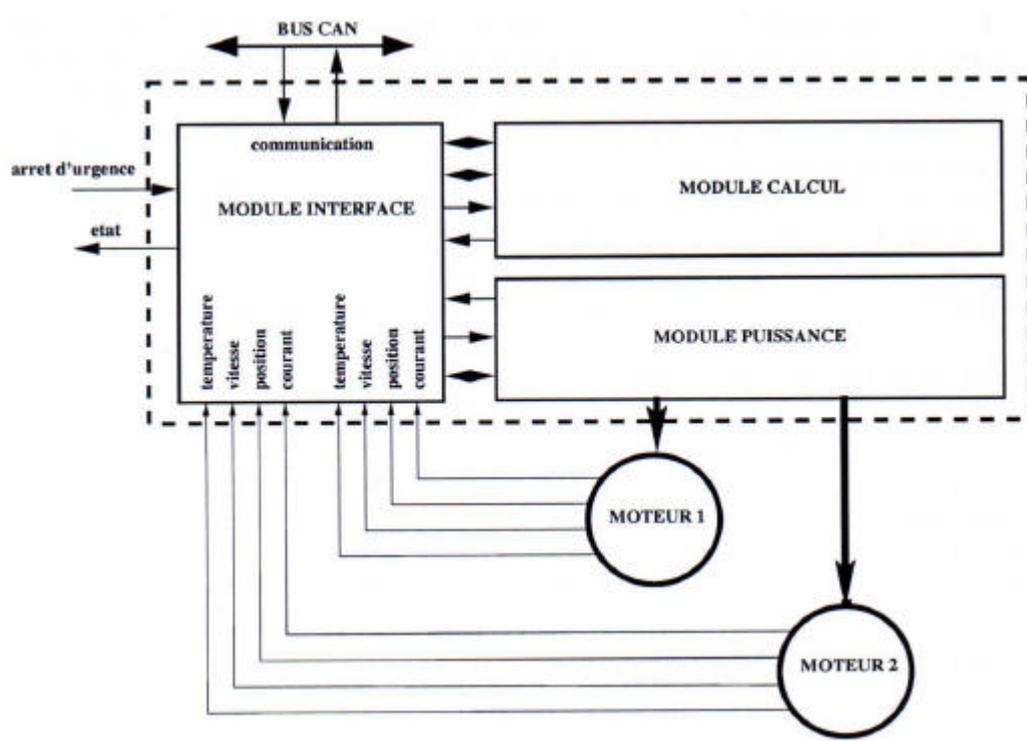


Figure III.2.3.1 : Schéma fonctionnel d'un nœud

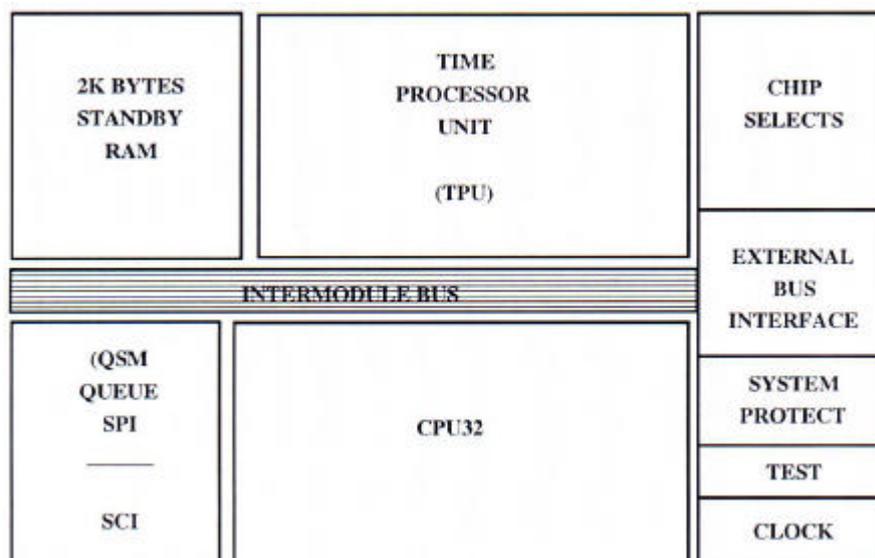
### 2.3.2. L'alimentation des nœuds

Toutes les tensions d'alimentation nécessaires au bon fonctionnement des différentes cartes constituant le nœud sont faites à partir de l'alimentation de 15 volts générée par le boîtier d'Alimentation Générale. Cette tension sert à fabriquer la tension principale  $V_{cc}$  à l'aide d'un circuit régulateur de puissance fournissant une tension variable de 5.1 Volts à 40 Volts pour un courant de 2.5 Ampère. Le circuit est protégé contre les surintensités au moment des commutations et contre les courts-circuits en sortie. Cette tension est utilisée par toutes les cartes.

### 2.3.3. Le module de calcul

Inspirée du module M68332BCC de Motorola, il est bâti autour du microcontrôleur 32 bits MC68332. Il en a les mêmes fonctionnalités mais il est pourvu d'une mémoire Flash ce qui permet de modifier le programme résidant sans être obligé d'ouvrir le nœud pour enlever l'EPROM pour l'effacer avant de la re-programmer.

Le MC68332 est un micro-contrôleur CMOS intégré de 32 bits combinant des possibilités avancées de manipulation de données avec des sous systèmes périphériques puissants. Le cœur opératoire du micro-contrôleur (bloc CPU32) s'apparente au MC68020 et est compatible avec les logiciels de la famille MC68000. Le schéma de la figure III.2.3.3 montre les principaux composants du MC68332.



**Figure III.2.3.3 : Schéma fonctionnel du microcontrôleur MC68332**

Le code pour le CPU32 est entièrement compatible avec le code 68010 et la plupart des extensions 68020 (support de mémoire virtuelle, loop mode opération, instructions "pipelinées", opérations mathématiques sur 32 bits). De nouvelles instructions (table lookup, interpolation, ...) supportent les requêtes spécifiques aux applications de contrôle.

Le MC68332 possède des sous-systèmes périphériques autonomes qui augmentent les performances du CPU pour les applications de contrôle/commande.

Le TPU (Time Processor Unit) contrôle les activités dépendant du temps en réduisant de façon importante l'intervention du CPU avec une unité d'exécution dédiée, un séquenceur à trois niveaux de priorité, une RAM de stockage, et une ROM micro-codée. Le TPU contrôle 16 canaux indépendants, chacun d'eux étant associé à une broche E/S.

Parmi les algorithmes micro-codés disponibles, on trouve :

- Entrées/Sorties discrètes ;
- Mesure de période avec détection de transition ;
- Génération d'impulsions synchronisées ;
- Contrôle de moteurs pas à pas ;
- Entrées comptage (capture) ;
- Correspondance de sortie (output match) ;
- Sorties périodiques à largeur d'impulsion modulée.

Le QSM (Queued Serial Module) contient deux ports série. Le QSPI permet des extensions périphériques aisées ou des communications inter processeur via un bus full-duplex synchrone (data in, data out et horloge série). Quatre broches programmables de sélection de périphériques permettent d'adresser jusqu'à 16 périphériques. Une queue contenue dans une petite RAM permet au QSPI de gérer 16 transferts série de 8 ou 16 bits chacun ou de transmettre un flot de 256 bits sans l'intervention du CPU. Un mode spécial permet à l'utilisateur de faire de façon continue l'échantillonnage d'un périphérique série en mettant automatiquement à jour la RAM QSPI pour une interface efficace avec des convertisseurs Analogiques Digitaux série par exemple.

Le SIM (System Integration Module) inclut une interface externe et des fonctions qui réduisent la glu logique autour du boîtier.

#### 2.3.4. La carte d'interface

Le rôle de cette carte est d'interfacer la carte processeur avec le monde extérieur : liaison série RS232, communication par bus CAN (Controller Area Network), conversion d'entrées analogiques, mise en forme et isolation galvanique des signaux Tout ou Rien aussi bien en entrée qu'en sortie.

Le bus CAN est un bus "série, asynchrone, à 2 fils, symétrique". Il se présente sous la forme normalisée des différentes couches ISO/OSI. Il en est à sa version 2.0 qui comprend deux parties A et B. La partie A décrit la trame CAN la plus courante qui permet d'adresser des identificateurs sur 11 bits (contre 29 bits pour la version B ou étendue).

Le bus CAN possède les principales propriétés suivantes :

- hiérarchisation des messages ;
- garantie des temps de latence ;
- souplesse de configuration ;
- réception de multiples sources avec synchronisation temporelle ;
- système multi-maîtres, détection et signalisation d'erreurs ;
- retransmission automatique des messages altérés dès que le bus est libéré ;
- distinction entre des erreurs temporaires et des non-fonctionnalités permanentes au niveau d'un nœud ;
- déconnexion automatique des nœuds défectueux.

Pour les mesures odométriques et de vitesse des roues (ou de braquage), les quatre moteurs de traction et le vérin motorisé de direction des Cycabs sont équipés de codeurs optiques incrémentaux qui délivrent en sortie deux ondes carrées en quadrature (500 cycles par tour) plus une sortie Index (top tour).

#### 2.3.5. Le nœud de direction

Son rôle est le contrôle/commande du vérin motorisé, l'interface avec le joystick, et la commande des moteurs de frein de parking.

Le position absolue du vérin de direction est obtenue par une mesure analogique de la tension aux bornes d'un potentiomètre linéaire fixé sur le vérin et monté dans un pont diviseur de tension.

La température du moteur est mesurée par une thermistance située sur le moteur. La commande du véhicule en mode manuel est donnée par un joystick connecté sur le nœud de direction.

Le nœud de direction commande aussi les moteurs de frein de parking. Le véhicule prototype dispose d'un frein motorisé par roue, mais les 4 moteurs commandés de la même manière et sans contrôle particulier. Ils sont vus par le nœud comme 2 entrées TOR pour indiquer l'état freiné ou l'état non-freiné et d'une sortie TOR pour la commande de changement d'état.

#### 2.3.6. Le nœud de traction

Le nœud de traction commande deux moteurs séparément. Chaque moteur est muni d'un codeur incrémental du même type que celui équipant le moteur du vérin motorisé et d'une thermistance pour la mesure analogique de la température du moteur. En outre un des deux moteurs est équipé d'un interrupteur thermique de sécurité utilisant une entrée TOR.

### 3. LOCALISATION PAR CATADIOPTRES

#### 3.1. Modèle géométrique

Il s'agit d'obtenir les informations suivantes :

- Position angulaire  $\alpha$  de la cible ;
- Distance  $d$  de la cible ;
- Angle  $\theta$ , orientation relative des axes des deux véhicules.

Nous avons installé un système de deux bandes réfléchissantes C1 et C2 (catadioptrés : Cf. Chapitre I 1.3) à l'arrière du véhicule cible (Cf. Figure III. 3.1).

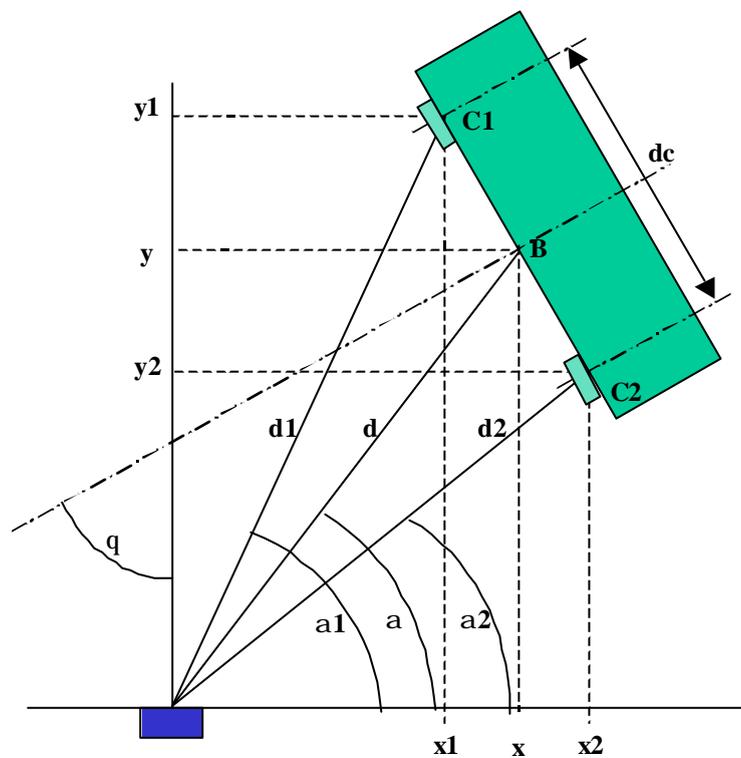


Figure III.3.1: Repérage de la cible

Le positionnement du véhicule cible est obtenu à partir du point milieu **B** des deux catadioptrés (**C1** et **C2**). Il en résulte les équations suivantes :

$$x_1 = d_1(\cos a_1) \text{ et } x_2 = d_2(\cos a_2)$$

$$y_1 = d_1(\sin a_1) \text{ et } y_2 = d_2(\sin a_2)$$

$$dc = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (20)$$

$$x = \frac{x_1 + x_2}{2} \text{ et } y = \frac{y_1 + y_2}{2} \quad (21)$$

$$a = \frac{a_1 + a_2}{2} \quad (22)$$

$$q = - \arg \sin \frac{\frac{x_2 - x_1}{d} \sin \theta}{\frac{y_2 - y_1}{d}} \quad (23)$$

avec :

$x$  et  $y$  : Coordonnées cartésiennes du véhicule suivi, relatives au véhicule commandé

$d$  : Distance entre les deux véhicules

$q$  : Angle entre les axes longitudinaux de chaque véhicule

$a$  : Position angulaire du véhicule suivi par rapport à l'avant véhicule commandé

$dc$  : distance entre les deux catadiopres C1 et C2

### 3.2. Détection et localisation des catadiopres

Pour localiser la cible il faut localiser les deux catadiopres installés à l'arrière de la cible.

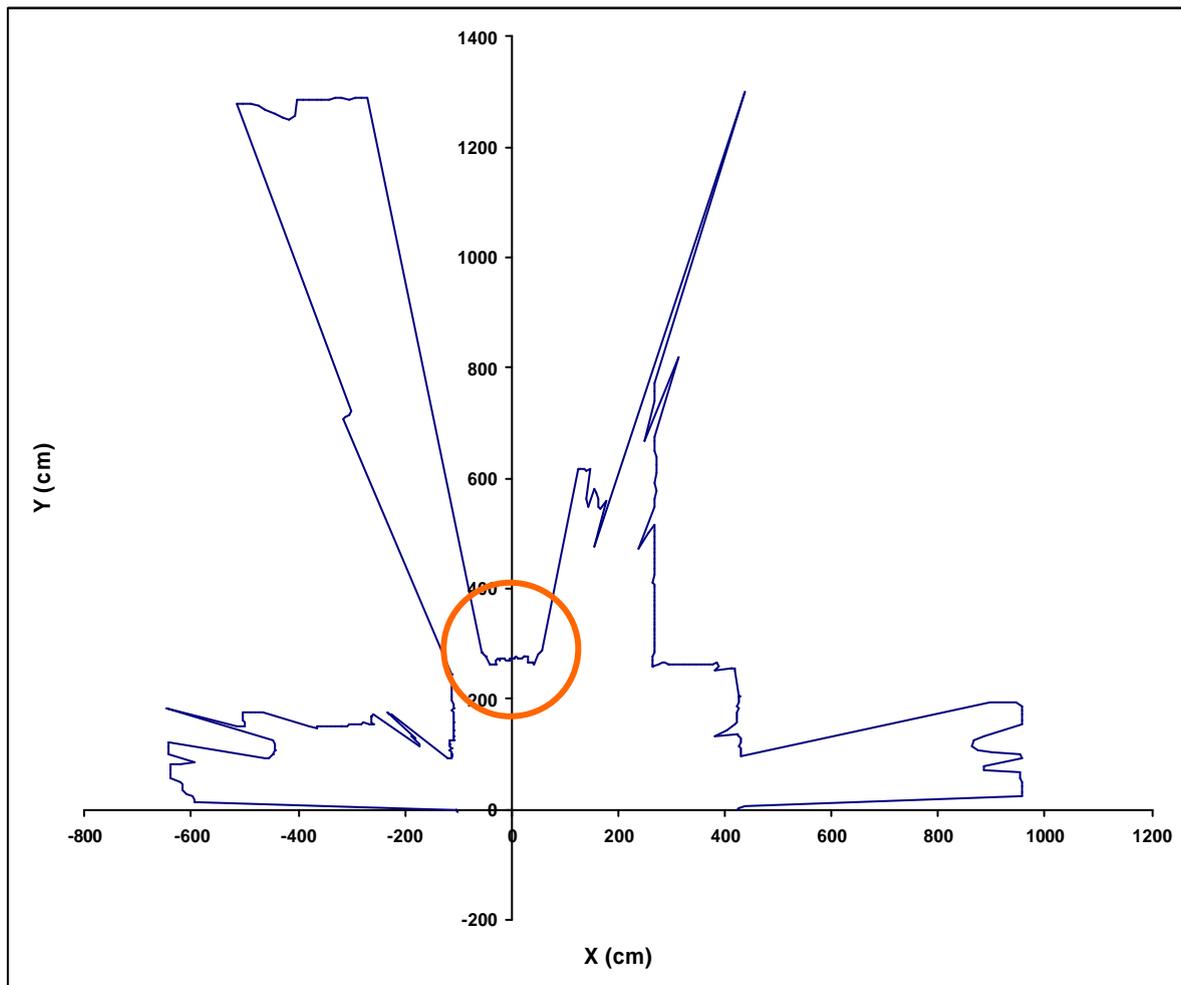
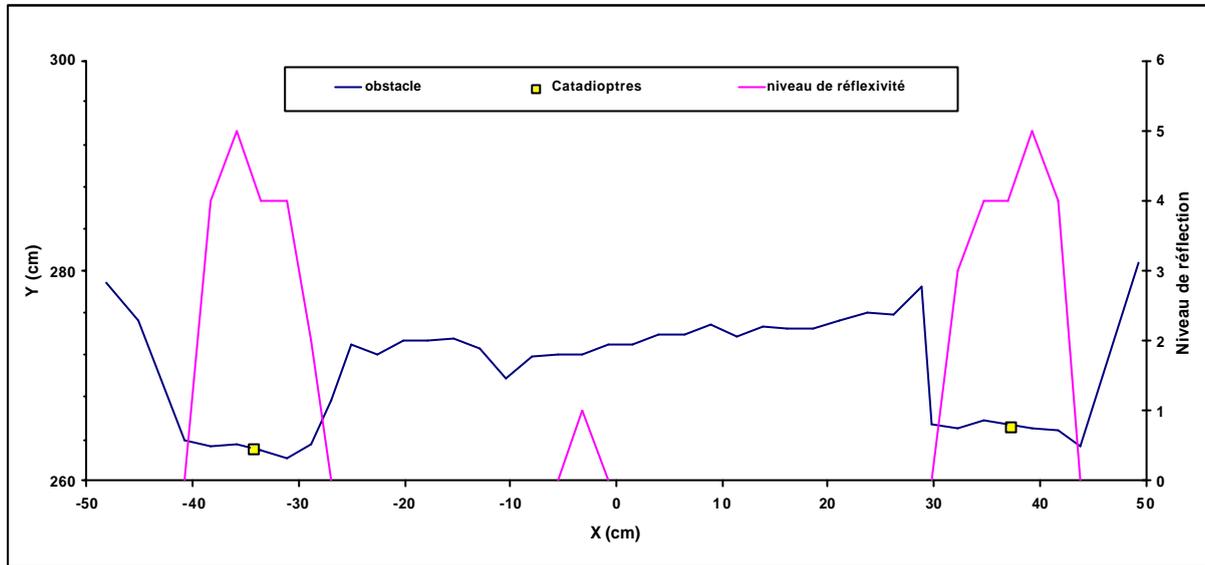


Figure III.3.2.a : Repérage de la cible

La figure III. 3.2.a présente le résultat d'une mesure du capteur télémétrique à balayage laser. Cette mesure a été effectuée dans un environnement fermé, à 3 mètres derrière le véhicule cible. On peut distinguer deux excroissances dues à la présence des catadioptres. Si on agrandit la zone concernant la cible et que l'on considère le niveau de réflexion (Cf. Figure III. 3.2.b), on constate qu'il est possible de positionner les catadioptres.



**Figure III. 3.2.b : Niveaux de réflexion**

On peut remarquer un niveau de réflexion non nul entre les deux catadioptres, imputable à la nature réfléchive du support en aluminium qui maintient les réflecteurs. Il suffira d'éliminer ce genre de parasites par un simple seuillage du niveau.

