

Réalisation d'une loi de commande pour le  
contrôle longitudinal d'un véhicule  
automatique

Jean-Philippe ROUSSEL

9 juillet 2001

wgsxwqy

## Table des matières

<b>1</b>	<b>Remerciements</b>	<b>5</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
<b>3</b>	<b>Présentation de l'INRIA</b>	<b>7</b>
3.1	Quelques chiffres . . . . .	7
3.2	Les activités . . . . .	8
3.3	L'INRIA Rhône-Alpes . . . . .	8
3.4	Le projet BIP . . . . .	9
3.5	Le service robotique . . . . .	10
<b>4</b>	<b>Sujet du stage</b>	<b>11</b>
<b>5</b>	<b>Présentation du Cycab</b>	<b>11</b>
5.1	Fiche technique . . . . .	11
5.2	laser télémétrique . . . . .	13
<b>6</b>	<b>Présentation du logiciel Orccad</b>	<b>16</b>
6.1	Utilité du logiciel . . . . .	16
6.2	L'architecture du logiciel . . . . .	17
<b>7</b>	<b>Loi de commande théorique</b>	<b>20</b>
7.1	Convention et noms de variables . . . . .	20
7.2	Conditions de validité et conditions initiales . . . . .	20
7.3	Loi de commande . . . . .	21
<b>8</b>	<b>Déroulement du stage</b>	<b>22</b>
<b>9</b>	<b>Travail réalisé</b>	<b>23</b>
9.1	Structure existante . . . . .	23
9.2	Modification de la structure existante . . . . .	25
<b>10</b>	<b>Problèmes rencontrés</b>	<b>28</b>
10.1	Problèmes logiciels . . . . .	28
10.2	Dérive mécanique . . . . .	28
<b>11</b>	<b>Les tests réalisés</b>	<b>29</b>
11.1	Mode opératoire . . . . .	29
11.2	Comparaison des résultats théoriques et expérimentaux . . . . .	29
11.3	Mise au point . . . . .	30

<i>TABLE DES MATIÈRES</i>	4
<b>12 Conclusion</b>	<b>38</b>
<b>A Annexes</b>	<b>40</b>
A.1 Module SickCataSuivi . . . . .	40
A.2 Module Cycab . . . . .	41
A.3 Module Dérivation . . . . .	42
A.4 Module Init . . . . .	43
A.5 Module AccelRule . . . . .	44
A.6 Module Integrator100ms . . . . .	45

## 1 Remerciements

*Je tiens à remercier en tout premier lieu MM. Alain Girault et Hervé Mathieu, mes maîtres de stage, pour leur disponibilité et leurs conseils avisés, toute l'équipe du projet BIP et des moyens robotiques pour leur chaleureux accueil et enfin l'ensemble du personnel et des stagiaires de l'INRIA Rhône-Alpes, et plus particulièrement Vincent Bozonnet pour sa patience et sa dextérité à la conduite du Cycab leader.*

## 2 Introduction

Dans le cadre des recherches sur les moyens de transport du futur et sur la route automatisée (<http://www.lara.prd.fr>), l'INRIA a conçu un véhicule automatique appelé Cycab [BGMPG99]. Un consortium de laboratoires de recherche et d'industries travaillent en collaboration dans ce grand programme français de Recherche et Développement.

Les techniques développées dans ce projet sont principalement basées sur des aides à la conduite et sur la sécurité. L'étude théorique d'une loi de commande a été réalisée afin d'assurer un suivi de véhicules. Ce suivi a un double intérêt, en assurant une sécurité optimum qui écarte les risques de collisions par régulation de l'accélération, et en optimisant les infrastructures existantes. En effet, l'automatisation du déplacement permet de réduire au minimum l'encombrement et d'augmenter le débit de véhicules sur la route. A partir de cette étude théorique, il était nécessaire de mettre en place cette loi de commande sur un véhicule automatique appelé Cycab, lui aussi développé dans le cadre de ce projet de recherche. Le but de ce stage de fin d'études de cycle ingénieur était donc d'assurer un suivi longitudinal de véhicules à partir de la loi de commande théorique. Le passage d'une étude théorique à la mise en place sur un matériel robotique n'étant généralement pas immédiate, il fallait donc réaliser, tester puis mettre au point sur le Cycab cette commande de suivi.

Ce travail s'est effectué grâce à Orccad, un environnement logiciel permettant de concevoir et de mettre en œuvre le contrôle et la commande de systèmes robotiques complexes. La sûreté de programmation et les possibilités qu'il propose fait de ce logiciel un outil particulièrement adapté lors de réalisations de programmes pour des applications embarquées.

## 3 Présentation de l'INRIA

L'INRIA (Institut National de Recherche en Informatique et Automatique) a l'ambition d'être au plan mondial un institut de recherche au cœur de la société de l'information. Sa volonté est de mettre en réseau des compétences et des talents de l'ensemble du dispositif de recherche français dans le domaine des STIC. Ce réseau permet de mettre l'excellence scientifique au service des progrès technologiques, créateurs d'emplois, de richesse et de nouveaux usages répondant à des besoins sociaux-économiques. Son organisation décentralisée (5 unités de recherche), ses petites équipes autonomes et évaluées régulièrement permettent à l'INRIA d'amplifier ses partenariats : 47 projets de recherche sur 87 sont communs avec les universités, les grandes écoles et les organismes de recherche ; il renforce son implication dans les travaux de valorisation des résultats de recherche et le transfert technologique : 600 contrats Recherche et Développement avec l'industrie et un presque d'une cinquantaine de sociétés sont issues de l'INRIA.

### 3.1 Quelques chiffres

Quelques chiffres (décembre 2000) :

- Ressources budgétaires
  - dotation de l'état : 442 MF HT
  - ressources propres : 174 MF HT
- Ressources humaines
  - titulaires INRIA : 724
  - post-doctorants et Contractuels : 256
  - doctorants : 550
  - chercheurs et enseignants d'autres organismes : 230
  - conseillers, collaborateurs divers et invités : 430
- Indicateurs
  - contrats de recettes actifs : plus de 600
  - contrats de recettes signés dans l'année : plus de 200
  - presque une cinquantaine de sociétés sont issues de l'INRIA, depuis Ilog, aujourd'hui cotée au Nasdaq, jusqu'aux toutes dernières, 5 en 1998, 6 en 1999, 11 en 2000.

- 7 brevets initiaux déposés en 1999 : 1 est en pleine propriété INRIA, les autres sont en copropriété, 5 avec des industriels et 1 avec une université.

### 3.2 Les activités

Créé en 1967 à Rocquencourt près de Paris, l'INRIA est un établissement public à caractère scientifique et technologique (EPST) placé sous la double tutelle du ministère de la recherche et du ministère de l'économie, des finances et de l'industrie. L'INRIA est organisée en différents projets de recherche. Un projet de recherche est une équipe rassemblant de 15 à 20 personnes autour d'une thématique forte et sur des objectifs scientifiques précis. Ces équipes gèrent de façon autonome leur budget. Les résultats obtenus et les retombées industrielles qu'ils induisent sont régulièrement évalués. Une action de recherche est un groupe de chercheurs poursuivant un travail commun sur une thématique spécifique qui peut aboutir à la création d'un projet de recherche.

### 3.3 L'INRIA Rhône-Alpes

L'INRIA Rhône-Alpes est une des cinq entités de L'INRIA. Créée en 1992, c'est la plus récente des cinq unités existantes. Plus de 330 personnes dont 220 chercheurs travaillent sur le site de Montbonnot. Cette unité accueille également 130 doctorants, ingénieurs et stagiaires. Les principales missions de L'INRIA Rhône-Alpes sont les suivantes :

- Entreprendre des recherches fondamentales et appliquées
- Réaliser des systèmes expérimentaux
- Organiser des échanges scientifiques internationaux
- Assurer le transfert et la diffusion des connaissances et du savoir-faire
- Contribuer à la valorisation des résultats de la recherche
- Contribuer, notamment par la formation, à des programmes de coopération pour le développement
- Effectuer des expertises scientifiques
- Contribuer à des actions de normalisation



### 3.4 Le projet BIP

C'est donc dans les locaux de l'INRIA Rhône-Alpes, au sein du projet BIP et en collaboration avec le service des Moyens Robotiques que le stage a été effectué. Le projet BIP, créé au 1er janvier 1994, a pour objectifs la conception de robots marcheurs de type bipède et, plus généralement, l'étude de techniques de contrôle/commande (algorithmes et architecture) de systèmes complexes :

