



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

655, Avenue de l'Europe
38330 Montbonnot



**Implémentation de la commande d'un véhicule
électrique autonome grâce à un capteur de
distance et d'angle basé sur une caméra linéaire**



Nicolas Cardilès
IUP de Mathématiques Appliquées et Industrielles
mai – août 1998



Table des matières

INTRODUCTION :	3
PRESENTATION GENERALE :	4
1. DOMAINES DE RECHERCHE :.....	4
2. LE PROJET SHARP :.....	5
3. LE SERVICE ROBOTIQUE :.....	6
3. PARTICIPATION DANS L'ACTION DE DEVELOPPEMENT PRAXITELE :.....	6
5. CADRE.....	7
6. AXES DE RECHERCHE :.....	8
7. BREF DESCRIPTIF DU TRAVAIL A REALISER :.....	8
DESCRIPTION DU MATERIEL UTILISE.....	9
1. BALISE INFRAROUGE A TROIS POINTS.....	9
2. LA CAMERA LINEAIRE :.....	10
CALCULS DE POSITIONNEMENT 2D PAR TRIANGULATION :.....	12
1. REPRESENTATION 2D :.....	12
2. CALCULS :.....	13
3. METHODE :.....	14
SYNCHRONISATION DE LA CAMERA :.....	15
1. POSITION DU PROBLEME :.....	15
2. METHODE UTILISEE POUR LA SYNCHRONISATION :.....	17
TRANSMISSION DE DONNEES VIA CAN.....	20
1. PRINCIPE :.....	20
2. MISE EN ŒUVRE :.....	20
GESTION DES INTERRUPTIONS.....	21
SUIVI DE VEHICULE.....	22
1. PRESENTATIONS DES LOGICIELS UTILISES :.....	22
2. PRINCIPE :.....	24
3. MISE EN ŒUVRE :.....	24
CONCLUSION :.....	28
ANNEXES.....	29
PRINCIPE DU FONCTIONNEMENT DIFFERENTIEL DE LA BALISE	
BALISE INFRAROUGE A TROIS POINTS	
FICHE TECHNIQUE DU CYCAB	
CONTROLE DU TEMPS D'EXPOSITION	
CAN	
CLEFS DE COMMUNICATION ET LEURS EFFETS	
SOURCES	

Introduction :

Ce stage de fin d'année s'inscrit dans le cadre de mes études à l'Institut Universitaire Professionnalisé de Mathématiques appliquées et Industrielles mention Génie Mathématique et Informatique de l'Université Joseph Fourier à Grenoble, pour l'obtention du diplôme d'Ingénieur Maître.

Il s'est déroulé à l'Unité de Recherche Rhône-Alpes (Montbonnot) de l'Institut National de Recherche en Informatique et en Automatique.

L'objectif de ce stage est de mettre au point un capteur de distance et d'angle entre deux véhicules, basé sur une caméra linéaire infrarouge programmable interfacée via un bus CAN d'une part, et sur une balise infrarouge d'autre part, puis d'implémenter la commande de véhicule pour la conduite en pelotons, grâce à ce capteur.

Ce stage est dans la continuité du stage que j'ai réalisé en fin de la licence, avec toutefois une importante amélioration du matériel utilisé, et un élargissement du travail demandé, notamment en ce qui concerne la partie commande de véhicule.

Présentation Générale :



1. Domaines de recherche :

L'INRIA est un établissement public à caractère scientifique et technique, placé sous la double tutelle du ministère de l'Education nationale, de l'Enseignement supérieur et de la Recherche, et du ministère de l'Industrie, des Postes et des Télécommunications.

Les activités de l'INRIA comprennent la réalisation de systèmes expérimentaux, la recherche fondamentale et appliquée, le transfert de technologie, l'organisation d'échanges scientifiques internationaux et la diffusion des connaissances et du savoir-faire.

Ces activités associent informaticiens, automaticiens, mathématiciens dans le cadre de projets de recherche regroupés dans six programmes :

1. Architectures parallèles, bases de données, réseaux et systèmes distribués.
2. Calcul symbolique, programmation et génie logiciel.
3. Intelligence artificielle, systèmes cognitifs et interaction homme-machine.
4. **Robotique, image et vision.**
5. Traitement du signal, automatique et productique.
6. Calcul scientifique, modélisation et logiciels numériques.

Depuis 1992, l'INRIA participe à de grands projets à finalité industrielle dans le cadre d'actions de développement : programmes de durée déterminée (généralement trois à cinq ans), menés en partenariat avec des industriels et des usagers des technologies de l'information. Faisant participer plusieurs équipes de recherche, ces actions constituent des occasions de coordination, voire de fertilisations croisées entre projets de recherche et sociétés de technologie.

Le projet Praxitèle a été le premier de ces grands projets industriels. Son objectif final à long terme est de développer un nouveau mode de transport public individuel avec des véhicules mixtes, c'est-à-dire à conduite manuelle ou automatique selon le besoin.

2. Le projet :

L'objectif de ce projet consiste à développer des méthodes informatiques aptes à accroître les capacités décisionnelles des robots et à faciliter leur mise en œuvre sur de nouvelles tâches, partiellement spécifiées et généralement non répétitives.

Ces deux points constituent l'un des verrous technologiques qu'il faut faire sauter pour permettre une utilisation plus étendue des robots sur des tâches non nécessairement manufacturières (le champ des applications manufacturières étant trop restrictif sur le plan socio-économique).

Sont particulièrement visées des tâches pour lesquelles il est difficile, voire impossible, de faire directement appel à l'homme (travaux dangereux et pénibles, milieux hostiles, sites éloignés), ou bien encore des tâches d'assistance à un opérateur humain (télé-robotique et *assistance à la conduite automobile* par exemple).

Ceci conduit à mettre l'accent sur les concepts de planification de mouvements et d'intégration système en robotique en orientant nos travaux de recherche vers la résolution de problèmes liés à la fois à la manipulation et à la mobilité. Par intégration système, il faut comprendre l'intégration des techniques de planification développées avec celles de la vision par ordinateur et du contrôle. Ceci est nécessaire pour être en mesure de développer des techniques réalistes vis-à-vis des types d'applications envisagées

Le projet SHARP a participé activement à l'action de développement Inria-INRETS (Institut National de Recherche sur les Transports et leur Sécurité) Praxitèle lancée en 1994 et qui implique des partenaires industriels aussi divers que la CGEA (Compagnie Générale d'Entreprise Automobile), Electricité de France, Renault et Dassault Electronique.

Praxitèle est un programme de recherche sur le concept du Transport Urbain Public et Individuel ; la contribution de SHARP a porté sur les aspects conduite automatique.

L'action de développement Praxitèle, désormais terminée, a été suivie du projet Cabby, qui inclut en plus la création d'un nouveau véhicule, puis du projet Cycab, toujours sur le même véhicule.

3. Le service robotique :

Le rôle des moyens robotiques est la mise en œuvre et la maintenance de plates-formes robotiques pour réaliser les expérimentations des projets robotiques.

Les missions qui lui sont attribuées sont de trois types :

- Activité de service :
 - maintenances des systèmes robotiques
 - installation et maintenance de logiciels spécialisés
 - interface entre les utilisateurs et le service informatique
 - assistance aux utilisateurs
- Activité de développement :
 - mise en place d'expérimentations
 - développement de logiciels dédiés à la robotique
- Activité de recherche :
 - conception de systèmes robotiques
 - confrontation théorie et expérimentation

Le but du Service Robotique est de fédérer l'effort expérimental en favorisant les expérimentations inter-projets, la mise en commun des moyens expérimentaux, les outils réutilisables (environnement de développement, machine de vision...)

Les moyens robotiques travaillent avec les projets des programmes 3 et 4 impliqués en robotique et vision : SHARP (Programmation automatique et systèmes décisionnels en robotique), MOVI (Modélisation, localisation, reconnaissance et interprétation en vision par ordinateur), BIP (Conception et contrôle de robots marcheurs et applications).

Les plates-formes robotiques sont regroupées dans une halle qui contient un bras manipulateur, une main tri-digitale, deux voitures électriques, un robot portique dédié à la vision, un pendule simple esquisse d'un robot bipède.

3. Participation dans l'action de développement Praxitèle :

Dès 1991, l'INRIA identifie les nouvelles technologies permettant de matérialiser un nouveau concept de transport à la fois public et individuel (projet DEDALE).

Il s'agit en particulier des avancées technologiques sur la communication, la localisation des mobiles, la billétique et l'automatisation de la conduite. Un brevet est déposé en 1992 sur ce concept.

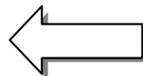
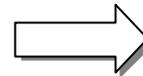
L'action Praxitèle est aussi en contact étroit avec les principaux centres de recherche travaillant sur des sujets similaires en France et à l'étranger (programme fédéral Automated Highways System, Université de Berkeley).

Huit équipes de l'INRIA, une trentaine de chercheurs (dont dix doctorants) sont aujourd'hui impliqués dans l'action Praxitèle dans les domaines suivants :

- Modélisation mathématique et contrôle.
- Traitement de l'image.
- Communication, automatisation.
- Conception de véhicules spéciaux pour une utilisation en libre-service.

Ces développements sont en cours d'intégration dans un nouveau concept de véhicule spécifiquement conçu par l'INRIA pour le transport public individuel et la conduite automatique en site propre.

Les recherches sur la conduite automatique ont été concrétisées sur des véhicules électriques de marque Ligier modifiés par l'INRIA :



La conduite en train avec accrochage immatériel a été démontrée en septembre 1994, le parking automatique en créneau, en juin 1995.

5. Cadre

Dans les villes d'aujourd'hui, la circulation asphyxie les centres et irrigue mal les quartiers périphériques. Il est impératif de préserver l'espace urbain tout en favorisant la fluidité des déplacements, sève indispensable à la vie économique ainsi qu'à la vie quotidienne de chaque citoyen.

Praxitèle est un service public de transport, complémentaire des transports en commun et des taxis qui vise à offrir une alternative aux déplacements en voiture individuelle privée. Ce nouveau service est destiné en priorité aux itinéraires et aux horaires où la demande est diffuse et où les transports publics sont insuffisants en terme de fréquence et de rentabilité. L'objectif est de réduire les problèmes de stationnement tout en gardant la souplesse du véhicule individuel.

Pour les utilisateurs, ce service peut constituer un frein à l'usage immodéré de la voiture particulière pour les trajets de proximité et pour les déplacements urbains liés à un stationnement difficile.

Ces véhicules pourraient avoir des droits spéciaux en terme de stationnement et de circulation dans le cadre de la réduction de la pollution atmosphérique et sonore des villes.

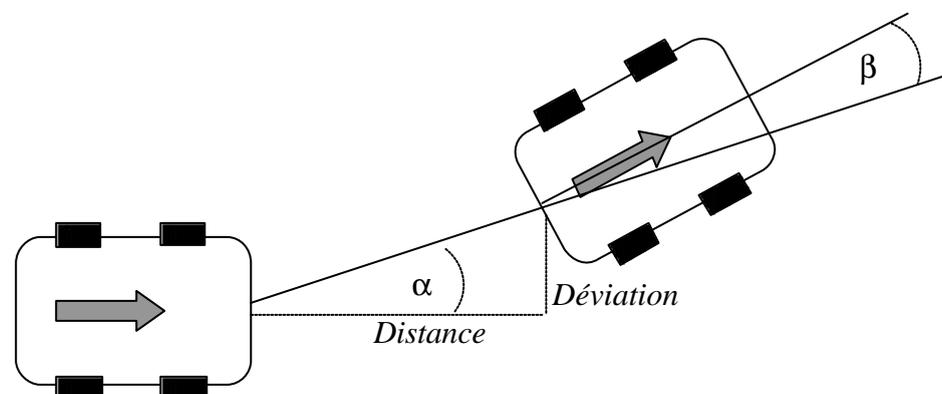
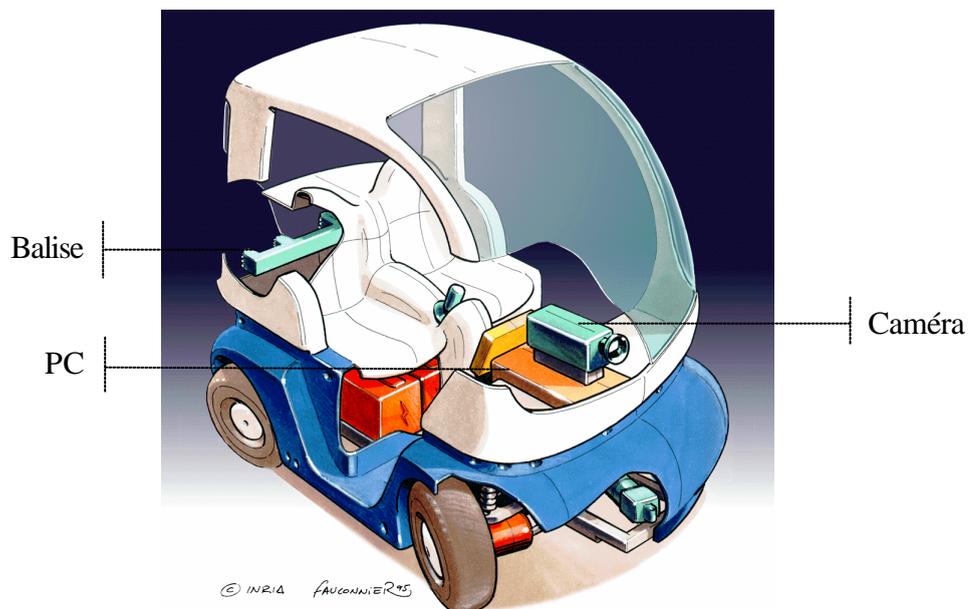
6. Axes de recherche :

Pour développer ce projet, plusieurs axes de recherche sont nécessaires : conception du véhicule, des stations de parking, ingénierie système, localisation, communication, gestion informatique, organisation et réglementation, automatisation des parkings, *conduite en pelotons*, conduite automatique sur voie propre et hors voie propre.