Le calcul du barycentre de chaque catadioptre permet d'obtenir un positionnement plus précis. Les barycentres s'obtiennent par les formules suivantes :

$$Distance\ du\ catadioptre\ i = d_i = \frac{\sum_{j=deb_i}^{j=fin_i} d_j n_j}{\sum_{j=deb_i}^{j=fin_i} n_j} \quad (24)$$

$$Angle\ du\ catadioptre\ i = a_i = \frac{\sum_{j=deb_i}^{j=fin_i} a_j n_j}{\sum_{j=deb_i}^{j=fin_i} n_j} \quad (25)$$

Les égalités suivantes définissent des quantités qui seront exploitées dans le cadre d'une évaluation qualitative de l'information "catadioptre" :

$$Nombre\ de\ points\ identifiants\ le\ catadioptre\ i = nbp_i = jfin_i - jdeb_i \quad (26)$$

$$\text{Niveau de réflexion globale du catadioptre } i = N_i = \sum_{j=\text{jdeb}_i}^{\text{jfin}_i} n_j \quad (27)$$

$$\text{Niveau de réflexion moyen du catadioptre } i = Nm_i = \frac{N_i}{nbp_i} \quad (28)$$

avec :

$d_j$  : distance du point  $j$

$n_j$  : niveau de réflexion du point  $j$

$a_j$  : position angulaire du point  $j$

$\text{jdeb}_i$  : indice  $j$  du premier point d'une série de points identifiants un catadioptre

$\text{jfin}_i$  : indice  $j$  du dernier point d'une série de points identifiants un catadioptre

### 3.3. Problématique liée à l'environnement non structuré

Lors des essais d'identification et de localisation de notre système cible à base de catadioptres, il est apparu que la présence d'autres catadioptres (intégrés dans les feux arrière de voitures) pose le problème du choix de la cible. Nous avons constaté que l'écartement entre les deux feux arrière des voitures conventionnelles se situait au delà du mètre. Nous avons donc choisi de fixer la distance inter-catadioptres  $dc$  à 70 cm.

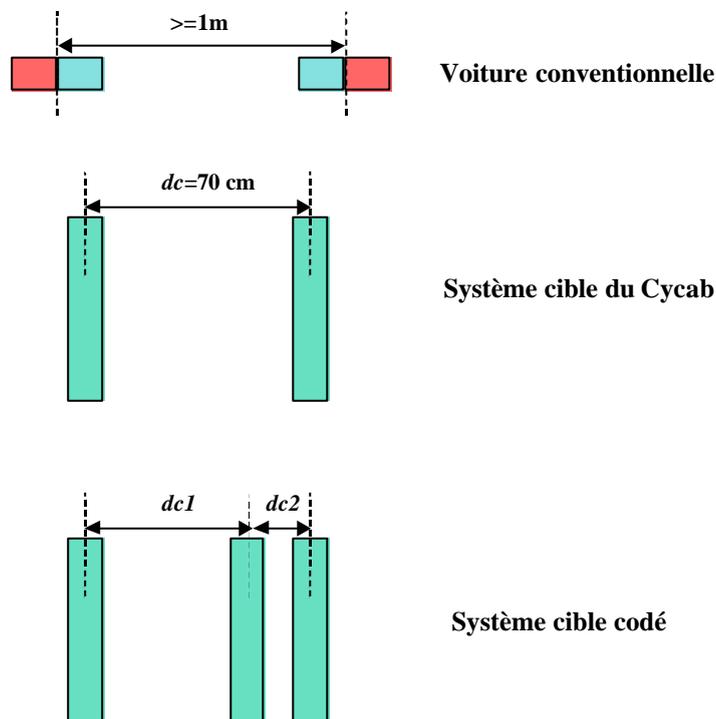


Figure III.3.3 : Niveaux de réflexion

En ajoutant un troisième catadioptre, on obtient un système permettant d'éliminer totalement le risque d'erreur de repérage de la cible.

### 3.4. Dimensionnement des catadioptres

Les amortisseurs du véhicule, la pente et les irrégularités de la route, engendrent une déviation verticale du plan de balayage du capteur. Une inclinaison de  $b=1^\circ$  engendre un déplacement du plan de 17,5 cm sur une cible située à  $d=10\text{m}$  (Cf. Figure III.3.4.a).

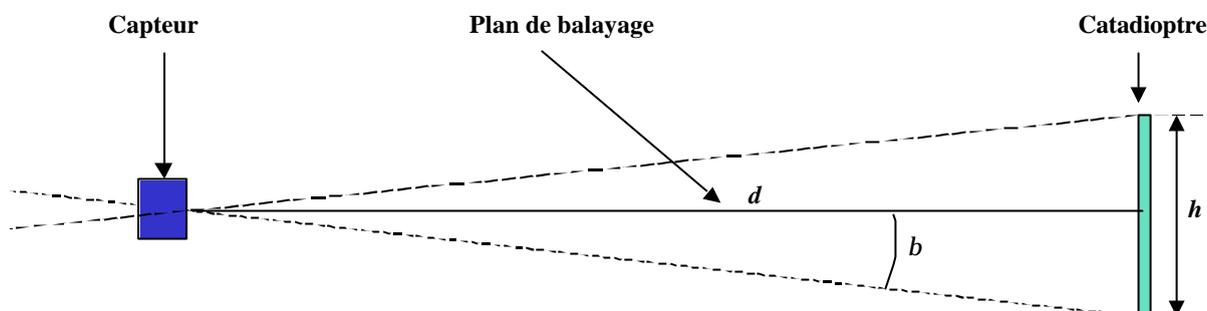


Figure III.3.4.a : Angle de déviation verticale

Des essais sur le terrain ont permis de fixer la hauteur  $h$  minimale des catadioptres à 35 cm, ce qui correspond à une déviation de plus ou moins  $1^\circ$ . Pour choisir un dimensionnement horizontal des catadioptres, il faut considérer les problèmes liés à l'espacement des spots. D'après les courbes de la figure I.1.4.5, on obtient les équations suivantes :

$$\text{Distance interspots (en m)} = di_{spot} = d(\arctan(Resa)) \quad (29)$$

$$\text{Diamètre d'un spot (en m)} = l_{spot} = d(\arctan(0.25^\circ)) + l_{spotinit} \quad (30)$$

Avec :

$Resa$  : Résolution angulaire en degrés ( $1^\circ$ ,  $0.5^\circ$ , ou  $0.25^\circ$ )

$l_{spotinit}$  : Largeur du spot à l'origine

Les figures 3.4.b (Resp. 3.4.c) donnent un aperçu des valeurs de  $l_{spot}$  (Resp.  $di_{spot}$ ).

<b><math>d</math> (m)</b>	5	10	20
<b><math>l_{spot}</math> (cm)</b>	3,3	5,5	9,8

Figure III.3.4.b : Diamètre des spots en fonction de la distance

<b><math>d</math> (m)</b>	<b><math>di_{spot}</math> (cm)</b>		
	<b><math>Resa=1^\circ</math></b>	<b><math>Resa=0,5^\circ</math></b>	<b><math>Resa=0,25^\circ</math></b>
5	8,7	4,4	2,2
10	17,5	8,7	4,4
20	34,9	17,5	8,7

Figure III.3.4.c : Distance interspots en fonction de la distance et de la résolution

Si on considère la distance  $l_{\min} = di_{spot} - l_{spot}$  (Cf. Figure III.3.4.d), on constate qu'il existe une zone de non recouvrement entre deux spots, pour les résolutions angulaires  $0.5^\circ$  et  $1^\circ$ , impliquant une taille minimale (largeur  $l_{\min}$ ) des objets pouvant être détectés. D'après la figure III.3.4.e, pour une distance capteur/cible de 20m et une résolution angulaire de  $0,5^\circ$ , on a une largeur minimale des objets détectables  $l_{\min} = di_{spot} - l_{spot} = 17.5 - 9.8 = 7.7\text{cm}$ . Lors d'essais sur le terrain avec des catadioptrés de largeur  $l_c = 8,5\text{cm}$ , nous avons vérifié une détection constante à une distance de 20m, tout en obtenant des niveaux de réflexion supérieurs au seuil de prise en compte.

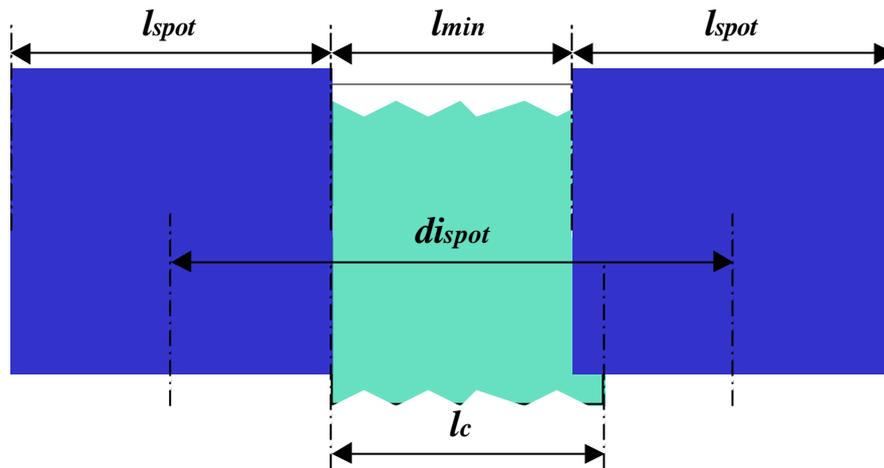


Figure III.3.4.d : Largeur minimale du catadioptré

$d(m)$	$l_{\min} (cm)$		
	$Resa=1^\circ$	$Resa=0,5^\circ$	$Resa=0,25^\circ$
5	5,4	1,1	-1,1
10	12	3,2	-1,1
20	25,1	7,7	-1,1

Figure III.3.4.e : Largeur minimale de détection en fonction de la distance

### 3.5. Erreur angulaire

Si on se place toujours dans le cas le plus défavorable, c'est à dire à une distance de 20m de la cible, l'analyse de l'erreur angulaire de détection des catadioptrés prend toute son importance. La figure III.3.5.a montre le cas d'une détection idéale et le cas d'une détection avec une erreur angulaire maximale.

Détection idéale : le catadioptré se situe au milieu de deux spots **Sp1** et **Sp2**. Chaque spot mesure un niveau de réflexion identique. La position détectée (calculée) correspond à la position réelle du catadioptré.

Détection avec erreur maximale : le catadioptré est placé de telle manière qu'un des deux spots (**Sp2**) mesure un niveau de réflexion nul et que la distance entre le second spot (**Sp1**) et le catadioptré est maximale. Dans ce cas, le spot **SP1** est seul à mesurer un niveau de

réflexion non nul. L'erreur de détection de la position du catadioptre est la distance entre le centre du spot **Sp1** et le centre du catadioptre.

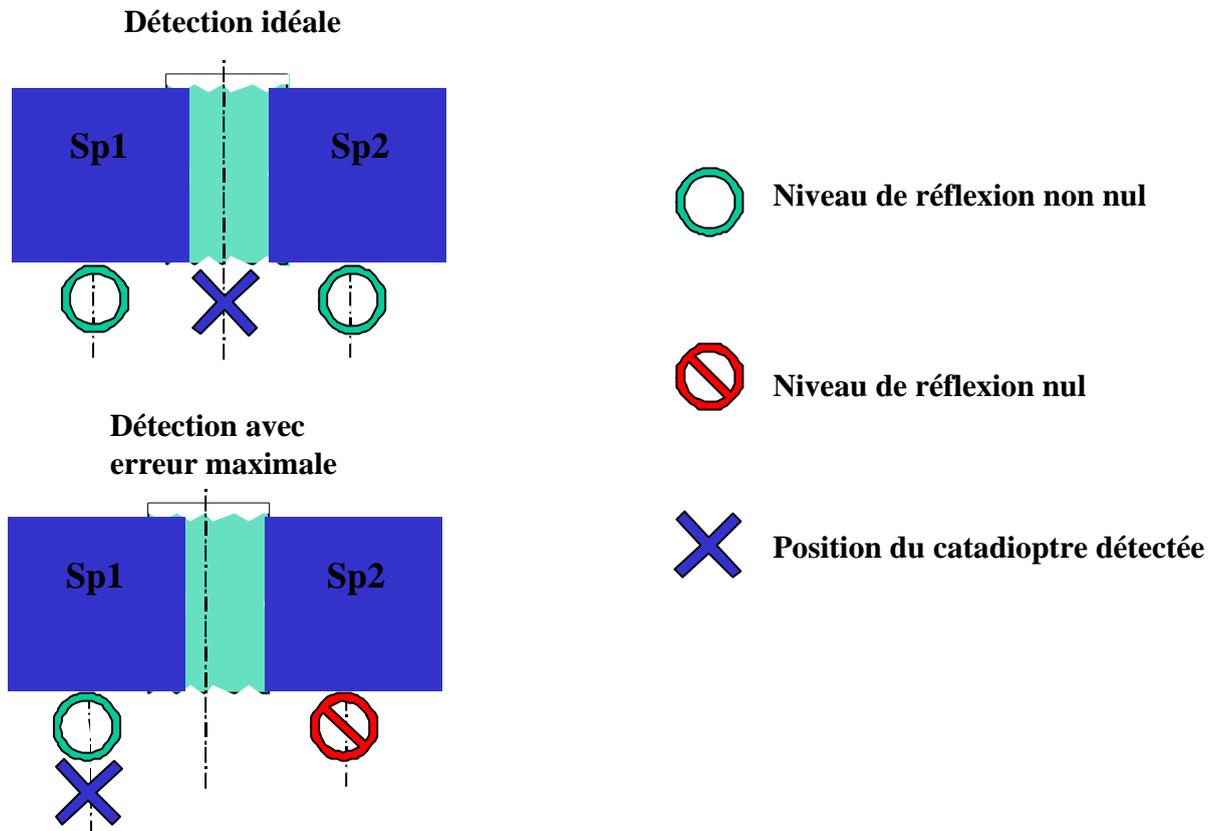


Figure III.3.5.a : Largeur minimale du catadioptre

Pour :

$$d = 20m, l_c = 8,5cm \text{ et Résolution angulaire } Resa = 0,5^\circ$$

On obtient :

$$\text{Erreur métrique} = e_m = di_{spot} - \frac{l_{spot}}{2} - \frac{lc}{2} = 17,5 - \frac{9,8}{2} - \frac{8,5}{2} = 8,35 \text{ cm}$$

Si on considère le système cible composé de deux catadioptres distants de  $dc = 70 \text{ cm}$ , on peut calculer l'erreur métrique de détection de la cible (erreur sur la mesure de  $dc$ ) à 20 mètres :

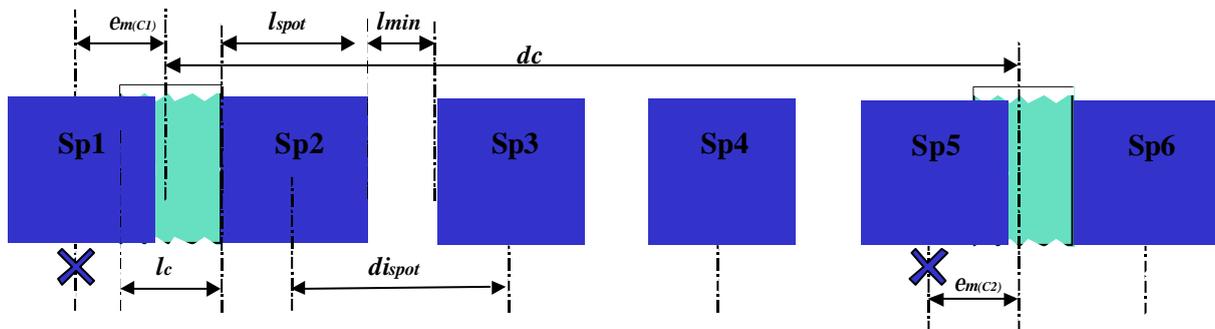


Figure III.3.5.a : Erreur de mesure de  $dc$

Erreur métrique sur  $dc = e_{dc} = e_{m(C1)} + e_{m(C2)}$

$$e_{dc} = \frac{\alpha}{\xi} di_{spot} - \frac{l_{spot}}{2} - \frac{lc}{2} \frac{\ddot{\theta}}{\theta} + \frac{\alpha lc}{\xi} \frac{l_{spot}}{2} + \frac{l_{spot}}{2} + n \cdot (di_{spot}) \frac{\ddot{\theta}}{\theta} - dc = ((n+1) \cdot di_{spot}) - dc$$

On s'aperçoit de l'annulation de l'erreur pour  $(n+1) \cdot di_{spot} = dc$ .

Pour obtenir des performances maximale en terme de détection de cible à une distance nominale (ici 20 m), nous choisirons une distance inter catadioptrés correspondant à un multiple de la distance inter spots (à cette même distance nominale).

Nous avons donc opté pour une distance inter catadioptrés  $dc = 17.5 \times 4 = 70 \text{ cm}$ .

## 4. COMMANDE

Les lois de commande que nous utilisons pour le suivi avec le capteur télémétrique sont les mêmes que celles utilisées avec la caméra linéaire. Même si la commande n'est pas optimale, nous l'avons conservé afin de simplifier de futur comparatifs entre les deux systèmes de capteurs.

La loi du suivi de véhicule, proposée dans [18], consiste à conserver un temps  $T$  constant entre deux véhicules (Cf. Figure III.4.a). En intervenant sur ce paramètre, appelé facteur de Davier-Parent, il nous est possible de contrôler la distance à laquelle les véhicules se suivent. S'agissant d'un temps, cela signifie que la distance entre les véhicules varie proportionnellement à leur vitesse relative. La consigne de vitesse est donc proportionnelle à la distance ( $d-d_0$ ) qu'il reste à parcourir pour atteindre la cible.

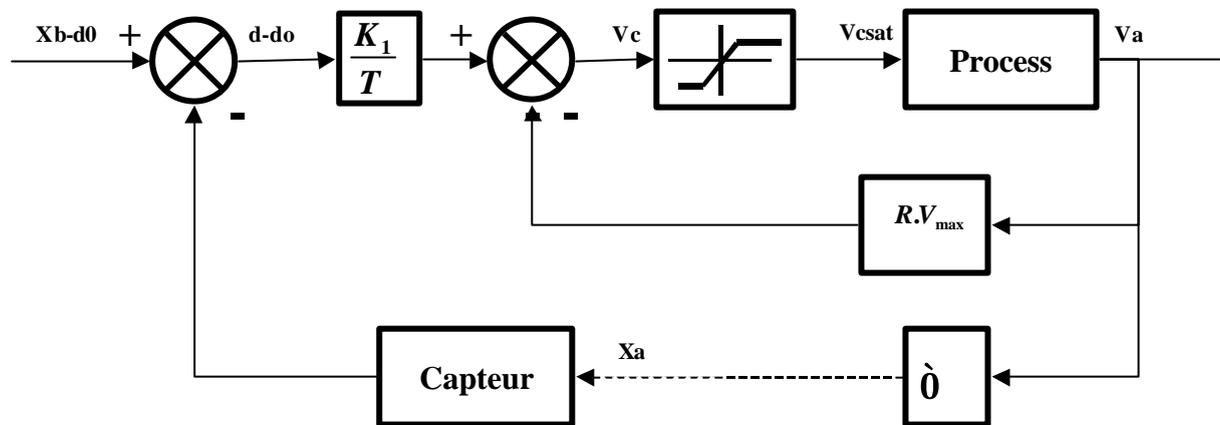


Figure III.4.a : Contrôle longitudinal

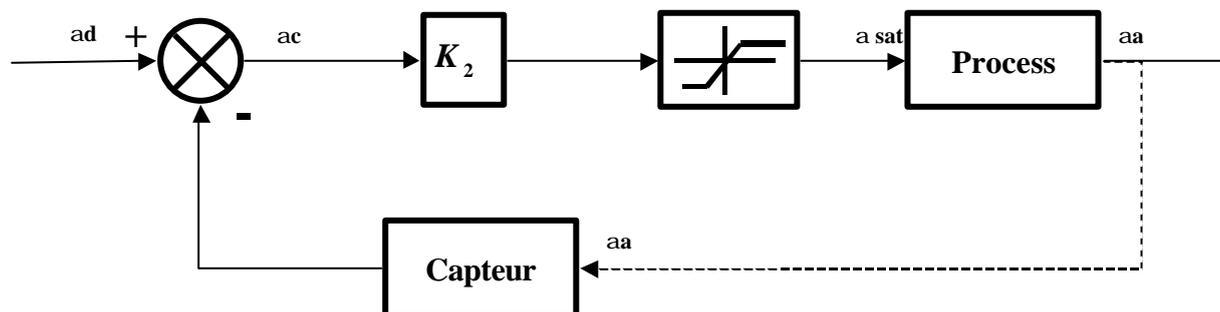


Figure III.4.b : Contrôle latéral



# Chapitre IV

## DEVELOPPEMENTS LOGICIELS

*Après un bref rappel sur l'architecture logicielle du Cycab, nous décrivons l'interface de communication et de prétraitement des données capteur. Les tâches robotique utilisant ces données sont implémentées à l'aide système contrôle commande ORCCAD. Nous présentons nos développements de tâches robotique à l'aide de ce système.*

## 1. ARCHITECTURE LOGICIELLE DU CYCAB

Pour le CYCAB Rhône-Alpes, nous disposons d'algorithmes de contrôle bas-niveau des moteurs au niveau des nœuds en EPROM et d'un protocole basique de communication entre les nœuds.

