



**HAL**  
open science

# Modèles multi-agents réactifs pour la navigation multi-véhicules : spécification formelle et vérification

Jean-Michel Contet

► **To cite this version:**

Jean-Michel Contet. Modèles multi-agents réactifs pour la navigation multi-véhicules : spécification formelle et vérification. Informatique [cs]. Université de Technologie de Belfort-Montbéliard, 2009. Français. NNT: . tel-00472415

**HAL Id: tel-00472415**

**<https://theses.hal.science/tel-00472415>**

Submitted on 12 Apr 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE

de l' Université de Technologie de Belfort-Montbéliard

numéro attribué par la bibliothèque 

1	2	3
---	---	---

pour obtenir le grade de

**DOCTEUR**

**DISCIPLINE : INFORMATIQUE**

## MODÈLES MULTI-AGENTS RÉACTIFS POUR LA NAVIGATION MULTI-VÉHICULES : Spécification Formelle et Vérification

par

**Jean-Michel CONTET**

Laboratoire Systèmes et Transport

Université de Technologie de Belfort-Montbéliard

Soutenue le 4 décembre 2009 devant le Jury composé de :

François Charpillet	Rapporteur	Directeur de Recherche au LORIA-INRIA Lorraine
Giovanna Di Marzo Serugendo	Rapporteur	Lecturer au Birkbeck college, Université de Londres
Philippe Lenca	Rapporteur	Habilité à Diriger des Recherches, Telecom Bretagne
Jean-Paul Haton	Examineur	Professeur des Universités Henri Poincaré, Nancy 1
Michel Parent	Examineur	Responsable Scientifique INRIA Paris
Abderrafiaa Koukam	Examineur	Professeur des Universités, UTBM
Pablo Gruer	Directeur de Thèse	Professeur des Universités, UTBM
Franck Gechter	Co-Directeur de Thèse	Maître de Conférences, UTBM



## Remerciements

Je tiens tout d'abord à remercier François Charpillet, Philippe Lenca et Giovanna Di Marzo Serugendo pour m'avoir fait l'honneur de rapporter sur ce travail et pour leurs remarques pertinentes qui ont fait progresser ce document. Je sais combien leur temps est précieux, et je leur exprime toute ma gratitude.

Je souhaite également remercier Jean-Paul Haton et Michel Parent d'avoir accepté de faire partie de mon jury.

Je ne saurais oublier ceux qui ont tant contribué à ces travaux, mes deux codirecteurs de thèse, Pablo Gruer et Franck Gechter. Je tiens à les remercier de leur soutien, leurs conseils précieux, leurs disponibilités et surtout leur encadrement exemplaire. Ces deux personnes ont été pour moi deux pères qui m'ont adopté durant mon stage de master, puis ont su m'encadrer activement durant la première année de ma thèse pour ensuite me laisser de plus en plus de liberté. A ce jour, je peux être comparé à un adolescent qui essaye de trouver sa personnalité en les imitant. Je leurs serait pour cela toujours reconnaissant.

Je tiens à remercier également Monsieur Abderrafiaa Koukam, directeur du SeT, pour sa gentillesse mais aussi sa rigueur. J'ai pu, au cours de ces trois ans, bénéficier de ses qualités scientifiques et humaines. Malgré la lourde charge qu'est la direction du SeT, il a toujours été disponible et a contribué substantiellement à nos travaux. Puissent ces lignes être l'expression de ma grande reconnaissance.

Un grand merci également à mes collègues doctorants Pawel Kmiotek, Sergio Nogueira ainsi que de nombreux jeunes ingénieurs comme Romain, Gillian, Cédric, qui ont été mes compagnons de bureau, pour leur compagnie et leur soutien.

Merci à toutes les personnes qui participe à la conception du véhicule intelligent et propre du laboratoire : David Bouquain, Cindy Cappelle, Cyril Meurie, Abdellatif Miraoui, Yassine Ruichek, Daniel Schlegel et Frederick Zann.

Je ne saurais oublier les personnes qui ont participé aux projets du laboratoire, aux démonstrations des véhicules intelligent et avec qui j'ai partagé de longs moments, Stéphane Galland, Mickael Goncalves, Olivier Lamotte, Alexandre Willaume et Ronan Zéo.

Merci à celles et ceux qui ont partagé mon quotidien au sein du laboratoire SeT et qui ont fait que ces années se soient écoulées dans la bonne humeur, je pense notamment à notre chère secrétaire Ariane Glatigny mais aussi à Nicolas Gaud, Vincent Hilaire et David Meignan.

En dernier lieu, je remercie ma famille et particulièrement ma future femme Jessica pour avoir partagé les bons moments et les plus difficiles pendant mes années d'études.



---

# Sommaire

---

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Contexte . . . . .	15
1.2	Objectifs de ces travaux . . . . .	16
1.2.1	Navigation multi-véhicules . . . . .	16
1.2.2	Un modèle multi-agents réactifs pour la conduite en convoi . . . . .	17
1.2.3	Modèle multi-agents réactifs pour la navigation autonome . . . . .	19
1.2.4	Approche formelle : spécification et vérification des systèmes multi-agents réactifs . . . . .	20
1.2.5	Simulation et expérimentation . . . . .	20
1.3	Plan de la thèse . . . . .	21
1.3.1	Première partie . . . . .	21
1.3.2	Deuxième partie . . . . .	22
1.3.3	Troisième partie . . . . .	22
<b>I</b>	<b>Problématique et modélisation</b>	<b>25</b>
<b>2</b>	<b>Les systèmes Multi-agents</b>	<b>27</b>
2.1	Introduction . . . . .	28
2.2	Les concepts fondamentaux . . . . .	29
2.2.1	Les agents . . . . .	29
2.2.2	L'environnement . . . . .	32
2.2.3	Les interactions . . . . .	33
2.2.4	L'organisation . . . . .	34
2.3	Applications de systèmes multi-agents réactifs . . . . .	34
2.3.1	Le contrôle du trafic aérien . . . . .	35
2.3.2	Problèmes de Positionnement . . . . .	36
2.3.3	Fourragement et tri de couvain chez les fourmis . . . . .	37
2.4	Évaluation et mesure des propriétés des SMA . . . . .	39
2.4.1	Évaluation des propriétés des SMA . . . . .	39
2.4.2	Mesure de propriétés globales d'un SMA . . . . .	39
2.5	Conclusion . . . . .	41

<b>3</b>	<b>La conduite en convoi</b>	<b>43</b>
3.1	Introduction . . . . .	44
3.2	Contrôle longitudinal . . . . .	45
3.2.1	Différents types de contrôles longitudinaux . . . . .	45
3.2.2	Exemple de contrôle longitudinal . . . . .	46
3.3	Contrôle longitudinal et latéral . . . . .	47
3.3.1	Contrôle latéral et longitudinal local . . . . .	47
3.3.2	Contrôle latéral et longitudinal global . . . . .	50
3.3.3	Bilan des méthodes pour la conduite en convoi . . . . .	52
3.4	Phases de transitions . . . . .	52
3.5	Conclusion . . . . .	53
<b>4</b>	<b>Modèle multi-agents réactifs pour la conduite en convoi</b>	<b>55</b>
4.1	Modèle abstrait de comportement . . . . .	56
4.2	Modèle d'interaction . . . . .	57
4.2.1	Mécanisme de perception . . . . .	57
4.2.2	Construction du modèle physique d'interaction . . . . .	58
4.2.3	Paramètres du modèle d'interaction . . . . .	61
4.3	Conclusion . . . . .	63
<b>5</b>	<b>Analyse du modèle physique d'interactions pour la conduite en convoi</b>	<b>65</b>
5.1	Analyse du modèle d'interaction . . . . .	66
5.1.1	Etude cinématique du modèle d'interaction : cas sans excitation . . . . .	66
5.1.2	Etude cinématique du modèle d'interaction : cas avec excitation . . . . .	69
5.1.3	Conditions pour la vérification de la sûreté . . . . .	70
5.2	Preuves de stabilité du train de véhicules . . . . .	71
5.2.1	Preuve de la stabilité d'inspiration physique . . . . .	71
5.2.2	Preuve de la stabilité inspirée de l'automatique . . . . .	72
5.3	Conclusion . . . . .	74
<b>6</b>	<b>Modèle multi-agents réactifs pour la navigation autonome</b>	<b>75</b>
6.1	Une approche pour l'évitement d'obstacles . . . . .	76
6.1.1	Introduction . . . . .	76
6.1.2	Caractéristiques de l'approche . . . . .	76
6.1.3	Représentation de l'environnement et attributs des agents . . . . .	78
6.2	Modèle d'interaction . . . . .	79
6.2.1	Description du modèle . . . . .	79
6.2.2	Calcul du comportement de l'agent décisionnel . . . . .	81
6.3	Evaluation globale pour la prise de décision . . . . .	82
6.3.1	Calcul du vecteur décision . . . . .	82

6.3.2	Modulation du vecteur décision à partir de la physique statistique . . .	83
-------	--	----

## **II Spécification formelle et vérification 85**

### **7 Langages formels 87**

7.1	Introduction . . . . .	88
7.2	Systèmes formels, vérification, preuve, model-checking . . . . .	88
7.3	Quelques langages formels . . . . .	90
7.3.1	Z et Object-Z . . . . .	90
7.3.2	La méthode B . . . . .	92
7.3.3	Les langages multi formalismes : <i>OZS</i> . . . . .	95
7.4	Conclusion . . . . .	97