Mon stage prend place dans le thème de recherche "*conduite en pelotons*" et porte plus particulièrement sur l'*accrochage immatériel* entre véhicules.

7. Bref descriptif du travail à réaliser :

Avec le matériel fourni (caméra linéaire, balise infrarouge, PC) je dois être en mesure de fournir la distance, la déviation et les angles séparant deux véhicules équipés l'un de la balise infrarouge à l'arrière, l'autre de la caméra à l'avant. De plus je dois faire fonctionner le suivi de véhicules grâce à ce capteur.



Description du matériel utilisé

Le capteur fonctionne avec un *émetteur* et un *récepteur*. La partie émetteur est constituée de la balise infrarouge, la partie récepteur de la caméra linéaire. Pour la mise au point de l'application j'ai aussi utilisé un PC industriel.

1. Balise infrarouge à trois points

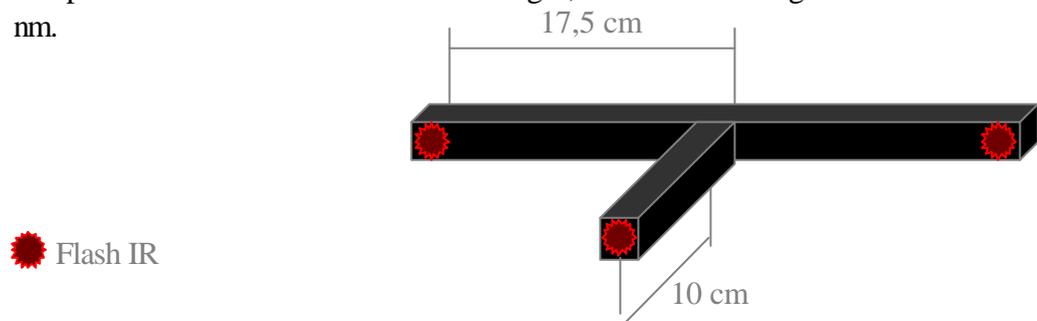
Pour améliorer les performances du capteur, j'ai décidé de transformer la partie émetteur, jusque là constituée d'une balise infrarouge à cinq points, en une balise à trois points.

Il est apparu en effet que trois points suffisent pour les calculs de positionnement. De plus la robustesse du capteur augmente quand on diminue le nombre de points d'émission de la balise. En effet, il faut que la totalité des sources d'émission soit visible pour pouvoir faire les calculs de positionnement, donc si une source parasite d'infrarouge, par exemple le soleil, se superpose à l'un des points d'émission de la balise cela a pour effet de rendre cette dernière invisible, situation dans laquelle on a moins de risques de se retrouver si on diminue le nombre de sources d'émission de la balise.

Enfin, deux sources d'émission en moins signifie aussi moins d'électricité utilisée pour le capteur : on passe de 1A pour la balise à 5 points à 0,6A pour la balise à trois points.

1.1. Principe de fonctionnement :

Afin de localiser précisément le véhicule qui précède, on équipe chaque véhicule d'une balise infrarouge placée à l'arrière. Cette balise en forme de "T" comporte trois sources d'émission d'infrarouges réparties comme sur le schéma. Les sources sont composées chacune de 18 diodes infrarouges, émettant à la longueur d'onde de 875 nm.



La caractéristique principale de la balise est que les trois sources n'émettent pas toutes en même temps. En effet il faut distinguer deux "groupes" : le premier constitué des deux sources extrêmes et le second de celle du milieu. Ces deux groupes émettent en alternance, chacun pendant 2 ms environ :



Ceci permet de créer une [image différentielle](#) de la balise et donc d'augmenter la robustesse du capteur.

1.2. Réalisation :

Afin de réaliser cette nouvelle balise, je me suis basé sur la balise existante. Il m'a fallu retrouver le schéma électronique de la balise à cinq points, en extraire la partie utile (en effet sur cette balise deux modes de fonctionnement étaient disponibles, dont un inintéressant car seulement deux points d'émission étaient différentiels), l'adapter pour une balise à trois points, et enfin réaliser la balise proprement dite.

D'autre part j'ai écrit une [documentation](#) sur cette balise, qui contient les données mécaniques, la liste des composants, leur schéma d'implantation sur la plaque électronique, ainsi que le schéma électronique de la balise et les data sheets des composants principaux.

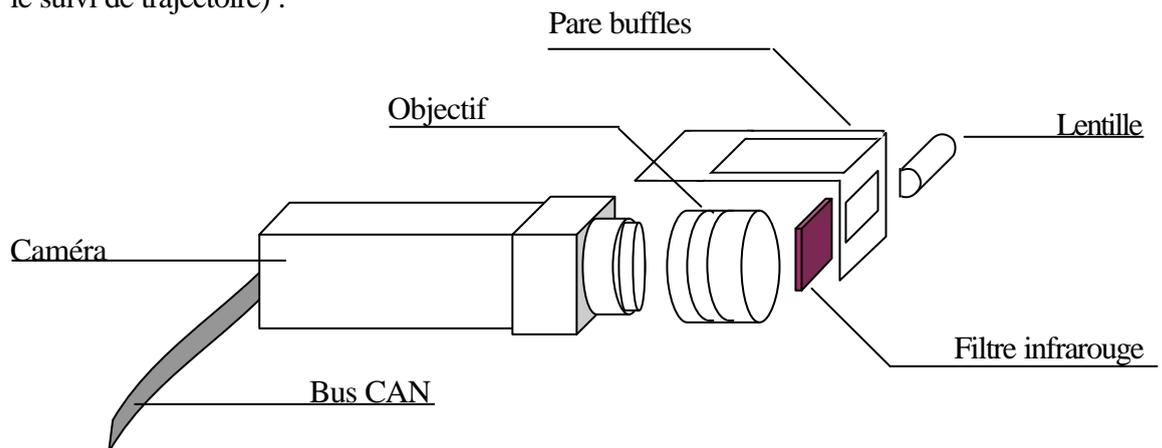


2. La caméra linéaire :

La Cil2048 et son moniteur



Pour localiser la balise précédemment décrite, on dispose d'une caméra linéaire (linéaire car le capteur CCD est constitué d'une ligne d'éléments photosensibles et non pas d'une matrice comme dans une caméra classique), intelligente (en fait programmable, donc en sortie on récupère directement l'information désirée, et non pas un signal vidéo non traité), et interfaçable via un bus CAN à 1 MBaud (afin de pouvoir communiquer avec le PC qui gère le suivi de trajectoire) :

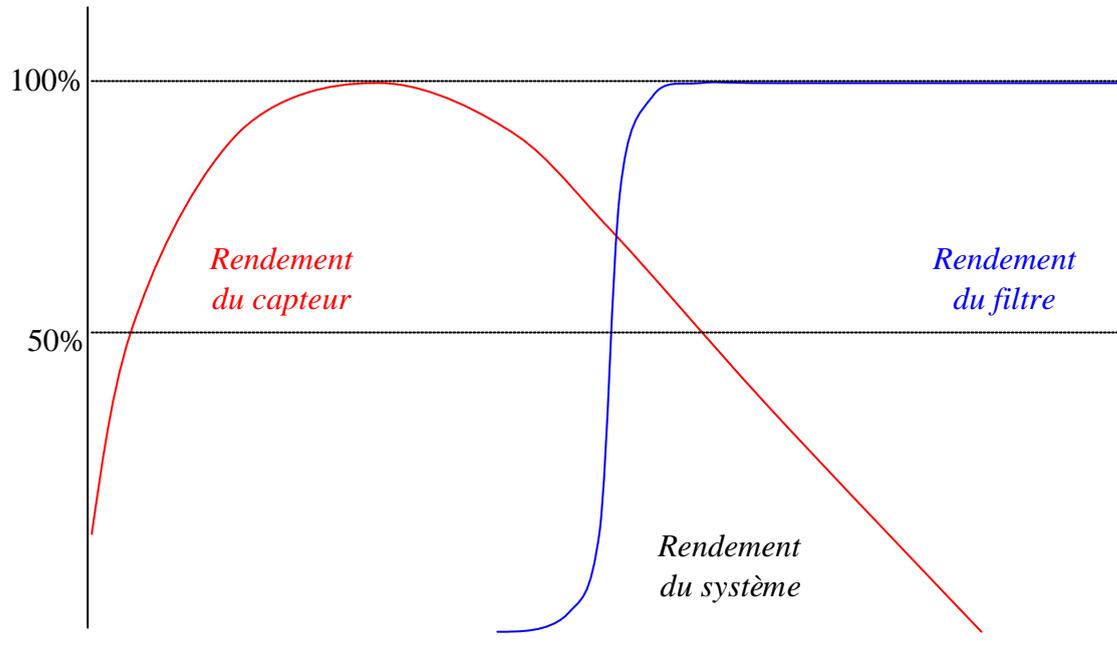


Comme on peut le voir, la caméra est équipée d'une optique particulière : un filtre infrarouge pour récupérer uniquement les longueurs d'ondes utiles, et une lentille cylindrique pour éviter de perdre la cible de vue en cas de variation de hauteur.

2.1. Le filtre infrarouge :

Ce filtre passe haut bloque les longueurs d'ondes inférieures à 780 nm. De plus la caméra est capable de percevoir les longueurs d'ondes jusqu'à 1100 nm.

On aboutit ainsi au filtre passe-bande équivalent :



2.2. La lentille cylindrique :

Dans le montage de l'optique, on a associé un objectif Cosina ($f=28$ mm, monture Nikon) pour la mise au point, et une lentille cylindrique d'axe vertical ($f=100$ mm). Le rôle de cette lentille est de rendre volontairement flou dans le plan vertical.