- **Conception de robots marcheurs :** Le projet a pour objectif l'étude générique des divers aspects intervenant dans le contrôle/commande des systèmes robotiques complexes. Il s'intéresse en particulier aux robots marcheurs de type bipède. L'intérêt des systèmes marcheurs réside dans leur capacité d'adaptation à des terrains ou des sols variés, leur permettant de se déplacer dans des milieux très contraints en dimensions, voire mal structurés. Les robots anthropomorphes sont donc particulièrement aptes à évoluer dans nos environnements courants, privés ou industriels, essentiellement conçus pour la bipédie. Ainsi, les domaines d'application visés sont-ils en priorité les robotiques personnelle, de service et d'intervention. Parallèlement, le projet s'attache à développer des activités de modélisation dans certains domaines de la biomécanique. Enfin, le projet ne souhaite pas que les applications de ses recherches se limitent au domaine des seuls robots marcheurs, dont le marché industriel est actuellement marginal. C'est pourquoi les techniques étudiées sont voulues suffisamment génériques pour faire l'objet de mises en oeuvre dans d'autres domaines.
- **Méthodes et outils pour le contrôle/commande :** Le contrôle-commande temps-réel de systèmes robotiques complexes relève de la problématique plus générale de la modélisation et programmation de systèmes mêlant aspects de commande (continue échantillonnée) et de contrôle (discret). On les trouve en robotique, mais aussi en avionique, dans l'automobile et dans les automatismes de production. Ils sont réactifs, ont des modes multiples, des cadences multiples, et doivent être tolérants aux pannes. Ces problèmes sont abordés sous l'angle des méthodes formelles, notamment l'approche synchrone. Dans un souci d'utilisabilité par des spécialistes du domaine cible, et non de celui des techniques formelles, le projet propose des langages spécialisés pour la spécification de missions, de tâches, et de propriétés. Ce dernier point est lié à l'utilisation de méthodes de vérification formelle. La tolérance aux pannes est abordée comme un problème d'ordonnancement réparti

avec réplication. Enfin, la génération automatisée de contrôleurs est fondée sur l'utilisation de la synthèse de contrôleurs discrets. Les Moyens Robotiques et le projet BIP développent le logiciel Orccad (Open Robot Controller Computer Aided Design [BCME<sup>+</sup>98, SKE97, SEKPG97, KPG]). L'environnement logiciel Orccad (Open Robot Controller Computer Aided Design) permet de concevoir et de mettre en oeuvre le contrôle et la commande d'un système robotique complexe.

### 3.5 Le service robotique

Les missions du service robotiques sont quant à elles de trois types :

- Activité de service
  - maintenance des robots
  - 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

## 4 Sujet du stage

Résumé du sujet :

Le sujet proposé consiste à étudier une loi de commande en accélération pour le contrôle longitudinal de véhicules automatiques, et à la mettre en œuvre sur les Cycab dont dispose le laboratoire de robotique de l'INRIA Rhône-Alpes.

L'étude théorique de la stabilité asymptotique de la loi de commande a déjà été effectuée [GY99]. Le travail demandé consiste à :

- Ecrire des drivers pour les capteurs de position et de vitesse. Pour cela le Cycab est équipé d'un laser télémétrique.
- Ecrire la boucle de commande et choisir la période d'échantillonnage. On utilisera pour cela le logiciel Orccad qui est déjà utilisé pour d'autres applications sur les Cycab.
- Tester et mettre au point de la loi de commande sur deux Cycab. Le premier aura un profil d'accélération constitué de phases successives d'accélération et de freinage, tandis que le second sera commandé par la loi d'accélération et devra éviter l'accident.

## 5 Présentation du Cycab

### 5.1 Fiche technique

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.

Les caractéristiques générales du véhicule sont les suivantes :

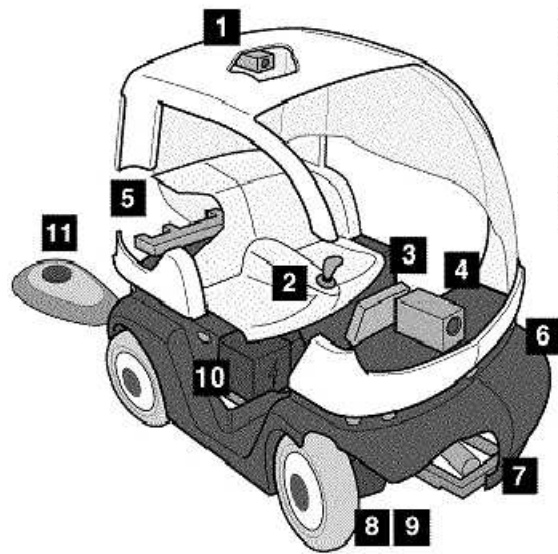
- Longueur hors tout : 1,90 m
- Largeur hors tout : 1,20 m
- Poids total avec batteries : 300 kg
- Motorisation : 4 moteurs électriques de 1 kW
- 4 roues motrices et directrices

FIG. 1: *Cycab*

- Vitesse théorique maximale : 20 km/h soit 5.5 m/s
- Autonomie : 2 heures d'utilisation continue
- Capacité d'accueil : 2 personnes avec bagages
- Conduite automatique ou manuelle

La vue éclatée permet d'observer les différents équipements du véhicule :

1. Caméra CCD pour la téléopération
2. Joystick central pour la conduite manuelle sécurisée
3. Écran tactile complétant l'Interface Homme Machine (IHM)
4. Laser télémétrique pour la localisation et le suivi de cible
5. Balises infrarouges pour l'accrochage immatériel
6. Vérin électrique de direction
7. Ceinture de capteurs à ultrasons pour la détection d'obstacles
8. Un moteur de traction par roue
9. Un frein à commande électrique par roue

FIG. 2: *Vue éclatée du Cycab*

10. Quatre batteries de 12 Volts et un gestionnaire automatique de charge

11. Borne de recharge électrique par induction fixée sur la voirie

leader est en effet essentielle pour assurer un bon suivi de véhicule.

## 5.2 laser télémétrique

Le Cycab est également équipé d'un laser télémétrique SICK dont les données sont exploitées pour la réalisation de la loi de commande. Une présentation de ce laser est donc nécessaire pour connaître ses caractéristiques et ses possibilités. La détection du véhicule leader est en effet primordiale pour assurer un bon suivi de véhicules.

- Principe optique. Le télémètre à balayage laser 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.

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



FIG. 3: *Laser télémétrique SICK*

de vol, mis par l'impulsion lumineuse à parcourir la distance capteur-objet puis objet-capteur. Le faisceau laser émis est dévié par un miroir tournant, permettant ainsi de balayer une surface de  $360^\circ$ .

Pour des raisons mécaniques, le faisceau sortant a une course limitée à  $100^\circ$  ou  $180^\circ$  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. La réception de ce signal arrête l'horloge. A partir de la mesure du temps séparant l'émission de la réception, on déduit la distance capteur-obstacle.

- Détection de catadioptrés. Le capteur SICK mesure le temps de vol, aller et retour, du faisceau laser. En supposant constante la vitesse de propagation de la lumière, 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 des catadioptrés pour le suivi de véhicule, qui sont installés sur l'arrière du véhicule leader.

- Portée théorique du faisceau :
  - 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.

En fait, lors des différents essais effectués, on a déterminé que la portée du laser sur les catadioptrés était de 10 mètres environ.

- 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. 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.

## 6 Présentation du logiciel Orccad

### 6.1 Utilité du logiciel

Orccad est un environnement logiciel permettant de concevoir et de mettre en oeuvre 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. 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 (Cycab) en passant par un robot sous-marin et un robot mobile terrestre avec une caméra.



## 6.2 L'architecture du logiciel

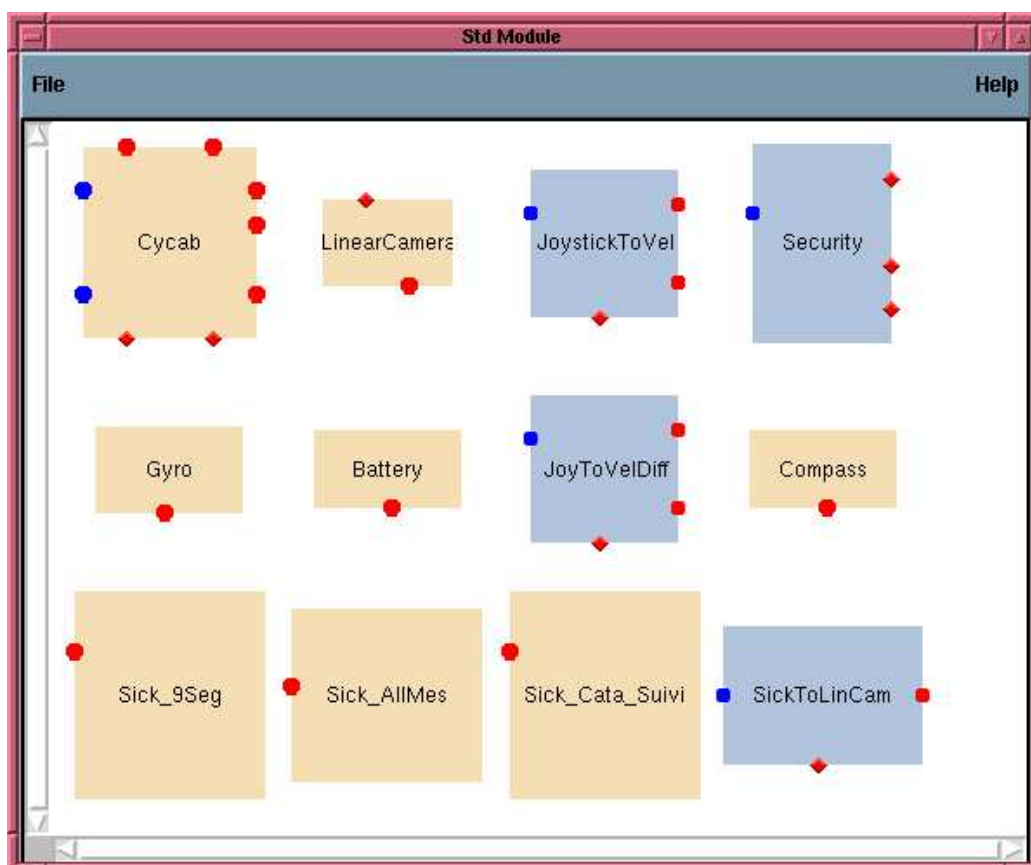
Une application Orccad est découpée hiérarchiquement en trois entités :