### **8 Une approche de vérification 99**

8.1	Introduction . . . . .	100
8.2	Les difficultés de la vérification et comment les contourner . . . . .	100
8.3	La boîte à outils SAL . . . . .	102
8.4	De <i>OZS</i> vers SAL . . . . .	103
8.4.1	Méthode de traduction . . . . .	103
8.4.2	Structure du système de transition associé à une classe <i>OZS</i> . . . . .	104
8.4.3	La méthode de vérification . . . . .	107
8.4.4	Vérification par model-checking . . . . .	109

### **9 Spécification et vérification de la conduite en convoi 111**

9.1	Introduction . . . . .	112
9.2	Modèle de spécification . . . . .	112
9.2.1	Modèle de spécification du véhicule . . . . .	113
9.2.2	Modèle de spécification de la perception . . . . .	113
9.2.3	Modèle de spécification du contrôle . . . . .	114
9.2.4	Spécification du modèle dynamique . . . . .	116
9.3	Traduction d' <i>OZS</i> vers SAL . . . . .	117
9.4	Vérification de la sûreté de fonctionnement d'un train de véhicules . . . . .	120
9.4.1	Propriété à vérifier . . . . .	120
9.4.2	Mise en pratique de la vérification compositionnelle . . . . .	120
9.4.3	Etude comparative des approches de vérification . . . . .	122
9.5	Conclusion . . . . .	123

## **III Simulation et expérimentation 125**

### **10 Les plate-formes de simulation et d'expérimentation 127**



10.1	Introduction . . . . .	128
10.2	Plate-forme de simulation 3D . . . . .	128
10.2.1	Présentation de la plate-forme . . . . .	128
10.2.2	Architecture du simulateur . . . . .	129
10.3	Plate-forme d'expérimentation . . . . .	131
10.3.1	Présentation de la plate-forme . . . . .	131
10.3.2	Capteurs . . . . .	133
10.4	Protocole de simulation et d'expérimentation . . . . .	134
10.4.1	Protocole de simulation . . . . .	135
10.4.2	Protocole d'expérimentation . . . . .	136
<b>11</b>	<b>Simulation et expérimentation</b>	<b>137</b>
11.1	Introduction . . . . .	138
11.2	Evaluation de la distance inter-véhicule . . . . .	138
11.2.1	Simulation : évaluation de la distance inter-véhicule . . . . .	139
11.2.2	Expérimentation : évaluation de la distance inter-véhicule . . . . .	141
11.2.3	Bilan : évaluation de la distance inter-véhicule . . . . .	143
11.3	Evaluation de l'erreur latérale . . . . .	143
11.3.1	Simulation : évaluation de l'erreur latérale . . . . .	143
11.3.2	Expérimentation : évaluation de l'erreur latérale . . . . .	145
11.3.3	Bilan : évaluation de l'erreur latérale . . . . .	146
11.4	Évitement d'obstacle simple . . . . .	147
11.4.1	Simulation : évitement obstacle simple . . . . .	148
11.4.2	Expérimentation : évitement obstacle simple . . . . .	149
11.5	Évitement d'obstacle multiple . . . . .	150
11.5.1	Simulation : obstacles multiples . . . . .	150
11.5.2	Expérimentation : obstacles multiples . . . . .	151
11.6	Conclusion . . . . .	152
<b>IV</b>	<b>Conclusion et Perspectives</b>	<b>155</b>
<b>12</b>	<b>Conclusion</b>	<b>157</b>
12.1	Bilan et apports . . . . .	157
12.1.1	Proposition d'un modèle multi-agents réactifs pour la conduite en convoi . . . . .	158
12.1.2	Proposition d'un modèle multi-agents réactifs pour la navigation autonome . . . . .	158
12.1.3	Spécification et vérification de la conduite en convoi . . . . .	159
12.1.4	Simulation et expérimentation . . . . .	159

12.2 Problèmes non résolus . . . . .	160
12.3 Perspectives . . . . .	160
12.3.1 Du point de vue du modèle multi-agents réactifs pour la conduite en convoi . . . . .	160
12.3.2 Du point de vue du modèle de SMAR proposé pour la navigation autonome . . . . .	161
12.3.3 Du point de vue de la spécification et la vérification . . . . .	161
12.3.4 Du point de vue applicatif . . . . .	161

## **V Annexes 163**

<b>A Interprétation de l'organisation émergente d'un SMAR basée sur la physique statistique</b>	<b>165</b>
A.1 Potentiels thermodynamique . . . . .	165
A.2 Exemple : oscillateur harmonique . . . . .	168
A.3 Utilisation de la physique statistique dans le cadre des systèmes multi-agents	169
<b>B Expérimentation des Cybercars</b>	<b>171</b>
B.1 Aéroport de Schiphol 1997 . . . . .	171
B.2 CyberCars/CyberMove 2001 . . . . .	172
B.3 Le projet BODEGA 2003 . . . . .	173
B.4 MobiVip 2005 . . . . .	173
B.5 CRISTAL 2008 . . . . .	174
<b>C Modèle d'interaction</b>	<b>177</b>
C.1 Paramètres du modèle d'interaction . . . . .	177
C.1.1 Force théorique du modèle d'interaction : . . . . .	177
C.1.2 Paramètre du modèle d'interaction : . . . . .	178
<b>D Capteurs</b>	<b>179</b>
D.1 Capteurs télémétriques . . . . .	179
D.1.1 LMS 200 . . . . .	180
D.1.2 IBEO . . . . .	181
D.2 Capteur de type GPS . . . . .	182
<b>E Approche de modélisation, spécification et vérification pour des SMAR d'ins- piration physique</b>	<b>183</b>
E.1 Perspective . . . . .	183
E.1.1 Vue générale . . . . .	184
E.1.2 Construction d'un modèle abstrait de comportement . . . . .	185

---

E.1.3	Construction du modèle physique d'interaction . . . . .	186
E.1.4	Analyse du modèle physique d'interaction . . . . .	187
E.1.5	Construction du modèle de spécification de la classe d'agent . . . . .	187
E.1.6	Verification . . . . .	188
E.1.7	Implémentation . . . . .	189

## **VI Bibliographie**

**191**

---

# Sommaire des Figures

---

1.1	Convoi de deux véhicules . . . . .	17
1.2	Modèle multi-agents réactifs pour la navigation autonome . . . . .	19
1.3	Organisation de la thèse . . . . .	21
2.1	Wall E : un agent cognitif . . . . .	30
2.2	Wall E : un agent réactif . . . . .	31
2.3	Représentation des composants d'un système multi-agents . . . . .	32
2.4	Force répulsive et force glissante, forces de glissements coordonnés . . . . .	36
2.5	Comparaison de la configuration proposée et du résultat de positionnement par l'approche agents . . . . .	37
2.6	Les fourmis en haut ont pris un plus court chemin. Plus elles ont franchi l'obstacle, plus de phéromones ont été déposées du côté supérieur ou inférieur. La fourmi venant de l'autre sens choisira donc le côté inférieur. . . . .	38
2.7	Méthodes d'évaluation . . . . .	39
3.1	Exemple de platooning avec des véhicules autonomes du projet PATH [Hedrick et al., 1994], évoluant sur une Automated Highway System (AHS) . . . . .	44
3.2	Image de synthèse : flotte de véhicules devant le laboratoire Système Et Transport (SET) . . . . .	45
3.3	Exemple de contrôle ACC Adaptive Cruise Control de Mercedes-Benz . . . . .	46
3.4	Contrôle longitudinal [Daviet and Parent, 1996, Laugier et al., 1999] . . . . .	47
3.5	Contrôle latéral [Daviet and Parent, 1996, Laugier et al., 1999] . . . . .	48
3.6	Contrôle latéral et longitudinal modélisé par des impédances non amorties . . . . .	48
3.7	Contrôle latéral et longitudinal modélisé par un modèle d'impédance contrôle . . . . .	49
3.8	Contrôle local VS contrôle global . . . . .	51
3.9	Caractérisation des espaces de satisfaction autour des véhicules. . . . .	52
3.10	Phases de transitions selon Halle and Chaib-draa [2005] . . . . .	53
4.1	Modèle de comportement du véhicule suiveur . . . . .	56
4.2	Zone de perception des véhicules . . . . .	57
4.3	Lien virtuel entre véhicules . . . . .	58
4.4	Anticipation du contrôle de suivi en courbe . . . . .	58
4.5	Courbure du ressort . . . . .	59
4.6	Force de torsion . . . . .	59

4.7	Représentation d'un train de véhicules . . . . .	60
4.8	Convoi de deux véhicules . . . . .	61
4.9	Paramètres du modèle d'interaction de suivi de véhicule . . . . .	64
5.1	Modèle d'interaction dans le cas d'un mouvement rectiligne . . . . .	66
5.2	Paramètres dynamique du véhicule et du modèle d'interaction . . . . .	67
5.3	Réponse dynamique du système "masse ressort" par rapport à une distance en dessous de la distance inter-véhicule souhaitée . . . . .	68
5.4	Réponse dynamique du système "masse ressort" par rapport à une distance au dessus de la distance inter-véhicule souhaitée . . . . .	68
5.5	Réponse dynamique du système "masse ressort" par rapport à une distance égale à la distance de sécurité . . . . .	69
5.6	Réponse dynamique du système "masse ressort" par rapport à une excitation . . . . .	70
5.7	Etude globale du modèle d'interaction . . . . .	71
5.8	Fréquences limites de stabilité du système . . . . .	73
6.1	Vue d'ensemble de l'approche pour l'évitement d'obstacle . . . . .	77
6.2	Zones de perception des agents décisionnels . . . . .	79
6.3	Comportement de répulsion Agent-Agent . . . . .	80
6.4	Comportement d'attraction Agent-point de passage (Gauche) point de passage dans la zone de perception, (Droite) point de passage en dehors de la zone de perception . . . . .	81
6.5	Calcul du vecteur décision $\vec{M}$ . . . . .	82
8.1	Classe <i>OZS</i> d'un agent réactif . . . . .	103
8.2	Contexte SAL associé à la classe <i>Agent</i> . . . . .	104
8.3	Comportement du rôle <i>Perception</i> . . . . .	106
8.4	Application de la règle de déduction . . . . .	110
9.1	Modèle de comportement du véhicule suiveur . . . . .	112
9.2	Sous statechart associé à l'état <i>INSERTED</i> . . . . .	112
9.3	Structure de la classe <i>InsertedVehicle</i> . . . . .	113
9.4	Modèle de perception du véhicule . . . . .	114
9.5	Modèle de contrôle du véhicule . . . . .	116
9.6	Modèle dynamique du véhicule . . . . .	117
9.7	Tableau de la relation entre le nombre de véhicules et nombre de nœuds parcourus lors d'une vérification non compositionnelle . . . . .	123
9.8	Tableau de la relation entre le nombre de véhicules et nombre de nœuds parcourus lors d'une vérification compositionnelle . . . . .	123
10.1	Simulation 3D temps réelle d'un train de véhicules en station . . . . .	128
10.2	Architecture du simulateur 3D temps réel . . . . .	129