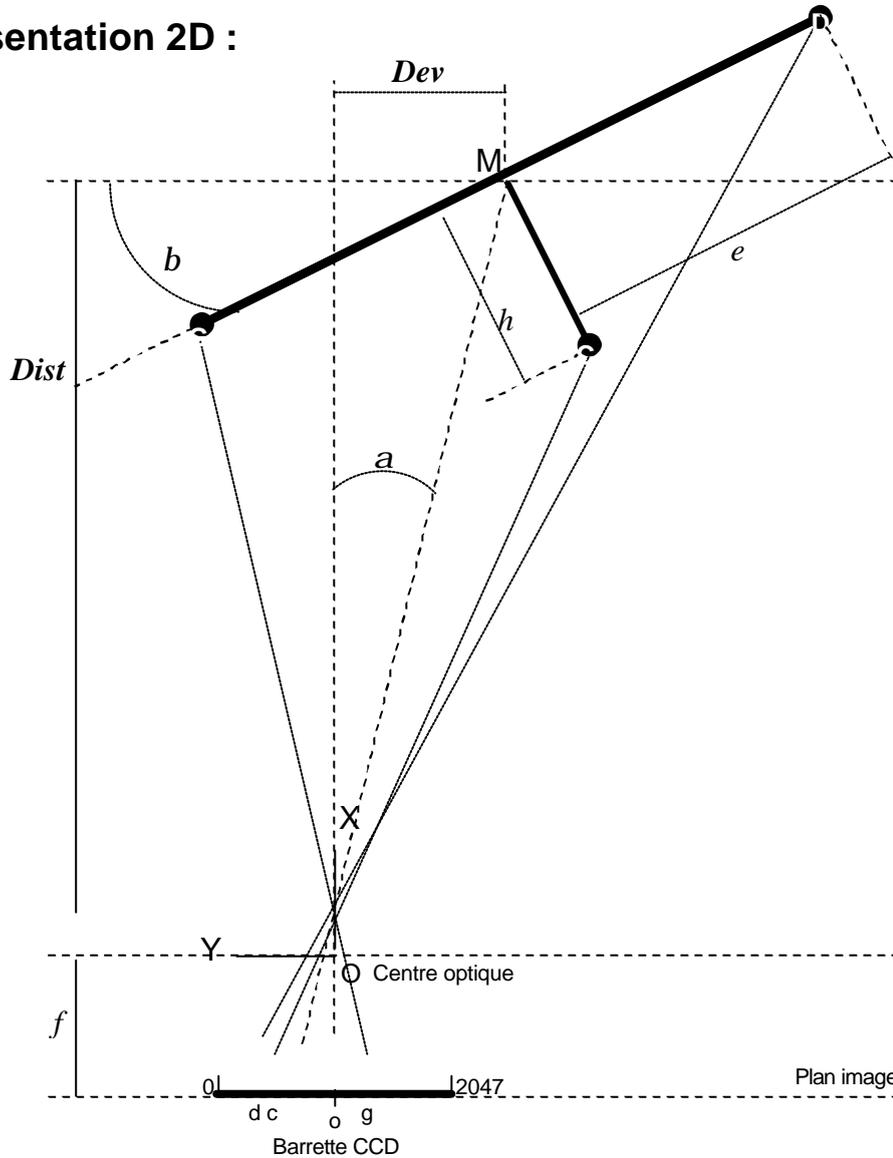
En effet, en utilisant une caméra linéaire, on a un champ de vision très réduit dans ce plan pour indication, à 1 m de distance le champ de vision vertical n'excède pas 5° . Grâce cette lentille, on effectue en quelque sorte une sommation de la lumière perçue dans le champ de vision sur une ligne virtuelle horizontale placée sur le même plan que la caméra. De plus cette lentille augmente le champ de vision vertical.

Le gain procuré par cette lentille se traduit par une utilisation plus facile du matériel pour l'application de suivi de véhicule. En effet, les cahots éventuels ou les cassis et autres dos d'ânes ne sont désormais plus une entrave au bon fonctionnement du capteur.

Cependant les inconvénients amenés par la lentille ne sont pas négligeables : en effet, on a désormais un angle de vue dans le plan vertical suffisant pour être obligé de prendre en compte la lumière parasite, typiquement celle d'origine solaire.

Calculs de positionnement 2D par triangulation :

1. Représentation 2D :



Variables à déterminer :

- **Distance** de la caméra à la balise.
- **Angle** entre la balise et la normale au plan image.
- **Déviation** entre la balise et la normale au plan image.
- **Rotation** de la balise sur elle-même.

Nom abrégé :

Dist

a

Dev

β

Variables connues :

- Demi longueur de la balise.
- Hauteur de l'avancée centrale.
- Distance focale de l'objectif.
- Projection des extrémités de la balise.

e

h

f

g, c, d

Variables intermédiaires :

- Extrémités de la balise.
- Position du milieu de la balise.

G, C, D

2. Calculs :

On cherche à déterminer des variables caractérisant la position de la balise. Pour y parvenir on procédera dans l'ordre suivant : ∂ Angle α | •Distance | \div Déviations | \neq Angle β .

On exprime en premier lieu pour $a=0$ les vecteurs OM, MG, MC, MD :

$$\mathbf{OM} = \begin{bmatrix} Dist \\ Dev \end{bmatrix} \quad \mathbf{MG} = \begin{bmatrix} 0 \\ e \end{bmatrix} \quad \mathbf{MC} = \begin{bmatrix} -h \\ 0 \end{bmatrix} \quad \mathbf{MD} = \begin{bmatrix} 0 \\ -e \end{bmatrix}$$

On définit ensuite la matrice de rotation par rapport à a :

$$\mathbf{R}\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} = \cos(\alpha) \times \begin{bmatrix} 1 & -\tan(\alpha) \\ \tan(\alpha) & 1 \end{bmatrix}$$

On calcule ensuite les vecteurs suivants :

$$\mathbf{OG} = \mathbf{OM} + \mathbf{R}\alpha \times \mathbf{MG} = \begin{bmatrix} Dist - e \times \tan(\alpha) \times \cos(\alpha) \\ Dev + e \times \cos(\alpha) \end{bmatrix}$$

$$\mathbf{OC} = \mathbf{OM} + \mathbf{R}\alpha \times \mathbf{MC} = \begin{bmatrix} Dist - h \times \cos(\alpha) \\ Dev - h \times \tan(\alpha) \times \cos(\alpha) \end{bmatrix}$$

$$\mathbf{OD} = \mathbf{OM} + \mathbf{R}\alpha \times \mathbf{MD} = \begin{bmatrix} Dist + e \times \tan(\alpha) \times \cos(\alpha) \\ Dev - e \times \cos(\alpha) \end{bmatrix}$$

Puis en appliquant le théorème de Thalès, on calcule la position de g, c, d :

$$g = f \times \frac{\mathbf{OG}_y}{\mathbf{OG}_x} = f \times \frac{Dev + e \times \cos(\alpha)}{Dist - e \times \tan(\alpha) \times \cos(\alpha)}$$

$$c = f \times \frac{\mathbf{OC}_y}{\mathbf{OC}_x} = f \times \frac{Dev - h \times \tan(\alpha) \times \cos(\alpha)}{Dist - h \times \cos(\alpha)}$$

$$d = f \times \frac{\mathbf{OD}_y}{\mathbf{OD}_x} = f \times \frac{Dev - e \times \cos(\alpha)}{Dist + e \times \tan(\alpha) \times \cos(\alpha)}$$

Enfin, on résout ce système d'équations et on obtient :

$$\tan(\alpha) = \frac{e \times f \times (g + d - 2 \times c) + h \times c \times (g - d)}{e \times (c \times (g + d) - 2 \times g \times d) + h \times f \times (g - d)}$$

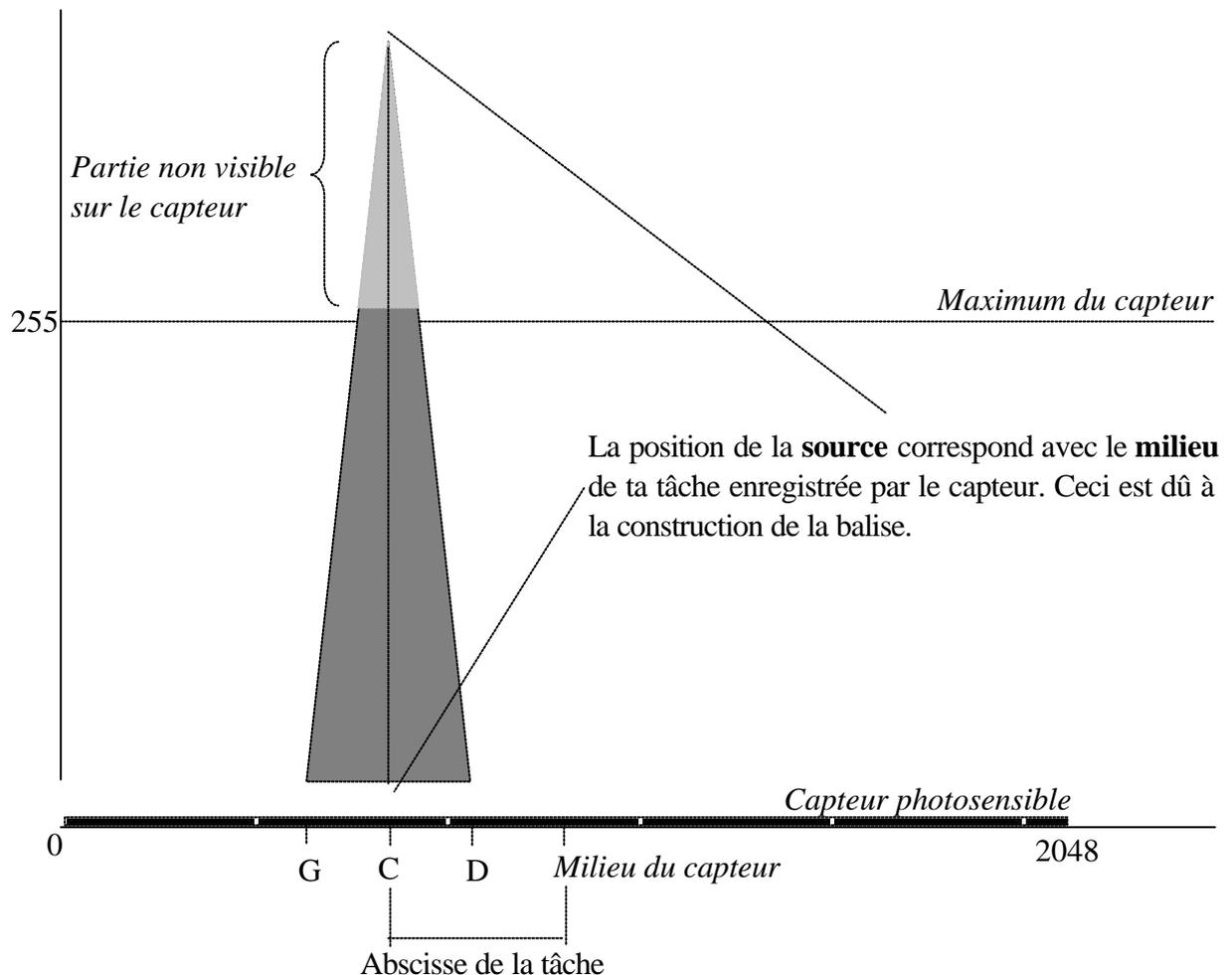
$$Dist = \frac{e \times \cos(\alpha) \times (2 \times f + \tan(\alpha) \times (g + d))}{g - d}$$

$$Dev = h \times \cos(\alpha) \times \left(\tan(\alpha) - \frac{c}{f} \right) + \frac{c \times Dist}{f}$$

$$\tan(\beta) = \frac{Dev}{Dist}$$

3. Méthode :

Une fois les coordonnées des pics récupérées sur la barrette CCD, on calcule les valeurs de g , c , et d (les valeurs de g , c , et d sont données par rapport au centre de la barrette) en prenant le milieu de chaque pic. En effet, dans l'intervalle spatial (environ de 2 à 10 mètres) où la mesure est demandée, l'amplitude des pics dépasse parfois le maximum du capteur, on ne peut donc utiliser que le centre de chaque pic pour estimer la position de la source avec exactitude.



On utilise ensuite ces données et les *données physiques* de la caméra et de la balise pour calculer les valeurs demandées :

<i>Caméra :</i>	Taille d'un pixel :	$14 \mu\text{m} \times 14 \mu\text{m}$
	Distance focale :	28 mm.
	Nombre de pixels :	2048
<i>Balise :</i>	Zone photosensible :	$28,7 \text{ mm} \times 14 \mu\text{m}$
	Demi longueur :	$e=200 \text{ mm}$
	Hauteur de l'avancée :	$h=180 \text{ mm}$



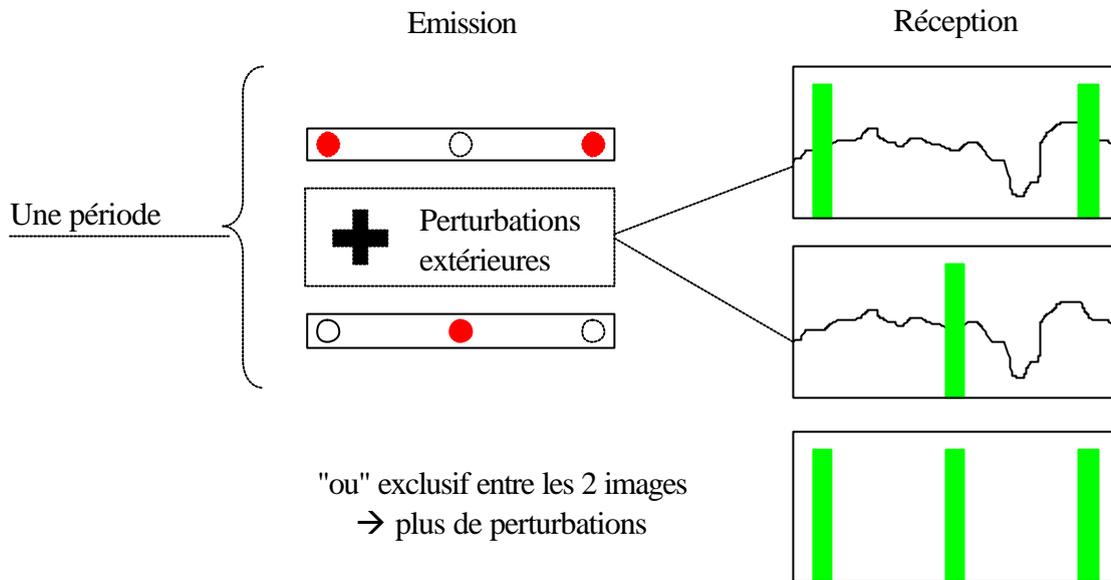
La compilation d'instruction manipulant des nombres à virgule flottante ayant posé des problèmes d'exécution sur la caméra, le calcul se fait encore sur le PC, la caméra se contentant de lui transmettre la position des tâches qu'elle perçoit.