### 1.1. Rappels sur l'informatique embarquée du CYCAB Rhône-Alpes

L'architecture matérielle du CYCAB est composée de :

- Une carte MVME162, avec l'OS VxWorks 5.3, un module IP-CAN (Controller Area Network et une liaison Ethernet HF ;
- Trois Modules (traction avant, traction arrière, direction-joystick) composés chacun d'un microprocesseur 68332, d'une interface CAN, et d'une interface RS232 (liaison série) ;
- Des entités peuvent être rapportées sur le bus CAN (ultrasons, caméra linéaire).

### 1.2. Choix sur la communication

Quelques règles générales :

- l'interface série des Modules ne sert qu'à la mise au point;
- le bus CAN est le seul moyen de communication entre la carte MVME162 et les Modules;
- la carte MVME162 dialogue avec le réseau (pour l'amorçage par exemple) via le lien Ethernet HF.

Du point de vue du bus CAN, la carte MVME162 est maître, alors que les Modules (ou les autres entités CAN) sont esclaves. C'est donc la carte MVME162 qui prend l'initiative de la communication, le ou les Modules se contentent de rester à l'écoute et de répondre si on le leur demande.

### 1.3. Environnement de Programmation

Pour la carte MVM162 (VxWorks), la chaîne de développement croisée Solaris-68k est utilisée (compilation, loader dynamique,...).

La partie GNU (compilateur croisé solaris-68k) est aussi utilisée pour développer le code des modules. En effet, le cœur du micro-contrôleur 68332 est celui du 68020. Le code est chargé dans la mémoire flash se trouvant dans le module via la liaison série ou via le bus CAN. A l'amorçage du 68332, le code en Flash est recopié en mémoire, puis exécuté.

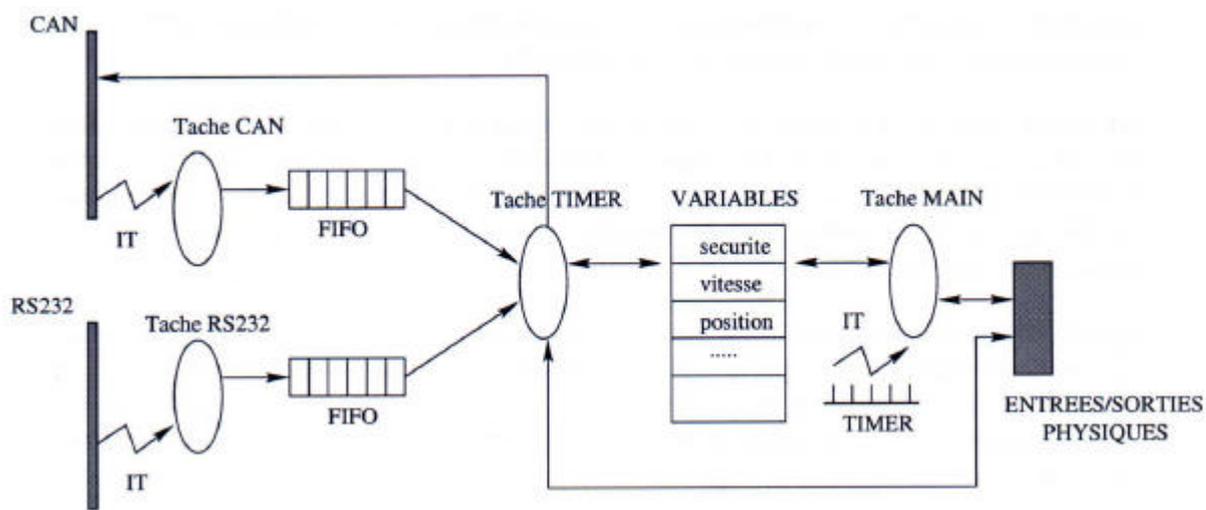
### 1.4. Modèle d'exécution du logiciel sur les nœuds

A l'amorçage du micro-contrôleur, quatre tâches se déroulent en parallèles :

- une routine d'interruption reçoit les messages venant du CAN et les empile dans une FIFO ;
- une routine d'interruption reçoit les messages venant de la liaison série et les empile dans une FIFO ;

- le programme principal, dépile et traite les messages. Il peut émettre un message CAN ou une sortie sur la liaison série s'il le faut ;
- une routine d'interruption déclenchée sur un TIMER à 10 ms, permet de dérouler des fonctions (périodiques). Celles-ci peuvent être autorisées ou non dynamiquement. A titre d'exemple, sur chaque Module de traction se trouve une routine TIMER qui annule la sortie puissance si celle-ci n'est pas mise à jour régulièrement. Sur le Module Direction-Joystick, une routine TIMER permet l'asservissement en position de la direction.

Il est à noter que le protocole de plus haut niveau, décrit par la suite, (un seul maître, acquittement permanent) permet d'éviter tout blocage sur le 68332.



**Figure IV.1.4 : Modèle d'Exécution**

Le nœud 68332 qui écoute et envoie des messages sur le bus CAN suit le protocole CAN-CYCAB que nous définissons dans la partie suivante.

### 1.5. Le Protocole CAN-CYCAB

Chaque entité sur le bus CAN doit se conformer au protocole suivant.

En résumé, les transferts sur le bus CAN sont basés sur des trames, constituées de 11 bits d'en-tête et de 8 octets de message.

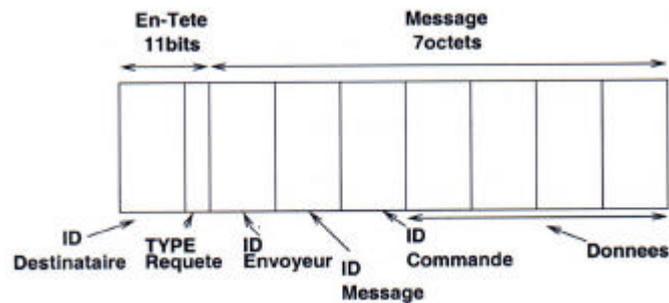
L'en-tête est utilisé pour identifier le destinataire de manière unique (8 bits) et le type de message (3 bits).

Les différents types de messages sont :

- envoi d'une requête, avec ou sans données et avec ou sans demande d'acquittement ;
- réponse à une requête, avec ou sans données.

Les 8 octets de messages se décomposent en :

- un octet pour l'identifiant de l'expéditeur ;
- un octet pour l'identifiant du message ;
- un octet pour le mot de commande à exécuter ;
- quatre octets pour les données éventuelles.



**Figure IV.1.5 : Trame CAN**

Dans la version actuelle du CYCAB, seules des requêtes avec demande d'acquittement sont utilisées.

### 1.6. Pilotage du CYCAB via le CAN

Une première méthode consiste à utiliser un menu du même type que celui de la liaison série. Il est lui aussi réservé au développement et à la maintenance. En plus des commandes décrites ci-dessus, apparaissent :

- La possibilité de changer de Module ;
- Le chargement d'un fichier type hex format en flash ;
- Un test de vitesse de transmission sur le bus CAN.

Une librairie C dont les fonctions sont décrites ci-après (extrait du fichier d'include C) est disponible pour réaliser l'interaction entre la commande et les capteurs/actionneurs :

```
int cycabPutBrake(int etat)
int cycabGetBrake(int *etat)
int cycabPutMotor(int etat)
int cycabGetMotor(int *etat)
int cycabPutPwm(int avg, int avd, int arg, int ard)
int cycabGetPos(int *avg, int *avd, int *arg, int *ard)
int cycabPutDir(int etat)
int cycabGetDir(int *etat)
int cycabPutDirPos(int centieme_degre)
int cycabGetDirPos(int *centieme_degre)
int cycabGetDirCod(int *top)
int cycabGetState(int *etat_ava, int *etat_arr, int *etat_dir)
int cycabGetJoystick(int *j_dir, int *j_tra)
```

## 2. INTERFACE CAPTEUR

### 2.1. Introduction

La société SICK fournit des bibliothèques logicielles pour le développement d'applications Windows que nous ne pouvons pas utiliser sur nos plates-formes. Compte tenu de l'architecture logicielle et matérielle du Cycab, il a été nécessaire de développer une interface capteur pour Linux ainsi qu'une interface de communication avec le système d'exploitation temps réel VxWorks.

### 2.2. Présentation de l'interface logicielle

L'interface entre le capteur et le système informatique est composée principalement de deux entités:

- Une application développée sous LINUX (appelée Interface Capteur), comprenant :
  - une Interface de communication série RS232 pour dialoguer avec le capteur,
  - un gestionnaire de télégrammes,
  - un module intégrant des fonctionnalités de traitement des données capteur,
  - un menu utilisateur avec des fonctionnalités avancées,
  - une interface TCP/IP (Serveur de socket) pour communiquer avec d'autres systèmes d'exploitation à travers le réseau.
- Une application développée sous VxWorks (appelée Client VxWorks), comprenant :
  - une interface TCP/IP (Client socket),
  - un interpréteur de commandes.

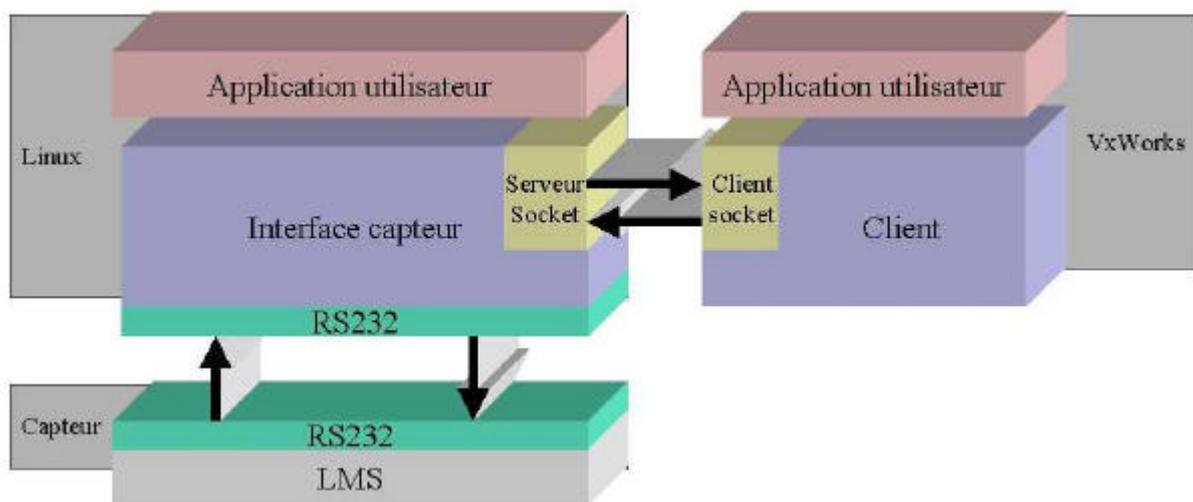


Figure IV.2.2 : Schéma bloc de l'interface logicielle capteur

### 2.3. Interface de communication série

L'interface de communication série (RS232) est basé sur les bibliothèques de communication série standard de LINUX (termios, stdio, unistd). L'écriture et la lecture des données d'un port série est réalisé par l'écriture et la lecture sur la ressource correspondante (ex : ttyS0).

Afin de simplifier son utilisation, la bibliothèque RS232 a été réécrite en tenant compte des paramètres spécifiques liés à l'utilisation du capteur. La liaison série du capteur n'intègre pas de contrôles de flux (XON, XOFF). La détection d'erreur de communication et de pertes de données doit être gérée par l'applicatif de niveau supérieur.

### 2.4. Gestionnaire de télégrammes

La communication avec le capteur est basée sur l'échange de télégrammes entre le capteur et le système maître (dans le cas présent : un PC Linux).

Les télégrammes peuvent être classés en différentes catégories :

- Télégrammes du PC vers le capteur
  - Initialisation du capteur,
  - Modification du mode de fonctionnement du capteur,
  - Modification de la configuration du capteur,
  - Demande de données de mesure,
  - Demande de données de configuration,
  - Demande de données de statut.
- Télégrammes du capteur vers le PC
  - Tous les télégrammes de réponse aux télégrammes envoyés par le PC,
  - Le télégramme de non-acquittement.

Suite à l'envoi d'un télégramme au capteur, celui-ci renvoie un caractère d'acquittement ACK (code hexadécimal : 06H). Si une erreur est détectée par le capteur, c'est un caractère de non-acquittement qui sera transmis NAK (code hexadécimal : 15H).

Contraintes temporelles :

- L'intervalle de temps entre deux caractères doit être compris entre 55  $\mu$ s et 6 ms. Le cas échéant, le télégramme est ignoré ;
- Les caractères ACK et NAK sont transmis dans un délai de 60 ms maximum. Une réponse à une demande de changement du mode du capteur peut prendre jusqu'à 3 secondes ;
- Suite à la réception d'un NAK, le PC doit attendre au moins 30 ms avant de pouvoir renouveler la demande.

### 2.5. Echanges de données entre le capteur et l'interface

Les données échangées entre l'interface et le capteur sont de différentes natures :

- Données de configuration du capteur ;
- Paramètres de spécification des mesures ;

- Les données mesurées.

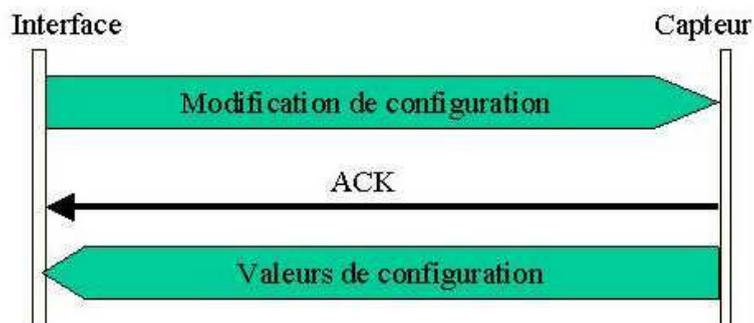
Seule la commande d'initialisation est de type simple sans retour de données (Cf. IV.2.5.a).



**Figure IV.2.5.a : Commande simple**

Le schéma d'échange le plus utilisé est le suivant :

- L'interface (maître) envoie un télégramme de commande (avec ou sans paramètres);
- Le capteur analyse le télégramme;
- Le capteur renvoie :
  - Un NACK si le CRC<sup>13</sup> du télégramme reçus et faux
  - Un ACK suivi de données informants sur le résultat de la commande



**Figure IV.2.5.b : Commande avec réponse**



**Figure IV.2.5.c : Commande avec mauvais CRC**

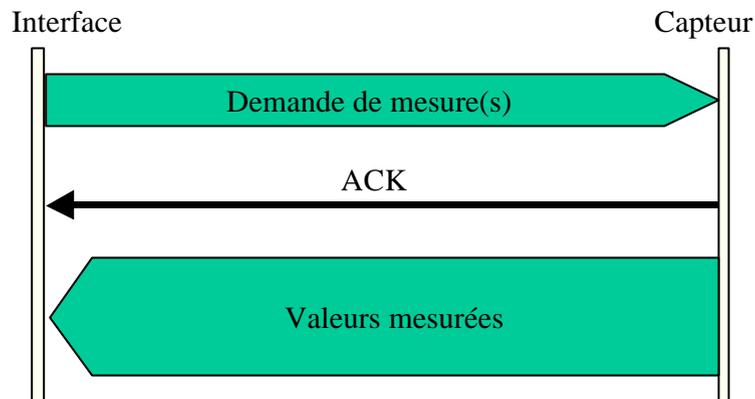
Dans le cas d'une commande de demande de valeurs de mesure, le type de réponse dépend du mode de fonctionnement utilisé.

<sup>13</sup> CRC : Cyclic Redondancy Check, test d'erreurs de type "check sum" consistant à valider un chaîne de données.

Deux modes se distinguent :

- Mode "sur demande" ;
- Mode "continu".

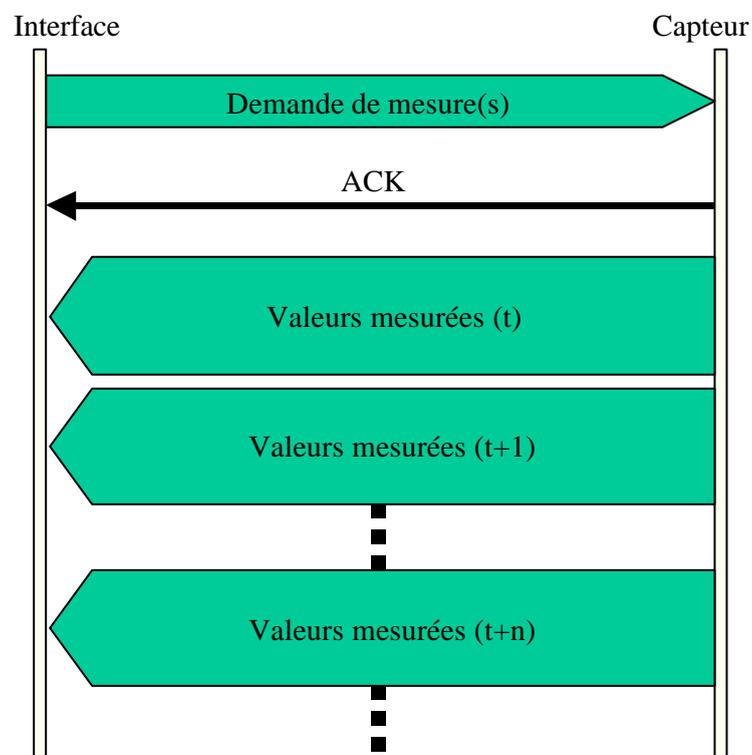
Dans le mode "sur demande", le capteur renvoie une et une seule fois le résultat de la mesure demandée.



**Figure IV.2.5.d : Mode "sur demande"**

Dans le mode "continu", le capteur renvoie continuellement le résultat de mesures successives.

Seule une commande de changement en mode "sur demande" peut arrêter l'envoi de données continues de mesures.



**Figure IV.2.5.e : Mode "continu"**

La durée des échanges est directement liée à la vitesse de transfert sur la ligne série et au temps de réponse du capteur. On peut définir un temps de réponse global ( $Tr$ ) qui est dépendant des données suivantes :

- Temps d'envoi d'un télégramme au capteur ( $Tenv$ ) : il dépend de la taille du télégramme et de la vitesse de transfert de la ligne série ;
- Temps d'analyse du télégramme par le capteur ( $Tack$ ) : temps qu'il faut au capteur pour lire le télégramme et répondre par un ACK ;
- Temps d'accessibilité ( $Tinit$ ) : temps nécessaire pour atteindre une configuration qui permet la mesure demandée. Ce temps dépend du type de mesure et de la position du miroir tournant lors de la réception de la demande ;
- Temps de mesure et d'envoi du télégramme ( $Tresp$ ) : il dépend de la taille du télégramme de retour, de la résolution angulaire et de la vitesse de transfert de la ligne série.