10.3	Modèle physique d'un train de véhicules en station . . . . .	130
10.4	Modèle <i>PhysX</i> d'un véhicule . . . . .	131
10.5	Véhicules intelligents du laboratoire SET . . . . .	132
10.6	Architecture du système embarqué du véhicule . . . . .	132
10.7	Capteurs installés sur le véhicule du laboratoire SET . . . . .	133
10.8	Zone de perception du véhicule . . . . .	134
10.9	Balise réfléchissante . . . . .	134
10.10	Scénario de simulation 3D et d'expérimentation réelle . . . . .	135
10.11	Paramètres dynamiques du véhicule et du modèle de contrôle de suivi . . . . .	135
11.1	Evaluation de l'écart longitudinal . . . . .	138
11.2	Evaluation de l'écart latéral . . . . .	138
11.3	Visualisation de la distance inter-véhicule dans un train de 4 véhicules . . . . .	139
11.4	Mesure de l'erreur longitudinale . . . . .	140
11.5	Mesure de l'erreur longitudinale en fonction de la vitesse du véhicule précédent . . . . .	140
11.6	Changement rapide de vitesse du premier véhicule en simulation . . . . .	141
11.7	Visualisation de la distance inter-véhicule sur un train constitué de 2 véhicules . . . . .	141
11.8	Mesure de l'erreur d'écart . . . . .	142
11.9	Changement rapide de la vitesse du premier véhicule en expérimentation . . . . .	143
11.10	Bilan de l'étude des cas critiques . . . . .	143
11.11	Erreur latérale lors de la sortie d'une station . . . . .	144
11.12	Simulation : relation entre la rotation des roues (rayon de courbure) et de l'erreur latérale . . . . .	144
11.13	Relation entre la rotation de la roue et l'erreur latérale . . . . .	145
11.14	Erreur latérale lors de la sortie d'une station . . . . .	145
11.15	Relation entre la rotation de la roue et l'erreur latérale . . . . .	146
11.16	Expérimentation : relation entre la rotation des roues (rayon de courbure) et de l'erreur latérale . . . . .	146
11.17	Configuration d'obstacle simple . . . . .	147
11.18	Evitement d'obstacle simple . . . . .	148
11.19	Distance de l'obstacle le plus proche avec une configuration d'obstacle simple . . . . .	148
11.20	Trajectoire du véhicule avec une configuration d'obstacle simple . . . . .	149
11.21	Distance de l'obstacle le plus proche avec une configuration d'obstacle simple . . . . .	149
11.22	Evitement d'obstacles multiples . . . . .	150
11.23	Trajectoire du véhicule lors de la simulation . . . . .	150
11.24	Vitesse du véhicule . . . . .	151
11.25	Trajectoire du véhicule lors l'expérimentation . . . . .	152

---

11.26	Vitesse du véhicule lors de l'expérimentation . . . . .	152
A.1	Potentiels thermodynamiques les plus couramment utilisés . . . . .	165
A.2	Exemple de mesure de stabilité d'un système de boïds [Contet et al., 2008] .	166
A.3	Propriété intéressante de la fonction de partition . . . . .	168
A.4	Oscillateur harmonique . . . . .	168
B.1	Véhicules automatisés de l'aéroport de Schiphol . . . . .	171
B.2	Cycabs de la présentation du projet CyberCars/CyberMove . . . . .	172
B.3	Projet MobiVip . . . . .	174
C.1	Repère utilisé . . . . .	177
C.2	Forces subies par deux éléments reliés par un piston . . . . .	178
D.1	Télémetre laser LMS 200 . . . . .	180
D.2	Télémetre laser IBEO . . . . .	181
D.3	GPS Pro Flex 500 . . . . .	182
E.1	Dépendance entre activités . . . . .	185
E.2	Composition des rôles . . . . .	186
E.3	Boucle perception-action . . . . .	187

# INTRODUCTION

---

## 1.1 Contexte

Le concept de véhicule intelligent a donné lieu à plusieurs travaux de recherche et d'innovation technologique visant à améliorer le confort et la sécurité des usagers et à promouvoir de nouvelles approches de mobilité. Ainsi de nombreuses applications ont vu le jour telles que l'évitement d'obstacle, la perception de l'environnement, la communication inter-véhicules, la conduite automatisée, la navigation autonome, etc... Par ailleurs, ce concept occupe une place importante dans la stratégie i2010<sup>1</sup> de la communauté Européenne et s'inscrit également dans la poursuite des recommandations du livre blanc de 2004 sur les enjeux de la politique des transports à l'horizon 2010.

Dans la majorité de ces applications, l'intelligence conférée au véhicule se matérialise par sa capacité à prendre des décisions de manière proactive et à interagir avec son environnement (l'infrastructure et les autres véhicules par exemple). D'un point de vue conceptuel, l'approche multi-agents offre un cadre bien adapté pour modéliser et mettre en œuvre ces capacités.

Le principal intérêt de ce type d'approche est sa capacité à résoudre des problèmes complexes, éventuellement distribués, tout en conservant une simplicité tant fonctionnelle que conceptuelle des entités. Les systèmes multi-agents sont caractérisés, en particulier, par des propriétés multi-échelles qui prennent source dans la définition locale des comportements individuels des agents et qui apparaissent, de façon émergente, au niveau global du système. Cependant, de part la complexité des problèmes à résoudre, il est difficile de prouver ou de vérifier les propriétés globales en ne connaissant que la définition locale des comportements des agents. Cette difficulté peut devenir critique pour les applications soumises à des contraintes sévères de sûreté de fonctionnement. Il est donc indispensable d'élaborer des méthodes permettant de définir des modèles multi-agents, principalement de façon locale, tout en assurant l'émergence d'états globaux caractérisés par de bonnes propriétés. Un pas dans cette direction peut consister en l'application de méthodes de spécification formelle et de vérification tout au long du processus de conception du système multi-agents.

En se situant dans le cadre des projets de recherche sur les véhicules intelligents, nos

---

<sup>1</sup>la société de l'information et les médias au service de la croissance et de l'emploi



travaux de recherche visent à montrer l'apport des systèmes multi-agents réactifs à la modélisation et la mise en œuvre de la navigation multi-véhicules.

## 1.2 Objectifs de ces travaux

L'objectif principal de cette thèse peut être résumé par l'énoncé suivant :

proposer des modèles multi-agents réactifs fondés sur un cadre formel pour la vérification de propriétés et les valider par la simulation et l'expérimentation en considérant la navigation multi-véhicules.

Cet objectif peut se décliner en quatre parties :

- La classe d'application cible de nos travaux : nous nous intéressons à la navigation multi-véhicules et en particulier à deux problèmes spécifiques : la conduite en convoi ou platooning et la navigation autonome.
- Les fondements de l'approche proposée : nous proposons une approche fondée sur les agents réactifs, inspirée de la physique.
- Un cadre formel pour la vérification des propriétés : conscient des contraintes de sécurité imposées par la classe d'applications cibles, nous proposons de spécifier de manière formelle nos modèles afin de vérifier certaines propriétés de sûreté.
- La simulation et l'expérimentation : nous avons contribué au développement d'un simulateur de la conduite en convoi et la navigation autonome. Le simulateur prend en compte les caractéristiques physiques des véhicules et est couplé à un outil de visualisation 3D. Nous avons aussi expérimenté nos modèles sur des véhicules tels que le Cycab et le GEM Car.

Nous allons préciser séparément chacun de ces points.

### 1.2.1 Navigation multi-véhicules

Ce domaine d'application soulève plusieurs problématiques : perception de l'environnement, communication inter-véhicule, évitement d'obstacle, ... Dans le cadre de cette thèse, nous nous sommes intéressés à deux applications :

- **La conduite en convoi ou platooning**

Il s'agit de la navigation le long d'une même trajectoire d'un groupe de véhicules formant un convoi tout en respectant une distance donnée. Bien évidemment, les véhicules du convoi ne sont pas liés par un accrochage matériel. Dans ce cas, on parle

d'accrochage immatériel entre les véhicules. Le véhicule de tête (leader) peut être en conduite manuelle comme c'est le cas dans notre étude ou complètement automatique. A noter que ces deux modes de conduite ne changent rien quant à l'approche que nous proposons dans la section suivante.

Nous considérons des véhicules capables de percevoir leur environnement. En particulier, chaque véhicule peut disposer de capteurs lui permettant de calculer la distance qui le sépare de son prédécesseur et l'angle de courbure  $\theta$  entre eux (cf. figure 1.1).