Synchronisation de la caméra :

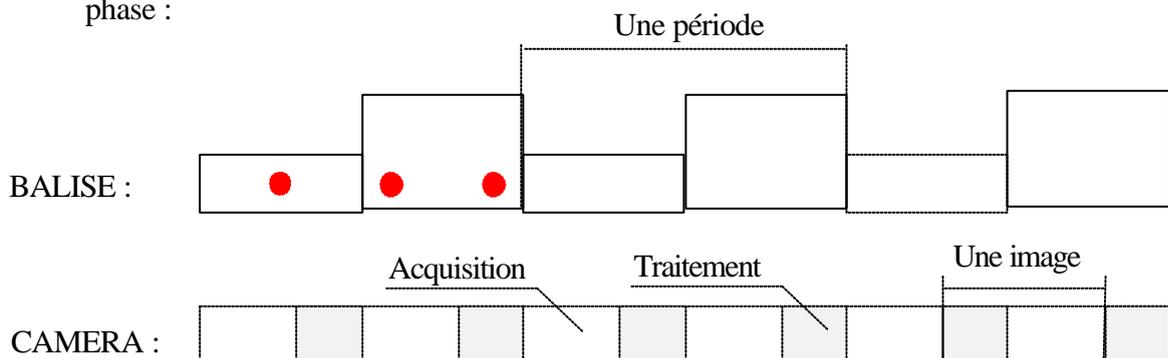
1. Position du problème :

Afin de rendre la mesure plus robuste pour que le système soit utilisable à l'extérieur, il faut arriver à synchroniser la caméra et la balise pour supprimer les parasites :

Fonctionnement de la balise et de la caméra en synchronisation :

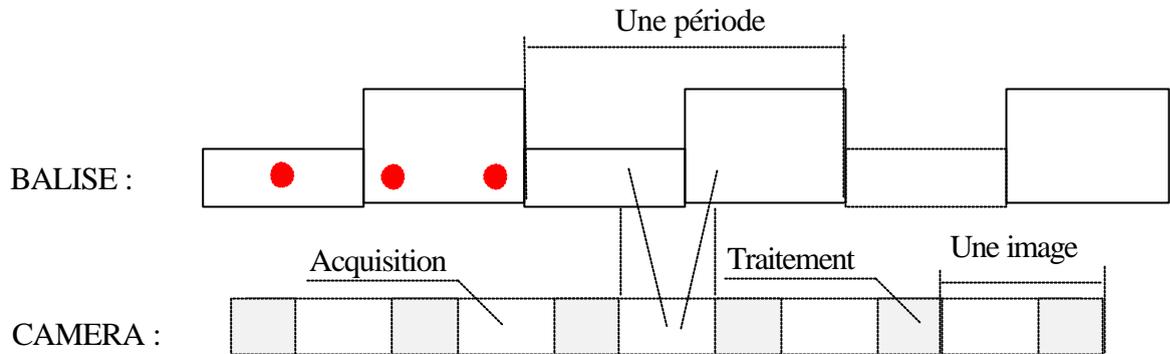


Cette technique de filtrage n'est utilisable que si la caméra et la balise sont parfaitement en phase :

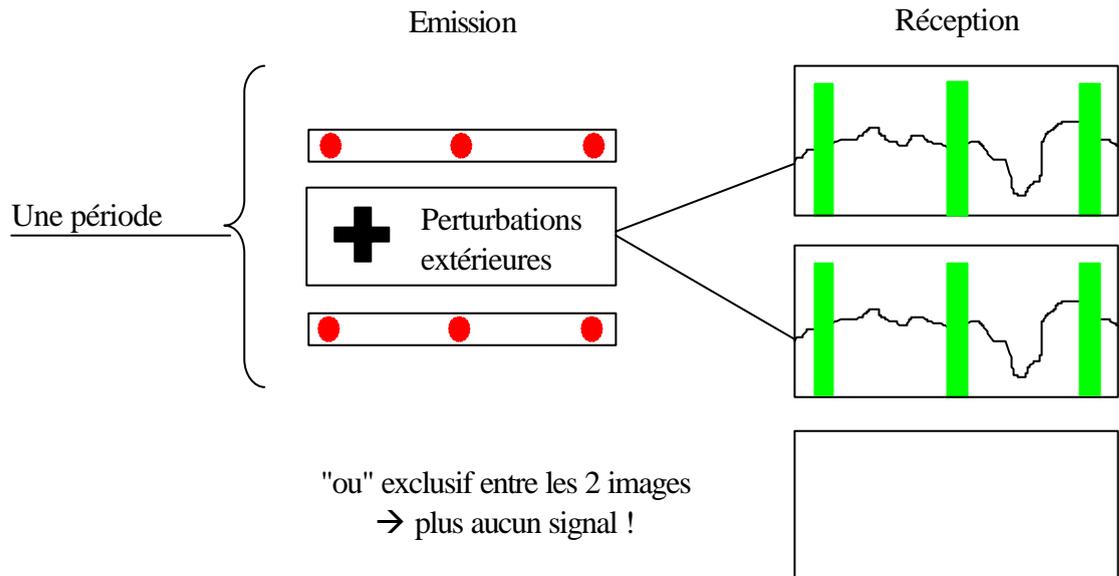


Il ne faut pas qu'une période d'acquisition/intégration soit commune à deux phases d'émission différentes.

Dans le cas contraire, si la caméra et la balise ne sont pas exactement en phase, chacune des deux images acquises rendra compte bien sur des perturbations extérieures, mais en plus elle contiendra cinq pics, puisque la période d'acquisition aura été commune à deux périodes d'émission différentes :



L'image résultante sera dépourvue de signal :



La mesure sera donc rendue plus robuste grâce à cette technique, seulement si la synchronisation entre la balise et la caméra est parfaite. Pour y parvenir, on connaît uniquement la durée de la période de scintillement des balises : 2 pics sont visibles pendant 2 ms, puis un pic pendant la même durée.

Il faut donc que le temps de cycle de la caméra soit : $TC = 2 \text{ ms}$ et pour une bonne dynamique de l'image il faut que le temps d'intégration/acquisition T_i soit égal à la moitié du temps de cycle.

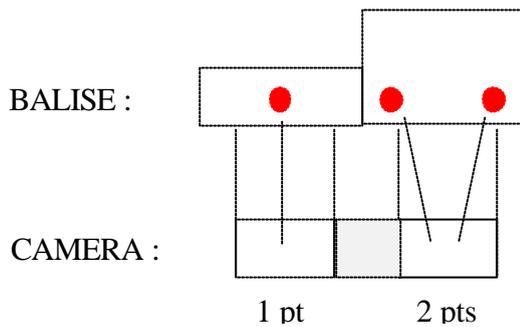
En effet si on prend un temps d'intégration trop important, on arrive très rapidement à saturer le capteur (on ne voit plus rien du tout). Mais il s'agit là d'un autre problème : [le contrôle du temps d'exposition](#), qui est fonction de la puissance d'émission de la balise, de l'ouverture du diaphragme de l'objectif, etc.



D

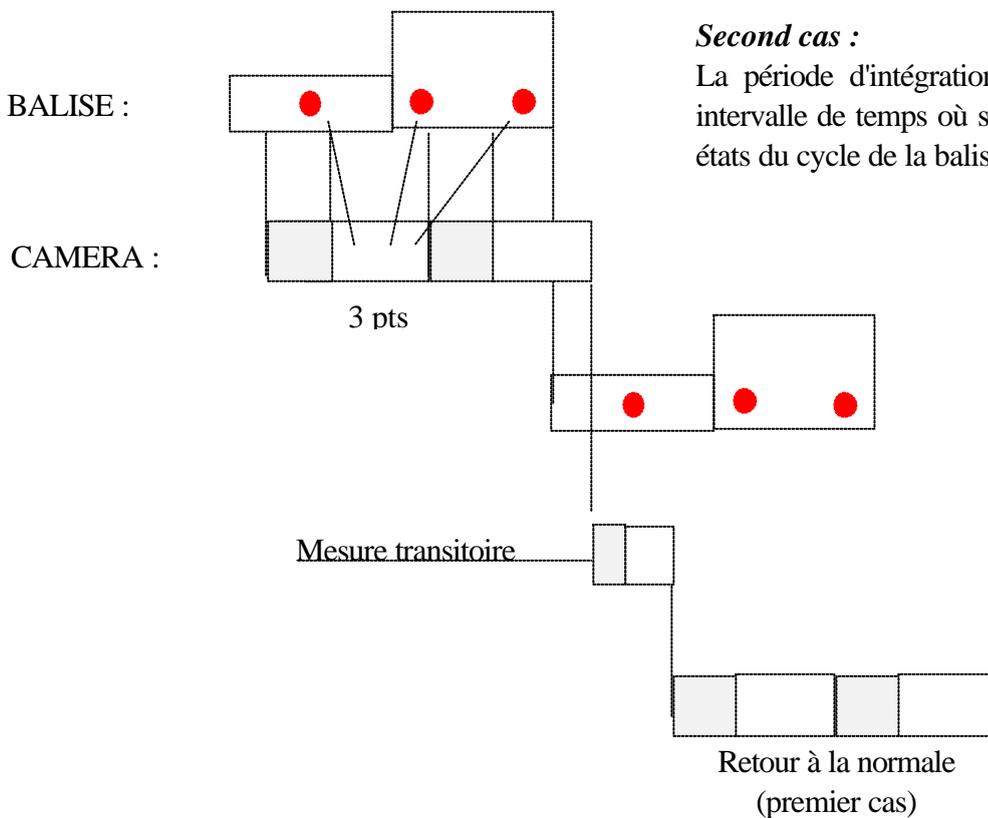
2. Méthode utilisée pour la synchronisation :

On fixe le temps de cycle $TC=2$ ms et le temps d'intégration $Ti=1$ ms à la moitié du temps de cycle. On sait qu'on a ainsi une chance sur deux en prenant une image de "tomber" à un moment où les trois points sont allumés, auquel cas il suffit de prendre une autre image avec TC diminué de moitié pour faire coïncider la période d'intégration avec une période de temps où les deux cycles d'émission de la caméra ne se chevauchent pas :



Premier cas :

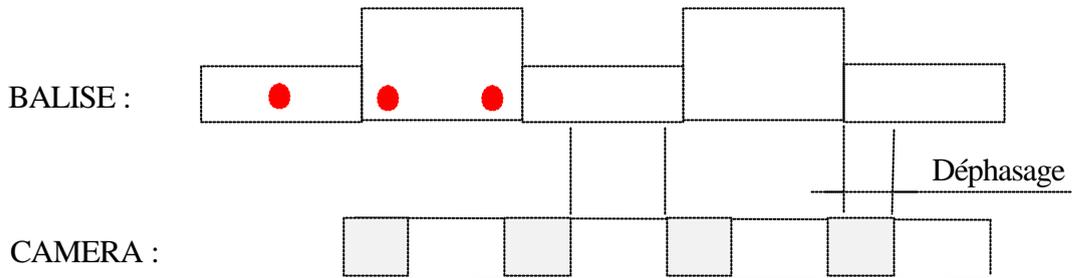
La période d'intégration se situe dans un intervalle de temps où les deux états du cycle de la balise ne se chevauchent pas.



Second cas :

La période d'intégration se situe dans un intervalle de temps où se chevauchent deux états du cycle de la balise.

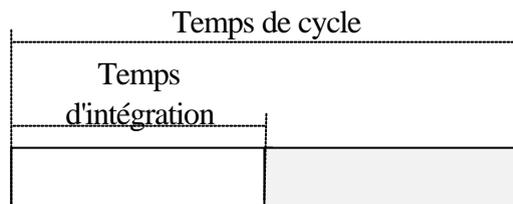
Une fois cette première étape réalisée, toute période d'intégration se situe dans un intervalle de temps où la balise est dans un seul état (1 ou 2 pics) :



Il ne reste qu'à faire coïncider le début du temps d'intégration avec un changement d'état de la balise, en augmentant petit à petit la durée du temps d'intégration.