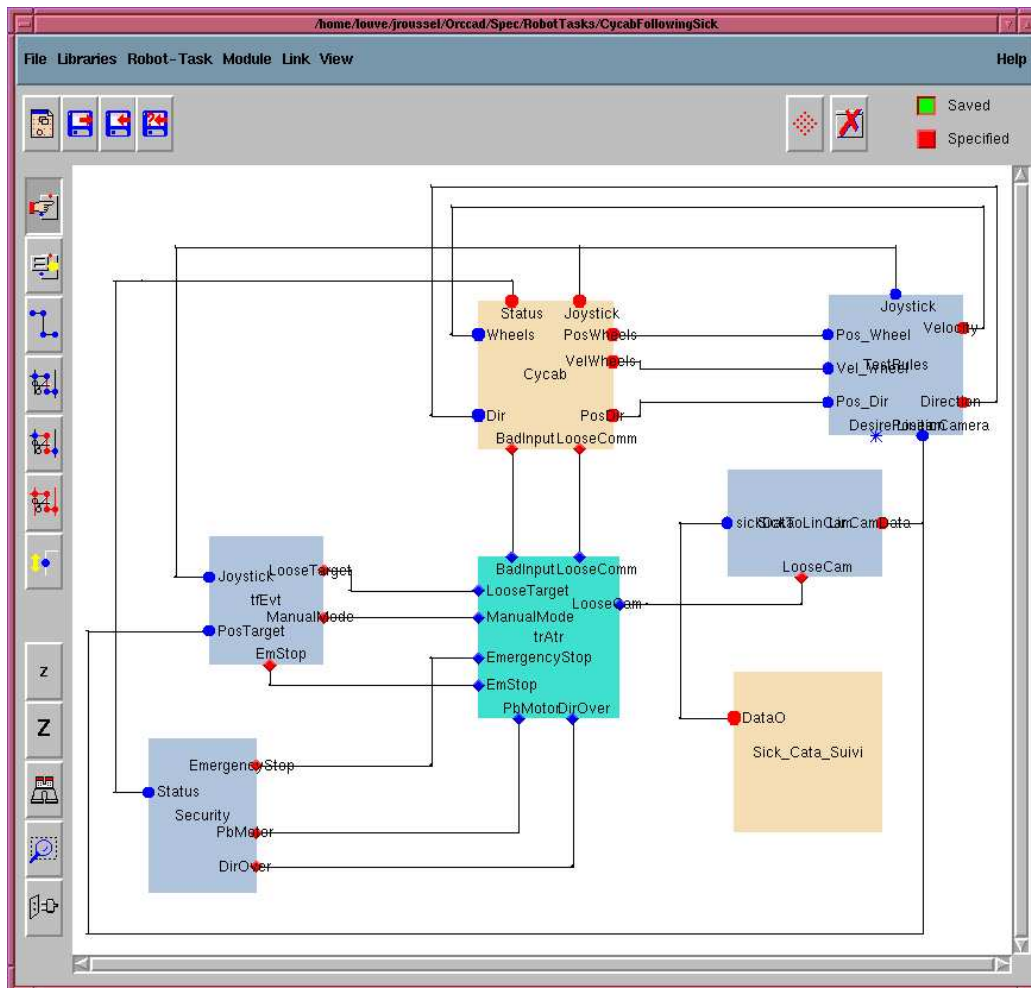
- le Module qui est le grain minimal de l'utilisateur. La Figure 4 montre différents types de Modules; ces Modules permettent d'écrire :
  - du code de calcul (Module de classe Algorithmique),
  - des interfaces avec les ressources physiques tel que robot ou capteur (Module de classe Ressource Physique),
  - 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 évènements 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 Figure 5 est un exemple de TR telle qu'on peut la trouver dans Orccad.
- la Procédure-Robot (PrR) qui décrit en Estérel (ou MaestRO) la composition de TRs. Cette composition permet de spécifier l'application que doit réaliser le robot.

La ressource physique assure le passage entre la vue Orccad et le Cycab proprement dit par connexion de ses différents ports qui appellent les fonctions pilotant les moteurs ou les capteurs. 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. La Figure 5 représente une TR Orccad telle qu'elle apparaît dans le logiciel. Cette figure représente les différents types de modules qui interviennent dans une loi de commande, avec par exemple:

- Cycab : Module de Ressource Physique
- Security : Module Algorithmique
- trAtr : Module Automate

Les Modules communiquent entre eux par des liens qui permettent de transmettre les données et les variables d'un Module à l'autre.

FIG. 4: *exemple de modules Orccad*

FIG. 5: *exemple de TR*

## 7 Loi de commande théorique

La loi de commande en accélération étudiée ici doit permettre à un véhicule autonome de circuler sur une autoroute sans provoquer d'accident. Cette loi optimise par ailleurs les conditions de circulation en garantissant une sécurité et un espacement minimal entre les véhicules. La philosophie choisie pour la mise en place de cette loi est la conservation d'un temps qui doit rester constant entre deux véhicules successifs. La conséquence est donc le maintien d'un espacement proportionnel à la vitesse des véhicules.

### 7.1 Convention et noms de variables

Dans toute la suite, le véhicule de devant sera appelé le leader, le second sera appelé le suiveur. On notera :

- pour le leader :
  - $z(t)$  la position
  - $\dot{z}(t)$  la vitesse
  - $\ddot{z}(t)$  l'accélération
- pour le suiveur :
  - $x(t)$  la position
  - $\dot{x}(t)$  la vitesse
  - $\ddot{x}(t)$  l'accélération

- la constante de temps  $h$ .

Dans un double souci d'optimisation, c'est-à-dire de décongestion d'autoroute et de sécurité, les différentes simulations effectuées ont montré que, dans les limites de technologie actuelles (limites physiques des capteurs et actionneurs), une valeur de  $h=0.6$  s semblait donner les résultats optimaux.

### 7.2 Conditions de validité et conditions initiales

- la fonction  $\ddot{z}(t)$  est au moins continue par morceaux
- le leader doit inévitablement rester devant le suiveur :

$$x(t) \leq z(t) \tag{1}$$

- les essais se font uniquement en marche avant et la vitesse du leader est limitée :

$$0 < \dot{x}(t) \quad (2)$$

$$0 < \dot{z}(t) < V_{max} \quad (3)$$

Note : la vitesse maximale théorique pour le cycab est de 20 km/h soit environ 5,5 m/s.

- les limites physiques impliquent que l'accélération du leader est bornée :

$$A_{min} = -0.5g \leq \ddot{z}(t) \leq A_{max} = 0.2g \quad (4)$$

### 7.3 Loi de commande

Un ratio peut être défini à partir de cette exigence de conservation d'un temps constant. Ce ratio, appelé  $r$ , va permettre de prendre en compte la position du suiveur par rapport au leader et va induire la correction appropriée selon le cas.

$$r(t) = \frac{z(t) - x(t)}{h\dot{x}(t)}$$

Ce ratio est un nombre non dimensionné, qui s'exprime donc sans unité. Il représente en effet le rapport de la distance qui sépare les deux véhicules sur un temps multiplié par la vitesse du suiveur. Une valeur supérieure à 1 pour  $r$  signifie que le suiveur est trop loin du leader, et qu'il doit donc accélérer; inversement, une valeur inférieure à 1 signifie que le suiveur est trop proche, et qu'il doit par conséquent freiner. Enfin, si  $r$  vaut 1, le véhicule suiveur est à la bonne distance du leader.

À partir de ce ratio, l'accélération va être donnée par la relation :

$$\ddot{x}(t) = \frac{\dot{z}(t) - \dot{x}(t)}{h} + \lambda(r(t) - 1) \quad (5)$$

où  $\lambda$  est un gain de dimension  $ms^{-2}$ .

L'équation différentielle obtenue (5) régissant l'accélération du suiveur n'est pas intégrable. Pour étudier la stabilité de notre système, une étude des variations de  $r(t)$  a été réalisée. Ceci a permis de valider la loi, mais sous certaines conditions uniquement, ce qui a donc entraîné la mise en place de conditions initiales supplémentaires.

- $\ddot{x}(t) > -\lambda$ . En effet, sous cette condition seulement, on a  $\lim_{t \rightarrow \infty} r(t) = 1$ . Le système est donc asymptotiquement stable lorsque la condition

$\ddot{x}(t) > -\lambda$  est vraie. La conséquence primordiale de ce fait est que la collision entre les deux véhicules est théoriquement impossible, le système tendant toujours vers 1 au final de l'évolution.

La Figure 6 montre la courbe représentative du ratio théorique au cours du déplacement simulé pour différentes valeurs du gain  $\lambda$ . On constate effectivement la convergence du ratio vers la valeur 1, quelles que soient les conditions d'essais (freinage ou accélération).