#### – La navigation autonome

Il s'agit de doter le véhicule d'une capacité lui permettant de se déplacer vers un objectif de façon autonome. L'autonomie est assurée par un comportement permettant au véhicule d'éviter les obstacles le séparant de son objectif. Comme dans la conduite en convoi, le véhicule autonome est doté de capacité de perception de l'environnement.

### 1.2.2 Un modèle multi-agents réactifs pour la conduite en convoi

Le concept d'agent réactif a fait l'objet de plusieurs travaux. Il stipule qu'il n'est pas nécessaire que l'agent soit doté de connaissances individuelles élaborées pour que le système montre un comportement collectif qualifié "d'intelligent". C'est ce concept qui constitue le fondement des modèles que nous proposons pour les deux problèmes évoqués précédemment.

Plus précisément, nous considérons chaque véhicule du convoi comme un agent réactif capable de percevoir son environnement et de réagir en maintenant une distance souhaitable avec le véhicule qui le précède. La conduite en convoi est le résultat des interactions directes entre chaque véhicule et son prédécesseur.

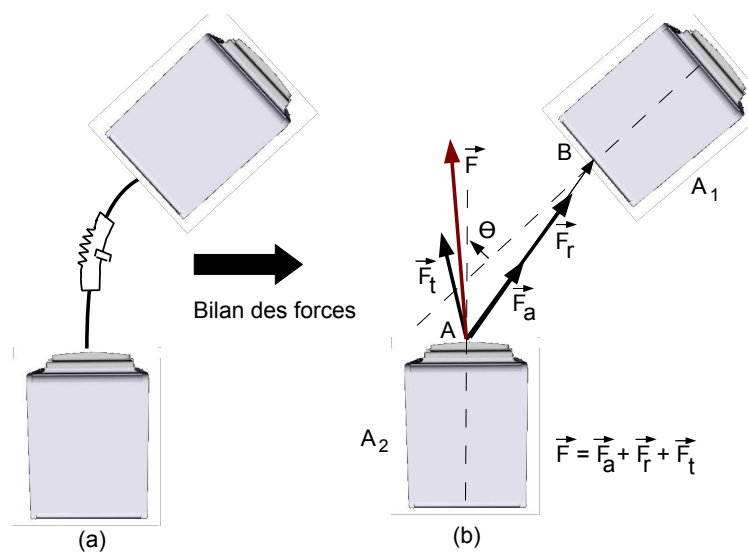


Figure 1.1: Convoi de deux véhicules

Nous proposons un modèle d'interaction inspiré de la physique. En effet, l'interaction entre deux véhicules qui se suivent est matérialisée par un lien virtuel de type “masse-ressort”, complété par un moment de torsion. La figure 1.1 illustre ce principe en considérant deux véhicules du convoi.

En fonction de la distance  $\|\vec{AB}\|$  et de l'angle de courbure  $\theta$  entre les deux véhicules  $A_1$  et  $A_2$ , une force  $\vec{F}$  résultante de la somme de la force du ressort  $\vec{F}_r$ , de l'amortisseur  $\vec{F}_a$  et de la force de torsion  $\vec{F}_t$  est appliquée au véhicule  $A_2$  (cf. figure 1.1). L'agent véhicule  $A_2$  élabore alors sa décision (accélération, direction) en appliquant simplement le principe fondamental de la dynamique (i.e.  $m\vec{\gamma} = \vec{F}$ ).

Par rapport aux travaux existants sur la conduite en convoi, notre approche peut être assimilée aux méthodes dites de “proche en proche”. Les deux principaux inconvénients de ce type de méthode sont :

- Le phénomène d'oscillation  
Il résulte de la non prise en compte de l'état du système multi-véhicules dans son intégralité. Pour remédier à cet inconvénient, nous avons mis en place un mécanisme permettant une adaptation dynamique des paramètres du modèle (raideur du ressort, coefficient d'amortissement, ...) en fonction de la dynamique du véhicule.
- L'anticipation en courbe  
La méthode de proche en proche fait que le véhicule suiveur anticipe la courbure du virage, provoquant ainsi une erreur de suivi de la trajectoire du précédent. Grâce au moment de torsion qui complète le modèle “masse ressort”, nous avons remédié à cet inconvénient.

En s'appuyant sur la physique, nous avons réalisé une étude cinématique du modèle d'interaction. Ceci permet de calculer la réponse du modèle ainsi que ses limites de fonctionnement. Ces limites servent, ensuite, à formuler les conditions de sûreté pour la vérification. Puis nous faisons des preuves classiques de stabilité inspirées de la physique et de l'automatique.

Ces preuves sont réalisées suivant deux critères. Le premier consiste à établir qu'en l'absence de perturbations extérieures le système converge vers un état stable. Grâce au théorème de Lyapunov, la stabilité du système “convoi” est étudiée en analysant l'évolution de l'énergie du système.

Le second critère dit qu'une perturbation affectant un composant du système “convoi” ne s'amplifie pas en se propageant vers les autres composants. Cette preuve est réalisée suivant la démarche classique adoptée pour le problème de la “string stability”.

### 1.2.3 Modèle multi-agents réactifs pour la navigation autonome

Contrairement à la conduite en convoi, nous nous intéressons ici au comportement d'un seul véhicule se déplaçant, de façon autonome, le long d'une trajectoire tout en évitant les obstacles. Doter le véhicule d'un tel comportement doit enrichir la conduite en convoi, en permettant au véhicule leader de naviguer de manière complètement autonome le long d'une trajectoire prédéfinie. Par ailleurs, ce comportement doit permettre aux autres véhicules du convoi d'éviter les obstacles (par exemple, un piéton qui traverse entre deux véhicules du convoi) tout en poursuivant la trajectoire de leur prédécesseur.

Pour traiter cette problématique, nous nous inspirons encore une fois de la physique et particulièrement de l'approche à base de forces. Contrairement au modèle précédent, cette approche n'associe pas un agent à chaque véhicule mais un système multi-agents (cf. figure 1.2) dont l'observation des caractéristiques émergentes permet de construire une décision. Dans ce système, l'environnement joue un rôle particulier puisqu'il agit dynamiquement sur le processus de décision. En fonction des positions du but et des obstacles éventuellement perçus, l'environnement génère des forces sur les agents du système. Ainsi, l'état but du véhicule induit une force d'attraction alors que les obstacles génèrent des forces de répulsion. Sous l'action de ces forces, les agents adoptent des configurations spatiales et temporelles caractéristiques permettant de construire une loi de commande pour le véhicule. Le modèle multi-agents réactifs que nous proposons est une adaptation de l'approche de suivi en robotique mobile présentée dans [Gechter, 2003].

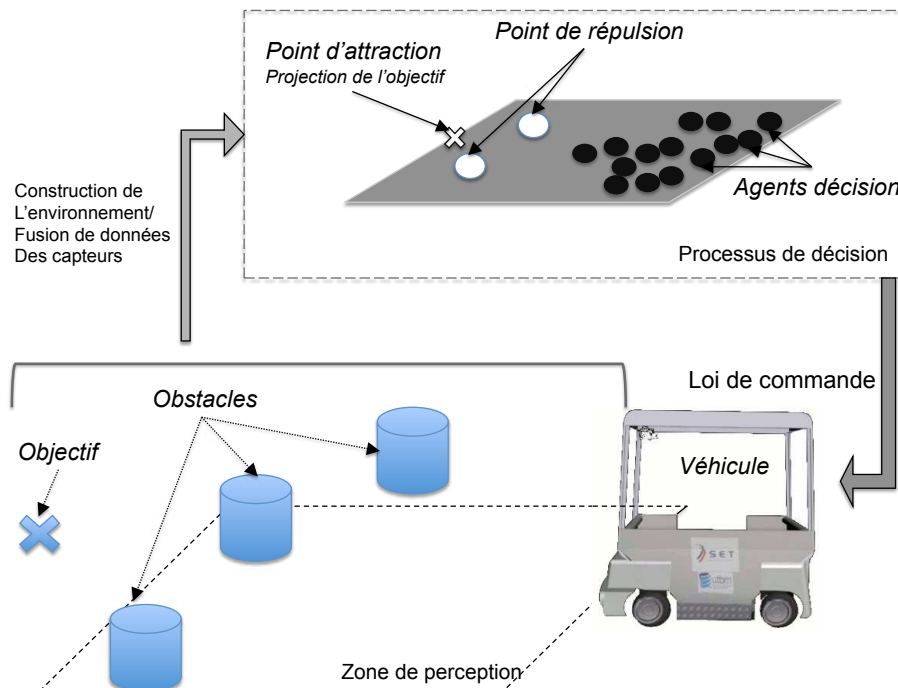


Figure 1.2: Modèle multi-agents réactifs pour la navigation autonome

Actuellement, nous n'avons pas encore réalisé l'intégration des deux modèles proposés. En effet, nous les avons analysés, simulés et expérimentés sur des véhicules réels de manière complètement séparée. Cette démarche d'intégration constitue une des perspectives de nos travaux.

### 1.2.4 Approche formelle : spécification et vérification des systèmes multi-agents réactifs

De nombreuses applications de systèmes multi-agents réactifs doivent satisfaire des propriétés de sûreté. C'est le cas des convois multi-véhicules, qui constituent la principale problématique de cette thèse.

Pour répondre à cette exigence, nous proposons une approche de vérification basée sur la spécification formelle et le model-checking. La spécification formelle est fondée sur le langage multi-formalisme *OZS* [Gruet et al., 2004, Koukam, 28 Janvier 2000], résultant de l'intégration du langage Object-Z (extension objet du langage Z) et les Statecharts (extension des automates à état fini). Un agent est alors spécifié par une classe Object-Z qui intègre un modèle de comportement exprimé par un statechart.