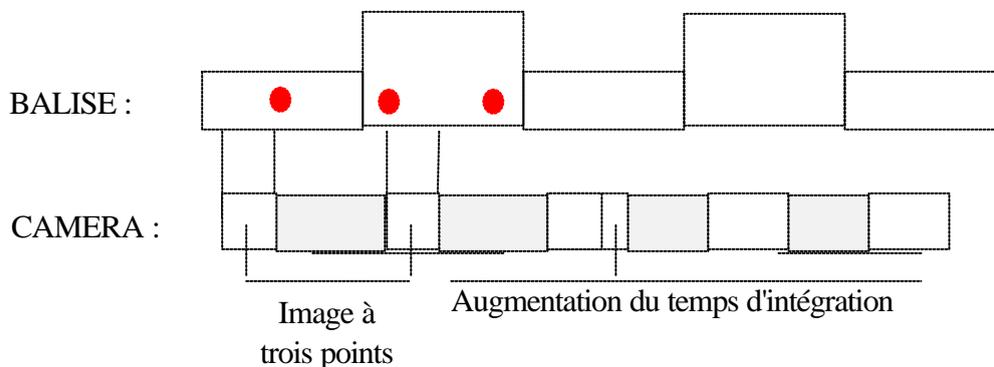


Le temps imparti à l'intégration se situe toujours au début du temps de cycle

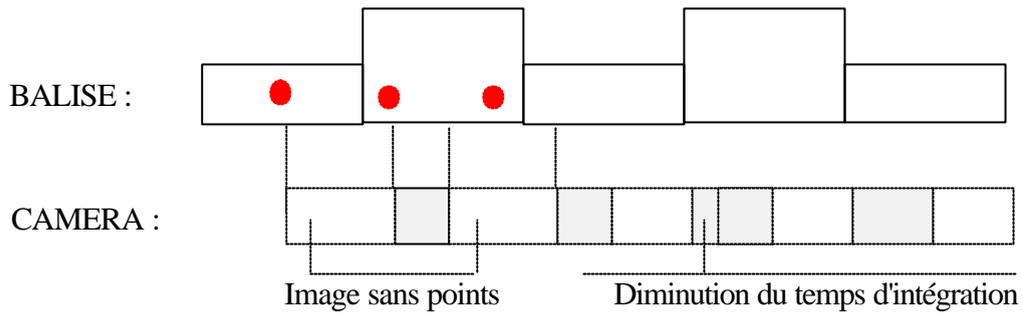


La recherche du temps d'intégration pour lequel la balise change d'état se fait par méthode dichotomique dans l'intervalle allant de $T_{i_0}=1$ ms à $TC=2$ ms :

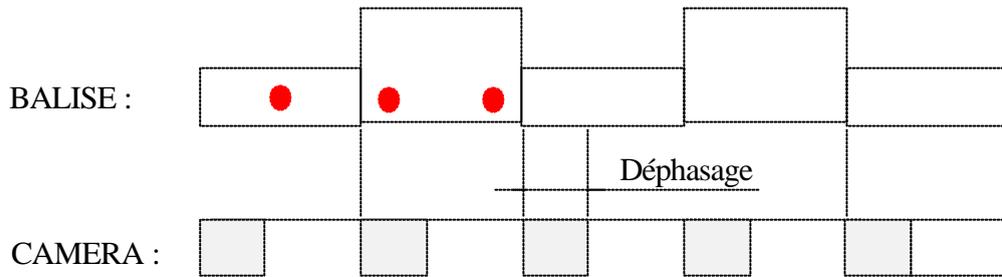
Pour T_{i_n} fixé, si on perçoit trois points sur deux images consécutives, on peut encore agrandir le temps d'intégration : $T_{i_{n+1}}=T_{i_n}+(TC-T_{i_n})/2$.



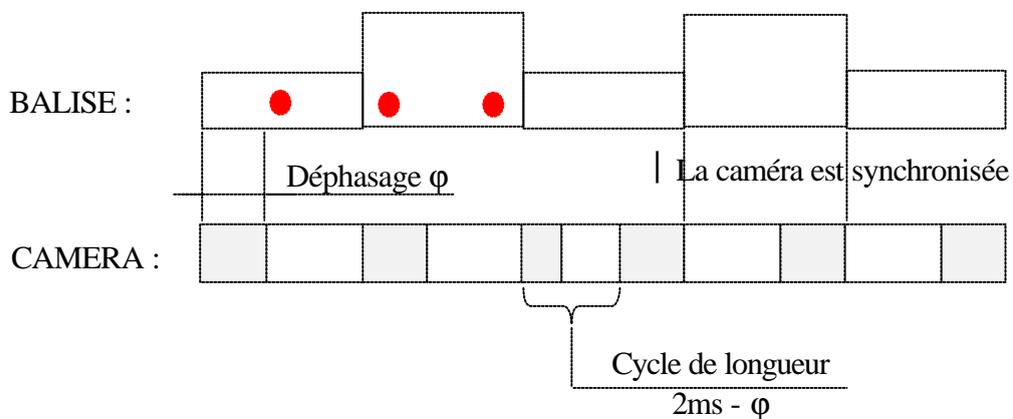
Si par contre l'image résultante est nulle, on doit diminuer le temps d'intégration pour revenir dans une zone où on reverra trois points : $T_{i+1} = T_i - (TC - T_i) / 2$.



On arrête la recherche à une certaine précision (ici 0.01 ms). A la fin de cette étape, la fin de chaque période d'intégration coïncide avec un changement d'état de la balise :



De plus, comme on connaît la durée du temps de cycle (qui est constante) et la durée du temps d'intégration, on peut déterminer la valeur du déphasage $f = TC - T_{final}$. Il suffit ensuite sur une acquisition de fixer TC à $2\text{ ms} - f$ puis de remettre TC à son ancienne valeur pour que la caméra et la balise soient en phase :



Transmission de données via CAN

1. Principe :



E

Pour relier entre eux les différents composants autonomes du CyCab, on utilise un bus de terrain : **le bus CAN** (Controleur Area Network). Ce bus permet de transmettre des informations d'un émetteur vers un récepteur. Le transit des informations se fait par une circulation de "trames" qui contiennent les informations suivantes :

Information	Type
• l'identifiant du récepteur du message	0..255
• l'identifiant de l'expéditeur du message	0..255
• l'identifiant du message	0..255
• le type du message (requête / acknowledgement)	0/1
• un drapeau indiquant si des données sont envoyées	0/1
• les éventuelles données	32 bits
• un drapeau indiquant si l'acknowledge a été reçu	0/1
• la clef de communication du message	0..255

2. Mise en œuvre :

La caméra linéaire a pour l'instant été testée seule sur un bus CAN ne lui permettant de communiquer qu'avec le PC, sans qu'il ne puisse y avoir de message provenant d'un autre module.

Par conséquent, tous les messages émis par la caméra sont destinés au PC, et inversement tous ceux émanant du PC sont reçus par la caméra. Les champs décrivant l'émetteur, le récepteur, et les drapeaux ne sont pas utiles, seules comptent les clefs de communication car elles définissent les fonctionnalités du produit (les ordres qu'on peut lui donner). Dans le cas de la caméra linéaire, **ces clefs** sont au nombre d'une vingtaine et s'organisent en trois catégories de messages :



F

- messages de fonctionnement
- messages d'accès aux résultats du calcul
- messages de commande de la caméra à distance pour des évolutions du soft ou du débogage

La mise en œuvre du CAN ne s'est pas déroulée sans problèmes, car il a fallu mettre en place un protocole de transmission de données que je ne connaissais pas.

Cependant ce mode de transmission de données s'est révélé être très performant quant au temps de transit des informations.

Gestion des interruptions

La caméra est équipée d'un microcontrôleur (68331) qui doit fonctionner de manière autonome, il s'est donc révélé nécessaire de programmer les routines d'interruption de la caméra en Assembleur, en faisant bien attention à ce qui se passait à l'exécution, chose que je n'avais encore jamais faite.

La caméra peut en effet interrompre son travail dans les trois cas suivants :

- début de l'acquisition d'une ligne vidéo,
- fin de l'acquisition d'une ligne vidéo,
- arrivée d'une trame sur le bus CAN.

Dans le cas où la caméra doit interrompre le programme en cours d'exécution pour traiter une interruption liée à l'un des trois événements cités ci-dessus, il faut sauvegarder l'état dans lequel se trouvaient toutes les variables internes, traiter l'interruption "au plus vite", puis restaurer l'environnement de travail du programme qui a été interrompu, avant de le laisser poursuivre son exécution.

Toutes les routines d'interruption de la caméra ont donc cette structure commune :

- sauvegarde des registres,
- traitement de l'interruption,
- restauration des registres.



Voir le code source de
cl_interrupt.s

La partie "traitement de l'interruption" est en général constituée d'un code très court, se contentant de mettre un indicateur à 0 ou à 1, pour signaler qu'un événement a eu lieu.

Cependant l'interruption liée à l'arrivée d'une trame CAN doit être traitée sans délai, car dans tous les cas de messages envoyés par le PC, il faut que la caméra réponde très vite, ne serait ce que pour signifier qu'elle a bien reçu le message.

On a donc une routine d'interruption dans laquelle on reste plus longtemps, et cela a participé à rendre le système moins stable dans certains moments de la mise au point logicielle.

Suivi de véhicule

1. Présentations des logiciels utilisés :

1.1. ORCCAD : Open Robot Controller Computer Aided Design

☞ *ORCCAD est un environnement logiciel 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.*

Participants :

ORCCAD a été à l'origine conçu par des chercheurs du projet Icare de l'INRIA et de l'ISIA, avec le support de la société Aleph Technologies et du Ministère de la recherche. Des applications en robotique sous-marine ont été soutenues par l'IFREMER. Le projet BIP, en relation avec l'action Praxitèle, a également contribué à la mise en œuvre du logiciel dans des expérimentations de conduite automatisée. Les développements actuels sont effectués conjointement par les projets BIP et ICARE et le service robotique de l'INRIA Rhône-Alpes.

Applications :

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.

Architecture logicielle :

ORCCAD est organisé en une boîte outils destinée à faciliter et factoriser le développement des logiciels de contrôle/commande de différents systèmes robotiques. Cette boîte fournit les outils pour accompagner l'utilisateur tout au long de la phase de conception du contrôleur : des spécifications aux codages. Les outils sont agencés autour d'un noyau. Ce noyau logiciel est l'ensemble de la hiérarchie des classes modélisant les Modules, tâches robot (TRs) et procédure robot (PrRs) du système.

Logiciels connectés :

Le noyau d'ORCCAD, qui permet la conception de divers niveaux de tâches et qui est réalisé selon une approche orientée objet, utilise ou est connecté à divers outils :

- ESTEREL , langage synchrone impératif, et son environnement de simulation discrète et de vérification (fc2tools, xes, et autograph);
- VxWorks, de WindRiver systems (et son environnement Tornado), OS temps réel industriel largement répandu.
- MAESTRO, un langage métier pour la robotique;
- SIMPARC, un simulateur hybride permettant de valider la discrétisation temporelle de schémas de commande et leur exécution sur une architecture cible.

1.2. SIMPARC : Simulation for Multi-Processor Advanced Robot Controller

☞ *SIMPARC permet de simuler, en fonction du temps, le comportement d'un système robotique commandé. Il est constitué d'un ensemble de classes C++ conçues pour créer des programmes de simulation, et permettant de simuler le comportement de robots commandés par des contrôleurs numériques.*

SIMPARC est capable d'effectuer plusieurs tâches :

- Simulation de la dynamique.
- Exécution des programmes utilisateur permettant de décrire des comportements.
- Prise en compte des temps de calcul.
- Simulation de contrôleur multiprocesseurs.

Les événements sont générés par l'exécution des programmes utilisateur sur l'architecture simulée. Une application SIMPARC est décrite par un ensemble de composants communicants :

- Les composants communiquent par l'intermédiaire de broches .
- Les broches sont connectées par des nets.

Chaque composant , broche ou net possède des propriétés qui lui sont propres. Les communications entre les broches des composants sont de type maître-esclave :

- Les broches maître comportent un processus qui initialise la communication
- Les broches esclaves possèdent des fonctions de services qui répondent aux communications.

Un mécanisme d'adressage permet de sélectionner les broches esclaves , et il ne doit pas y avoir de recouvrement d'adresse sur un même net.