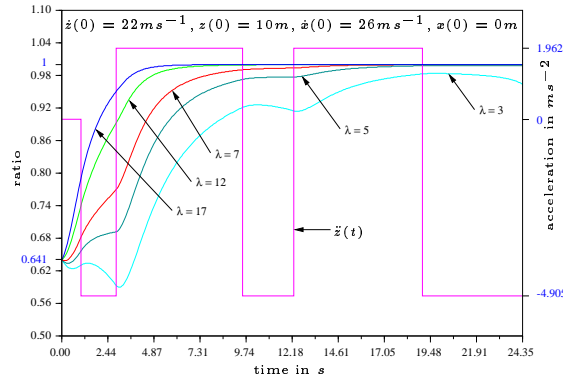


FIG. 6: Tracé du ratio théorique

## 8 Déroutement du stage

Les deux premières semaines du stage ont été consacrées en majeure partie à la lecture de documentation, à l'étude de bibliographie (cf Bibliographie) et à l'apprentissage du logiciel Orccad.

Ces deux semaines ont également été mises à profit pour prendre en main le Cycab et en maîtriser l'utilisation tant au niveau de la conduite qu'au niveau logiciel. La prise en main du logiciel Orccad s'est faite relativement rapidement à partir d'un tutorial disponible sur le site internet de l'INRIA (<http://www.inrialpes.fr/iramr/local/Orccad/>). Ce tutorial, spécialement développé pour un apprentissage rapide du logiciel Orccad, s'est révélé efficace et m'a permis de maîtriser, bien qu'imparfaitement les premiers temps, le logiciel au bout de quelques jours d'utilisation. Il permet en effet un auto-apprentissage efficace et rapide, à partir de l'exemple d'asservissement d'un moteur.

Une fois le logiciel pris en main, le travail sur la loi de commande proprement dite a pu commencer. Ce travail a débuté par l'étude et la compréhension du code existant. Un test avec la loi de commande existante a été effectué

avec le Cycab afin de mesurer son action et pouvoir faire la comparaison avec celle qui sera implémentée lors du stage.

Il a ensuite fallu mettre en place la nouvelle structure choisie pour le contrôle désiré en modifiant la TR existante, puis coder les différents modules afin d'implémenter la loi. Ceci a été réalisé à partir de la loi de commande existante. Une semaine environ a été nécessaire pour réaliser cette étape. Après cette semaine, les premiers tests réels ont été réalisés. Des modifications et tests ont alors été réalisés durant une semaine afin de relever les premiers résultats d'une part, et d'apporter les modifications et réglages nécessaires d'autre part. La base de la loi de commande était alors implémentée. Dans toute la suite du stage, de nombreux essais ont été effectués et de nombreuses modifications et améliorations apportées, avec des modifications de la TR, des mises en place de filtrage, etc... Toute cette phase de mise au point s'est faite parallèlement à la rédaction du rapport.

## 9 Travail réalisé

### 9.1 Structure existante

Une Tâche Robot, appelée CycabFollowingSick, existe et assure un suivi longitudinal et latéral correct. Cette TR est basée sur la détection du véhicule leader par le SICK. La loi de commande jusqu'alors implémentée est rudimentaire mais fonctionne. La vitesse injectée au Cycab est proportionnelle à la distance séparant les deux véhicules détectée par le laser. Cette loi de commande est modélisée par la Figure 7.

Un seul Module Algorithmique est nécessaire pour cette loi, la vitesse calculée étant injectée directement aux moteurs du Cycab.

C'est à partir de cette structure existante que la nouvelle loi a été mise en place. Les modules de Ressource Physique n'ont pas été modifiés. En revanche, des Modules Algorithmiques existants ont été modifiés tandis que d'autres ont été ajoutés. Le détail du rôle des modules qui ont été repris pour la mise en place de la nouvelle loi est nécessaire à la bonne compréhension du projet.

- La Ressource Physique **Cycab**. La Ressource Physique **Cycab** assure le passage entre Orccad et le Cycab proprement dit. La Ressource Physique **Cycab** correspond, d'un point de vue spécification, au robot à commander. On constate que le Cycab ne peut être commandé qu'en vitesse angulaire sur les roues et en direction. Ces limites du matériel, à savoir l'impossibilité de commander la machine en accélération, ont posé des problèmes lors de la réalisation de la nouvelle loi de commande,

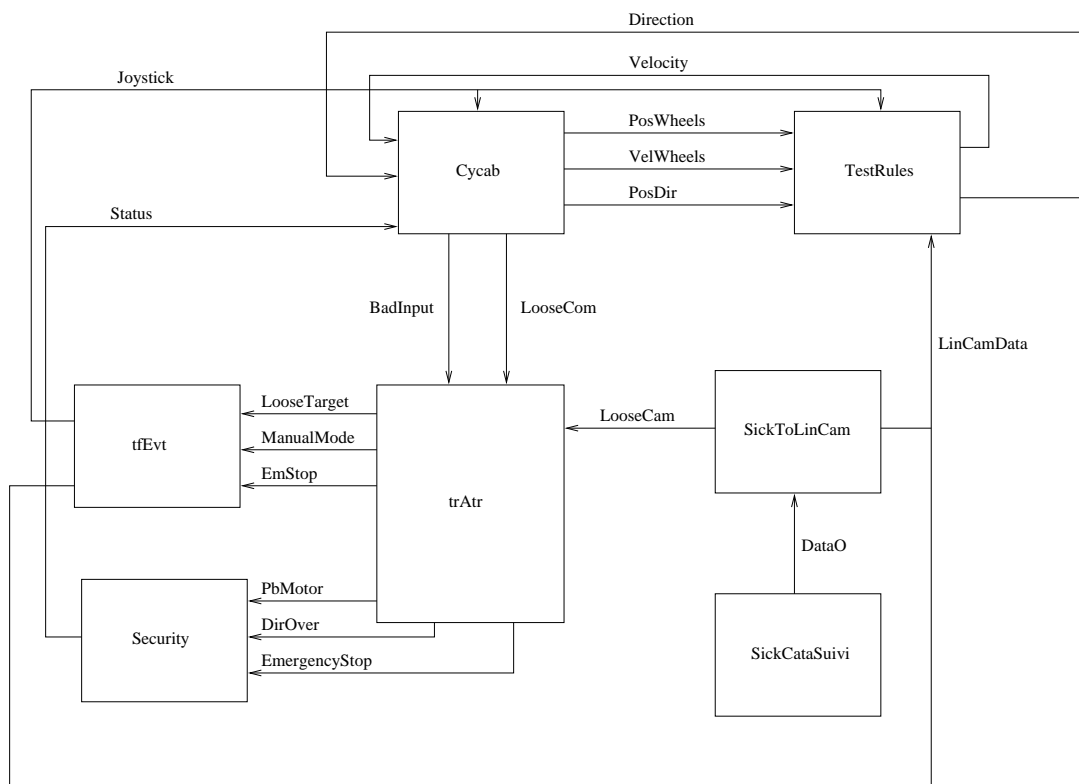


FIG. 7: TR de suivi excitante: *CycabFollowingSick*



car cette exigence de fournir une consigne de vitesse va nécessiter une intégration de l'accélération à injecter. Ces problèmes seront détaillés dans le paragraphe suivant. La Ressource Physique **Cycab** 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.

Les divers ports d'entrée/sortie, ainsi que les fonctions associées à chaque Module sont consultables en annexes.

- La Ressource Physique **SickCataSuivi**. Ce Module représente le laser télémétrique, et donne en sortie la distance qui sépare le laser de la cible.
- Ces données passent ensuite dans le Module **SickToLinCam**. Ce Module a pour rôle de transformer les données du laser. Historiquement, c'est une caméra linéaire qui permettait de mesurer la distance entre les véhicules. Cette caméra a été remplacée par le laser, qui produit donc des données sous une forme différente de celle de la caméra. Ce Module sert donc à mettre les données dans le format de la caméra afin qu'elles puissent être traitées sans modifier les autres Modules Algorithmiques.
- Le Module **Security** permet de gérer les situations liées à un problème entraînant un arrêt d'urgence.
- Le Module **tfEvt** permet de passer du mode automatique au mode de conduite manuelle avec le Joystick.
- Le Module automate **trAtr** permet de sélectionner le Module approprié **Security** ou **trAtr** en cas de détection de défaut.

## 9.2 Modification de la structure existante

D'autres Modules ont été créés pour implémenter la nouvelle loi de commande.