L'approche de vérification que nous proposons consiste à traduire la spécification *OZS* en un système de transition pouvant être analysé par l'environnement de vérification SAL. Une des caractéristiques importante de cette approche est l'introduction d'une règle de vérification compositionnelle. En effet, cette règle permet de réduire le risque d'explosion combinatoire lors du model-checking.

Nous avons ainsi pu vérifier la propriété de sûreté relative à l'absence de collision dans un convoi de véhicules.

### 1.2.5 Simulation et expérimentation

Compte tenu du domaine d'application cible de ce travail, le développement d'un simulateur prenant en compte les caractéristiques physiques des véhicules est primordial. Le simulateur doit permettre d'évaluer différents scénarii critiques avant de passer à une expérimentation sur des véhicules réels. C'est pourquoi, nous avons implanté nos modèles dans un simulateur développé au sein de notre laboratoire.

Nous avons ainsi pu valider par simulation notre approche de conduite en convoi et la navigation autonome en étudiant plusieurs scénarii.

Enfin, pour compléter notre étude, nous avons procédé à une étape d'expérimentation de nos modèles sur des véhicules de type CYCAB et GEM. Cette expérimentation réelle, nous a permis de concrétiser et de confirmer les résultats obtenus lors de la vérification et la simulation.

## 1.3 Plan de la thèse

Suivant les objectifs dégagés au paragraphe 1.2, nous décomposons ce mémoire en trois parties. La figure 1.3 montre les parties qui caractérisent notre contribution.

La **première** dresse un état de l'art sur les systèmes multi-agents, la navigation multi-véhicules et présente deux modèles de système multi-agents réactifs. Le premier constitue notre contribution à la conduite en convoi, le deuxième modèle est relatif à la navigation autonome.

La **deuxième partie** présente notre approche de spécification formelle du modèle de conduite en convoi et la vérification des propriétés de sûreté.

La **troisième partie** est consacrée à la simulation des modèles que nous avons proposés dans la partie I et à leur expérimentation sur des véhicules du laboratoire.

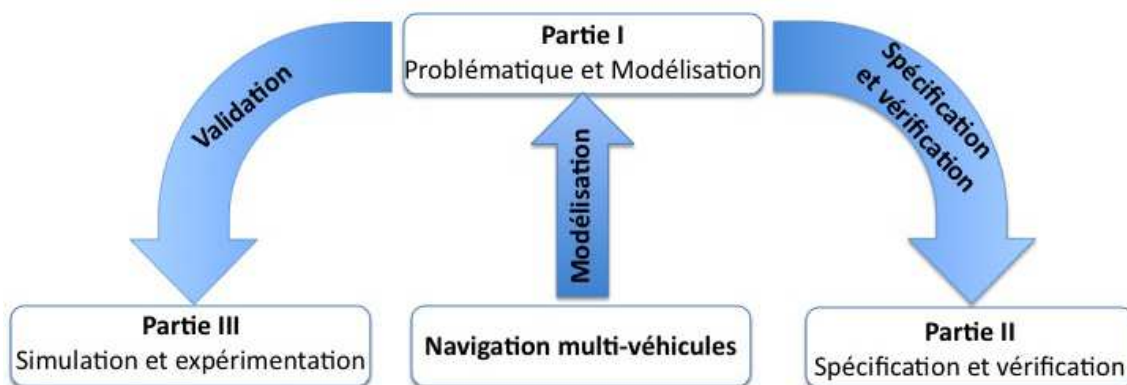


Figure 1.3: Organisation de la thèse

### 1.3.1 Première partie

Dans le **chapitre 2**, nous présentons d'abord les concepts de base des systèmes multi-agents. L'accent est mis ensuite sur les systèmes multi-agents réactifs. Une discussion sur l'évaluation de leurs propriétés est présentée à la fin de ce chapitre.

Au **chapitre 3**, nous présentons les différentes approches proposées pour la conduite en convoi ou platooning. Ces approches se distinguent par le type de contrôle qu'elles mettent en œuvre : contrôle longitudinal, latéral ou une combinaison des deux.

Le **chapitre 4** présente l'approche que nous proposons pour la conduite en convoi. Il s'agit d'une approche fondée sur les agents réactifs et un modèle d'interaction inspiré de la physique. L'interaction entre deux agents ou véhicules est assimilée à un lien virtuel de type "masse ressort", complété par un moment de torsion.

Le **chapitre 5** présente une analyse du modèle d'interaction présenté au chapitre 4. En effet, nous exploitons les résultats issus de la physique pour analyser la cinématique du modèle et la stabilité du convoi de véhicules.

Le **chapitre 6** expose le modèle de systèmes multi-agents réactifs que nous proposons pour la navigation autonome d'un véhicule. Il s'inspire d'une approche réactive fondée sur les forces d'attraction et de répulsion développée pour le suivi en robotique mobile. Cette contribution doit permettre d'enrichir le modèle de conduite en convoi (chapitre 4) en permettant à un véhicule de naviguer de manière autonome tout en évitant les obstacles.

### 1.3.2 Deuxième partie

Cette partie vise à donner un cadre formel à nos modèles en vue de prouver certaines propriétés.

Le **chapitre 7** est une introduction aux approches de spécification formelle et de vérification de propriétés. Il présente le langage Z, son extension orientée objet (Object-Z) et le formalisme B. Ensuite le langage multi-formalismes *OZS*, fondé sur l'intégration des deux langages Object-Z et Statecharts est présenté. Développé au sein de notre laboratoire, c'est ce langage que nous utilisons pour spécifier nos modèles.

Le **chapitre 8** présente notre approche de vérification formelle fondée sur le modèle checking et l'environnement SAL que nous avons utilisé pour la mettre en œuvre. L'approche consiste à traduire la spécification *OZS* en un système de transition, point d'entrée de l'environnement SAL. Afin d'éviter le problème de l'explosion combinatoire, cette approche introduit une règle de vérification compositionnelle.

Le **chapitre 9** est consacré à la spécification formelle et la vérification du modèle du système multi-agents réactifs pour la conduite en convoi. Il présente la spécification formelle du modèle en utilisant le formalisme *OZS*. Ensuite, il expose la vérification des propriétés de sûreté.

### 1.3.3 Troisième partie

C'est dans cette partie que nous validons nos contributions par la simulation et l'expérimentation sur les véhicules du laboratoire.

Le **chapitre 10** présente une plateforme de simulation dotée d'un outil de visualisation 3D et d'un moteur physique prenant en compte de manière fine les caractéristiques des véhicules : éléments mécaniques, amortisseurs, adhérence, capteurs... etc. Ensuite, il expose la plateforme d'expérimentation composée de deux véhicules électriques de type GEM et d'un CYCAB. Les protocoles de simulation et d'expérimentation constituent la dernière partie de ce chapitre.

Le **chapitre 11** montre les résultats de la simulation et de l'expérimentation de plusieurs scénarii de conduite en convoi et d'évitement d'obstacle.

Enfin, le mémoire termine par **une conclusion** présentant le bilan et les perspectives de nos travaux.





---

PREMIÈRE PARTIE

---

# **Problématique et modélisation**

---



# LES SYSTÈMES MULTI-AGENTS

---

**C**E CHAPITRE présente les concepts fondamentaux des systèmes multi-agents (SMA) en portant une attention particulière à la définition des systèmes multi-agents réactifs (SMAR). Des exemples sélectionnés seront présentés pour illustrer les différentes inspirations possibles dans l'utilisation des SMA. Enfin, diverses méthodes sont exposées pour effectuer l'évaluation et la mesure de propriétés d'un SMA.

---

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>28</b>
<b>2.2</b>	<b>Les concepts fondamentaux</b>	<b>29</b>
2.2.1	Les agents	29
2.2.2	L'environnement	32
2.2.3	Les interactions	33
2.2.4	L'organisation	34
<b>2.3</b>	<b>Applications de systèmes multi-agents réactifs</b>	<b>34</b>
2.3.1	Le contrôle du trafic aérien	35
2.3.2	Problèmes de Positionnement	36
2.3.3	Fourragement et tri de couvain chez les fourmis	37
<b>2.4</b>	<b>Évaluation et mesure des propriétés des SMA</b>	<b>39</b>
2.4.1	Évaluation des propriétés des SMA	39
2.4.2	Mesure de propriétés globales d'un SMA	39
<b>2.5</b>	<b>Conclusion</b>	<b>41</b>

---

## 2.1 Introduction

Les Systèmes Multi-agents, ou SMA, proposent une nouvelle métaphore basée sur des concepts sociaux. Cette métaphore repose sur la notion d'agent, entité qui exhibe des comportements flexibles et autonomes dans des environnements dynamiques et non prédictibles. La métaphore sociale d'agents qui interagissent afin de satisfaire des objectifs locaux ou globaux constitue un paradigme prometteur pour la modélisation, l'analyse et la conception appliquées à de nombreux domaines comme : la simulation de systèmes complexes, la résolution distribuée de problèmes, l'optimisation de problèmes possédant une forte dynamique, les systèmes ouverts, les agents assistants, ...

Cette grande diversité de domaines d'applications a donné lieu à de nombreux points de vues sur la façon de définir et construire les agents et leurs interactions qui les lient au sein d'un SMA.

On peut distinguer deux grandes « écoles » de pensée dans la façon de concevoir les agents. La première école essaye de reproduire une société de penseurs. Chaque agent est en soi une entité capable de mener un processus de délibération, on parle alors d'agents cognitifs. Ce type d'agent, relativement complexe, donne lieu à des SMA composé d'un faible nombre d'agents.