SIMPARC peut effectuer deux types de simulations :

- Simulation analogique
- Simulation échantillonnée

2. Principe :

Le capteur constitué de la caméra et de la balise est dédié à une application de suivi de véhicule, pour lequel j'ai pu mettre en place une simulation à partir d'une méthode connue pour sa simplicité et ses performances dans ce domaine, reposant essentiellement sur l'algorithme de Kanayama.

Cet algorithme "trace une droite" entre la cible et le véhicule suiveur et donne ainsi la direction à suivre (orientation des roues).

Une amélioration est possible en interpolant la trajectoire par un polynôme de degré 2, qui permettrait de moins "lisser" la trajectoire, et ainsi de suivre la cible en passant au plus près de là où elle est passée.

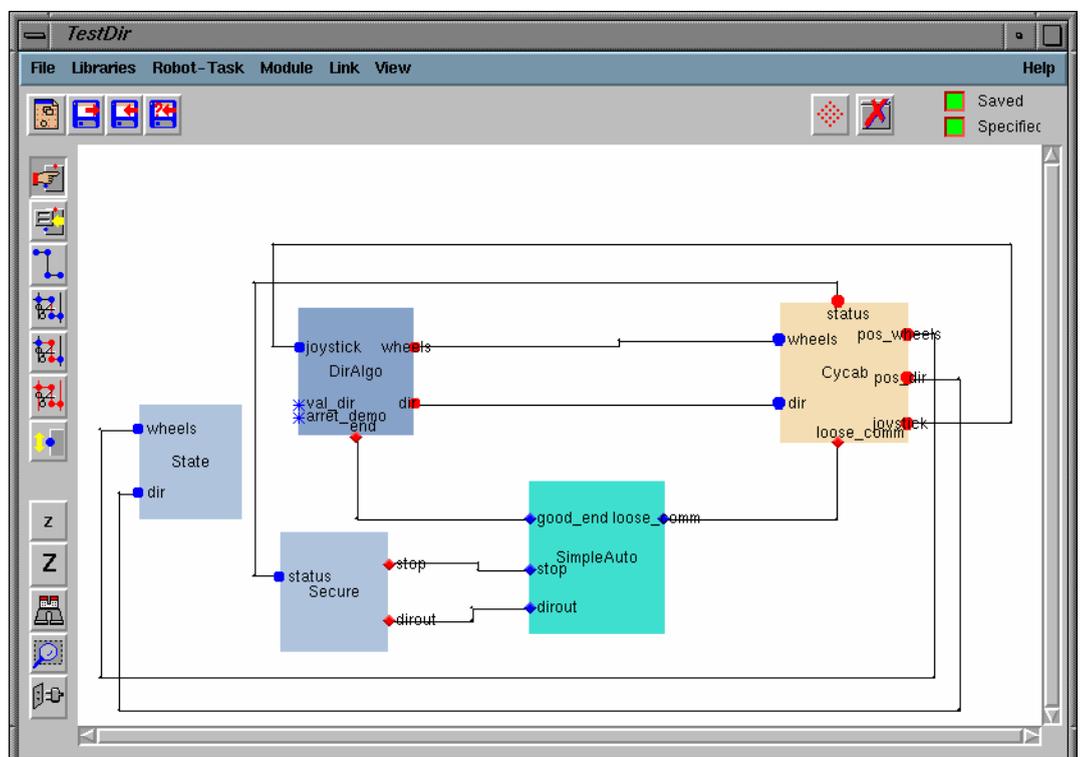
Cela améliorerait le suivi dans les courbes, où le véhicule suiveur a encore tendance à l'heure actuelle à "couper" les virages, d'où des problèmes possibles dans un environnement strict (voie étroite avec barrières, etc.).

3. Mise en œuvre :

3.1. Partie ORCCAD :

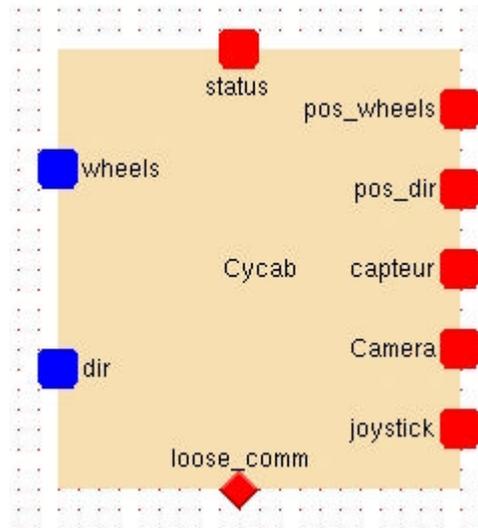
Grâce à ORCCAD, et avec l'aide d'un autre stagiaire de la halle robotique chargé de participer à la programmation du véhicule électrique en utilisant ORCCAD, j'ai pu mettre au point la simulation de suivi et tester quelques mouvements.

Ceci a été rendu possible grâce à une application existante sous ORCCAD qui permet de commander manuellement la voiture électrique.



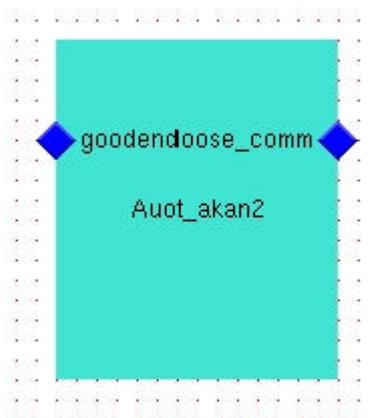
Cette application comme toutes les autres applications développées sous ORCCAD se compose de trois type de modules :

- Les modules ressource physique qui rassemblent les entrées et sorties du robot :



Ici, en sortie nous avons la position des roues, la position de la direction du braquage, la position du joystick, le status qui indique l'état de la voiture et le loose_com qui avertit l'automate si la communication est perdue. Pour les ports d'entrée nous avons les roues (les roues sont commandées en vitesse) et la direction de braquage. Pour chacun de ces ports il faut écrire un driver qui est le lien entre la ressource réelle et celle de ORCCAD. Pour la simulation, il faut écrire un deuxième driver pour chacun des ports. Le choix des drivers se fait lors de la compilation.

- Les modules automates qui gèrent les démarrages et arrêts de la tâche robot :



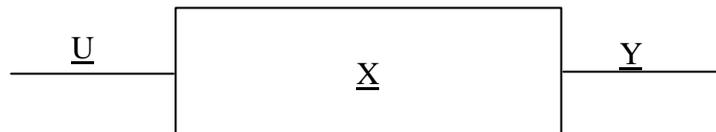
Comme montre le schéma ci-dessus chaque élément réel est représenté par un module spécifique, et ces modules communiquent entre eux via des BUS.

Pour la modélisation d'objets dynamiques, il faut avoir le modèle cinématique de l'objet.

Par exemple pour le CyCab, les équations du modèle utilisées dans la simulation :

$$\begin{cases} x'_f = V_r * \cos(\theta + k * \phi) \\ y'_f = V_r * \sin(\theta + k * \phi) \\ \theta' = V_r * \sin(\theta + k * \phi) / (l_w * V_r * \cos(\theta + k * \phi)) \end{cases}$$

Pour entrer ce modèle dans la simulation, il faut suivre le schéma suivant :



X vecteur d'état du système

U vecteur de commande appliqué en entrée du système

Y vecteur de mesures fournies en sortie par le système

Chaque roue du CyCab peut être commandée en vitesse angulaire et l'angle de braquage est commandé en position. Pour plus de réalisme, un retard a été ajouté à chacune de ces commandes. En effet le temps de réaction de la mécanique n'est pas négligeable.

Enfin, il est possible de visualiser le résultat des différentes simulations grâce à un logiciel qui permet de créer un environnement en 3D. Les objets se déplacent selon les données enregistrées dans les fichiers, chaque objet étant repéré dans l'espace par ses six coordonnées x, y, z et les trois angles de rotation par rapports à chaque axe.

3.3. Tests :

Des tests ont été effectués pour vérifier le bon fonctionnement de la simulation, et ont répondu à nos attentes, malgré le problème précédemment cité (trajectoires courbes).

Cependant ces tests étaient restreints : un seul véhicule suiveur, environnement très protégé, mais ils donnent néanmoins un bon aperçu de ce qu'il est possible de réaliser en matière de suivi.

Des tests plus poussés seront donc à faire avant de pouvoir réellement implanter l'application sur les véhicules.

Conclusion :

Bien que le sujet de ce stage ait été très semblable au précédent, dans la partie "capteur de distance et d'angle" du moins, ce que j'ai réalisé cette année m'a semblé beaucoup plus compliqué, cela étant probablement dû à l'évolution du matériel utilisé vers un matériel de pointe.

La dernière partie de mon stage, la commande de véhicule et la simulation étaient des notions très nouvelles pour moi, et leur découverte m'a beaucoup intéressé.

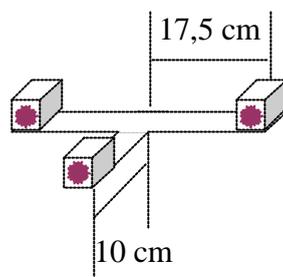
Enfin la réalisation de la balise infrarouge à trois points m'a pris une grande partie de mon temps car là aussi j'ai dû acquérir "sur le tas" les connaissances nécessaires en électronique pour y parvenir.

Ce projet de fin d'année a été très enrichissant car j'ai découvert un domaine d'application où les enseignements que j'ai reçus m'ont été d'un grand secours mais m'ont aussi permis d'acquérir de nouvelles connaissances.

Annexes

Principe du fonctionnement différentiel de la balise

Afin de localiser précisément le véhicule qui précède, on équipe chaque véhicule d'une balise infrarouge placée à l'arrière. Cette balise en forme de "T" comporte trois sources d'émission d'infrarouges réparties comme sur le schéma. Les sources sont composées chacune de 2x9 LED IR en série, émettant à la longueur d'onde de 875 nm.



1. Position du problème :

Dans un environnement protégé (i.e. sans source parasite d'infrarouges) on peut distinguer la balise infrarouge avec la caméra sans aucun problème. Mais ce type de situation étant rare, on est obligé de trouver un moyen robuste de repérer la balise même s'il y a des parasites. On rend donc la balise "différentielle", c'est à dire que toutes ses sources d'émission n'émettent pas en même temps, et on utilise cette caractéristique pour extraire l'information utile du bruit ambiant.

2. Méthode :

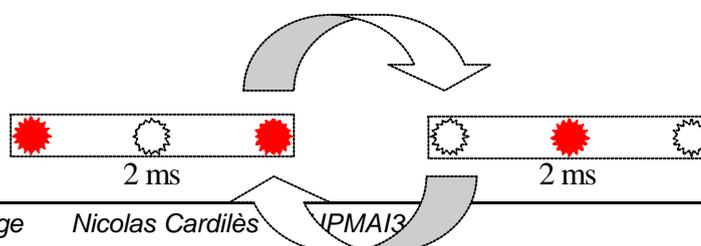
2.1. Du point de vue de la balise :

La balise a trois sources d'émission, on distingue deux groupes : le premier constitué des deux sources extrêmes, le second constitué de la source médiane.

Ces deux groupes de sources d'émission émettent en alternance, avec une période de 4 ms.

Cette période de fonctionnement a été choisie afin d'être d'une part suffisamment longue pour permettre à la caméra de réaliser les acquisitions d'images et les calculs de position, et d'autre part suffisamment courte pour ne pas constituer un facteur limitant en temps la délivrance des informations nécessaires pour la partie commande du véhicule.

La balise émet donc de la façon suivante :



2.2 Du point de vue de la caméra :

Pour mieux identifier la balise infrarouge au milieu du bruit ambiant (typiquement d'origine solaire) on utilise cette caractéristique de différenciation.

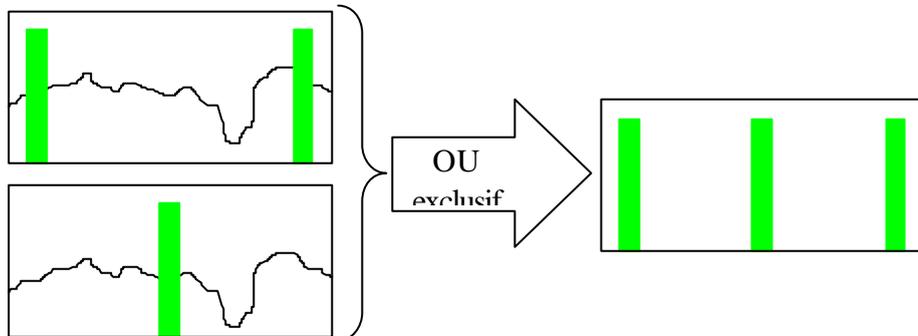
En effet, si la caméra et la balise sont en phase, en prenant deux images consécutives de la balise, on peut, par un "ou" exclusif effectué entre ces deux images, supprimer toutes les perturbations :

Rappel :

XOR, le "ou" exclusif... ou l'éloge de la différence :

X	Y	X XOR Y
1	1	0
1	0	1
0	1	1
0	0	0

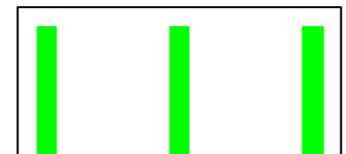
Pour ce que voit la caméra, cela se traduit de la façon suivante :



On arrive donc à supprimer les perturbations qui sont communes aux deux images enregistrées :



Et on garde la partie significative du signal : les trois pics de lumière infrarouge correspondant aux positions des sources de la balise.