Figure IV.2.5.f : Temps de réponse global

$$Tr = Tenv + Tack + Tinit + Tresp$$

$$Tenv = \frac{NbEnv \cdot 8}{Vsérie}$$

$$Max(Tenv + Tack) = 60ms$$

$$Max(Tinit) = 6,5ms$$

Avec :

$NbEnv$  : nombre de caractères du télégramme d'envoi

$Vsérie$  : vitesse de transfert de la ligne série

L'évaluation exacte de ces temps est cruciale lors de l'utilisation du capteur lorsque il est nécessaire de dater les données. Le temps  $Tresp$  inclut la mesure des données télémétriques, les calculs de mise en forme de ces données et l'envoi des données par la ligne série.

$$\text{Temps d'envoi des données par la ligne série : } Tret = \frac{NbResp \cdot 8}{Vsérie}$$

Avec :

$NbResp$  : nombre de caractères du télégramme de réponse

Le temps  $Tresp$  est variable en fonction de la résolution angulaire :

- Résolution angulaire =  $1^\circ$  (Cf. Figure IV.2.5.g  $\delta$ ) : Toutes les mesures sont faites lors d'une demi rotation du miroir tournant ;

- Résolution angulaire = 0,5° (Cf. Figure IV.2.5.g ●) : Deux rotations sont nécessaires pour les mesures. Sur chaque tour, les mesures sont espacées de 1°. Un décalage de 0,5° entre les mesures du premier et du deuxième tours permet d'obtenir une résolution de 0,5° ;
- Résolution angulaire = 0,25° (Cf. Figure IV.2.5.g ÷) : Quatre tours et un décalage de 0,25° sont nécessaires pour les mesures.

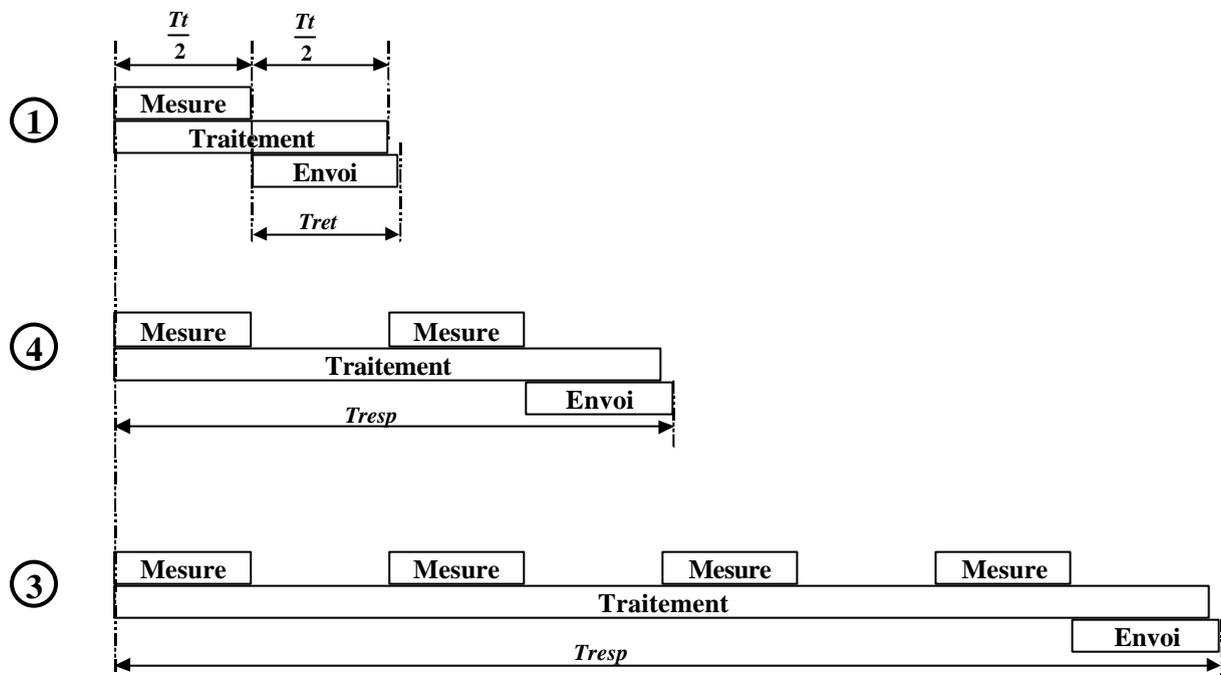


Figure IV.2.5.g : *Tresp* en fonction de la résolution angulaire

On obtient l'équation suivante :

$$Tresp = \frac{1}{Resa} - \frac{1}{\theta} + \frac{1}{2\theta} \cdot Tt + Tret$$

Avec :

*Resa* = résolution angulaire (1, 0.5 ou 0.25)

*Tt* = temps d'une rotation du miroir

L'échange de données d'une trame de mesure sur 180° avec une résolution angulaire de 0.5° et une vitesse de transfert série de 38400 bauds, donne les résultats théoriques suivants :

$$NbResp = \frac{Angle}{Resa} + \frac{1}{\theta} \cdot 2 + 8 = 730$$

$$Tret = \frac{NbResp \cdot 8}{Vsérie} = 0.152 \text{ S}$$

$$Tresp = \frac{3 \cdot \ddot{u}}{2 \cdot \ddot{u}} \cdot Tt_{\ddot{u}} + Tret = (1,5 \cdot 13) + 152 = 171,5 \text{ mS}$$

$$Tenv = \frac{NbEnv \cdot 8}{Vsérie} = 4,16 \text{ mS}$$

$$Tr = Tenv + Tack + Tinit + Tresp$$

$$238 \text{ mS} > Tr > 180 \text{ mS}$$

Ces valeurs ont été confirmés en pratique.

## 2.6. Menu utilisateur

Pour banaliser l'utilisation du capteur, nous avons développé un menu à plusieurs niveaux qui comprend :

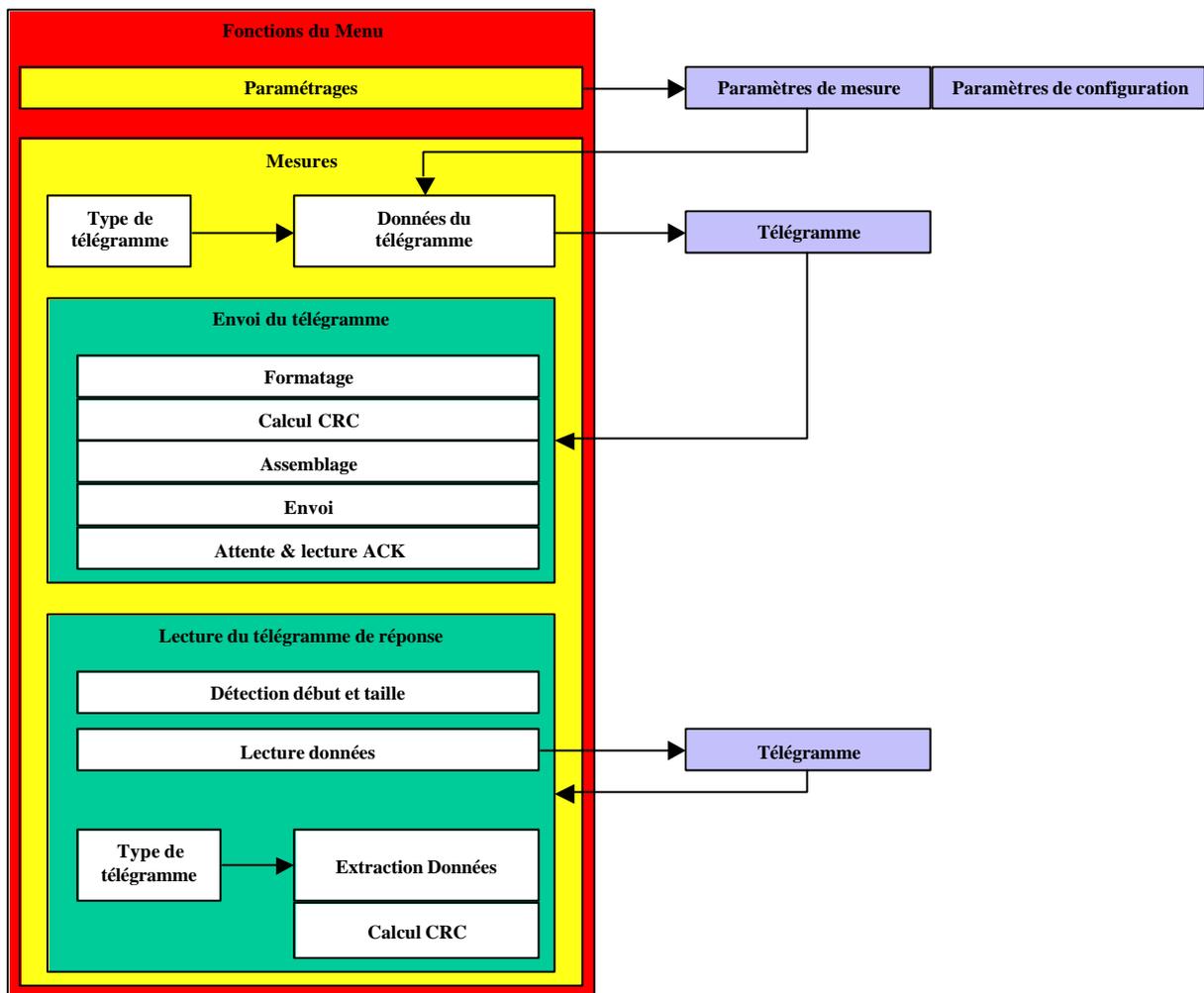
- L'accès aux fonctionnalités de base du capteur ;
- L'accès à des fonctionnalités et traitements spécifiques.

Le tableau en Annexe A.2.3 présente le détail du menu utilisateur. Toutes les fonctionnalités du menu peuvent être utilisées à travers des fonctions de la librairie *Sickio* (qui comprend le menu utilisateur).

## 2.7. Structures et variables globales

Les données de configuration ou les données de mesure se manipulent à travers les structures suivantes :

- **SENSORSTATUS** : structure contenant les informations liées au statut du capteur ;
- **SENSORFIELDS** : structure contenant les informations de configuration des champs du capteur ;
- **SENSORCONFIG** : structure contenant les informations de configuration du capteur ;
- **SENSORVALUES** : structure des valeurs de paramétrage des mesures ;
- **SICK\_TELEGRAM** : structure contenant les données d'un télégramme émis ou reçu ;
- **SENSORMEASUREDVALUES** : structure contenant les données d'une mesure ;
- **SENSORMEASUREDMULTIVALUES** : structure contenant les données de plusieurs mesures ;
- **SENSORMEASUREDCATA** : structure contenant les données de positionnement de catadioptrés, d'une mesure.



**Figure IV.2.7 : Partage de données**

Les structures de données de mesure et de paramétrage sont aussi bien utilisées en tant que arguments de fonctions de bas niveau (Envoi et réception de données sur le port série) que dans des fonctions évoluées (Calcul de la position des catadioptrés).

## 2.8. Interface TCP/IP (Serveur de socket)

La partie contrôle/commande du système est implémentée sur une plate-forme VxWorks. Il est donc nécessaire de réaliser une passerelle entre VxWorks et Linux afin d'échanger les données capteur. Le véhicule Cycab étant muni d'un réseau Ethernet, nous avons développé une liaison par "Sockets" permettant le dialogue entre l'application Linux et un client VxWorks (Cf. Figure IV.2.2).

Un serveur de sockets, exécuté du côté interface Linux, scrute en permanence le réseau. Si le client côté VxWorks envoie un message, le serveur interprète celui-ci et exécute les fonctions associées.

Nous avons utilisé une structure de message très simple. Il existe deux types de messages qui transitent entre le client VxWorks et l'application linux :

- Les messages envoyés par le client se composent de :
  - Un code numérique de trois chiffres : ce code correspond à une fonctionnalité du menu utilisateur,

- Trois paramètres de type entier,
  - Un paramètre de type chaîne de 20 caractères.
- Les messages envoyés en retour par le serveur se composent de :
    - Un code numérique de trois chiffres : ce code est 999 et ne sert que à l'acquittement de la demande du client.
- ou
- Les données : dans le cas où le client a demandé une mesure au capteur, le résultat des mesures est renvoyé sous forme d'une chaîne de caractères. Cette chaîne de caractères sera interprétée par le client afin d'en extraire les données de mesure.

## 2.9. Architecture Multi-thread

Le serveur de sockets est exécuté de manière indépendante dans un thread<sup>14</sup>, afin de ne pas bloquer le reste de l'application Linux.

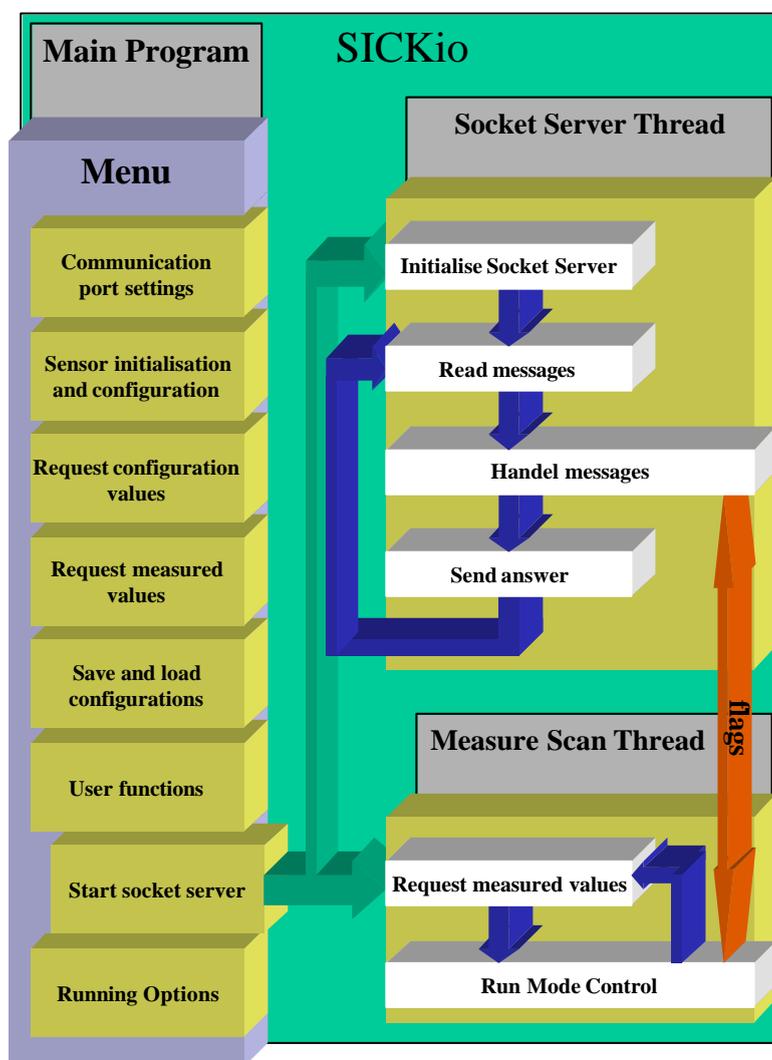


Figure IV.2.9a : Architecture multi-threads

<sup>14</sup> Thread : Processus "léger" qui partage les ressources (mémoire, cpu, ...) avec les autres processus d'une application Linux.

Pour des raisons de performances nous avons aussi fait le choix d'exécuter les opérations de mesure capteur dans un autre thread. On s'assure, par l'utilisation de flags<sup>15</sup>, de ne pas avoir de conflits lors de l'écriture et de la lecture de données par les différents threads (Cf. figure IV.2.9.a). La figure 2.9.b montre le cheminement de l'information lors de la demande de données de mesure par l'application de commande.

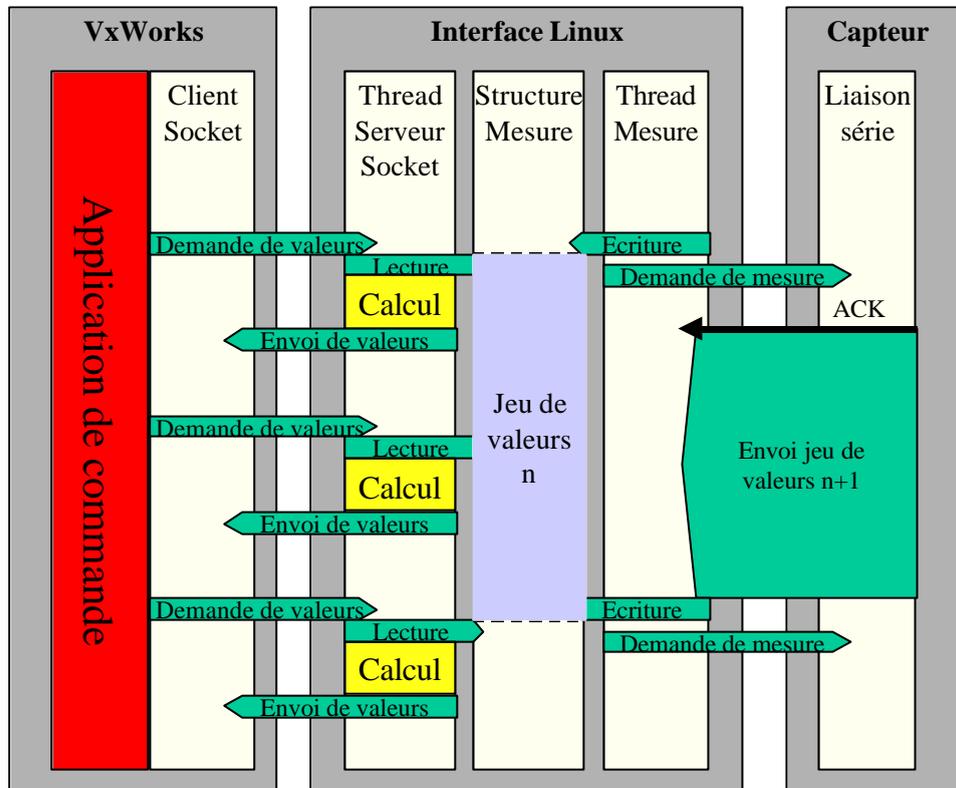


Figure IV.2.9b : Architecture multi-threads

Les résultats des mesures du capteur sont stockés dans une structure dont les données sont mises à jour à chaque nouvelle mesure. Du côté du serveur de socket, les données à transmettre sont toujours disponibles. Le client VxWorks peut demander des valeurs à volonté sans être bloqué par les latences des mesures capteur.

Les threads de mesure et de serveur socket sont décrits en détail en annexe A.

## 2.10. Modes de mesures

L'emploi de threads et de structures de données partagées permet d'optimiser les temps de réponse lors de la demande de mesures par l'application de commande. Deux paramètres interviennent :

- Le mode de mesure des données capteur :

<sup>15</sup> Flag : variable partagée servant à synchroniser plusieurs processus entre eux.

- REQUEST : le thread de mesure demande une seule mesure au capteur. Le déclenchement d'une nouvelle mesure devra être demandé explicitement par l'envoi d'un message de la part du client.
- CONTINUE : le thread de mesure demande constamment de nouvelles mesures au capteur. Chaque nouvelle mesure met à jour la structure des données de mesure.
- Le mode de lecture des données capteur par le client :
  - BLOCKMODE : lors d'une demande de mesures capteur par le client, le serveur socket bloque l'envoi d'une réponse tant que le thread de mesure n'a pas fourni une nouvelle valeur.
  - NOBLOCKMODE : toute demande de mesure de la part du client donne lieu à une réponse avec les valeurs de la structure de mesure.

La figure IV.2.10.a présente les chronogrammes d'échanges de données dans chacun des modes.