La deuxième école prend comme exemple les insectes sociaux ou certains phénomènes physiques et donne lieu à des agents simples dits réactifs. Ces agents implémentent un mécanisme de type stimulus/réponse. Chaque agent ne possède pas de représentation des autres et le SMA est en général composé d'un grand nombre d'agents.

Les SMA Réactifs, ou SMAR, exhibent des propriétés de robustesse et d'émergence qui en font des candidats de choix pour l'approche présentée dans ce manuscrit. Un des inconvénients des SMAR est la difficulté liée à la conception, à partir de comportements simples en interaction, fréquemment inspirés de phénomènes biologiques ou physiques, d'un système qui va accomplir les buts pour lesquels il a été conçu. Cette problématique génère deux niveaux de représentation, local au niveau de l'agent et global au niveau du système. Le niveau local est la cause du niveau global puisque le résultat des interactions des agents constitue le résultat au niveau système. Afin de résoudre ce problème nous étudions le concept d'indicateurs qui permettent d'évaluer localement et globalement l'évolution des SMAR.

Ce chapitre est structuré comme suit : la section 2.2 définit les concepts d'agents et SMA ainsi que les concepts qui en dépendent. La section 2.3 propose trois exemples de SMAR afin d'illustrer les concepts précédemment présentés et la section 2.4 recense les indicateurs de la littérature.

## 2.2 Les concepts fondamentaux

Cette section présente les principales définitions qui permettent d’appréhender la notion d’agent et de système multi-agents en reprenant la description donnée par Demazeau [1995].

- Agents (A)
- Environnement (E)
- Interactions (I)
- Organisations (O)

### 2.2.1 Les agents

La littérature propose deux principales définitions : la première par Russell and Norvig [1995] qui définit un agent comme « *Tout ce qui peut être vu comme percevant son environnement à l’aide de capteurs et agissant sur cet environnement à l’aide d’effecteurs, de façon autonome* ». Cette première définition qui se veut généraliste a été “étendue” par Ferber [1995] :

**Définition** : “ On appelle agent, une entité physique ou virtuelle :

1. Qui est capable d’agir dans un environnement ;
2. Qui peut communiquer directement avec d’autres agents ;
3. Qui est mue par un ensemble de tendances (Sous forme d’objectifs individuels ou de fonctions de satisfaction, voire de survie, qu’elle cherche à optimiser) ;
4. Qui possède des ressources propres ;
5. Qui est capable de percevoir son environnement (mais de manière limitée) ;
6. Qui ne dispose que d’une représentation partielle de cet environnement (et éventuellement aucune) ;
7. Qui possède des compétences et offre des services ;
8. Qui peut éventuellement se reproduire ;
9. Et dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications quelle reçoit. “

Il existe deux écoles concernant la conception de systèmes multi-agents : l’école cognitives et l’école réactive [Ferber, 1995]. La principale différence entre les approches réactif/cognitif réside dans la représentation du monde que les agents disposent. Le type cognitif possède sa propre représentation du monde, de la tâche qu’il doit accomplir et éventuellement de faculté de communication avec les autres agents. L’agent réactif, quant à lui,

possède seulement une perception limitée de l'environnement et fonctionne selon un principe Action/Réaction [Ferber, 1995] ou Stimulus/Réponse [Bonabeau et al., 1999].

### Agents cognitifs

L'approche cognitive est la plus représentée dans la communauté de l'intelligence artificielle distribuée. Elle trouve son origine dans la volonté de faire communiquer et coopérer des entités intelligentes.

Chaque agent dispose d'une base de connaissances et de règles nécessaires pour exécuter sa tâche et gérer les interactions avec les autres agents et l'environnement (cf. figure 2.1).

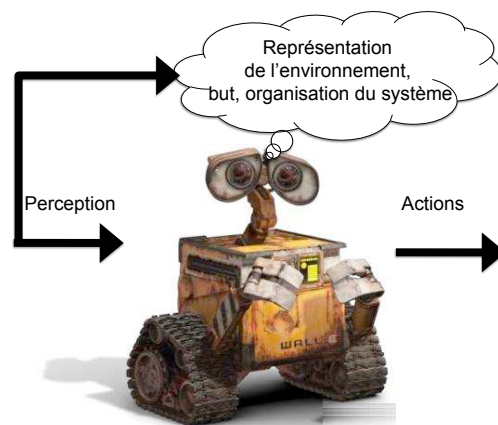


Figure 2.1: Wall E : un agent cognitif

L'inspiration anthropomorphe conduit à concevoir les agents cognitifs comme des entités possédant des croyances, des intentions, des désirs et même dans certains cas des principes éthiques [Shoham, 1993, Wooldridge and Jennings, 1995]. Comme *Shoham* le précise, "une perspective mentale ou intentionnelle des agents intelligents n'est pas simplement une autre invention, mais un paradigme utile pour décrire les systèmes répartis complexes". De tels agents, principalement caractérisés par un niveau symbolique de représentation des connaissances et par des états mentaux, sont nommés **agents intelligents**.

L'une des caractéristiques majeures des agents cognitifs, de part leurs capacités représentationnelles et de raisonnement, est qu'ils peuvent travailler indépendamment et dans certains cas, se comporter comme des organisations d'individus devant coordonner leurs activités pour résoudre leurs conflits [Jennings et al., 1998].

### Agents réactifs

L'approche réactive, quant à elle, considère que les agents ne doivent pas obligatoirement posséder une certaine forme d'intelligence pour que le système complet se comporte

de façon intelligente. On observe souvent ce type de comportement dans la nature (abeilles, termites, ...). Ces agents sont dépourvus de connaissances explicites concernant son but propre et le but global du système. Ils évoluent en interagissant mutuellement et avec l'environnement. Selon Drogoul [1993], on les qualifie **d'agent réactif**. Contrairement aux agents cognitifs, avec ce type d'agent, il n'y a pas de représentation de but global du système ni de planification permettant de prévoir des propriétés d'émergence [Parunak, 1999].

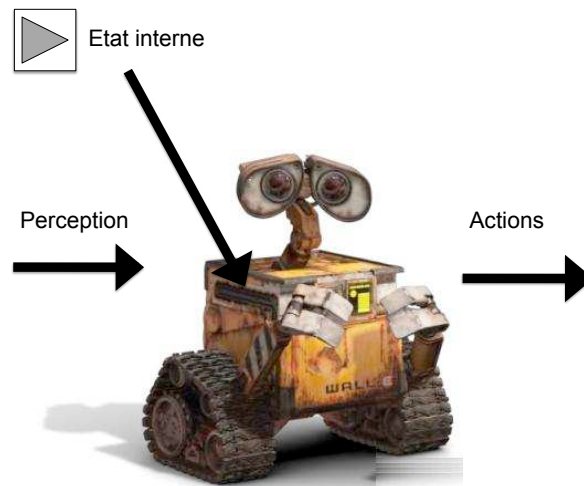


Figure 2.2: Wall E : un agent réactif

Ainsi dans de tels systèmes, la notion d'interaction surpasse la notion d'agent. Le comportement des agents réactifs est calculé en fonction de ses interactions et non pas de sa propre connaissance (cf. figure 2.2).

### Les Systèmes multi-agents

Un système multi-agents est “un système composé d'agents qui interagissent pour réaliser leurs objectifs individuels et/ou collectifs“. Ferber [1995] propose une définition plus détaillée :

**Définition** : On appelle système multi-agents, un système composé des éléments suivants :

1. Un environnement  $E$ , c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets  $O$ . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans  $E$ . Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble  $A$  d'agents, qui sont des objets particuliers ( $A \subseteq O$ ), lesquels représentent les entités actives du système.



4. Un ensemble d'opérations  $Op$  permettant aux agents de  $A$  de percevoir, produire, consommer, transformer et manipuler des objets de  $O$ .
5. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

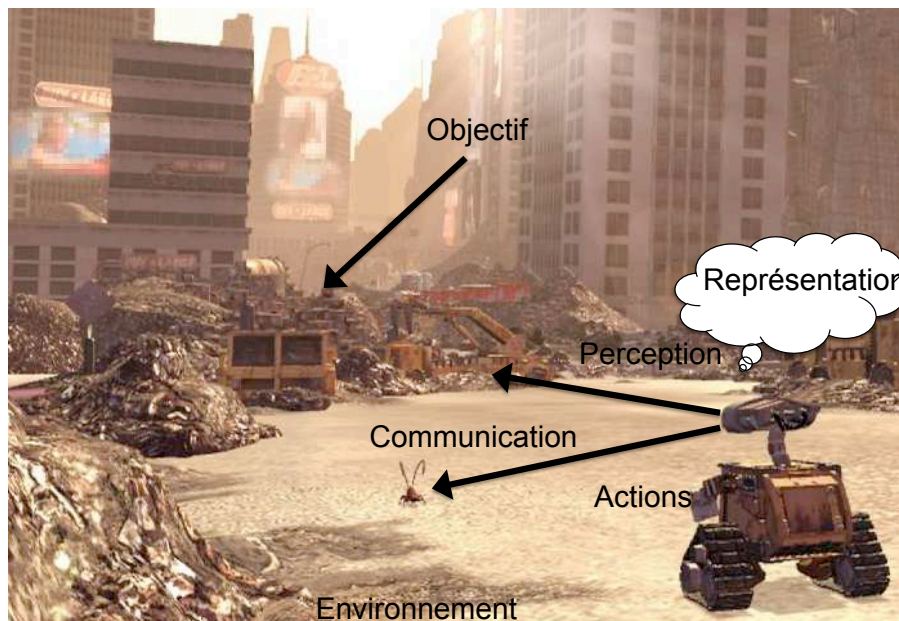


Figure 2.3: Représentation des composants d'un système multi-agents

Un système multi-agents est un système composé d'agents qui communiquent et collaborent pour accomplir des objectifs spécifiques individuels ou collectifs (cf. figure 2.3). La communication implique l'existence d'un espace partagé support de cette communication. Cet espace est généralement qualifié d'environnement.