- Le Module **Dérivation** doit permettre de traiter les informations fournies par les différents capteurs du Cycab. Il doit en particulier permettre de traiter les informations fournies par le laser. Les données concernant la distance séparant les deux véhicules sont essentielles pour la réalisation d'une bonne commande du véhicule. Les opérations effectuées dans

ce Module permettent d'avoir en sortie les trois grandeurs suivantes :

- La vitesse **SpeedFollow**, vitesse du véhicule suiveur, obtenue à partir de la donnée de la vitesse radiale relevée sur chaque roue du véhicule. A ce titre, les quatre roues étant indépendantes et possédant chacune un capteur propre, les quatre vitesses obtenues ne sont pas rigoureusement identiques. C'est la moyenne des quatre vitesses qui va être exploitée, et qui va servir de valeur de base dans le reste des calculs. De même, la donnée concernant la distance parcourue pour chaque roue n'est pas la même sur les quatre roues. Ceci vient du fait de la précision des codeurs incrémentaux optiques utilisés en tant que capteurs sur chaque roue. Des différences de pression de gonflage entre les quatre roues, ou la répartition non homogène du poids, surtout lorsqu'il y a un conducteur, peuvent expliquer les variations observées bien que les roues aient sensiblement parcouru la même distance. En revanche, la consigne de vitesse retournée au véhicule sera identique sur les quatre roues.
- La distance entre le véhicule leader et le véhicule suiveur. Cette distance est appelée **DiffDistance**, et est retournée par le laser. On peut l'exprimer comme une différence de position entre les deux véhicules. On suppose donc que  $\text{DiffDistance} = z(t) - x(t)$ , mais on ne connaît pas les positions absolues  $z(t)$  ni  $x(t)$ .
- La différence des vitesses des deux véhicules. La vitesse du véhicule leader n'étant pas connue, la différence des vitesses peut être obtenue en dérivant la différence de positions déterminée précédemment. Cette différence **DiffSpeed** va ensuite être injectée dans le Module suivant, appelé Module **AccelRule**, dans lequel est implémentée la loi de commande, qui nécessite cette grandeur. Cette différence de vitesse a été obtenue de la façon suivante:

$$\begin{aligned}
 \text{DiffSpeed} &= \dot{z}(t) - \dot{x}(t) \\
 &= \lim_{h \rightarrow 0} \frac{z(t+h) - z(t)}{h} - \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h} \\
 &= \lim_{h \rightarrow 0} \frac{\text{DiffDistance}(t+h) - \text{DiffDistance}(t)}{h}
 \end{aligned}$$

- Le Module **AccelRule** contient la loi de commande proprement dite. Ce Module s'articule donc autour du calcul de l'accélération à fournir en fonction des divers paramètres pris en compte. La valeur de  $r$  est calculée dans ce Module à partir des informations données par le

Module précédent. Les différents essais effectués confirment des valeurs cohérentes pour  $r$ . Plus précisément, les valeurs relevées tendent effectivement vers 1. Selon la valeur de  $r$ , le véhicule devra donc réduire sa vitesse ou l'augmenter. Pour ce faire, on va déterminer une valeur de l'accélération à injecter dans le système.

- Le Module **Integrator100ms** doit réaliser une intégration de l'accélération calculée précédemment. En effet, on rappelle que les moteurs du Cycab ne peuvent qu'accepter une consigne de vitesse. Une intégration est donc absolument nécessaire. Cette étape est réalisée de la manière suivante : connaissant la vitesse actuelle du suiveur ainsi que la valeur de l'accélération à injecter, on détermine la valeur suivante de la vitesse en effectuant un développement limité à l'ordre 1 de la fonction représentant l'évolution de la vitesse en fonction du temps.

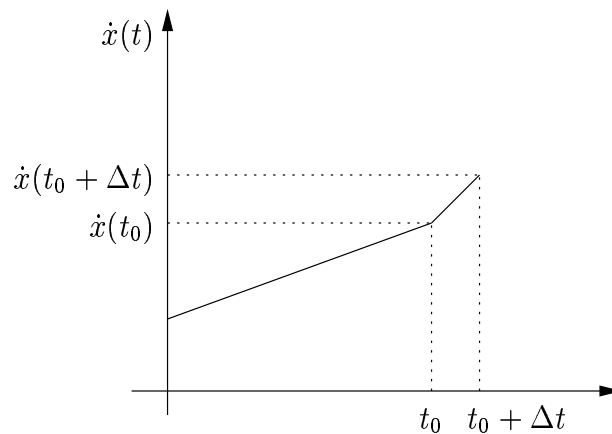


FIG. 8: *Tracé de la vitesse en fonction du temps*

On connaît en effet  $\dot{x}(t)$  la vitesse du suiveur obtenue à la sortie de la Ressource Physique **Cycab**, l'accélération calculée à partir de la valeur du ratio qui est déterminée dans le Module Algorithmique **AccelRule**, ainsi que le temps  $\Delta t$  qui correspond à la période de l'horloge du Module d'intégration. Il faut d'ailleurs bien noter l'importance de ce terme d'intégration  $\Delta t$ . L'intégration sera toujours réalisée sur une période  $\Delta t$  qui dépendra de l'horloge du Module. Toute modification de la valeur de cette horloge devra être prise en compte dans l'implémentation de la phase d'intégration qui prend en compte ce temps. Dans toute la TR, il a été choisi de prendre une période de 100 ms pour chacun des Modules. Le nom du Module, contenant la valeur de l'intervalle d'intégration, rappelle la valeur de ce dernier. A partir de ces données, il

est possible d'effectuer une intégration de façon à obtenir la valeur de la vitesse à injecter aux moteurs. Cette valeur,  $\dot{x}(t_0 + \Delta t)$  peut être approximée par un développement limité à l'ordre 1.

$$\dot{x}(t_0 + \Delta t) = \dot{x}(t_0) + \ddot{x} \times \Delta t \quad (6)$$

La valeur de  $\dot{x}(t_0 + \Delta t)$  une fois déterminée sera injectée sur chaque roue du Cycab, après avoir été convertie en vitesse radiale.

De plus, ce Module permet de limiter la vitesse du suiveur. En effet, un problème est apparu lors des premiers essais alors que le véhicule n'était pas bridé. Le Cycab ne peut accepter une consigne de vitesse trop grande sur les roues. Ainsi, il faut limiter la vitesse d'entrée à moins de 20 rad/s. Dans le cas présent, la vitesse maximum autorisée est de 15 rad/s, ce qui correspond à une vitesse linéaire du Cycab de l'ordre de 3 m/s. Cette vitesse paraît un compromis acceptable à plusieurs titres. D'une part, les essais se font à une vitesse qui n'est pas toute faible, ce qui permet de tester des conditions réelles de circulation, à faible allure certes. D'autre part la vitesse est encore raisonnable en cas de défaillance. La sécurité humaine et matérielle, le Cycab étant doté de matériels onéreux, ont donc été un autre facteur de ce choix.

## 10 Problèmes rencontrés

### 10.1 Problèmes logiciels

Les premiers essais ne se sont pas révélés concluants, le véhicule automatique ne fonctionnant pas avec la loi de commande mise en place. C'est en fait un défaut du logiciel qui empêchait une bonne communication entre les Modules. Chaque Module doit en fait générer une variable correspondant à chaque port d'entrée/sortie. Cette étape n'ayant pas été correctement réalisée par le logiciel, toute communication entre Modules était impossible. Il a donc fallu régler le problème en écrivant directement le nom des variables dans le code.

### 10.2 Dérive mécanique

Lors des premiers essais effectifs, un problème gênant est apparu rapidement. En effet, une forte dérive mécanique empêche ce véhicule de garder une trajectoire linéaire. Il dévie fortement sur la droite, et une correction

logicielle se révèle nécessaire pour éliminer ce défaut. Lorsque la donnée de l'angle de direction des roues relevé dévie de la valeur limite autorisée, une consigne est injectée au système de façon à ce que le véhicule roule droit. Cet asservissement corrige le défaut de trajectoire inhérent au Cycab.

## 11 Les tests réalisés

### 11.1 Mode opératoire

Lorsque les conditions météorologiques le permettaient, les essais étaient réalisés sur le parking extérieur de l'INRIA. Les essais prenaient alors la forme suivante : un conducteur prenait place dans le Cycab leader, tandis qu'un second lançait le programme depuis le Cycab suiveur, avec la charge d'arrêter ce véhicule en cas de problème par le biais du bouton d'arrêt d'urgence. Le test était lancé avec départ arrêté pour les deux véhicules. Le leader imprimait alors un rythme, avec différentes phases d'accélération et de freinage. Des tests avec véhicule sur cales ont également dû être mis en place. D'une part, les mauvaises conditions climatiques persistantes et les besoins des autres stagiaires d'autre part expliquent cette solution provisoire. Le véhicule suiveur se trouvait donc sur cales, le leader étant quant à lui libre. La vitesse du suiveur étant fonction de la distance qui le sépare du leader, il suffisait alors d'avancer ou reculer le leader afin de provoquer des phases d'accélération et de freinage. La simulation des essais extérieurs était alors reproduite. En revanche, les paramètres liés à la liaison au sol et à l'inertie ne pouvaient plus être pris en compte. Toutefois, cette solution proche des essais réels a permis d'effectuer les tests nécessaires aux moments voulus.

### 11.2 Comparaison des résultats théoriques et expérimentaux

Les premiers essais ont mis en évidence des problèmes au lancement de la loi de commande. Les essais de simulation réalisés avec la loi de commande théorique ont été réalisés dans des conditions particulières, à savoir sur des véhicules ayant une vitesse initiale non nulle.

Lors des premiers essais réels, on a constaté une forte accélération au départ, au moment du déclenchement de la loi de commande. Le véhicule suiveur démarre donc très rapidement, la consigne d'accélération au départ étant très élevée. Pour palier ce problème, mais également pour se mettre facilement dans les conditions des simulations réalisées antérieurement, une modification

de la structure jusqu'alors mise en place a été décidée. Pour ce faire, plusieurs solutions ont été envisagées.