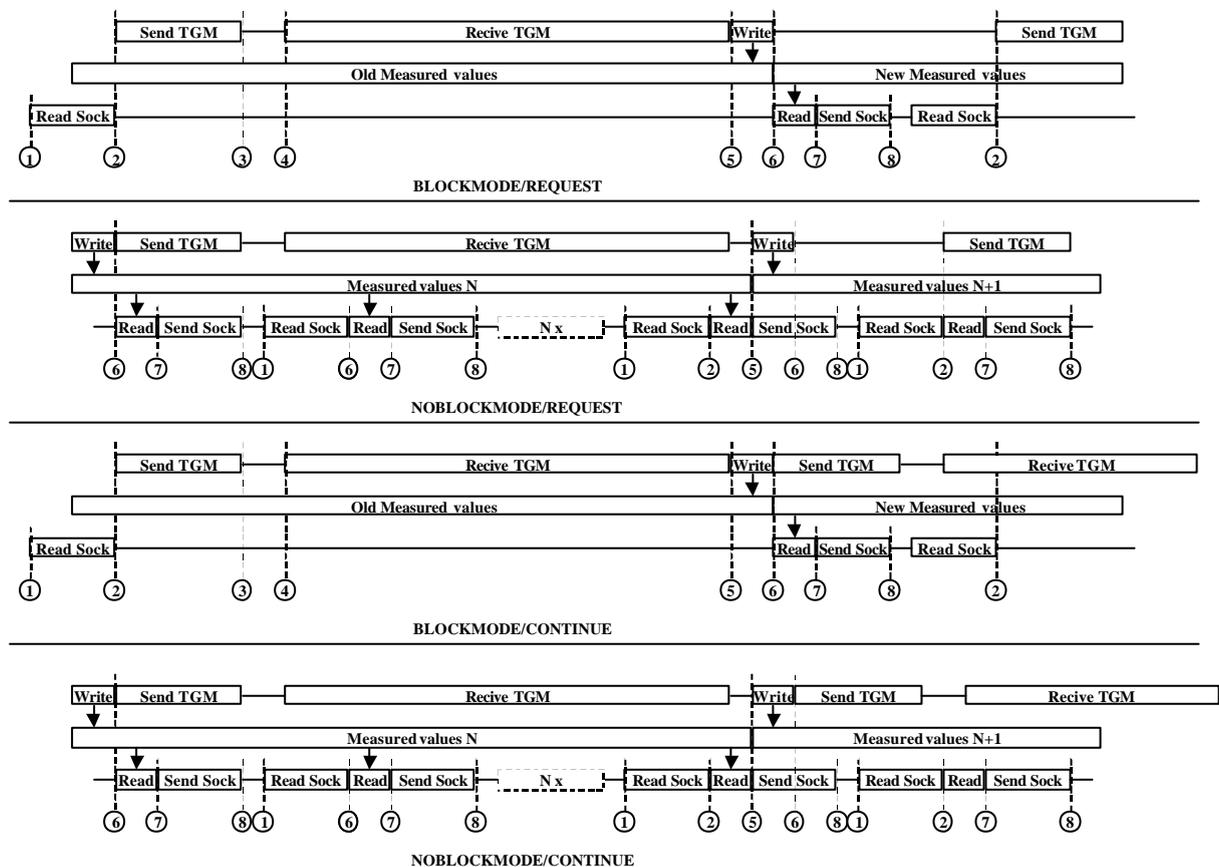


Figure IV.2.10.a : Modes de mesure

- ∂ : Message reçu en provenance du client ;
- : Spec = TRUE & measurementmode != SLEEPMODE ;
- ÷ : En attente d'un ACK ;
- ≠ : ACK reçu ;
- ≡ : OKtoWrite = TRUE ;
- ≈ : OKtoRead = TRUE.

Si on considère les résultats du chapitre III.2.5, on peut définir un temps de réponse  $Trcl$  vu du coté du client et un temps de réponse  $Trca$  vu du coté capteur.

En posant les définitions suivantes, nous pouvons définir les temps  $Trcl$  et  $Trca$  donnés respectivement par les tableaux des figures 2.10.b et 2.10.c :

$Tread$  = temps de lecture de la structure de données de mesure

$Twrite$  = temps d'écriture de la structure de données de mesure

$Tss$  = temps de transfert d'un message du client vers le serveur de sockets

$Trs$  = temps de transfert d'un message du serveur de sockets vers le client

	BLOCKMODE	NOBLOCKMODE
REQUEST	$Tr + Tread + Twrite + Tss + Trs$	$Tr + Tread + Twrite + Tss + Trs$
CONTINUE	$Tr + Tread + Twrite$	$Tr + Tread + Twrite$

**Figure IV.2.10.b :  $Trcl$  suivant les modes de mesure**

	BLOCKMODE	NOBLOCKMODE
REQUEST	$Tr + Tread + Twrite + Tss + Trs$	$Tread + Twrite + Tss + Trs$
CONTINUE	$Tenv + Tack + Tinit + Tresp + Tread + Twrite + Tss + Trs$	$Tread + Twrite + Tss + Trs$

**Figure IV.2.10.c :  $Trca$  suivant les modes de mesure**

Il apparaît que le mode le plus performant est NOBLOCKMODE/CONTINUE. Nous remarquons cependant, que malgré la fréquence d'obtention de mesures élevée de la part du client, celle ci ne représentent pas pour autant une nouvelle mesure capteur.

Pour permettre une distinction entre une ancienne et une nouvelle mesure, nous avons ajouté une date.

## 2.11. Datation des mesures

La datation d'un jeu de valeurs (mesure) est réalisé lors de l'écriture des données dans la structure des données de mesure. La date est représentée par un nombre entier, multiple de 10 ms.

Toutes les mesures peuvent ainsi être différenciées à 10 ms près. La date adjointe à chaque mesure est locale à l'application linux. Pour les applications nécessitant une datation des mesures (localisation, cartographie), il est conseillé d'évaluer les temps  $Tss$  et  $Tr$  et d'en déduire une date de mesure par rapport au système de commande.

les temps  $Tss$  et  $Tr$  dépendent de la taille des messages échangés et de la bande passante du réseau utilisé. Nous préconisons de procéder à l'échantillonnage de ces temps par des échanges de messages test lors de l'initialisation de l'application de commande, voire périodiquement pour d'éventuels recallages.

A titre indicatif, lors de mesures sur  $180^\circ$  avec une résolution angulaire de  $0.5^\circ$  et une bande passante de 10 Mo, on obtient les résultats pratiques suivants :

***T<sub>ss</sub> + T<sub>r</sub> = 7 à 8 ms***

### 3. ORCCAD

ORCCAD [28] est un logiciel développé à l'INRIA permettant de concevoir et de mettre en œuvre le contrôle et la commande d'un système robotique complexe. Il permet également la spécification et la validation des missions à réaliser par ce système. La philosophie d'ORCCAD est la définition formelle d'une action robotique. Ceci est basé sur les trois principes suivants :

- La plupart des actions physiques accomplies par un robot peuvent être ramenés à des problèmes de contrôle/commande et peuvent être résolus en temps réel ;
- L'action physique n'est pas suffisante pour définir complètement une action robotique : les instants de début et de fin doivent être considérés comme des réactions à des événements significatifs observés durant l'exécution de la tâche ;
- Pour une exécution temps réel sûre, un intérêt particulier sera donné à la spécification et à la validation.

#### 3.1. Introduction à ORCCAD

ORCCAD est principalement destiné aux applications temps réel critiques en robotique, dans lesquelles les aspects relevant de l'automatique (les asservissements, les commandes) sont amenés à interagir étroitement avec ceux manipulant des événements discrets. De tels systèmes sont souvent qualifiés d'hybrides.

Dans cette classe d'applications, ORCCAD s'adresse particulièrement aux systèmes présentant une forte interaction avec l'environnement par le biais de nombreux capteurs et actionneurs. Le contrôle/commande de ces systèmes est souvent embarqué, et le caractère critique de l'application apparaît dans le coût extraordinairement élevé attaché à une défaillance : l'impossibilité ou la difficulté d'intervention sur un sous-marin autonome à longue portée, sur un engin intervenant après un incident technologique majeur ou sur un véhicule planétaire rendent impératif la minimisation du risque de non réalisation de la mission. A cet effet, ORCCAD offre sûreté de programmation et possibilités de validation par simulation extensive ou vérification formelle.

A titre d'exemple, les applications aujourd'hui traitées en laboratoire avec ORCCAD vont du bras manipulateur au véhicule électrique automatique en passant par un robot sous-marin et un robot mobile terrestre avec une caméra.

Une application Orccad est découpée hiérarchiquement en trois entités :

- le Module qui est le grain minimal de l'utilisateur. Il permet de décrire :
  - du code de calcul (module de classe algorithmique),
  - des interfaces avec les ressources physiques telles que robot ou capteur (module de classe Ressource Physique),
  - des événements pour gérer la sécurité de l'entité de niveau supérieur (module de classe automate),

- la Tâche Robot (TR) qui décrit une action robotique élémentaire qui contient un asservissement et la gestion des événements pour assurer l'intégrité de l'action (erreurs typées, début et fin de l'action). Cette action est spécifiée en interconnectant un certain nombre de Modules.
- la Procédure Robot (PrR) qui décrit en langage Estérel (ou Maestro) la composition de tâche robot. Cette composition permet de spécifier l'application que doit réaliser le robot.

### 3.2. Les modules

Les modules peuvent être de types suivants :

- Ressource physique : ce type de module permet de réaliser la jonction entre l'environnement de spécification de Orccad et l'ensemble des actionneurs et capteurs. Ces modules disposent chacun d'une fonction d'initialisation et d'une fonction de fin. L'écriture ou la lecture de données sur chaque port est réalisée à travers des fonction associées.
- Module algorithmique : les modules algorithmiques servent à réaliser le traitement des données. Ces modules disposent d'une procédure d'initialisation, d'une procédure de fin et d'une procédure de traitement.
- Module comportemental : ce type de module permet de manipuler des signaux d'exception qui déclenchent l'arrêt ou le changement d'une tâche robot. Aucune procédure n'est associée à ce type de modules. Les ports d'entrée (et il n'y a que des ports d'entrée) ne véhiculent uniquement que des signaux de type événementiels servant à la surveillance du système, permettant l'arrêt d'une tâche robot et le passage à une nouvelle tâche.

#### 3.2.1. La Ressource Physique Cycab

La ressource physique Cycab (Cf. Figure IV. 3.2.1.a ) assure le passage entre ORCCAD et le Cycab proprement dit, on connecte ses différents ports qui appellent les fonctions pilotant les moteurs ou les capteurs. La ressource physique Cycab correspond, d'un point de vue spécification, au robot à commander.

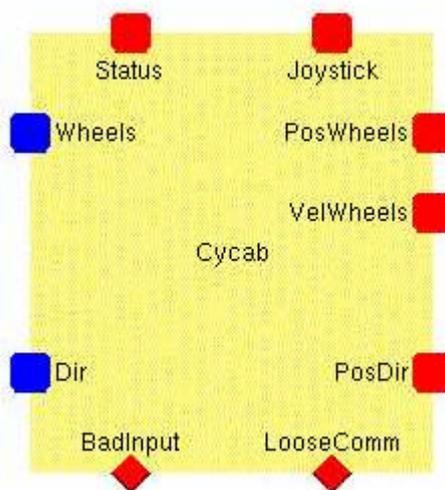


Figure IV.3.2.1.a : Ressource physique Cycab

C'est le point de départ des applications Cycab développées sous ORCCAD. Une application comportera donc cette ressource physique, les ressources physiques des capteurs utilisés, des modules algorithmiques, et un faisceau de connections.

Le tableau de la figure IV. 3.2.1.b énumère et décrit les divers ports d'entrée et de sortie de la ressource physique Cycab. Les données échangées sur un port sont véhiculées par des variables.

Lorsque un port est sollicité pour la lecture (Resp. l'écriture) d'une variable, la fonction associée est appelée pour mettre à jour cette variable avec la nouvelle valeur (Resp. appliquer la valeur de la variable au niveau de l'actionneur correspondant).

Ports		Variables			Fonction appelée	
Nom	Type	Nom	Type	Dimension	Nom	Description
Wheels	Entrée	WheelsI	Double	Vecteur(4)	putWheels	Réception de consignes en vitesse (radian par seconde)
Dir	Entrée	DirI	Double	Scalaire	putDir	Réception de consigne de direction (radian par seconde)
PosWheels	Sortie	PosWheelsO	Double	Vecteur(4)	getPosWheels	Lecture de la position des roues (radian)
VelWheels	Sortie	VelWheelsO	Double	Vecteur(4)	getVelWheels	Lecture de la vitesse de rotation des roues (radian par seconde)
PosDir	Sortie	PosDirO	Double	Scalaire	getPosDir	Lecture de la position de la direction (radian)
Status	Sortie	StatusO	Integer	Scalaire	getStatus	Lecture du statut renseignant sur le bon fonctionnement de la direction et des trains de roues avant et arrière
Joystick	Sortie	JoystickO	Double	Vecteur(2)	getJoystick	Lecture de la position (latérale et longitudinale) du joystick
BadInput	Événement	BadInputE				Signalisation du dépassement de la valeur maximale d'une consigne
LooseComm	Événement	LooseCommE				Signalisation d'une perte de communication sur le port CAN

**Figure IV. 3.2.1.b : Ports d'entrée et de sortie de la ressource physique Cycab**

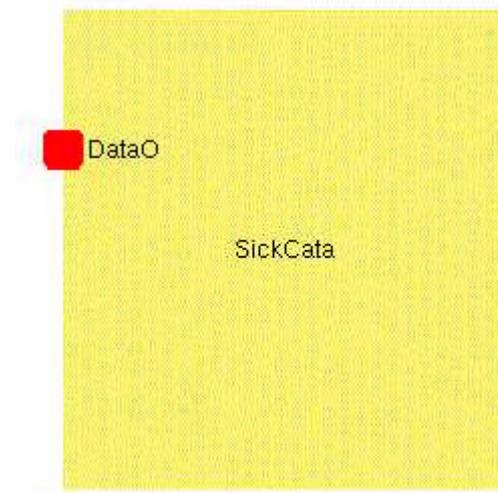
Les ports de type événement permettent de gérer les anomalies de fonctionnement.

### 3.2.2. La Ressource physique SickCata

La ressource physique SickCata permet d'intégrer le capteur télémétrique dans l'application ORCCAD de suivi. Ce module ne comporte pas de port d'entrée. Par rigueur de spécification, une seule ressource physique d'une tâche robot doit représenter le robot à commander. Seul une ressource physique robot peut donc posséder des ports d'entrée.

Il est donc "impossible" d'implémenter une ressource physique pour le capteur, qui permettrait de garantir les fonctionnalités nécessaires à toutes les missions. La ressource physique SickCata est donc spécifique à la tâche robot de suivi de véhicule. La configuration du capteur a lieu à l'initialisation du module et sera définitive pour toute la durée d'exécution de la tâche robot. Lors du développement du module pour le capteur télémétrique, cette

restriction a conduit à des remarques et suggestion d'évolution du logiciel Orccad. Plusieurs solutions ont été étudiées afin de conserver la philosophie initiale de Orccad et de garantir une adaptation de l'outil face aux spécificités de chaque application robotique.



**Figure IV.3.2.2 : Ressource physique SickCata**

Le seul port DataO fournit un vecteur de 7 données de type double :

- DataO[0] : distance  $d$  entre les deux véhicules (en centimètres) ;
- DataO[1] : Angle  $\theta$  entre les axes longitudinaux de chaque véhicule (en dixièmes de degrés) ;
- DataO[2] : Position angulaire  $\alpha$  du véhicule suivi par rapport au véhicule commandé (en dixièmes de degrés) ;
- DataO[3] : Coordonnée cartésienne  $X$  du véhicule suivi, relative au véhicule commandé (en cm) ;
- DataO[4] : Coordonnée cartésienne  $Y$  du véhicule suivi, relative au véhicule commandé (en cm) ;
- DataO[5] : Date d'enregistrement des données par l'interface (résolution de 10 ms) ;
- DataO[6] : Statut, 0 si pas de détection cible, 1 si une cible est détectée, 2 si seulement un catadioptré a été détecté.

Les données de ce vecteur sont mises à jour à chaque appel de la fonction *getData*.

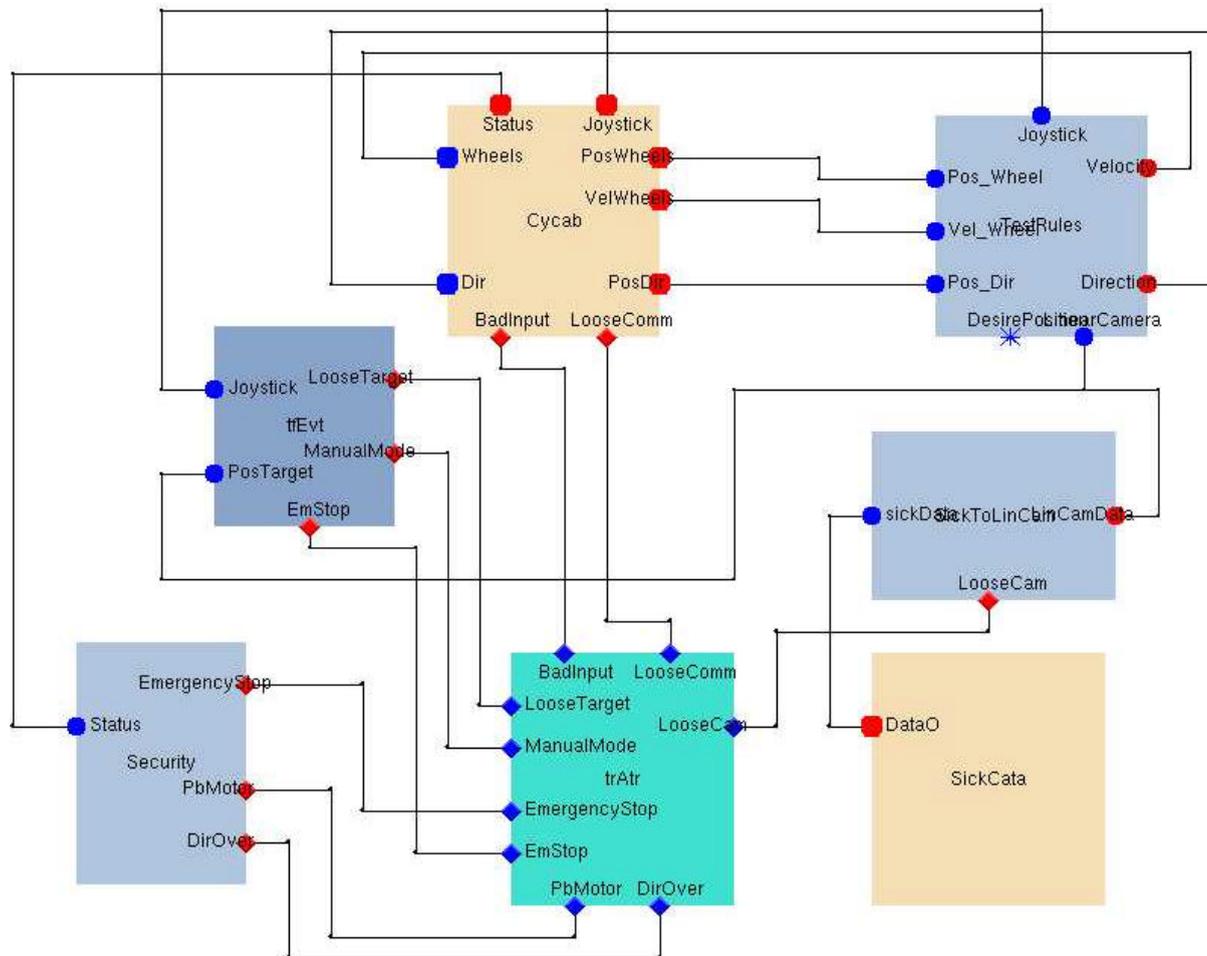
### 3.3. La Tâche Robot de suivi

On connecte via des ports d'entrées-sorties, la ressource physique et des modules algorithmiques, contenant la partie opératoire de l'application. Ici le suivi de véhicule.

Les spécifications de la mission de suivi exigent le passage en mode de conduite manuelle suite à la perte de la cible pendant un temps paramétrable. Cette fonctionnalité pourrait être réalisée par une deuxième tâche. On passerait alors de la tâche de suivi à une tâche de conduite manuelle lorsque la cible est perdue. Pour des raisons de simplification, les deux modes de fonctionnement (suivi et conduite manuelle) ont été implémentés dans une même tâche. Le passage du mode *suivi* au mode *conduite manuelle* est pris en charge par le module *tfEvt*.

On retrouve la ressource physique du *Cycab*, le module *SickToLinCam* qui adapte les données du capteur Sick en données identiques au format de la caméra, un module *Security* qui, en fonction du statut (mot représentant l'état du véhicule) envoie des messages au module *trAtr* qui représente la machine d'état de la tâche robot.

La boucle d'asservissement et de commande est assurée par le module *TestRules* qui reçoit en entrée les mesures odométriques du véhicule (*VelWheels*, *PosWheels* et *PosDir*) ainsi que les consignes provenant du joystick (pour la conduite manuelle).



**Figure IV.3.3 : Tâche robot de suivi**

Toutes les opérations liées à la commande sont programmées dans le module *TestRules*.

Remarque : La Procédure Robot n'est pas décrite ici car élémentaire puisque ne contenant qu'une seule tâche robot.

### 3.4. Compilation et intégration

Une fois que tous les modules ont été assemblés de façon graphique, Orccad permet de réaliser des vérifications et simulations afin de déceler des incohérences de spécification.

Une ultime étape permet de choisir la plate-forme cible (ici VxWorks) pour laquelle Orccad crée un exécutable. Une fois assemblée et compilée, l'application est prête pour le

chargement. L'intégration au niveau du véhicule Cycab consiste à charger en mémoire la librairie du client de sockets (qui permet de communiquer avec l'application linux) et l'application VxWorks de suivi.