### 2.2.2 L'environnement

L'environnement est un élément important dans le processus dynamique d'un système multi-agents. Il peut être défini comme étant tout ce qui entoure les agents [Weyns et al., 2005]. L'environnement peut posséder une dynamique propre qui influe directement ou indirectement sur les actions des agents.

Dans [Russell and Norvig, 1995] les auteurs proposent une caractérisation des différents types d'environnement.

- accessible versus inaccessible : un environnement est dit accessible lorsque les agents ont accès à l'intégralité de son état. L'accessibilité de l'environnement n'est pas une propriété souhaitable pour un SMA où la portée des actions et de la perception est locale.

- déterministe versus indéterministe : un environnement est déterministe si un changement de son état est déterminé uniquement par son état courant et les actions des agents. Un environnement indéterministe peut produire deux résultats différents pour une même action.
- statique versus dynamique : un environnement est dit statique si il n'évolue que sous l'effet des actions des agents. Au contraire, un environnement dynamique possède ses propres processus d'évolution qui peuvent le faire changer d'état sans intervention des agents.
- discret versus continu : on parle d'environnement discret lorsque la fonction de perception est discrète.

Pour certains systèmes, l'environnement peut se réduire à des canaux de communications, comme pour les agents web. Cependant, dans les systèmes réactifs d'inspiration physique ou biologique, l'environnement est essentiel à l'accomplissement des interactions entre les agents. Il permet une immersion réelle des agents. L'environnement offre ainsi la possibilité, de prendre en compte la dynamique de l'agent, les contraintes, la métrique, ... Dans le cas d'une inspiration physique, l'environnement dispose d'une métrique et de propriétés intrinsèques permettant l'applicabilité des principes dictés par la physique [Gechter et al., 2004]. Dans le cadre de la simulation d'insectes [Muller, 2004, Parunak, 1997], l'environnement d'une colonie de fourmis contient des zones pour représenter les nids, la nourriture, ... De plus, il gère les évaporations des phéromones déposées par les fourmis.

Dans [Weyns et al., 2005] les auteurs distinguent deux types d'aspects pour les environnements : les aspects structurels (structuration de l'environnement, métrique, propriétés fondamentales,...) et les aspects dynamiques (perception, action, dynamique,...).

Dans notre cas, l'environnement doit pouvoir rendre compte des propriétés dynamiques et métriques du monde réel ainsi que des véhicules utilisés.

### 2.2.3 Les interactions

Le concept d'interaction est fondamental pour les SMA. En effet, sans interactions aucune métaphore sociale n'est possible. Dans le cadre des SMA, une interaction peut être définie comme la mise en relation dynamique de deux ou plusieurs éléments du système par le biais d'un ensemble d'actions réciproque [Gechter, 2003]. Ces éléments peuvent être des agents ou des éléments de l'environnement ou encore l'environnement lui-même. Les interactions peuvent être directes, Agent-Agent, ou indirecte, Agent-Environnement. En d'autres termes, soit l'agent reçoit cette information directement, par envoi de message par exemple, ou indirecte, par le biais de l'environnement ou d'une structure partagée. Ce dernier type d'interaction est parfois qualifié de stigmergie. Il peut prendre la forme de phéromones, de champs de potentiels, etc. A la différence d'une communication directe où les interlocuteurs sont identifiés, dans une communication indirecte il n'y a pas de notion de destinataire mais plutôt d'une perception de traces déposées dans l'environnement.

L'interaction dépend alors de la capacité de perception des agents liée à la métrique de l'environnement et sa topologie ainsi qu'à une notion de temps. Une trace peut ainsi disparaître comme une phéromone qui s'évapore ou n'avoir qu'une durée instantanée comme un signal émit ponctuellement.

La communication directe a donné lieu à de nombreuses recherches sur les langages de communications comme les théories sur les actes de langage [Austin, 1970] et les protocoles d'interaction comme le réseau contractuel [Smith, 1988]. Les communications directes sont en général utilisées dans le cadre des SMA cognitifs tandis que les interactions indirectes sont utilisées dans les SMAR.

### 2.2.4 L'organisation

L'organisation au sein des SMA peut être vue de deux façons. La première consiste à considérer l'organisation comme une structure issue d'un arrangement des composants d'un système. Dans le cadre des systèmes multi-agents, l'organisation est considérée comme une configuration des agents et de leurs comportements en rapport avec les contraintes imposées par l'environnement.

Lorsque cette structure est la conséquence émergente de la conjonction des interactions locales entre les éléments du système, on parle d'auto-organisation. Dans le cadre de la résolution de problème, ce phénomène complexe représente une solution du problème à résoudre par le système multi-agents.

La deuxième définition possible d'une organisation est issue des travaux plaçant l'organisation au centre des préoccupations de l'ingénierie des SMA, parmi ces travaux on peut citer [Cossentino et al., 2009, Ferber et al., 2003, Hübner et al., 2007]. Dans ce cas, l'organisation fait référence aux théories organisationnelles et à la métaphore sociale des SMA. Cette perspective organisationnelle propose d'analyser et concevoir les SMA en utilisant les concepts de rôles ou comportement abstrait au sein d'une organisation et/ou de norme sociale qui impose des règles au sein d'une organisation.

## 2.3 Applications de systèmes multi-agents réactifs

Comme nous l'avons énoncé dans l'introduction, nous proposons dans cette thèse un modèle multi-agents réactifs pour la navigation. Dans ce contexte, nous nous concentrons sur l'approche réactive et proposons des exemples d'utilisation de SMAR. Ils permettent d'une part, d'illustrer les concepts définis précédemment et d'autre part de présenter des solutions à des problèmes ayant des propriétés communes avec la solution proposée dans cette thèse.

Le choix des applications a été guidé par la volonté d'illustrer deux types d'inspirations pour le modèle d'interaction des agents. Le premier et le deuxième exemples montrent

l'utilisation d'un SMA réactif d'inspiration physique pour d'une part un système de gestion du trafic aérien et d'autre part une solution aux problèmes de positionnement. Tandis que le troisième exemple illustre un modèle d'inspiration biologique adapté au fourragement et au tri de couvain.

### 2.3.1 Le contrôle du trafic aérien

Les organismes du contrôle aérien sont responsables de l'écoulement du trafic dans l'espace. Les contrôleurs doivent offrir une sécurité parfaite avec un débit optimum. Pour cela, l'espace est découpé en secteurs, à l'intérieur de chaque secteur, les contrôleurs maintiennent chaque avion séparé du reste du trafic en donnant aux pilotes des instructions. Le contrôleur assure la coordination avec les secteurs voisins et anticipe les conflits. Le projet <sup>1</sup> ATLAS23 est parmi les premiers à avoir envisagé des avions autonomes ou hybrides. Cette hypothèse a été étudiée par **Karim Zeghal** dans sa thèse [Zeghal, 1994] où il introduit la notion de coordination d'actions grâce à des forces s'exerçant sur des agents (cf. figure 2.4). On associe à chaque agent un modèle d'interaction d'inspiration physique. Comme [Reynolds, 1987], on définit trois types de forces :

- **Les forces attractives** qui permettent aux avions d'atteindre leur objectif (une balise ou leur destination finale).
- **Les forces répulsives** qui permettent aux avions d'éviter un obstacle proche et par conséquent dangereux.
- **Les forces de glissement** qui permettent de contourner les obstacles.

La force de glissement est définie de la manière suivante : si l'on observe l'équipotentielle de danger passant par l'avion, une force de glissement est tangente à cette équipotentielle alors que la force répulsive est normale à celle-ci.

Dans le cas d'un obstacle mobile, *Karim Zeghal* définit une action coordonnée d'évitement. Si deux avions arrivent au même point, ils subissent des forces de glissement complémentaires. Dans le cas simple de deux avions, on projette par exemple la vitesse relative sur l'équipotentielle de danger pour chaque avion et on obtient ainsi deux forces de glissement coordonnées. Il s'agit ensuite, pour limiter les changements brutaux de direction, de gérer l'intensité des différentes forces. En effet, les avions étant limités par leurs angles de virage, de montée et de descente, ils ne peuvent pas effectuer de manœuvre trop rapide. Les différentes forces s'exerçant sur les avions s'additionnent donc avec des coefficients variables suivant l'imminence du danger.

---

<sup>1</sup><http://www.recherche.enac.fr/opti/papers/thesis/NICOLAS/rapport002.html>

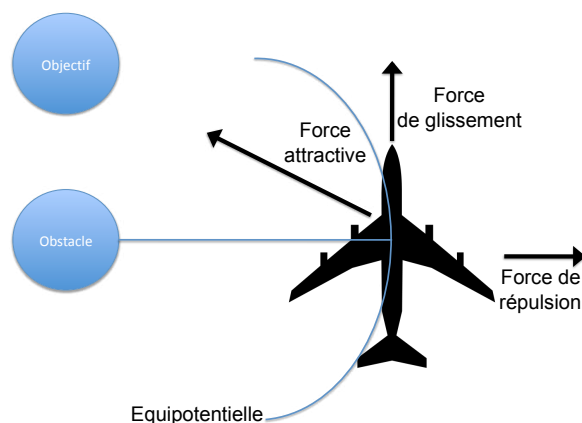


Figure 2.4: Force répulsive et force glissante, forces de glissements coordonnés

La robustesse des solutions est un des atouts de ces travaux. Cependant, ces résultats ne sont ni vérifiables ni démontrables, il est par conséquent impossible d'envisager l'utilisation de ce type de système dans un cas réel.