La première solution envisagée consistait à créer un Module ayant un double rôle :

- démarrer le cycab puis le faire atteindre une vitesse déterminée.
- enclencher la loi de commande lorsque les conditions initiales désirées sont atteintes.

Une deuxième solution était de créer une procédure robot (PR), composée de deux tâches robots (TR) distinctes. La philosophie de cette solution serait alors la suivante: la première TR, ou tâche **Accelerate** est chargée de démarrer et atteindre une vitesse désirée. Une fois cette vitesse atteinte, une exception de type 2 est lancée par le Module **Automate** afin que la PR commute et enclenche la seconde TR **Follow**, qui assure le suivi longitudinal.

Cette solution est sans nulle doute la plus lourde et la plus compliquée à réaliser. En effet, l'exception doit être lancée sur un évènement, évènement qui doit correspondre en fait à l'arrivée du suiveur à une distance prédéfinie du leader, et ce à une vitesse elle aussi fixée. Par ailleurs, le Cycab suiveur devant atteindre une certaine vitesse et la garder constante pendant un certain temps, il faut qu'il se trouve au départ suffisamment loin du Cycab leader pour ne pas entraîner la procédure d'arrêt d'urgence. En revanche, il doit être suffisamment proche tout de même pour que le laser télémétrique atteigne la cible située à l'arrière du leader, et puisse donc retourner la distance mesurée. C'est donc la première solution qui a été retenue.

### 11.3 Mise au point

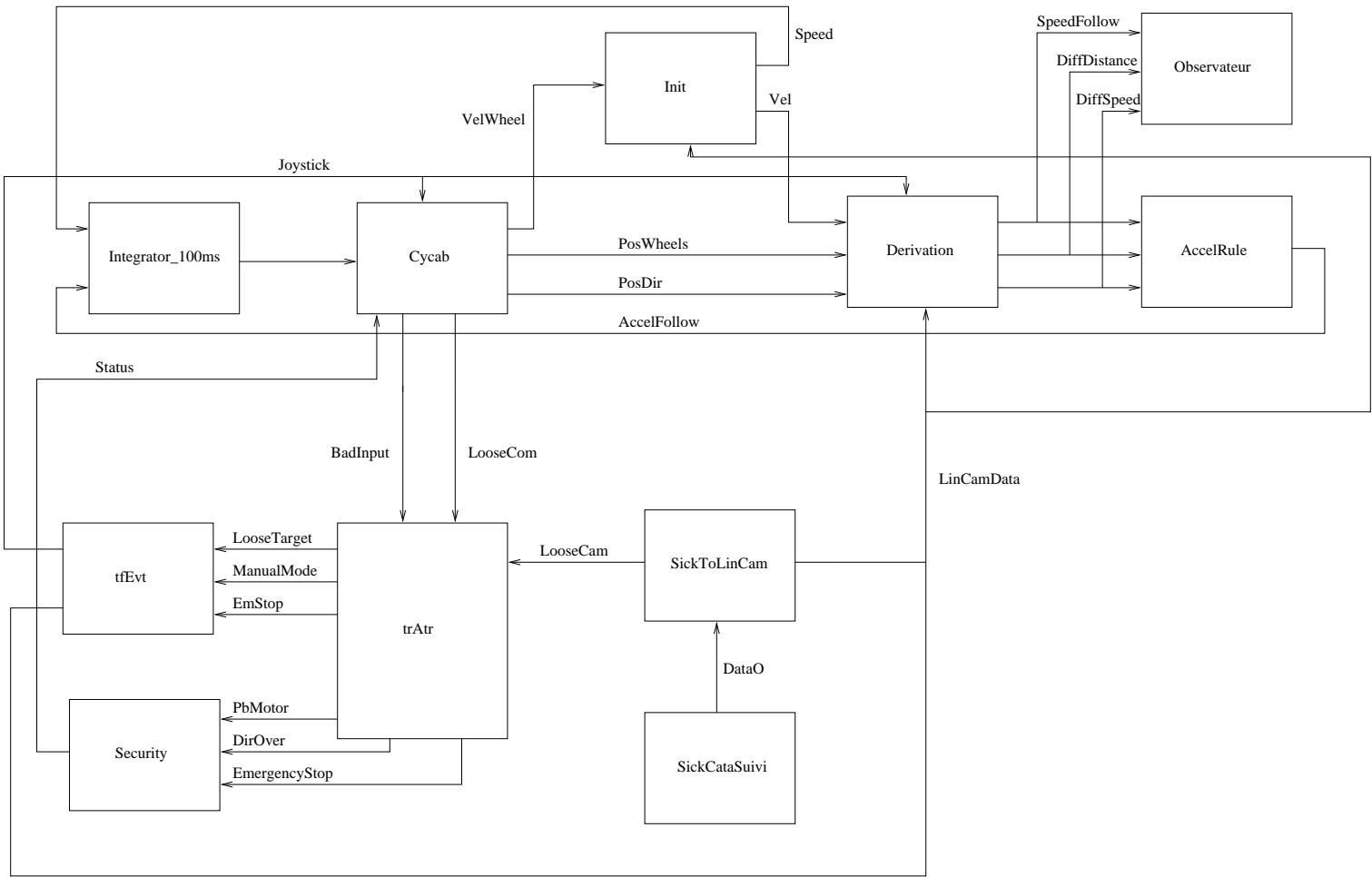
Un Module nommé **Init** est donc chargé de mener le véhicule aux conditions initiales désirées puis d'enclencher la loi de commande initialement mise en place. La loi de commande proprement dite n'est active qu'à partir du moment où la vitesse du véhicule désirée sera atteinte. En dessous des conditions initiales désirées, le Module **Init** reboucle directement sur le Module de Ressource Physique **Cycab** en lui injectant une rampe en vitesse.

La nouvelle structure se présente alors sous la forme de la Figure 9.

Les essais alors réalisés ont donné des résultats satisfaisants en ce qui concerne de nombreux points :

- Tout d'abord, le Module **Init** améliore sensiblement les conditions de démarrage. La montée en vitesse est beaucoup moins violente, ce qui améliore bien sûr le confort des passagers. Par ailleurs, il permet d'atteindre une vitesse déterminée à partir de laquelle la loi de commande

FIG. 9: TR de suivi définitive



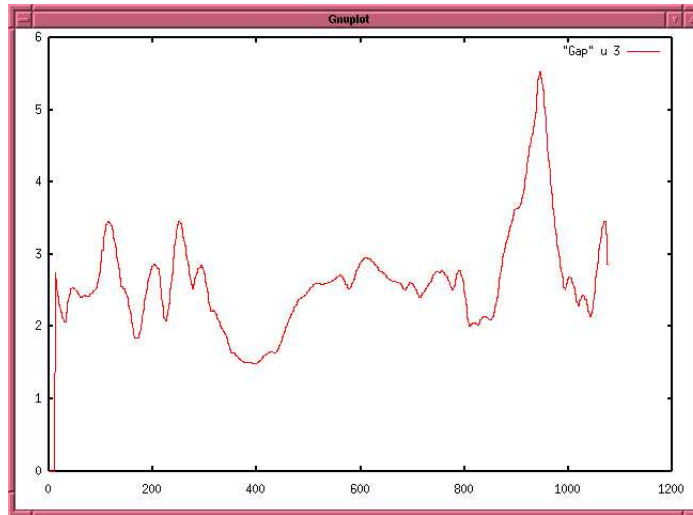


FIG. 10: *Données du laser : distance en m entre les deux véhicules en fonction du temps en  $10^{-1}s$*

initiale va entrer en action. Les conditions initiales peuvent donc maintenant être fixées à des valeurs désirées.

- Les prévisions de suivi sont, elles aussi, bien réalisées. On constate en effet les caractéristiques attendues par la loi de commande mise en place. Lorsque la vitesse des deux véhicules est élevée, de l'ordre de 3 m/s, la distance entre les deux véhicules est grande. A vitesse plus faible, la distance de suivi est plus petite. De façon plus précise, la Figure 10 représentant la différence de distance entre les véhicules permet de visualiser les fluctuations de ce gap.

Lors des essais, on constate donc cette évolution du gap, qui se trouve donc être d'autant plus grande que la vitesse est élevée. On constate également un problème d'inertie. La réaction du suiveur n'est pas instantanée lors d'un changement de vitesse assez brusque du véhicule leader. Ce problème d'inertie est particulièrement gênant. Lors des tests avec véhicule sur cales, les différentes phases d'accélération sont bien reconstituées. L'évolution de la vitesse des roues dépend bien de la distance qui sépare les deux véhicules. Les valeurs des variables relevées montrent d'ailleurs la bonne exécution de la loi de commande. En revanche, on verra par la suite que d'autres problèmes se posent lors des essais sur route. Le Cycab entraîné par sa vitesse et son inertie n'a pas un comportement aussi fluide que le laissaient supposer les essais sur cales.