L'application se déroule dans l'ordre suivant :

- Démarrage de l'application Interface Linux ;
- Démarrage du serveur de socket ;
- Démarrage de l'application VxWorks de suivi ;
  - Initialisation de l'application VxWorks,
  - Envoi d'une commande d'initialisation du capteur vers l'interface Linux,
- Initialisation du capteur et renvoi d'un message de confirmation;
- Démarrage de la partie opératoire de l'application de suivi.

Lors de l'exécution de l'application de suivi, les données échangées entre chaque module sont recalculées à une période fixe (ici 100 ms). A chaque période, toutes les consignes d'actionneurs sont mises à jour. Ainsi, le module *SickCata* appelle la fonction de demande de valeurs de mesure capteur toutes les 100 ms.

# Chapitre V

## RESULTATS

*Ce chapitre résume les principales caractéristiques du capteur en présentant les résultats obtenus lors d'expérimentations. L'analyse des performances est orientée autour de l'application de suivi de véhicule, en comparaison avec les performances obtenues à l'aide d'une caméra linéaire.*

## 1. PERFORMANCES DU SYSTEME DE DETECTION DE CIBLE

Lors des premiers essais de qualification du capteur, des distances de plus de 80 m ont pu être enregistrées. Cependant, la portée propre du capteur est remise en question, par la particularité du système à détecter (système à deux catadioptrés).

Les paramètres tels que la largeur des catadioptrés, la distance entre deux catadioptrés, l'angle d'incidence, la résolution angulaire, la largeur des spots, la fréquence de rafraîchissement des données et la nature de l'environnement d'expérimentation (contraintes temps réel) sont autant de facteurs à prendre en compte pour garantir des performances réelles.

L'étude du système de suivi de véhicule a nécessité une évaluation constante de l'impact relatif de ces paramètres.

Des choix judicieux ont dû être faits en ce qui concerne :

- la géométrie et le dimensionnement de la cible ;
- la configuration du capteur ;
- les moyens de développements logiciels utilisés
- le respect du système de commande initial

L'ensemble de ces choix ont permis d'aboutir à une fonctionnalité opérationnelle dont les performances sont décrites dans les prochaines sections.

### 1.1. Distance de détection de la cible

La qualité de détection de la cible est directement liée à la qualité de détection des catadioptrés utilisés, qui dépend de :

- **Le niveau de réflexivité du catadioptré** : nous avons testé plusieurs types de catadioptrés. Des catadioptrés sur support souple fournis par la société Sick ont délivrés les meilleurs résultats ;
- **Le dimensionnement des catadioptrés** : un compromis a dû être fait entre la largeur des catadioptrés et la distance inter-catadioptrés. En effet, un catadioptré large présente une qualité de détection supérieure à un catadioptré étroit. Cependant, il est apparu le besoin éventuel d'utiliser des systèmes de cible avec plus de deux catadioptrés afin de différencier plusieurs véhicules (Cf. Chapitre III.4.3) ;
- **La résolution angulaire du capteur** : l'utilisation d'une résolution angulaire de  $0,25^\circ$  aurait permis d'augmenter la qualité de détection des catadioptrés. Dans ce mode de fonctionnement, le capteur ne permet pas de mesures sur plus de  $100^\circ$ . Afin de préserver un angle de détection de  $180^\circ$ , nous avons opté pour une résolution de  $0,5^\circ$ .

Un compromis entre largeur de catadioptré (Cf. Chapitre III.4.4) et distance maximale de détection a abouti à une distance maximale de détection de la cible de 20 m.

## 1.2. Angle de détection de la cible

Deux angles sont à considérer :

- La position angulaire  $\alpha$  de la cible ;
- L'angle d'orientation  $\theta$  de la cible.

La position angulaire minimale et maximale est fixée par le choix de configuration du capteur à  $180^\circ$ .

Les limites d'angle d'orientation de la cible dépendent de la largeur et des caractéristiques géométriques des catadioptrés. L'utilisation de catadioptrés cylindriques plutôt que plats, aurait permis d'élargir ces limites. Cependant, le calcul du barycentre de chaque catadioptré aurait demandé sensiblement plus de calculs.

Avec des catadioptrés plats et à une distance de fonctionnement courante de 8 m, nous avons mesuré l'influence de l'angle  $\theta$  sur la qualité de détection des catadioptrés.

Les courbes de la figure V.1.2 présentent le Niveau de réflexion globale  $N_i$  (Cf. Chapitre III.4.2) des catadioptrés  $i$  en fonction de l'angle  $\theta$ .

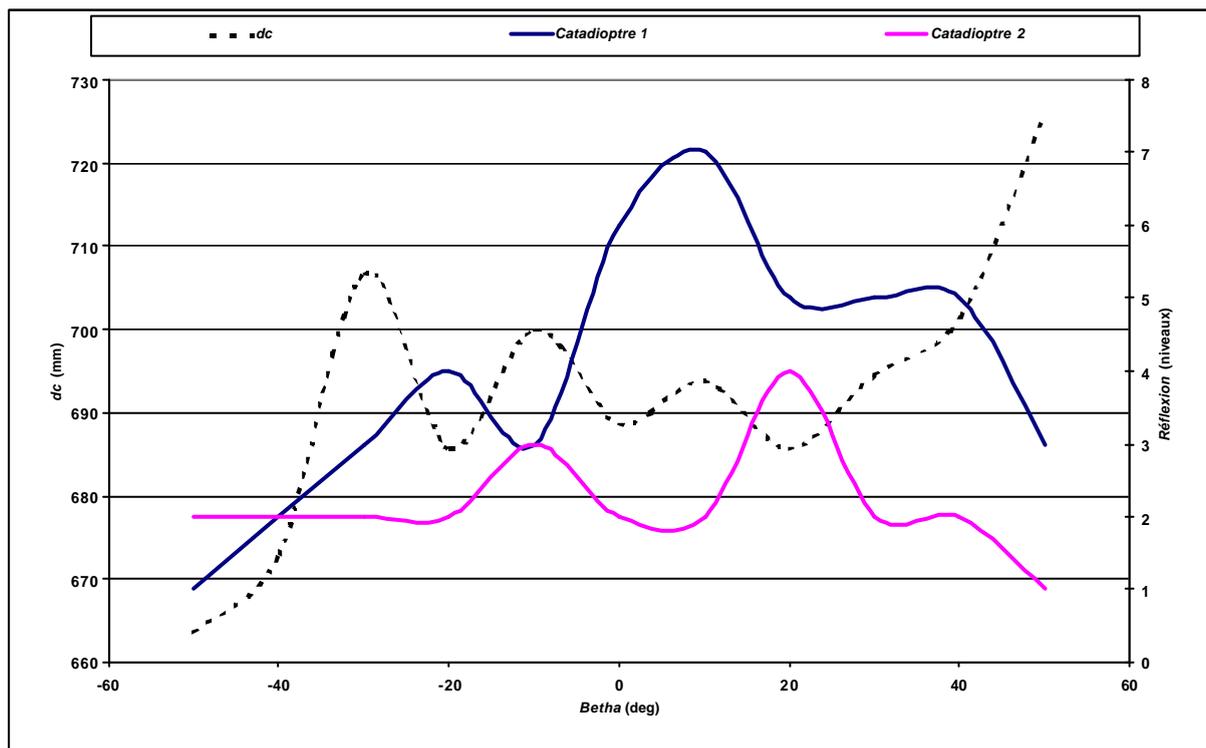


Figure V.1.2 : Angle d'orientation de la cible

La courbe de  $dc$ , distance calculée entre les deux catadioptrés, permet d'évaluer l'impact sur la qualité des mesures. La distance  $dc$  réelle étant de 695 mm, nous pouvons affirmer que  $\theta$  varie de  $\pm 40^\circ$  avec une erreur sur  $dc$  inférieure à  $\pm 2,5\%$ . Une perte totale de la cible est constatée vers  $\pm 50^\circ$ .

Remarque : les variations de type sinusoïdales des courbes sont dues à l'écartement des spots (Cf. Chapitre III.4.5).

### **1.3. Invariance au parasites**

Nous avons vérifié que lors des essais en extérieur, la présence d'un fort ensoleillement (qui pose un problème lors de l'utilisation de la caméra linéaire) n'est pas un élément perturbateur.

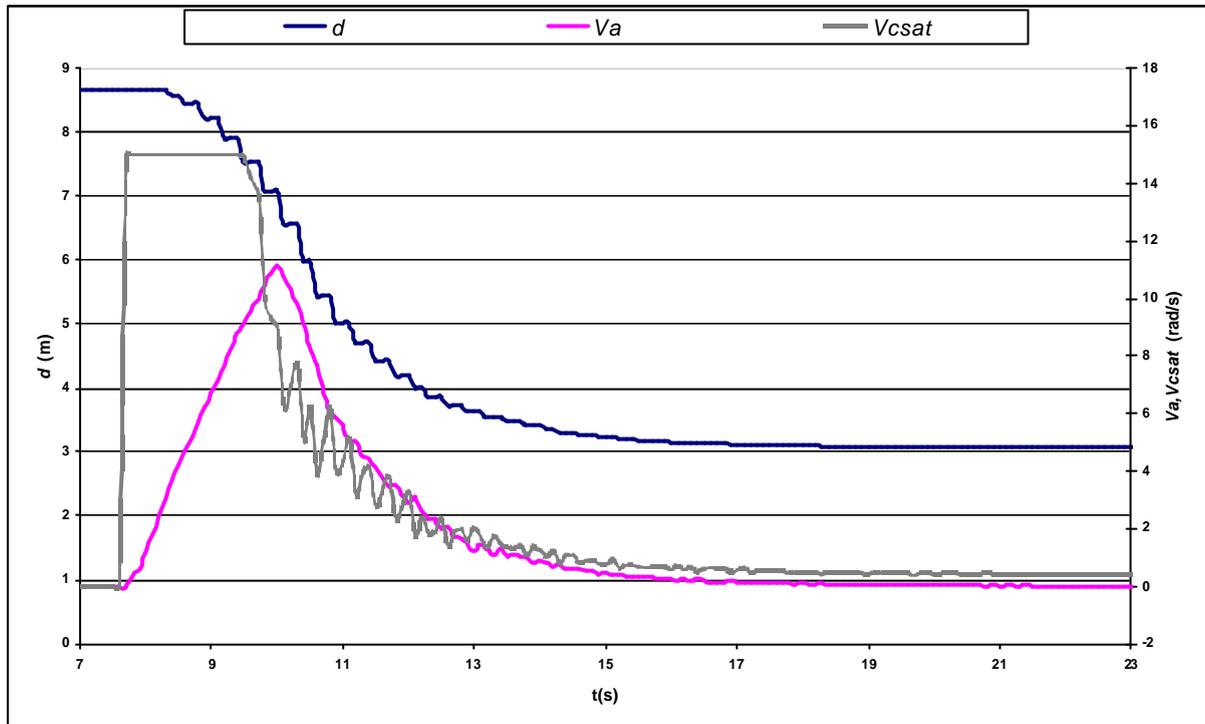
### **1.4. Détection verticale**

Comme nous l'avions prévu au Chapitre III.4.4, la détection des catadioptrés sur un plan vertical pose des problèmes lors de fortes perturbations du terrain. Nos essais de suivi ont été réalisés avec succès sur le parking de l'INRIA Rhône-Alpes. Sur des terrains plus accidentés, il est conseillé d'augmenter la hauteur des catadioptrés.

## 2. EXPERIMENTATIONS DU SUIVI DE VEHICULE

Pour la validation du système de suivi, nous avons comme contrainte de réutiliser la commande existante (Cf. Chapitre III.5).

Afin d'observer les caractéristiques des asservissements, nous présentons les courbes résultantes du suivi d'une cible inerte.



**Figure V.2.a : Asservissement longitudinal**

Le véhicule Cycab est placé à 8,64 m de la cible. Nous avons pris soin, pour cet essai, de placer le véhicule de telle manière que l'angle de positionnement de la cible soit nul, ceci afin de n'observer que l'asservissement longitudinal.

Nous vérifions un dépassement nul et une légère erreur (6,9 cm) due à la saturation inférieure de la consigne  $V_c$  (Figure V.2.a).

Les courbes de la Figure V.2.b montrent la réponse de l'asservissement latéral. Pour cet essai, la cible inerte est placée dans un angle de  $44,76^\circ$  par rapport à l'axe longitudinal du véhicule Cycab.

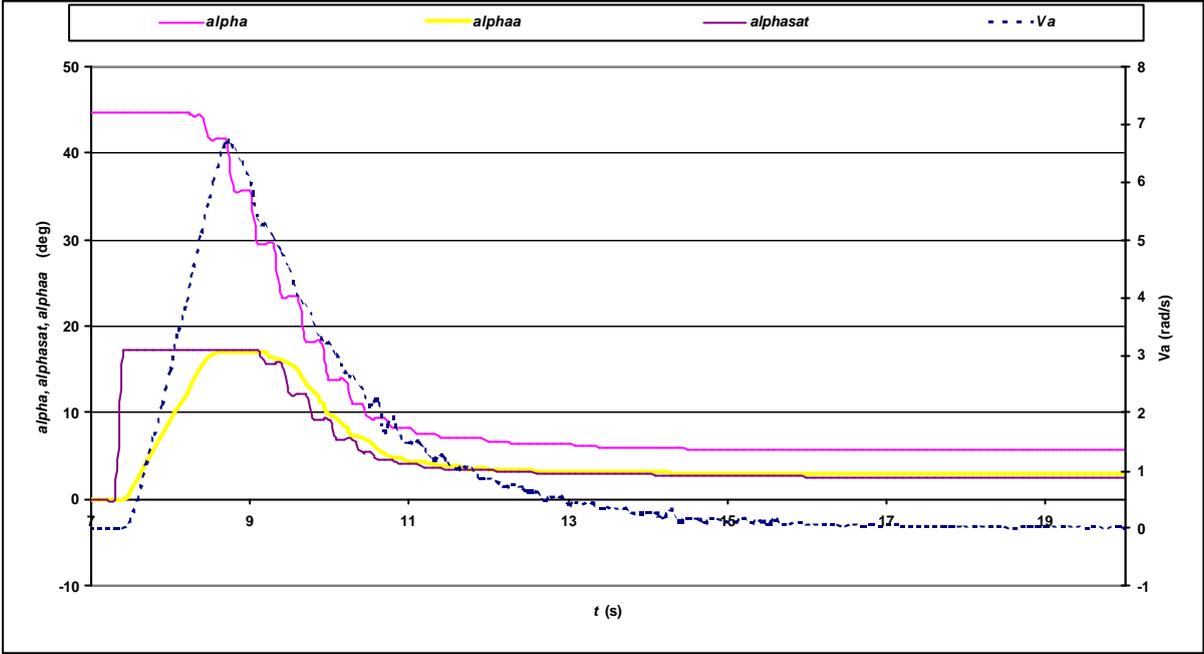


Figure V. 2.b : Asservissement latéral de direction

## Conclusion

---

Notre travail a été réalisé dans le cadre des missions du service des moyens robotiques de l'INRIA Rhône-Alpes. Dans ce mémoire nous avons proposé une solution orientée capteur pour obtenir des données sensorielles télémétriques complémentaires à celles délivrées par les capteurs utilisés habituellement lors d'expérimentations pratiques dans le domaine de la recherche sur l'autonomie des véhicules.

Il s'agit donc d'intégrer un capteur télémétrique à balayage laser sur un véhicule électrique de type Cycab afin d'obtenir plus d'informations sur l'environnement du véhicule. La problématique revient à qualifier le capteur et à argumenter des choix d'intégration et de mise en œuvre. Le but est une exploitation expérimentale dans plusieurs applications de robotique mobile. Nous avons constaté que les applications pour ce type de capteur sont extrêmement larges. Les travaux de qualification ont donc été basés sur la spécification et les besoins liés à des applications déjà développés à l'INRIA.

Différentes expérimentations ont montré les apports de ce type de capteur dans l'estimation des erreurs de l'odométrie et pour des applications spécifiques telles que le convoi de véhicule par accrochage immatériel, l'anti-collision, le positionnement ou la cartographie de l'environnement.

Nous présentons une solution originale permettant de remplacer une caméra linéaire dans une application de convoi de véhicules à accrochage immatériel. Les spécifications de l'application d'origine ont été respectées. L'originalité de la solution repose sur l'utilisation d'un système cible à base de catadioptrès. Notre solution a permis de mettre en évidence l'intérêt de ce type de capteur par l'augmentation des performances.

Le contrôle du capteur - qui dispose de son propre micro-contrôleur -, le traitement et l'interprétation des données sont réalisées sur une plate-forme Linux. Une architecture client/serveur à base de socket permet l'échange des données avec l'application temps réel (sous VxWorks).

L'étude du système de suivi de véhicule a nécessité une évaluation constante de l'impact relatif des paramètres liés à l'environnement et à la plate-forme d'expérimentation. Nous discutons et argumentons les choix faits en terme d'intégration de la solution.

La solution a été validée lors d'expérimentations réelles, confirmant les hypothèses et résultats théoriques de l'étude de qualification. Le capteur télémétrique à balayage laser apporte des solutions à valeur ajoutée dans des applications nécessitant des informations sur l'environnement extéroceptif. La richesse des informations délivrées par ce capteur implique par contre une connaissance approfondie de son fonctionnement ainsi que des contraintes du système. A cet effet, le présent mémoire fournit les éléments théoriques et pratiques nécessaires à l'étude particulière que nécessite chaque application.

Suite au succès d'implémentation du capteur sur le Cycab pour le suivi de véhicules, d'autres applications issues de projets de recherche tel que SHARP sont en cours d'étude et de développement. Ainsi, des travaux sur la cartographie et sur la localisation ont débutés et devraient permettre d'avoir, dans un plan horizontal, des données précises – jusqu'à 1 cm – de l'environnement.

Nos travaux ont permis d'offrir des perspectives nouvelles en terme de validation de concepts théoriques, de performance d'applications réelles et d'équipement des plates-formes expérimentales.

## Annexe

*Dans cette annexe on trouvera, dans une première partie, une description des différents éléments logiciels (structures, fonctions, variables globales) qui ont été développés afin de fournir toutes les fonctionnalités nécessaires à l'exploitation du capteur laser en environnement hétérogène. Une deuxième partie fournit des données sur la cinématique du Cycab.*

## 1. INTERFACE CAPTEUR

L'interface capteur a été développée en vue d'une exploitation en environnement Linux. La compilation du code source a été réalisée avec le compilateur gcc GNU (egcs-2.7.2.3) et un noyau Linux version 2.0.36.