Balise infrarouge à trois points

Voilà une brève description du travail que j'ai réalisé pour créer la balise infrarouge à trois points, suivie des schémas électroniques de montage, et des Data Sheets des composants principaux.

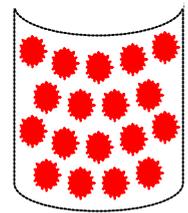
1. Implantation des diodes infrarouges

De façon à assurer une implantation optimale des diodes infrarouges sur la nouvelle balise, quelques valeurs clefs ont été calculées, à partir des contraintes imposées par les dimensions et les capacités des différents composants.

1.1 Données matérielles et contraintes physiques :

Données physiques de l'émetteur (balise)

Demi angle d'émission des diodes	12,5°
Nombre de blocs de diodes	3
Nombre de diodes par bloc	18
Disposition des diodes sur chaque bloc	5-4-5-4



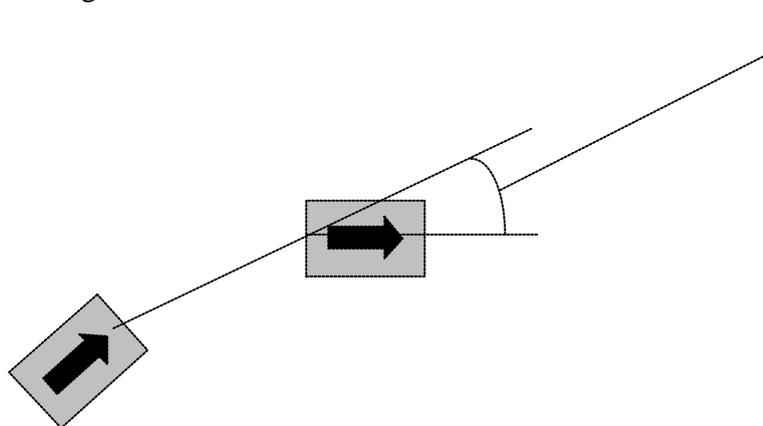
Données physiques du capteur (caméra)

Dimension de chaque pixel	14 x 14 μm
Nombre de pixels en ligne	2048
Taille du capteur linéaire	14 x 28672 μm



Contraintes dues au suivi de véhicules

Distance minimum entre l'émetteur et le récepteur	1,5 m
Distance maximum entre l'émetteur et le récepteur	10 m
Angle maximum de rotation du véhicule à suivre	45°



1.2. Calculs :

Demi angle de vision maximum du capteur : $\alpha \approx 27^\circ$

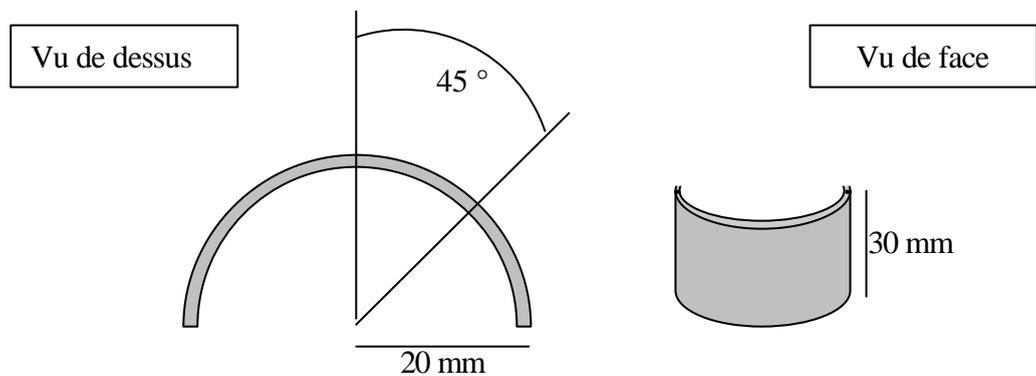
Résolution du capteur à la distance maximum (10 m) : $r \approx 5\text{mm}$

Demi angle d'émission maximum des blocs de diodes : $\beta \approx 45^\circ$

1.3. Implantation des diodes :

Pour placer correctement les diodes sur le support de fixation, il faut déterminer les angles que doivent faire entre eux les axes d'émission des diodes afin d'obtenir un demi angle d'émission final de 45° .

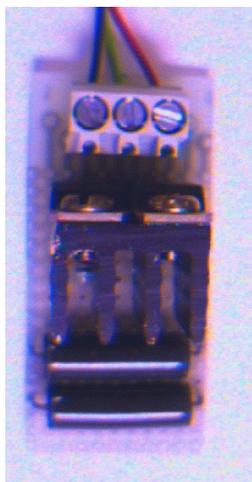
Le support de fixation est constitué d'une moitié de cylindre en PVC, d'axe vertical, de 20 mm de rayon, et de 30 mm de hauteur :



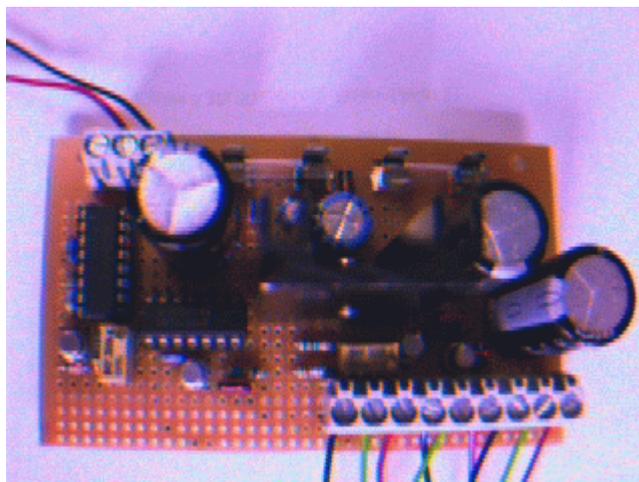
Les angles séparant les axes d'émission des diodes sont donc d'environ 16° .

2. Schémas électroniques :

Les schémas électroniques des cartes d'alimentation et de commande de puissance permettent de monter la balise complète. Une fois ce montage effectué, le résultat est celui présenté ci dessous :

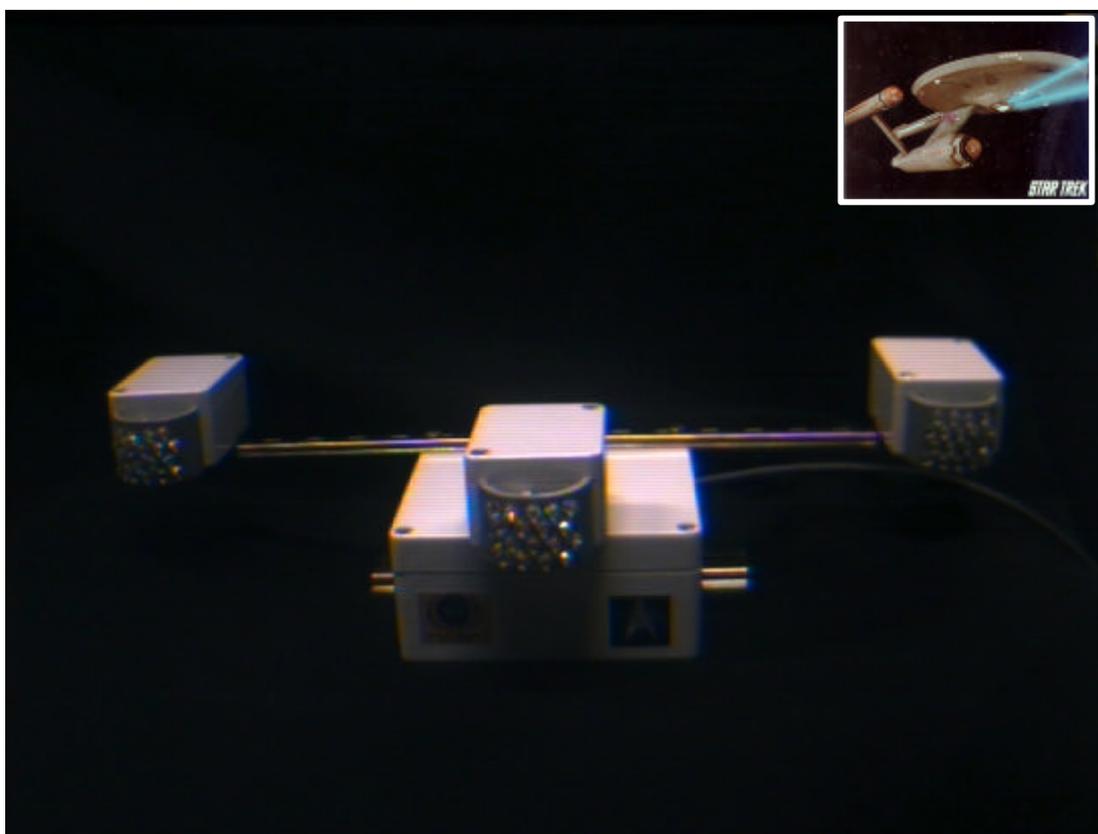


Carte de commande
de puissance



Carte d'alimentation et de génération des
impulsions nécessaires pour la balise.

La balise, terminée, ressemble un peu au vaisseau "Enterprise" de la série "Star Trek" :



Liste des composants utilisés (balise à 5 points) :

Résistances :			
Valeur	Nb ut	Réf. fabricant	Marque
4,75 Ohms +/-1% 100ppm/°C	6		
47 Ohms bobinée vitrifiée - 5W	2	Série W22	Welwyn
100 Ohms +/-1% 50ppm/°C	1		
330 Ohms	1		
1 KOhms	2		
12 KOhms	2		
33,2 KOhms +/-1% 50ppm/°C	6		
Potentiomètre 200 Ohms	2		
Condensateurs :			
Valeur	Nb ut		
3300 µF 25V	1		
2200 µF 25V	1		
1000 µF 25V	1		
100 µF 63 V	1		
10 µF 63V	2		
10 nF	2		
1,2 nF	2		
Diodes :			
Description	Nb ut	Réf. fabricant	Marque
Rectifieur	1	1N4003	Motorola
Diode	2	BZX79C2V7PH	Phillips
Diode infrarouge	54	TSHA5201	TFK
Fusibles :			
Description	Nb ut		
Fusible 1,6A	1		
Retard 2A	1		
Divers :			
Description	Nb ut	Réf. fabricant	Marque
Oscillateur à quartz	1	SPG8640BN	Seiko
Inverseur	1	SN74LS04N	TI
Transistor	4	2N2222A	SGS
Darlington	6	TIP130	SGS
Régulateur de tension	1	MC7805CT	Motorola

Liste des composants utilisés (balise à 3 points) :