L'évolution des différentes variables qui sont mises en jeu lors de la tâche de suivi paraît des plus cohérentes. En effet, on constate que l'évolution de la vitesse du suiveur est bien fonction de la distance qui sépare les deux véhicules. La détection d'un espace plus grand par le laser télémétrique entraîne une augmentation de la vitesse du suiveur. Inversement, lors d'une phase de freinage du leader, l'espace diminue, ce qui a pour conséquence de diminuer la vitesse du suiveur. L'évolution du ratio n'est en revanche pas exactement conforme aux prévisions théoriques.

On imprime un certain rythme au Cycab leader, avec des phases d'accélération, de freinage et de vitesse constante. La Figure 10 retrace ces conditions d'expérience, à partir de la variable DiffDistance. C'est en effet à partir de cette donnée que toute la loi de commande va pouvoir être exécutée. Cette variable étant à la base de la commande, on peut par son intermédiaire visualiser le comportement du Cycab leader.

A partir de cette distance qui sépare les Cycab, le ratio est calculé et va permettre de déterminer la valeur de l'accélération à injecter au suiveur. La Figure 11 permet d'observer l'évolution de ce ratio. On constate alors que, comme le prévoit la théorie, le ratio calculé tend effectivement vers 1. Toutefois, les résultats ne sont pas exactement conformes à ceux attendus. Si la valeur de ce ratio est proche de 1, elle oscille entre 0.85 et 1. On constate par ailleurs que le ratio diverge sous certaines conditions expérimentales. En effet, le véhicule étant bridé à partir d'une certaine vitesse (3 m/s), la vitesse effective du suiveur n'est pas celle qui est calculée. Il est donc inévitable que le ratio ne tende pas vers 1 dans ce cas de figure. Le résultat global est cependant honorable, de nombreux facteurs étant susceptibles d'engendrer des différences entre la loi théorique et la mise en place sur le Cycab.

Ces différents facteurs peuvent être :

- Les opérations internes. Comme on l'a vu, la mise en place de la loi de commande nécessite une dérivation puis une intégration. Ces opérations induisent des erreurs non négligeables qui n'apparaissent pas lors de l'étude théorique. Les petites variations sur les données du laser par exemple sont après dérivation susceptibles d'entraîner de fortes variations sur la dérivée.
- Les données à disposition. Un traitement des variables a déjà apporté une amélioration des résultats. Un lissage des résultats qui interviennent dans le calcul du ratio a été envisagé à partir d'un filtrage numérique des données. Les premiers résultats observés

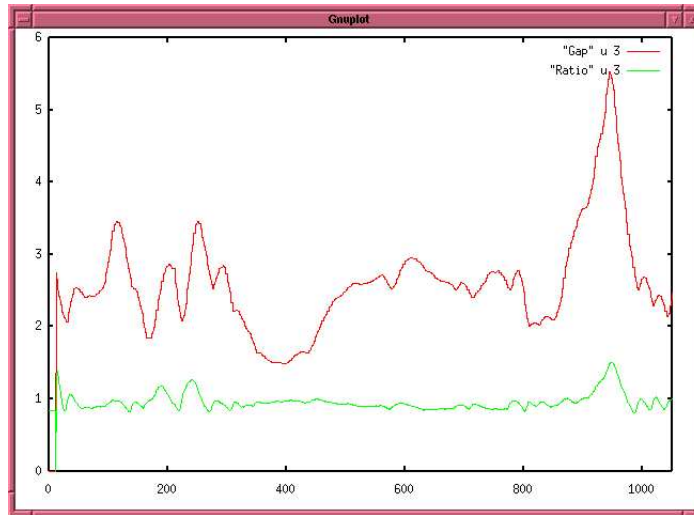


FIG. 11: Evolution du gap en m et du ratio en fonction du temps en  $10^{-1}s$

ont été amélioré de façon non négligeable par ce traitement. Les valeurs de ratio oscillaient en effet initialement entre 0.7 et 1.2. Les détails sur l'opération de filtrage seront détaillés dans le paragraphe suivant.

- Les non-linéarités. Dans un système complexe et réel comme l'est le Cycab, il existe de nombreux paramètres qui peuvent induire des non-linéarités qui peuvent bien sûr être préjudiciables au bon fonctionnement de la loi de commande, et entraîner des différences par rapport à l'étude théorique. Les liaisons au sol, les actionneurs, les capteurs sont autant de paramètres qui peuvent induire des non-linéarités.

Enfin, la Figure 12 permet de rendre compte de la proportionnalité qui existe entre la vitesse du Cycab suiveur et le gap qui sépare les deux Cycab. Comme on le vérifie, la distance de suivi est fonction de la vitesse du Cycab leader. Lorsque celui-ci roule doucement, la distance de suivi, représentée par le gap, est faible, et inversement, lorsque le leader accélère, la distance de suivi s'accroît.

- Filtrage. Les premiers tests ont donc révélé certains problèmes, malgré un bon fonctionnement global. Un filtre numérique a donc été mis en place dans le Module **AccelRule** de façon à améliorer les résultats. En effet, la sécurité, et plus précisément la non-collision entre les Cycab est d'autant mieux assurée que le ratio est proche de 1. Un filtre passe-bas

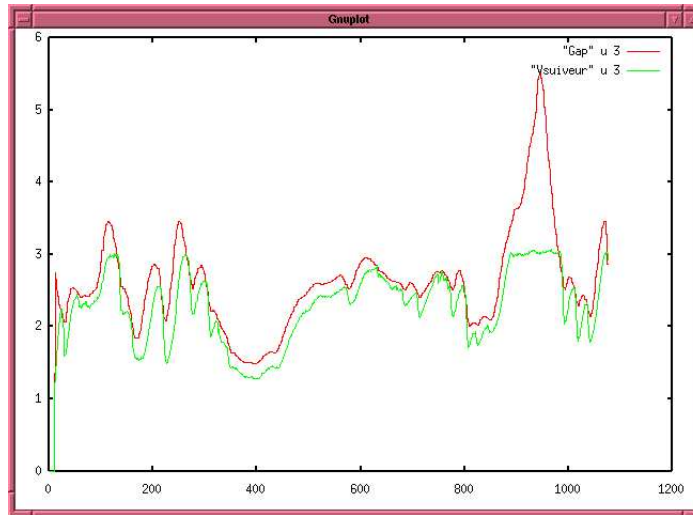


FIG. 12: Evolution du gap en m et de la vitesse du suiveur en m/s en fonction du temps en  $10^{-1}s$

a été implémenté. Ce filtre IIR (Réponse Impulsionnelle Infinie) d'ordre 1 est de la forme :

$$Y_n = \alpha y_n + (1 - \alpha)y_{n-1} \quad (7)$$

avec  $\alpha = 0.3$ .

$Y_n$  est donc la valeur calculée à partir des deux dernières valeurs mesurées que sont  $y_n$  et  $y_{n-1}$ . Ce type de filtre permet donc de filtrer les données suivantes:

- DiffDistance
- SpeedFollow
- r

Les deux premières variables interviennent dans le calcul du ratio, tandis que r est également filtré avant d'être injecté dans la relation qui permet de calculer l'accélération.

Cette valeur de 0.3 a définitivement été retenue après plusieurs essais. C'est en effet avec cette valeur que le meilleur compromis entre un comportement fluide et des courbes proches des courbes théoriques est le meilleur.

On constate donc une diminution sensible du bruit sur les données grâce à cette étape de filtrage.

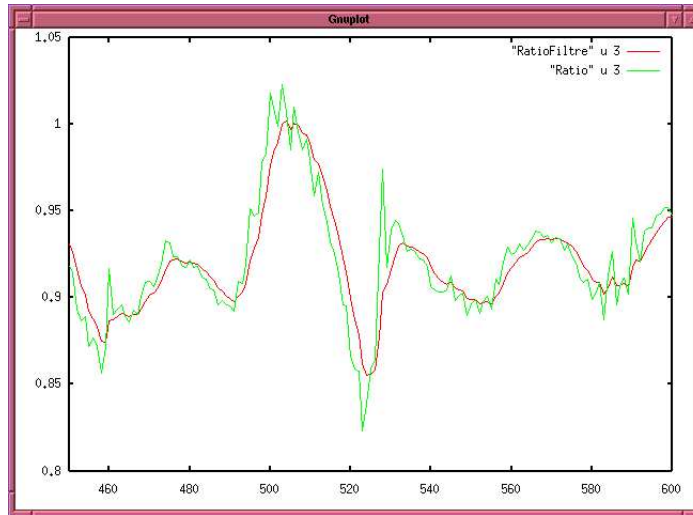


FIG. 13: Zoom de la superposition du ratio filtré et non filtré

De nouveaux essais ont donc pu être réalisés avec filtrage. On observe sur la figure 13 l'effet du filtrage qui lisse les données en diminuant le bruit et qui améliore par conséquent la sécurité.

Le Module **Observateur** trouve ici son utilité. Ce Module a en fait été créé suite à un défaut d'initialisation du filtre mis en place. Le filtre étant récursif, les résultats obtenus étaient dès lors incohérents. Dans ce type de systèmes embarqués, un observateur peut être une méthode de détection d'erreur efficace. Ce qui a été le cas ici, l'observation des données filtrées et non filtrées ayant permis de trouver rapidement la cause du problème. Ce Module n'intervient donc pas dans la loi de commande proprement dite.