### 1.1. Liste des télégrammes

#### 1.1.1. Télégrammes du PC vers le capteur

<b>Télégramme</b>	<b>Code hexa</b>	<b>Description</b>
INIT_TGM	0x10	Commande d'initialisation du capteur
BM_TGM	0x20	Commande de changement de mode
MWANF_TGM	0x30	Commande de demande de mesures
SSANF_TGM	0x31	Commande de demande de statut du capteur
ERRANF_TGM	0x32	Commande de demande d'information sur l'erreur
MMWANF_TGM	0x36	Commande de demande de mesures moyennées
MWPANF_TGM	0x37	Commande de demande de mesures partitionnées
TYPANF_TGM	0x3A	Commande de demande du type de capteur
VARDEF_TGM	0x3B	Commande de modification du type de mesure du capteur
MMWPANF_TGM	0x3F	Commande de demande de mesures moyennées et partitionnées
FIELDKFG_TGM	0x40	Commande de modification de la configuration des champs du capteur
FIELDCHG_TGM	0x41	Commande de modification du groupe de champs actif du capteur
FIELDDEF_TGM	0x45	Commande de demande de configuration des champs du capteur
LMSKFGANF_TGM	0x74	Commande de demande de configuration du capteur
MWXYANF_TGM	0x76	Commande de demande de mesures en coordonnées cartésienne
LMSKFGDEF_TGM	0x77	Commande de modification de la configuration du capteur (LMS-2xx-xxx06)

## 1.1.2. Télégrammes du capteur vers le PC

<b>Télégramme</b>	<b>Code hexa</b>	<b>Description</b>
PWON_TGM	0x90	Message envoyé par le capteur après la mise sous tension
INIT_ACK_TGM	0x91	Réponse au télégramme INIT_TGM
NACK_TGM	0x92	Refus de la commande
BMACK_TGM	0xA0	Réponse au télégramme BM_TGM
MW_TGM	0xB0	Réponse au télégramme MWANF_TGM
SS_TGM	0xB1	Réponse au télégramme SSANF_TGM
ERR_TGM	0xB2	Réponse au télégramme ERRANF_TGM
MMW_TGM	0xB6	Réponse au télégramme MMWANF_TGM
MWP_TGM	0xB7	Réponse au télégramme MWPANF_TGM
TYPE_TGM	0xBA	Réponse au télégramme TYPANF_TGM
VARDEFACK_TGM	0xBB	Réponse au télégramme VARDEF_TGM
MMWP_TGM	0xBF	Réponse au télégramme MMWPANF_TGM
FIELDACK_TGM	0xC0	Réponse au télégramme FIELDKFG_TGM
FIELDCHGACK_TGM	0xC1	Réponse au télégramme FIELDCHG_TGM
FIELDDAT_TGM	0xC5	Réponse au télégramme FIELDDEF_TGM
LMIKFGACK_TGM	0xF0	Réponse au télégramme LMIKFGDEF_TGM
LMSKFG_TGM	0xF4	Réponse au télégramme LMSKFGANF_TGM
MWXY_TGM	0xF6	Réponse au télégramme MWXYANF_TGM
LMSKFGACK_TGM	0xF7	Réponse au télégramme LMSKFGDEF_TGM

## 1.2. Fonctionnalités du menu utilisateur

menu level I	menu level II	menu level III	N°	
Exit program	upper menu level		0 0 0 0	
Setting Communication port	Open (9600)		1 1 1	
	Open (38400)		1 1 2	
	Close port		1 1 3	
	Set Baudrate (9600)		1 1 4	
	Set Baudrate (38400)		1 1 5	
Sensor Initialisation and reset			1 2	
Select or change Sensor Operating mode	Installation mode for configuration and setting parameters		1 3 1	
	Calibration mode		1 3 2	
	Reset to default password for installation and maintenance		1 3 3	
	Diagnostic mode		1 3 4	
	Monitoring mode	Minimum measured values per segment(continuously)		1 3 5 1
		Minimum measured values per segment(on request)		1 3 5 2
				1 3 5 3
		Minimum perpendicular distance to sensor (continuously)		1 3 5 4
		Minimum perpendicular distance to sensor (on request)		1 3 5 4
		All measured values of a scan (continuously)		1 3 5 5
		All measured values of a scan (on request)		1 3 5 6
		Averaged measured values (continuously)		1 3 5 7
		Partitioned measured values (continuously)		1 3 5 8
	Averaged partitioned measured values (continuously)		1 3 5 9	
	Baudrate	Set Baudrate to 9600 baud		1 3 6 1
		Set Baudrate to 19200 baud		1 3 6 2
		Set Baudrate to 38400 baud		1 3 6 3
		Set Baudrate to 500000 baud		1 3 6 4
	Change Sensor Configuration	Change Sensor Configuration	Global settings	1 4 1 1
			Restart settings	1 4 1 2
			Field A settings	1 4 1 3
			Field B settings	1 4 1 4
			Field C settings	1 4 1 5
Evaluation settings			1 4 1 6	
Change of Sensor variant			1 4 2	
Configuration of fields A, B or C			1 4 3	
Change active field group			1 4 4	
Teach mode for field configuration			1 4 5	
Configuration of a dynamic rectangular field			1 4 6	
Configuration of a dynamic segmented field			1 4 7	
Activation / deactivation of outputs			1 4 8	
Define permanent baudrate or sensor type			1 4 9	
Request for configuration values		Read Sensor status		1 5 1
		Read Sensor configuration		1 5 2
		Read Sensor Type		1 5 3
		Read Sensor fields configurations		1 5 4
Request for mesured values		Request for mesured values	only send minimum measured value per segment	1 6 1 0
			send all of scan's measured values	1 6 1 1
	send minimum perpendicular distance from sensor		1 6 1 2	
	reserved		1 6 1 3	
	reserved		1 6 1 4	
	send minimum measured values for defined number of segments		1 6 1 5	
	send every 2nd measured value		1 6 1 6	
	send every 20th measured value		1 6 1 7	
	Request for average mesured values		1 6 2	
	Request for partitioned mesured values		1 6 3	
	Request for average partitioned mesured values		1 6 4	
	Request for mesured values in Cartesian co-ordinates		1 6 5	
	Reserved		1 7 1	
	Handle Data	Save data set (last in use)		1 7 2
Save Sensor Configuration to file			1 7 3	
Load Configuration from file and send to Sensor			1 7 4	
Special functionalities			1 8	
Options	Run mode	Debug mode	1 9 1 0	
		Info mode	1 9 1 1	
		Silent mode	1 9 1 2	
	Password settings		1 9 2	
	Measure thread mode	Request mode		1 9 3 0
Continue mode			1 9 3 1	
Sleep mode			1 9 3 2	
Measure block-if-old mode	block client if old values		1 9 4 0	
	no blocking		1 9 4 1	

1.3. Thread serveur de socket

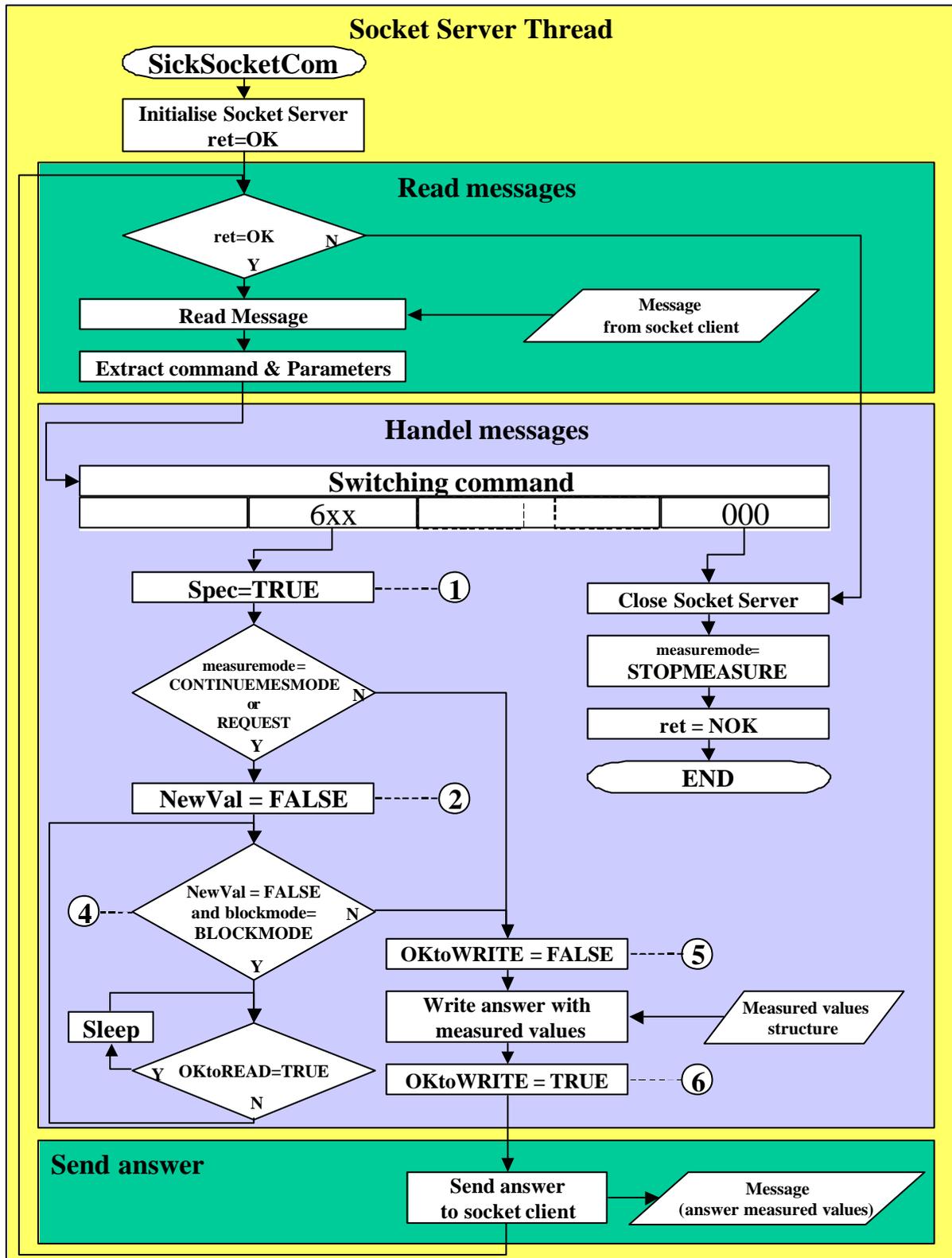


Figure A.1.3 : Thread serveur de socket

1.4. Thread de mesure

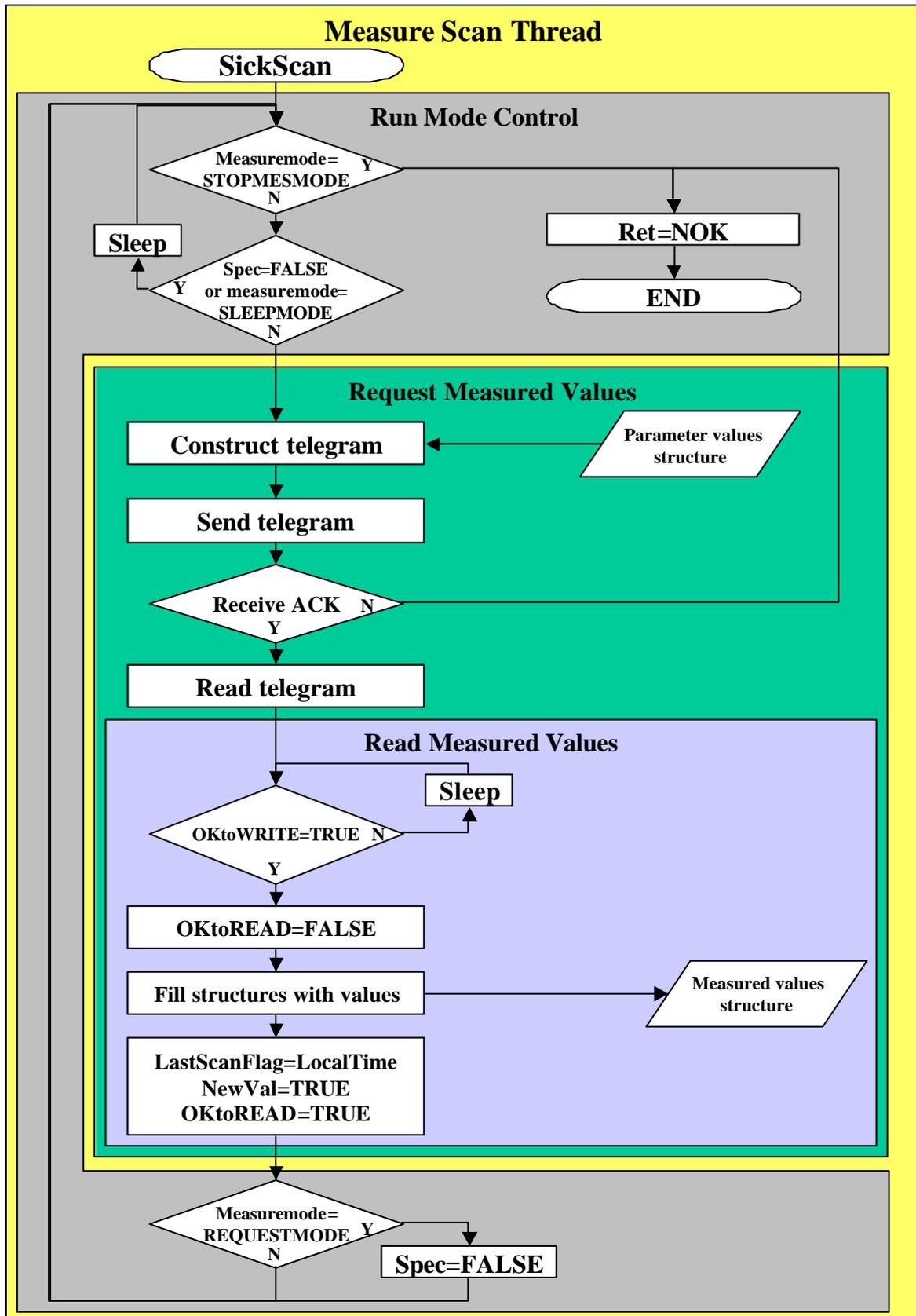


Figure A.1.4 : Thread de mesure

## 2. DONNEES EN VUE D'EXPERIMENTATIONS

Ce chapitre donne des éléments théoriques sur le modèle cinématique du véhicule.

### 2.1. Cinématique du Cycab

#### 2.1.1. Notations

Soient  $F = (x_F, y_F)$  (resp.  $R = (x_R, y_R)$ ) les coordonnées du centre de l'essieu avant (resp. arrière). De même,  $v_F$  (resp.  $v_R$ ) représente la vitesse instantanée au point  $F$  (resp.  $R$ ).

$f$  représente le braquage moyen des roues de l'essieu avant.  $l_w$  correspond à l'empattement du véhicule.  $q$  caractérise l'angle que fait l'axe longitudinal du véhicule avec l'axe des abscisses du repère lié à l'environnement. Enfin,  $r_F$  (resp.  $r_R$ ) est le rayon de giration instantané associé au point  $F$  (resp.  $R$ ).

#### 2.1.2. Modélisation

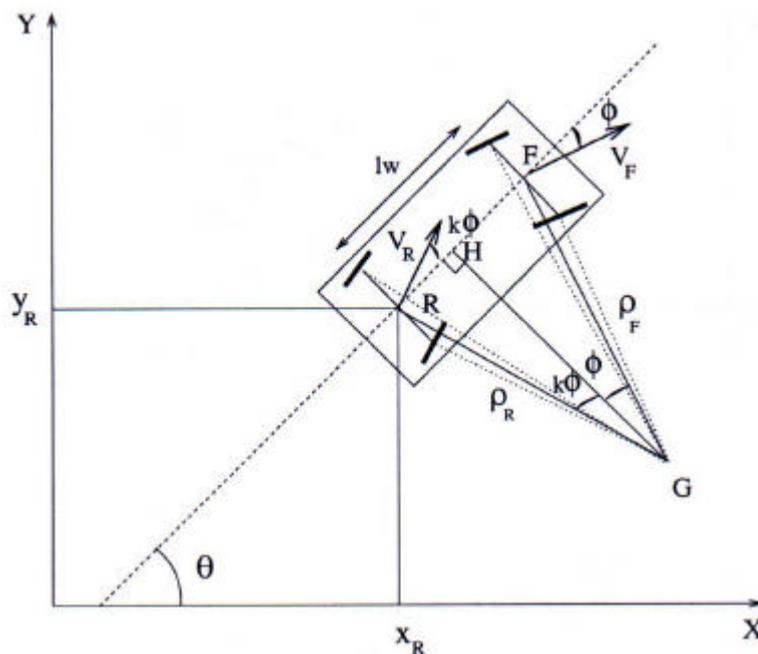


Figure A.2.1.2 : Modélisation du véhicule Cycab

Les roues avant et arrière sont directrices avec un coefficient de proportionnalité entre les deux braquages : si le braquage avant est de  $f$ , le braquage arrière est de  $kf$  (avec  $k \in [0,1]$ ). La valeur de  $k$  sera à déterminer par la suite. On introduit le point  $H$  défini comme la projection orthogonale du centre de giration instantané  $G$  sur l'axe longitudinal du véhicule. La distance  $GH$  peut être exprimée de la manière suivante :

$$GH = \frac{RH}{|\tan(kf)|} = \frac{HF}{|\tan(f)|} \quad (1)$$

ce qui permet d'obtenir :

$$RH = HF * \frac{\tan(kf)}{\tan(f)} \quad (2)$$

de plus :

$$RH + HF = l_w \quad (3)$$

On en déduit les expressions de RH et HF en fonction de  $l_w$  et  $f$  :

$$\begin{cases} RH = \frac{l_w}{1 + \frac{\tan(f)}{\tan(kf)}} = l_w * \frac{\cos(f) * \sin(kf)}{\sin(f + kf)} \\ HF = \frac{l_w}{1 + \frac{\tan(kf)}{\tan(f)}} = l_w * \frac{\cos(kf) * \sin(f)}{\sin(f + kf)} \end{cases} \quad (4)$$

Les rayons de braquage  $r_F$  et  $r_R$  sont donnés par :

$$\begin{cases} r_R = \frac{RH}{|\sin(kf)|} = l_w * \frac{\cos(f)}{|\sin(f + kf)|} \\ r_F = \frac{HF}{|\sin(f)|} = l_w * \frac{\cos(kf)}{|\sin(f + kf)|} \end{cases} \quad (5)$$

La vitesse de rotation instantanée peut s'écrire :

$$\dot{\mathbf{q}} = \frac{v_R}{r_R} = \frac{v_F}{r_F} \quad (6)$$

A partir de (5) et (6), on déduit :

$$v_R = v_F * \frac{r_R}{r_F} = v_F * \frac{\cos(f)}{\cos(kf)} \quad (7)$$

Les équations du mouvement en  $R$  sont obtenues de manière géométrique :

$$\begin{cases} \dot{x}_R = v_R * \cos(\mathbf{q} + k\mathbf{f}) \\ \dot{y}_R = v_R * \sin(\mathbf{q} + k\mathbf{f}) \\ \dot{\mathbf{q}} = v_R * \frac{\sin(f + kf)}{l_w * \cos(f)} \end{cases} \quad \text{ou} \quad \begin{cases} \dot{x}_R = v_F * \frac{\cos(f)}{\cos(kf)} * \cos(\mathbf{q} + k\mathbf{f}) \\ \dot{y}_R = v_F * \frac{\cos(f)}{\cos(kf)} * \sin(\mathbf{q} + k\mathbf{f}) \\ \dot{\mathbf{q}} = v_F * \frac{\sin(f + kf)}{l_w * \cos(f)} \end{cases} \quad (8)$$

De même, on en déduit les équations du mouvement en  $F$ :

$$\left\{ \begin{array}{l} \dot{x}_F = v_F * \cos(\mathbf{q} + \mathbf{f}) \\ \dot{y}_F = v_F * \sin(\mathbf{q} + \mathbf{f}) \\ \dot{\mathbf{q}} = v_R * \frac{\sin(\mathbf{f} + k\mathbf{f})}{l_w * \cos(\mathbf{f})} \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} \dot{x}_F = v_R * \frac{\cos(k\mathbf{f})}{\cos(\mathbf{f})} * \cos(\mathbf{q} + \mathbf{f}) \\ \dot{y}_F = v_R * \frac{\cos(k\mathbf{f})}{\cos(\mathbf{f})} * \sin(\mathbf{q} + \mathbf{f}) \\ \dot{\mathbf{q}} = v_F * \frac{\sin(\mathbf{f} + k\mathbf{f})}{l_w * \cos(\mathbf{f})} \end{array} \right. \quad (9)$$

remarque: valeurs particulières de  $k$

- $k=0$  : on est dans le cas des roues arrières fixes;
- $k=1$  : les points  $R$  et  $F$  ont exactement la même cinématique. Cela est également vérifié à partir des équations (7) et (9).

### 2.1.3. Détermination des vitesses de rotation des roues

On définit  $L$  comme la voie du véhicule (i.e. longueur des essieux) et  $r$  comme le rayon d'une roue. Les points  $RL$ ,  $RR$ ,  $FL$  et  $FR$  correspondent aux centres des roues arrière gauche, arrière droit, avant gauche et avant droit.

A partir de la figure A.2.1.3.a, on peut aisément déduire :

$$\left\{ \begin{array}{l} x_{RL} = x_R - \frac{L}{2} * \sin(\mathbf{q}) \\ y_{RL} = y_R + \frac{L}{2} * \cos(\mathbf{q}) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_{RR} = x_R + \frac{L}{2} * \sin(\mathbf{q}) \\ y_{RR} = y_R - \frac{L}{2} * \cos(\mathbf{q}) \end{array} \right. \quad (10)$$

pour les roues arrières.