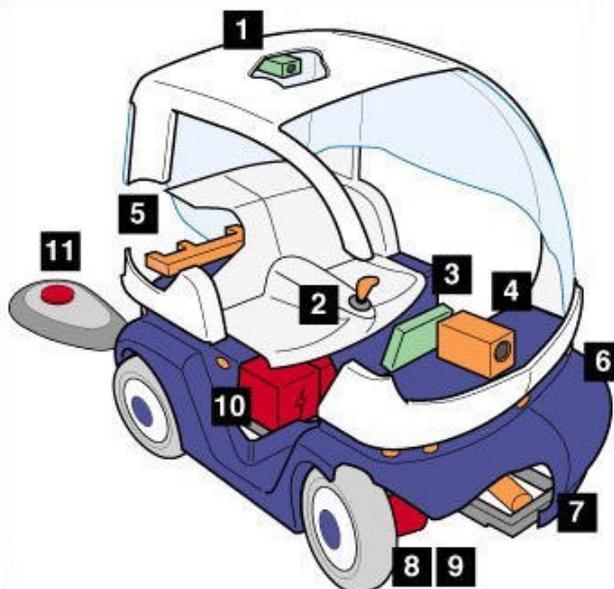
Résistances :			
Valeur	Nb ut	Réf. fabricant	Marque
4,7 Ohms	6		
47 Ohms bobinée vitrifiée 5W	6	Série W22	Welwyn
100 Ohms	1		
330 Ohms	1		
1 KOhms	2		
12 KOhms	2		
33,2 Kohms	6		
Potentiomètre 200 Ohms	2		
Condensateurs :			
Valeur	Nb ut		
3300 µF 25V	1		
2200 µF 25V	1		
1000 µF 25V	1		
100 µF 63 V	1		
10 µF 63V	1		
10 nF	4		
1,2 nF	2		
Diodes :			
Description	Nb ut	Réf. fabricant	Marque
Rectifieur	1	1N4003	Motorola
Diode	2	BZX79C2V4PH	Phillips
Diode infrarouge	54	TSHA5201	TFK
Fusibles :			
Description	Nb ut		
1,6A	1		
2A Retard	1		
Divers :			
Description	Nb ut	Réf. fabricant	Marque
Oscillateur à quartz	1	SPG8640BN	Seiko
Inverseur	1	SN74LS04N	TI
Transistor	4	2N2222A	SGS
Darlington	6	TIP131	SGS
Régulateur de tension	1	MC7805CT	Motorola
Boîtiers :			
Description	Nb ut		
Petit boîtier polycarbonate IP65	3		
Gros boîtier polycarbonate IP65	1		

Fiche technique du CyCab

Longueur : 1,90 m
Largeur : 1,20 m
Poids : 300 Kg



Motorisation : 4 moteurs électriques de 1 kW
4 roues motrices et directrices
Vitesse maximum 30 km/h
Autonomie : 2 heures d'utilisation continue
Capacité d'accueil : 2 personnes avec bagages en version de base
Conduite automatique ou manuelle
Accès par carte magnétique personnelle
Recharge automatique par induction



- 1 Caméra CCD pour la télé-opération
- 2 Joystick central de commande pour la conduite sécurisée
- 3 Terminal multimédia pour accéder aux informations touristiques et commerciales
- 4 Caméra linéaire pour l'accrochage immatériel
- 5 Balises Infra rouges pour l'accrochage immatériel
- 6 Ceintures de capteurs ultrason pour la détection d'obstacles
- 7 Vérin de direction électrique
- 8 1 moteur électrique par roue
- 9 1 frein électrique par roue
- 10 Batteries + gestionnaire de charge
- 11 Borne de recharge par induction

Contrôle du temps d'exposition

1. Position du problème :

Le contrôle de l'exposition consiste à réguler la quantité de lumière perçue par la caméra. Pour éviter de saturer le capteur ou pour avoir une image avec une dynamique suffisante, il faut contrôler le temps d'exposition de chaque prise d'image.

Cependant, ce contrôle ne peut se faire qu'en utilisant des outils de prévision, car on ne peut pas mesurer la quantité de lumière que va percevoir le capteur, sans prendre d'image. Comme les images sont prises en continu, avec un intervalle de temps très court entre deux images, on va tenter de prévoir la quantité de lumière qui sera perçue au $n^{\text{ième}}$ coup à partir des maximums des images précédentes.

Précision importante : la quantité de lumière perçue varie linéairement avec le temps d'exposition (i.e. : avec une source de lumière d'intensité constante, si on double le temps d'exposition, on double la quantité de lumière perçue par le capteur, et donc l'amplitude du pic traduisant la présence de la source lumineuse).

2. Méthodes :

Compte tenu des travaux réalisés auparavant et des observations précédentes, j'ai développé deux méthodes de contrôle du temps d'exposition.

2.1. Première méthode :

J'ai d'abord repris une méthode existante et déjà utilisée pour ce problème, qui consiste à fixer une amplitude "idéale" et à tenter en jouant sur le temps d'exposition, d'obtenir des mesures dont l'amplitude s'en rapproche le plus.

Cette méthode est basée sur l'équation suivante (les d_i sont des temps d'exposition) :

$$\delta_n = \delta_{n-1} \times \frac{\text{Max}_{\text{voulu}}}{\text{Max}_{d_{n-1}}}$$

Cependant ce contrôle seul présente un très fort risque d'oscillation, qu'on va tenter de "calmer" en ralentissant l'asservissement, c'est à dire en utilisant plusieurs boucles pour contrôler la lumière.

On obtient alors, sur deux mesures consécutives :

$$\delta_n = \delta_{n-1} \times \frac{\text{Max}_{\text{voulu}}}{\max(\text{Max}_{d_{n-1}}, \text{Max}_{d_{n-2}})}$$

Le problème de cette méthode est cependant de produire de fortes oscillations même en utilisant plusieurs mesures consécutives pour le contrôle du temps d'exposition.

2.2. Deuxième méthode :

On ne tente plus maintenant de se rapprocher d'un temps d'exposition idéal, mais simplement en se basant sur le calcul du maximum perçu sur chaque ligne, et compte tenu de la relation linéaire entre temps d'exposition et maximum d'une ligne, on utilise une équation différentielle du premier ordre avec second membre pour estimer le temps d'exposition de la prise de vue suivante, en créant le minimum d'oscillations :

Equation générale :
$$T \times \frac{\partial y}{\partial t} + y = x$$

Les temps d'exposition successifs peuvent s'écrire sous forme d'une suite :

$$Y_{n+1} = T \times X_{n+1} + (1 - T) \times Y_n$$

où les Y_i sont des temps d'exposition et X_i représente le maximum de la ligne i .

En identifiant : $\frac{\partial y}{\partial t} = \frac{Y_{n+1} - Y_n}{T_0}$

On obtient :
$$Y_{n+1} = \frac{T_0}{T} \times X_{n+1} + \left(1 - \frac{T_0}{T}\right) \times Y_n$$

D'où :
$$Y_{n+1} = K \times X_{n+1} + (1 - K) \times Y_n$$

En jouant sur la valeur de K , on influe sur la vitesse de réaction de l'asservissement, et on peut ainsi éviter les oscillations.

3. Conclusions :

Le contrôle du temps d'exposition n'a pas été implanté au niveau du soft car la caméra était déjà suffisamment lente bien que l'utilité de ce contrôle soit évidente. J'ai donc préféré trouver la bonne ouverture du diaphragme de la caméra pour ne pas avoir à implanter cet asservissement. Cependant l'asservissement fonctionne correctement (la deuxième méthode surtout) et une évolution du soft pouvant accélérer les calculs sur la caméra, je l'ai joint au rapport.

CAN

<http://www.can-cia.de/ican.htm>

The Controller Area Network (CAN) is a serial bus system especially suited for networking "intelligent" devices as well as sensors and actuators within a system or sub-system.

The Attributes of CAN

CAN is a serial bus system with multi-master capabilities, that is, all CAN nodes are able to transmit data and several CAN nodes can request the bus simultaneously. The serial bus system with real-time capabilities is the subject of the ISO 11898 international standard and covers the lowest two layers of the ISO/OSI reference model. In CAN networks there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted. A transmitter sends a message to all CAN nodes (broadcasting). Each node decide on the basis of the identifier received whether it should process the message or not. The identifier also determines the priority that the message enjoys in competition for bus access. The relative simplicity of the CAN protocol means that very little cost and effort need to be expended on personal training; the CAN chips interfaces make applications programming relatively simple. Introductory courses, function libraries, starter kits, host interfaces, I/O modules and tools are available from a variety of vendors permitting low-cost implementation of CAN networks. Low-cost controller chips implementing the CAN data link layer protocol in silicon and permitting simple connection to microcontrollers have been available since 1989. Today there are more than about 50 CAN protocol controller chips from more than 15 manufacturers announced, and available. The use of CAN in most of European passenger cars and the decision by truck and off-road vehicle manufacturers for CAN guarantees the availability of CAN chips for more than 10 years. Other high volume markets, like domestic appliances and industrial control, also increase the CAN sales figures. Up to spring 1997 there are more than 50 million CAN nodes installed. One of the outstanding features of the CAN protocol is its high transmission reliability. The CAN controller registers a stations error and evaluates it statistically in order to take appropriate measures. These may extend to disconnecting the CAN node producing the errors.

Each CAN message can transmit from 0 to 8 bytes of user information. Of course, you can transmit longer data information by using segmentation. The maximum transmission rate is specified as 1 Mbit/s. This value applies to networks up to 40 m. For longer distances the data rate must be reduced: for distances up to 500 m a speed of 125 kbit/s is possible, and for transmissions up to 1 km a data rate of 50 kbit/s is permitted.

CAN Applications

CAN networks can be used as an embedded communication system for microcontrollers as well as an open communication system for intelligent devices. The CAN serial bus system, originally developed for use in automobiles, is increasingly being used in industrial as well as building automation, medical equipment and maritime electronics. If the requirements for passenger cars networks are compared with those for industrial field bus systems, the similarities are remarkable. In both cases some of the major requirements are: low cost, the ability to function in a difficult electrical environment, a high degree of real-time capability and ease of use. Some users, for example in the field of medical engineering, opted for CAN because they have to meet particularly stringent safety requirements. Similar problems are faced by manufacturers of other equipment with very high safety or reliability requirements (e. g. robots, lifts and transportation systems).

Clefs de communication et leurs effets

Clef de communication

Action associée

Fonctionnement Standard

CL_KEY_RUN	Lancement du programme de localisation sur la caméra
CL_KEY_START_READ_RESULT	Si de nouvelles valeurs sont disponibles, bloque l'accès aux résultats du calcul pendant leur envoi via CAN et retourne 1, sinon retourne 0 (sans bloquer l'accès aux valeurs)
CL_KEY_READ_RESULT_ALPHA	Envoie l'angle alpha en radians
CL_KEY_READ_RESULT_BETA	Envoie l'angle bêta en radians
CL_KEY_READ_RESULT_DIST	Envoie la distance entre les deux véhicules en mètres
CL_KEY_READ_RESULT_DEV	Envoie la déviation entre les deux véhicules en mètres
CL_KEY_SEND_SPOTS_POSITION	Envoie la position des centres des tâches en nombre de pixels sur la barrette CCD
CL_KEY_STOP_READ_RESULT	Ré-autorise l'accès aux résultats du calcul
CL_KEY_END	Arrêt du programme de localisation
CL_KEY_STOP	Arrêt de la caméra

Mode DEBUG

CL_KEY_GET_BUFFERS	Provoque l'acquisition des deux buffers vidéo
CL_KEY_SEND_BUFFERS	Envoie deux des pixels mémorisés de chaque buffers
CL_KEY_SEND_NB_ACQ	Envoie le nombre d'acquisitions réalisées
CL_KEY_READ_PARAMETER_TC	Envoie le paramètre "temps de cycle"
CL_KEY_WRITE_PARAMETER_TC	Modification du paramètre "temps de cycle"
CL_KEY_READ_PARAMETER_TI	Envoie le paramètre "temps d'intégration"
CL_KEY_WRITE_PARAMETER_TI	Modification du paramètre "temps d'intégration"

Sources

Formule de base du copyright :

« Logiciel de gestion de la caméra linéaire ©INRIA dans sa version 1 du 07/08/98, ci-après dénommé "Le Logiciel ".

L'INRIA détient tous les droits de propriété sur le Logiciel, il souhaite que la communauté scientifique l'utilise afin de le tester et de l'évaluer. L'INRIA donne gracieusement le droit d'utiliser ce Logiciel. Toute utilisation dans un but lucratif ou à des fins commerciales est interdite sauf autorisation expresse et préalable de l'INRIA.

L'INRIA autorise toute reproduction de ce Logiciel :

- dans les limites définies aux articles 9 et 10 de la convention de Berne pour la protection des œuvres littéraires et artistiques pris respectivement dans leurs alinéas 2 et 3, c'est-à-dire à condition :
 - "qu'une telle reproduction ne porte pas atteinte à l'exploitation normale de l'œuvre, ni ne cause un préjudice injustifié aux intérêts légitimes de l'auteur,
 - que les citations à titre d'illustration et/ou d'un enseignement soient conformes aux bons usages et qu'il soit fait mention de la source et du nom de l'auteur, si ce nom figure dans la source",
- à condition que toute reproduction soit accompagnée du présent fichier.

Toute demande de droits d'exploitation plus étendus doit être faite par e-mail à :
copyright@inria.fr.

Toute utilisation hors des limites précisées ci-dessus et réalisée sans l'accord exprès préalable de l'INRIA constituerait donc le délit de contrefaçon.

Le Logiciel étant un produit en cours de développement, l'INRIA ne saurait assurer aucune responsabilité et notamment en aucune manière et en aucun cas, être tenu de répondre d'éventuels dommages directs ou indirects subis par l'utilisateur. »