- Période d'échantillonnage. Un autre facteur possible d'amélioration réside dans le choix de la période d'échantillonnage. Initialement, la TR existante avait une période d'échantillonnage de 100 ms. Ces tests ont alors été réalisés avec la nouvelle loi de commande afin de voir s'il était possible de diminuer cette période d'échantillonnage, sans provoquer de défaillances ou de dysfonctionnement. Plusieurs essais ont donc été effectués en diminuant la valeur de la période d'échantillonnage jusqu'à 20 ms. Les résultats sont sensiblement identiques. Il est évident que deux essais similaires ne peuvent être réalisés. Cependant, dans des conditions sensiblement identiques, aucune amélioration, tant du point de vue confort de suivi qu'analyse des résultats n'a été observée. Une hypothèse a alors été avancée afin d'expliquer ce phénomène. Cette

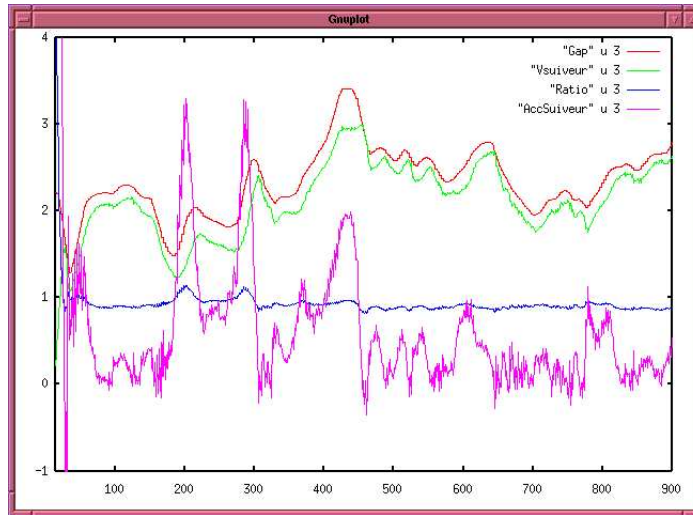


FIG. 14: *tracé des différentes variables en fonction du temps*

situation serait en fait certainement liée au matériel, et plus particulièrement au laser SICK. En effet, la période minimale de rafraîchissement des données du laser est de 300 ms. La TR étant échantillonnée à 100 ms, c'est donc 3 fois la même mesure de distance qui est injectée dans la loi de commande. Or, le véhicule s'est déplacé d'une distance non négligeable pendant ce laps de temps. Par exemple, à 3 m/s :

$$d = t * vd = 0.3 * 3 = 0.9 \text{ m} \quad (8)$$

Si dans le même temps le véhicule leader a diminué sa vitesse de façon significative, la nouvelle valeur donnée par le laser provoque un freinage brusque.

La Figure 14 représente un autre essai réalisé dans des conditions globalement identiques. Cette figure permet de superposer sur un même graphe les variables les plus importantes pour l'exécution de la loi. L'accélération calculée a également été rajoutée sur cette figure.

## 12 Conclusion

Ce stage a été particulièrement intéressant, et ce à de nombreux titres. Le travail nécessaire pour mener à bien le projet a été complet et diversifié, de nombreux concepts et domaines techniques ont été mis en jeu et abordés. L'étude de la loi de commande s'est révélée très intéressante, le codage des modules de la TR également. Mais c'est surtout au niveau de la mise au point et de la possibilité de faire des tests grandeur nature que cette expérience a été enrichissante. La résolution des différents problèmes robotiques qui se sont posés m'a été très profitable, et le concret du projet m'a beaucoup plu. Par ailleurs, les résultats obtenus sont bons, comme l'attestent les différents essais effectués et filmés. Le suivi aussi bien latéral que longitudinal de Cycab est en effet bien réalisé. En effet, le suivi latéral a lui aussi été mis en place. C'est en fait pour faciliter les conditions d'essais et pour faire le tour du parking que ce suivi latéral a été réalisé. L'arrêt d'urgence fonctionne également en cas de défaillance. La loi de commande mise en place est fiable et aucune collision n'a été observée.

L'INRIA, lors de la venue de visiteurs, fait maintenant des démonstrations de ce suivi de véhicules à partir de la procédure mise en place lors de ce stage. De plus, le stage s'est déroulé dans une ambiance sérieuse et conviviale que j'ai également beaucoup appréciée. La disponibilité et l'écoute de tout le personnel de l'INRIA, aussi bien en cas de problème technique qu'administratif est particulièrement agréable.

Enfin, la rédaction du rapport m'a permis de découvrir le mode d'édition latex et de développer mes connaissances de l'environnement UNIX.

## Références

- [BCME<sup>+</sup>98] J.-J. Borrelly, E. Coste-Manière, B. Espiau, K. Kapellos, R. Pissard-Gibollet, D. Simon, and N. Turro. The Orccad architecture. *International Journal of Robotics Research*, 17(4):338–359, avril 1998.
- [BGMPG99] G. Baille, P. Garnier, H. Mathieu, and R. Pissard-Gibollet. Le Cycab de l'INRIA Rhône-Alpes. Rapport technique 0229, INRIA, Rocquencourt, France, avril 1999.
- [GY99] A. Girault and S. Yovine. Stability analysis of a longitudinal control law for autonomous vehicles. In *IEEE Conference on Decision and Control, CDC'99*, Phoenix, AZ, décembre 1999. IEEE. Rapport de recherche INRIA 3498.
- [KPG] K. Kapellos and R. Pissard-Gibolet. *Open Robot Controller Computer Aided Design*. Disponible à <http://www.inrialpes.fr/iramr/pub/Orccad>.
- [SEKPG97] D. Simon, B. Espiau, K. Kapellos, and R. Pissard-Gibollet. Orccad: Software engineering for real-time robotics: A technical insight. *Robotica*, 15(1):111–115, janvier-février 1997.
- [SKE97] D. Simon, K. Kapellos, and B. Espiau. Control laws, tasks, and procedures with Orccad. *International Journal of Systems Science*, 17(10):1081–1098, octobre 1997.

## A Annexes

### A.1 Module SickCataSuivi

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  $q$  entre les axes longitudinaux de chaque véhicule (en dixièmes de degrés) ;
- DataO[2] : Position angulaire  $a$  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.



## A.2 Module Cycab

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

### A.3 Module Dérivation

Ports		Variables			Fonction appelée	
Nom	Type	Nom	Type	Dimension	Nom	Description
PosWheel	Entrée	PosWheelI	Double	Scalaire	GetPosWheels	Lecture de la position des roues
VelWheel	Entrée	VelWheelI	Double	Scalaire	GetVelWheels	Lecture de la vitesse de rotation des roues
PosDir	Entrée	PosDirI	Double	Scalaire	GetPosDir	Lecture de la position de la direction
Joystick	Entrée	JoystickI				
LinearCamera	Entrée	LinearCameraI				
Direction	Sortie	Direction	Double	Scalaire	PutDir	envoi la consigne de direction au moteur de direction
SpeedFollow	Sortie	SpeedFollowO	Double	Scalaire	PutSpeedFol	retourne la valeur de la vitesse du suiveur
DiffDistance	Sortie	DiffDistanceO	Double	Scalaire	PutDiffDist	retourne la distance entre les deux véhicules
DiffSpeed	Sortie	DiffSpeedO	Double	Scalaire	PutDiffSpeed	retourne la différence de vitesse entre les deux véhicules par dérivation de Diff-Distance

## A.4 Module Init

Ports		Variables			Fonction appelée	
Nom	Type	Nom	Type	Dimension	Nom	Description
LinearCamera	Entrée	LinearCameraI				
VelWheel	Entrée	VelWheelI	Double	Scalaire	GetVelWheels	Lecture de la vitesse de rotation des roues
Speed	Sortie	SpeedO	Double	Scalaire	PutSpeed	Sélection d'une des sorties selon que l'initialisation ait été réalisée ou non
Vel	Sortie	VelO	Double	Scalaire	PutSpeed	Sélection d'une des sorties selon que l'initialisation ait été réalisée ou non

## A.5 Module AccelRule

Ports		Variables			Fonction appelée	
Nom	Type	Nom	Type	Dimension	Nom	Description
SpeedFollow	Entrée	SpeedFollowI	Double	Scalaire	RatioCalculation	calcul du ratio
DiffDistance	Entrée	DiffDistanceI	Double	Scalaire	RatioCalculation	calcul du ratio
DiffSpeed	Entrée	DiffSpeedI	Double	Scalaire	AccelCalculation	calcul de l'accélération
AccelFollow	Sortie	AccelFollowO	Double	Scalaire	PutAcceleration	retourne la valeur de l'accélération calculée
RatioValue	Sortie	RatioValueO	Double	Scalaire	RatioCalculation	retourne la valeur du ratio calculé

## A.6 Module Integrator100ms

Ports		Variables			Fonction appelée	
Nom	Type	Nom	Type	Dimension	Nom	Description
AccelFollow	Entrée	AccelFollowI	Double	Scalaire	Integration	Intégration de l'accélération en vitesse
Speed	Entrée	SpeedI	Double	Scalaire		Initialisation du cycab
AccelFollow2	Entrée	AccelFollow2I	Double	Scalaire	Integration	Intégration de l'accélération en vitesse
Velocity	Sortie	VelocityO	Double	Vecteur(4)	GetVelocity	retourne la valeur de la vitesse à injecter