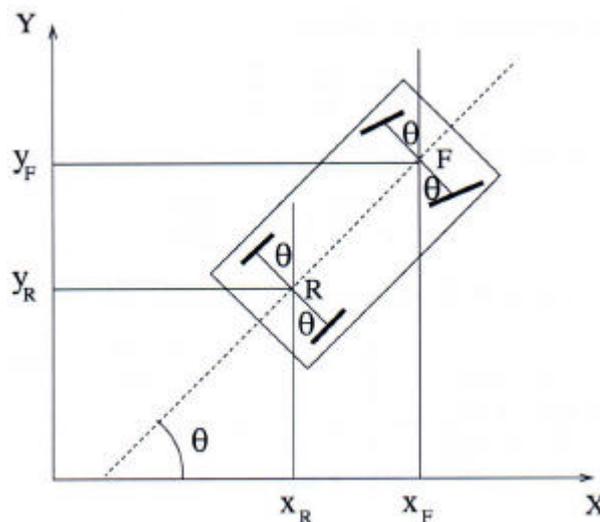


Figure A.2.1.3.a : Mise en évidence de l'orientation  $q$  au centre des roues

$$\left\{ \begin{array}{l} x_{FL} = x_F - \frac{L}{2} * \sin(\mathbf{q}) \\ y_{FL} = y_F + \frac{L}{2} * \cos(\mathbf{q}) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} x_{FR} = x_F + \frac{L}{2} * \sin(\mathbf{q}) \\ y_{FR} = y_F - \frac{L}{2} * \cos(\mathbf{q}) \end{array} \right. \quad (11)$$

pour les roues avant.

On en déduit les vitesses en ces points :

$$\left\{ \begin{array}{l} \dot{x}_{RL} = \dot{x}_R - \frac{L}{2} * \dot{\mathbf{q}} * \cos(\mathbf{q}) \\ \dot{y}_{RL} = \dot{y}_R + \frac{L}{2} * \dot{\mathbf{q}} * \sin(\mathbf{q}) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \dot{x}_{RR} = \dot{x}_R + \frac{L}{2} * \dot{\mathbf{q}} * \cos(\mathbf{q}) \\ \dot{y}_{RR} = \dot{y}_R - \frac{L}{2} * \dot{\mathbf{q}} * \sin(\mathbf{q}) \end{array} \right. \quad (12)$$

pour les roues arrière.

$$\left\{ \begin{array}{l} \dot{x}_{FL} = \dot{x}_F - \frac{L}{2} * \dot{\mathbf{q}} * \cos(\mathbf{q}) \\ \dot{y}_{FL} = \dot{y}_F - \frac{L}{2} * \dot{\mathbf{q}} * \sin(\mathbf{q}) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \dot{x}_{FR} = \dot{x}_F + \frac{L}{2} * \dot{\mathbf{q}} * \cos(\mathbf{q}) \\ \dot{y}_{FR} = \dot{y}_F + \frac{L}{2} * \dot{\mathbf{q}} * \sin(\mathbf{q}) \end{array} \right. \quad (13)$$

pour les roues avant.

Dans tous les cas, les vitesses des roues correspondent à la vitesse du centre de l'essieu concerné (i.e.  $v_F$  ou  $v_R$ ) ou avec :

$$\Delta v = \frac{L}{2} * \dot{\mathbf{q}} \quad (14)$$

On en déduit donc les vitesses linéaires et de rotation des 4 roues :

$$\left\{ \begin{array}{l} v_{RL} = v_R - \frac{L}{2} * \dot{\mathbf{q}} \\ v_{RR} = v_R + \frac{L}{2} * \dot{\mathbf{q}} \\ v_{FL} = v_F - \frac{L}{2} * \dot{\mathbf{q}} \\ v_{FR} = v_F + \frac{L}{2} * \dot{\mathbf{q}} \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \dot{\mathbf{y}}_{RL} = \dot{\mathbf{y}}_R - \frac{L}{2r} * \dot{\mathbf{q}} \\ \dot{\mathbf{y}}_{RR} = \dot{\mathbf{y}}_R + \frac{L}{2r} * \dot{\mathbf{q}} \\ \dot{\mathbf{y}}_{FL} = \dot{\mathbf{y}}_F - \frac{L}{2r} * \dot{\mathbf{q}} \\ \dot{\mathbf{y}}_{FR} = \dot{\mathbf{y}}_F + \frac{L}{2r} * \dot{\mathbf{q}} \end{array} \right. \quad (15) \text{et} (16)$$

avec  $\dot{\mathbf{y}}_R = \frac{v_R}{r}$  et  $\dot{\mathbf{y}}_F = \frac{v_F}{r}$ .

### Détermination du paramètre $k$ :

La détermination du rapport entre les braquages moyens des roues avant et arrière (ou plutôt entre les braquages des roues virtuelles situées respectivement au centre de chacun des essieux) n'est pas chose aisée. Un programme a alors été écrit pour déterminer ce coefficient. La figure A.2.1.3.c montre, entre autre, l'évolution de  $k$  en fonction de l'angle  $f$

correspondant à l'orientation de la manivelle avant de direction par rapport à l'axe longitudinal du véhicule (Cf. figure A.2.1.3.b).

Comme on peut le voir, ce coefficient n'est pas constant mais il semble raisonnable de le ramener à une valeur constante soit :

$$k = 0.69 \quad (17)$$

sur l'ensemble des valeurs admissibles pour  $f$  c'est à dire environ  $[-0.4, 0.4]$ .

remarque: La notation  $f$  caractérise indifféremment le braquage moyen avant (i.e. braquage de la roue virtuelle située au centre de l'essieu avant) et l'angle entre la manivelle avant de direction et l'axe longitudinal du véhicule (cf. figure A.2.1.3.b).

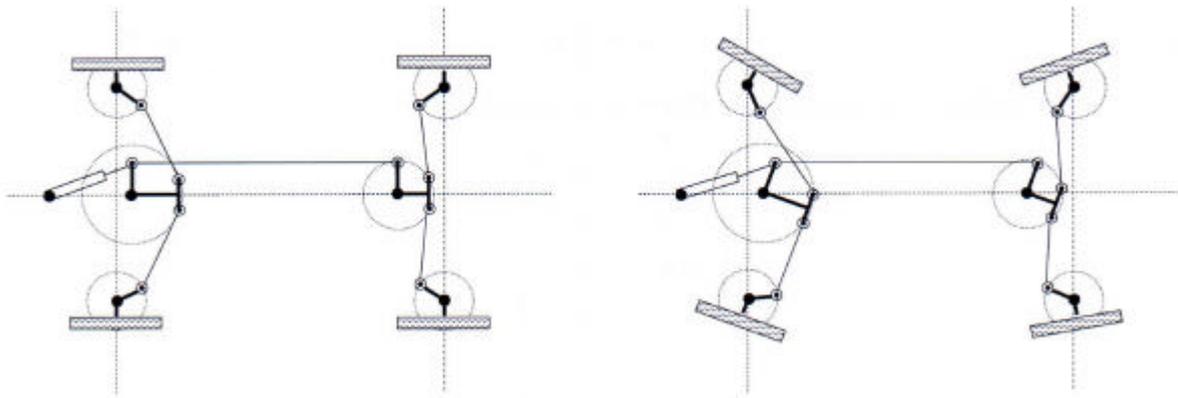


Figure A.2.1.3.b : Mécanisme de direction du Cycab

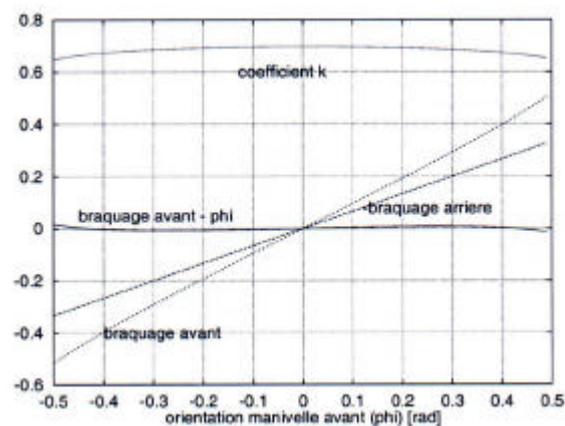
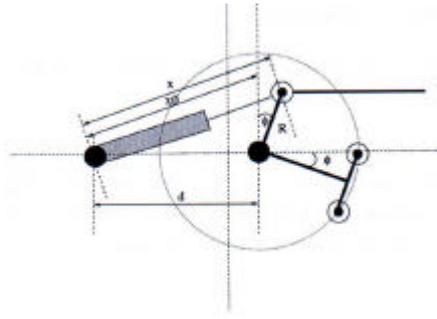


Figure A.2.1.3.c : Evolution des braquages en fonction de  $f$



**Figure A.2.1.3.d : Détail du mécanisme de direction sur l'essieu avant**

### **Relation entre la longueur du vérin hydraulique et l'angle $f$ :**

D'après les notations de la figure A.2.1.3.d, on peut écrire :

$$x^2 = (d - R \sin(f))^2 + R^2 \cos^2(f)$$

$$x^2 = d^2 + R^2 \sin^2(f) - 2dR \sin(f) + R^2 \cos^2(f)$$

$$x^2 = d^2 + R^2 (\sin^2(f) + \cos^2(f)) - 2dR \sin(f)$$

$$x^2 = d^2 + R^2 - 2dR \sin(f)$$

$$x^2 = x_0^2 - 2dR \sin(f)$$

$$\text{On en déduit la valeur de } f : f = \arcsin\left(\frac{x_0^2 - x^2}{2dR}\right) \quad (18)$$

## **2.2. Calibration du Cycab**

### **2.2.1. Calibration de la translation**

Le but est ici de déterminer le rapport top \ codeurs sur avance linéaire des roues.

Le déplacement du Cycab n'est pas rectiligne, pour des raisons d'imprécisions multiples (donnée joystick, parallélisme des roues,...), on peut cependant admettre que les quatre roues ont parcouru sensiblement la même distance. Pour cette calibration, une hypothèse d'une variation de moins de 0.5 % est retenue.

Les données mécaniques sont :

- codeurs, 2048 pas par tours;
- réduction moteur/roue de 20;
- diamètre  $D_R$  de la roue d'environ 40 centimètres.

La résolution codeur et le rapport de réduction nous donne le nombre de tops par tour de roue, 40960. Le diamètre de la roue n'est qu'indicatif, en effet, ce diamètre dépend de la pression de gonflage, de plus, la valeur qui nous intéresse est la distance entre le centre de la roue (centre de rotation) et le sol, cette valeur dépend de la pression de gonflage et de la charge sur chaque roue.

- première méthode avec les données théoriques :  
 $\text{top/mm} = \text{Top}/(\Pi * D_r) = 40960/(3.14*400) = 32.59$
- seconde méthode expérimentale :  $\text{top/mm} = \text{Moyenne\_Top}/\text{Distance} = 32651.75/925 = 35.30$

Soit 8.3 % de  $((35.30 - 32.59) / 32.59 = 0.083)$  de différence.

Il est à noter que les différences de lecture codeur sur les roues sont assez importantes. Si cela était dû à une distance parcourue différente, elle serait dans le cas extrême de l'ordre de  $(33725 - 31814) / 35.3 = 54$  millimètres, ce qui n'est pas envisageable.

Ces variations sont plutôt à mettre à l'actif des distances roues sol. Si on prend l'hypothèse que toutes les roues ont roulé sur la même distance, et à partir de la formule suivante:

$$\text{distance} = \text{roue\_sol} * 2 * \Pi * \text{Top\_mesure} / \text{Top\_par\_tour}.$$

On obtient alors pour chaque roue une distance roue\_sol (rayon) :

Roue	AVG	AVD	ARG	ARD	Moyenne
Distance Roue-sol	189.54	182.90	187.85	178.80	184.77

### 2.2.2. Calibration de la direction

Le but est ici de déterminer le rapport de la valeur du potentiomètre sur l'angle de rotation de la direction. L'approche est purement expérimentale et ne repose sur aucune donnée géométrique de la mécanique.

La partie précédente sur la cinématique donne une version analytique de l'angle de direction en fonction de l'allongement du vérin, mais il manque la relation entre l'allongement du vérin et la valeur du potentiomètre (qui doit être linéaire), de plus les jeux ne sont pas pris en compte. Une confrontation des deux approches a été faite mais n'est pas présentée ici, néanmoins les résultats obtenus sont proches.

La calibration ne sera pas précise car plusieurs paramètres n'ont pas été pris en compte :

- la géométrie de JANTEAU ;
- les jeux mécaniques de la direction n'ont pas été quantifiés.

La géométrie de JANTEAU, qui implique que la roue intérieure vire plus que la roue extérieure devrait faire apparaître deux zones sur les données expérimentales.

Une autre conséquence est que la direction recherchée sera celle de la roue virtuelle décrite dans la cinématique (moyenne des deux angles des roues).

En posant :

- $M$  une mesure;
- $P$  la valeur du potentiomètre correspondant à cette mesure;

- $M_0$  la mesure initiale;
- $P_0$  la valeur du potentiomètre correspondant à cette mesure initiale;
- $\mathbf{a}$  l'angle de direction;
- $D_{ml}$  la distance mur/laser.

On peut en déduire la loi potentiomètre/angle de direction  $\mathbf{a}$  suivante:

$$\tan(\mathbf{a}) = \frac{M - M_0}{D_{ml}}$$

soit

$$\tan(\mathbf{a}) = \frac{\text{pente} * (P - P_0)}{D_{ml}}$$

soit

$$\mathbf{a} = \text{Arc tan} \left( \frac{\text{pente} * (P - P_0)}{D_{ml}} \right)$$

soit

$$\mathbf{a} = \text{Arc tan}(0.00206 * (P + 40.0) / 2.335) \quad (19)$$

# Bibliographie

## 2.3. Articles

- [1] Wijk, O; Christensen : Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data, Robotics and Autonomous Systems Vol. 31, # 1, 2000
- [2] Thrun, Sebastian : Learning metric-topological maps for indoor mobile robot navigation, Artificial Intelligence 99 (1998) 21-71
- [3] Lee, Kyong-Ho; Cho, Sung-Bae; Choy, Yoon-Chul : Automated vectorization of cartographic maps by a knowledge-based system, Engineering Applications of Artificial Intelligence 13 (2000) 165-178
- [4] Krotkov, Eric; Hebert, Martial; Henriksen, Lars; Levin, Paul; Maimone, Mark; Simmons, Reid; Teza, James : Evolution of a Prototype Lunar Rover: Addition of Laser-Based Hazard Detection, and Results from Field Trials in Lunar Analog Terrain, Autonomous Robots 7 (1999) 119-130
- [5] Adams, M D : High Speed Target Pursuit and Asymptotic Stability in Mobile Robotics, IEEE Transactions on Robotics and Automation, vol. 15, #2 (1999) 230-237
- [6] Li, Shigang; Ochi, Akira; Yagi, Yasushi; Yachida, Masahiko : Making 2D Map of Environments Based upon Routes Scenes, Autonomous Robots Vol.8, #2, (2000) 117-128
- [7] Ma, Y; Kosecký, J; Sastry, S : Vision Guided Navigation for A Nonholonomic Mobile Robot, IEEE Transactions on Robotics and Automation, Vol. 15, #3, (1999) 521-536
- [8] Vendittelli, M; Laumond, J-P; Nissoux, C : Obstacle Distance for Car-Like Robots, IEEE Transactions on Robotics and Automation, Vol. 15, #4, (1999) 678-691
- [9] Kwon, Young D; Lee, Jin S : A Stochastic Map Building Method for Mobile Robot using 2-D Laser Range Finder, Autonomous Robots, Vol. 7, #2, (1999) 187-200
- [10] Pears, N E : Modeling of a scanning laser range sensor for robotic applications, Advanced Robotics, Vol. 13, #5 (1999)
- [11] Anousaki, G C; Kyriakopoulos, K J : Simultaneous Localization and Map Building for Mobile Robot Navigation, IEEE Robotics and Automation Magazine, Vol. 6, #3, (1999) 42-53

## 2.4. Rapports techniques

- [12] Gerard Baille, Philippe Garnier, Herve Mathieu, Roger Pissard-Gibollet : Le Cycab de l'INRIA Rhone-Alpes, # 0229, avril 1999
- [13] J.Borenstein, H. R. Everett, L. Feng : Sensors and Methods for Mobile Robot Positioning, University of Michigan, 1996

## 2.5. Conférences

- [14] C. Laugier, I. E. Paromtchik, Ph. Garnier and Th. Fraichard : Motion control of an autonomous vehicle through sensor-guided manoeuvres, INRIA Rhone-Alpes, GRAVIR
- [15] Sofiane Abdou, Michel Parent, Bernard Espiau : Spécification et implémentation logicielles pour des véhicules automatiques, INRIA
- [16] Sofiane Abdou, Michel Parent, Bernard Espiau : Mission programming : application to the distribution of empty vehicles in the PRAXITELE project, INRIA
- [17] D.P. Tsakiris, K. Kapellos, C. Samson, P. Rives and J.-J. Borrelly : Experiments in real-time vision-based point stabilisation of a nonholonomic mobile manipulator, Proceedings of the fifth International Symposium on Experimental Robotics (ISER'97)
- [18] M. Parent and P. Daviet : Automated urban vehicles – towards a dual mode PRT (personal rapid transit), In Proc. Of the IEEE Int. Conf. On Robotics and Automation, Minéapolis (MN), Avril 1996
- [19] F. Large, J. Hermosillo, S. Sekhavat, C. Laugier : Using Neuronal Networks to improve Sensor-Based Maneuvers for a Car-Like Vehicle, Int. Conf. On intelligent Autonomous Systems, Juillet 2000

## 2.6. Thèses

- [20] Philippe Garnier : Contrôle d'exécution réactif de mouvements de véhicules en environnement dynamique structuré, INPG, 30 mars 1992
- [21] Ali Siadat : Perception et modélisation de l'environnement par télémétrie laser. Application a la navigation autonome d'un fauteuil roulant, INPL Nancy, 26 janvier 1999

- [22] Stephane Leroy : Outils geometriques pour la planification de chemins de robots mobiles non holonomes, Laboratoire d'analyses et d'architecture des systèmes du CNRS, # 3239, 3 Decembre 1998
- [23] Eric Gauthier, Utilisation des Réseaux de Neurones Artificiels pour la Commande d'un Véhicule Autonome, INPG, 25 Janvier 1999

### **2.7. Rapports de stage**

- [24] Brieuc Desoutter, Localisation par odométrie, INRIA Rhone-Alpes, 29 septembre 1999
- [25] Funck Philippe, Casadoumecq Christophe : Conduite automatique et assistée d'un véhicule urbain, ISTG 1997-1998
- [26] Jean-Christophe DENIS, Tarek Msadaa : Essai de faisabilité d un Stop-&-Go, Ecole Nationale Supérieure des Mines de Paris, 1994

### **2.8. Documents constructeur**

- [27] SICK : LMS/LMI 400 – Definition of telegrams between the user interface and LMS or LMI systems via RS 422/RS 232
- [28] R. Pissard-Gibollet, K. Kapellos : Open Robot Controller Computer Aided Design – ORCCAD V3.0 $\alpha$ , User Manual, INRIA Rhône Alpes, Fevrier 1998
- [29] Wind River Systems : VxWorks Reference Manuel, 5.3.1, Edition 1, 1997





# Qualification d'un télémètre à balayage laser pour la robotique mobile : Intégration et expérimentations

Marcel RIESS

Mémoire d'Ingénieur CNAM, GRENOBLE 2000

## Résumé (en cours de rédaction)

Ce travail a été réalisé dans le cadre des missions du service des moyens robotiques de l'INRIA Rhône-Alpes. Dans le cadre de recherche sur l'autonomie des véhicules, plusieurs expérimentations ont fait apparaître le besoin de données sensorielles télémétriques complémentaires à celles délivrées par les capteurs en place : caméras, ultrasons.

Afin d'obtenir plus d'informations sur l'environnement du véhicule, nous avons intégré un capteur télémétrique à balayage laser sur un véhicule électrique de type Cycab. La problématique revient à qualifier le capteur et à argumenter des choix d'intégration et de mise en œuvre. Le but est d'exploiter expérimentalement le capteur dans plusieurs applications de robotique mobile.

Le contrôle du capteur - qui dispose de son propre micro-contrôleur - le traitement et l'interprétation des données sont réalisés sur une plate-forme Linux. Une architecture client/serveur à base de socket permet l'échange des données avec l'application temps réel (sous VxWorks).

Différentes expérimentations ont montré les apports de ce type de capteur dans l'estimation des erreurs de l'odométrie et pour des applications spécifiques telles que le convoi de véhicule par accrochage immatériel.

Suite à la présentation du capteur et des choix techniques d'intégration, nous proposons une évaluation des résultats obtenus lors de cas pratiques.

Mots clés : Robotique mobile, convoi de véhicules, capteur laser.

Key words : Mobil Robotics, platoon of vehicles, laser sensor.