### 2.3.2 Problèmes de Positionnement

L'objectif du travail de Moujahed [2007] est de proposer une approche auto-organisée d'inspiration physique pour le problème de positionnement mono et multi-niveaux. Ce type de problème concerne le positionnement d'un ensemble de facilities<sup>2</sup> par rapport à des demandes de façon à optimiser un objectif global.

L'enjeu de cette approche est la conception d'un comportement intelligent d'un point de vue global à partir d'agents ayant des comportements réactifs simples. La solution au problème de positionnement peut être considérée comme un phénomène émergent dû à l'organisation spatiale issue des interactions des agents entre eux et avec leur environnement. Le point nodal est alors la mise au point de comportements élémentaires simples afin d'obtenir un résultat émergent correspondant aux objectifs désirés. Les comportements des agents facilities se basent sur des lois physiques simples d'attraction et de répulsion. Ces agents sont attirés vers la demande pour satisfaire leurs objectifs locaux et repoussés entre eux pour assurer une répartition cohérente sur l'environnement. La solution globale du problème correspond à un agencement cohérent des agents par rapport à la demande.

Cette approche a été appliquée pour positionner au mieux, par rapport à un objectif fixé, des arrêts de bus sur des lignes déjà existantes [Moujahed et al., 2006]. Les arrêts de bus doivent être positionnés de telle sorte que la distance moyenne parcourue par les passagers ou la demande pour accéder à l'arrêt le plus proche soit minimisée (cf. figure 2.5).

<sup>2</sup>Le terme facility désigne un objet capable de fournir un service tel que par exemple, des arrêts de bus, des entrepôts, des hôpitaux, des usines ou des écoles.

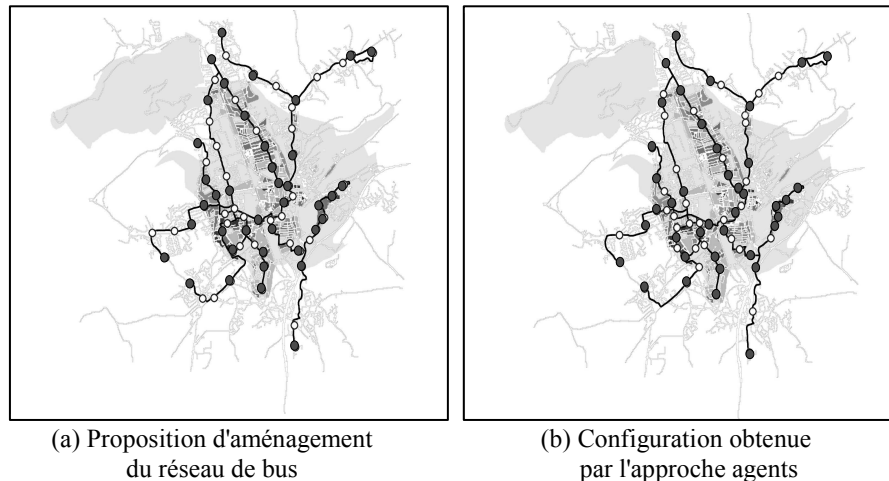


Figure 2.5: Comparaison de la configuration proposée et du résultat de positionnement par l'approche agents

Les travaux présentés dans la thèse de Moujahed [2007] permettent de proposer une solution réactive d'inspiration physique intéressante pour résoudre des problèmes complexes. Cependant les solutions proposées ne sont pas prouvées.

### 2.3.3 Fourrage et tri de couvain chez les fourmis

Le troisième exemple sélectionné présente un modèle d'inspiration biologique. Les insectes<sup>3</sup> sociaux comme les fourmis, les termites ou encore les abeilles présentent au niveau collectif des capacités supérieures à la somme des capacités individuelles. Cette performance est le résultat des interactions directes ou indirectes entre les individus. La compréhension de ces phénomènes est indispensable pour la compréhension de la vie artificielle et des mécanismes de l'émergence.

Dans la nature, les fourmis construisent un ensemble de chemins qui connectent le nid aux différentes sources de nourriture. La construction de ces trajets a été expliquée par les biologistes sous la forme d'un enchaînement de comportements élémentaires de type stimulus/réponse. D'un point de vue mathématique, les réseaux formés correspondent à des arbres couvrants minimaux [Goss et al., 1990]. Les chemins suivis par les fourmis correspondent aux chemins les plus courts. Cependant, la méthode de construction des réseaux ne peut pas être identifiée à un algorithme existant en théorie des graphes. Cette structure optimale du réseau émerge d'un ensemble d'actions individuelles simples exécutées par les fourmis.

Chaque fourmi possède des comportements élémentaires déclenchés en fonction de son environnement local [Parunak, 1999] :

- **Éviter** les obstacles

<sup>3</sup><http://www.vieartificielle.com/article/index.php?action=article&id=48>

- **Explorer** l'environnement de manière aléatoire
- **Déposer** des phéromones tout en se dirigeant vers le nid si la fourmi porte de la nourriture
- **Prendre** de la nourriture si la fourmi se trouve sur une source et si elle n'en transporte pas
- **Poser** la nourriture si la fourmi en transporte et se trouve sur le nid

De part la définition des ces comportements, tous les chemins de phéromones conduisent vers une source de nourriture. Par exemple, sur 6 fourmis, 3 fourmis pourront être passées par le chemin le plus court alors que les 3 fourmis ayant empruntées le chemin le plus long n'auront pas encore terminées de contourner l'obstacle. Une fourmi arrivant en sens inverse sentira donc plus de phéromones sur le chemin le plus court et elle le suivra (cf. figure 2.6).

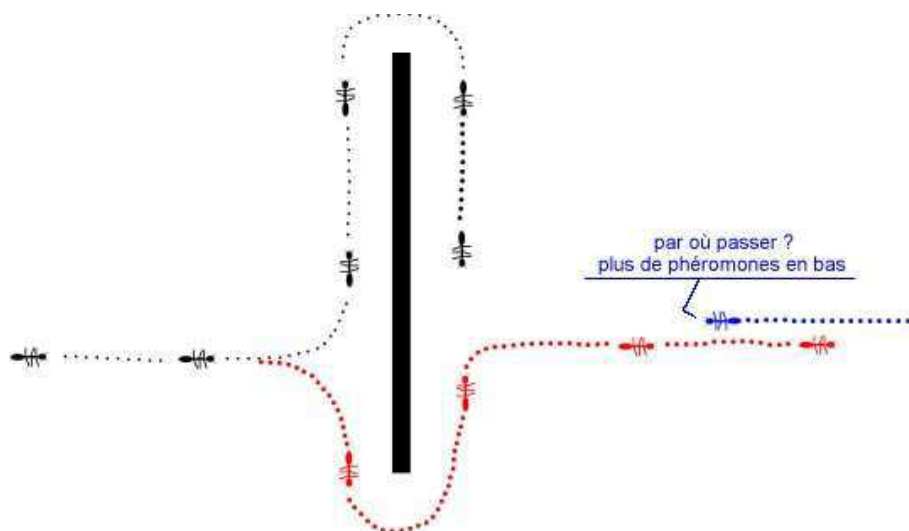


Figure 2.6: Les fourmis en haut ont pris un plus court chemin. Plus elles ont franchi l'obstacle, plus de phéromones ont été déposées du côté supérieur ou inférieur. La fourmi venant de l'autre sens choisira donc le côté inférieur.

L'inspiration biologique permet de retrouver les propriétés intéressantes telles que la robustesse, la flexibilité, et l'adaptation. En effet, lorsqu'un des chemins reliant le nid à la source de nourriture est obstrué, la quantité de phéromones présente dans celui-ci va diminuer jusqu'à disparaître. De façon simultanée, cette quantité va augmenter dans les autres chemins alternatifs jusqu'à ce que l'un d'entre eux soit privilégié.

Composées de plusieurs entités très simples, ces systèmes expriment un comportement global complexe et coordonné. Dans ce système, la solution émerge de la collaboration de l'ensemble des agents avec l'environnement. Cet exemple montre, de la même manière que le précédent, que les SMAR donnent une solution très intéressante face à des problèmes complexes sans permettre de prouver qu'ils apportent une solution pertinente et fiable à tout instant.

## 2.4 Évaluation et mesure des propriétés des SMA

### 2.4.1 Évaluation des propriétés des SMA

La notion d'émergence a été employée à l'origine par les physiciens et les biologistes pour exprimer l'apparition d'un phénomène au sein d'un système. Dans le cadre des SMA, selon M.R.Jean [1997], si le système dispose d'une dynamique, i.e. "une itération entre interprétation de leur environnement local par les agents, action des agents sur cet environnement, nouvelle interprétation de l'environnement modifié, nouvelles actions, etc. Quand une telle dynamique (ou certaines de ses composantes) se stabilise on peut parler d'émergence d'une structure ou de fonctionnalité globale."

Ce phénomène est une manifestation qui ne peut être simplement expliquée par la connaissance des constituants du système. Plus généralement, on peut formuler l'émergence comme l'idée qu'il existe dans un système des propriétés présentes à un niveau global d'organisation qui ne peuvent être déduites des propriétés du niveau local. L'émergence est vue par conséquent comme une apparition (phénomène physique comme l'émergence de la vie). Par définition même, il est très difficile de caractériser l'émergence dans les SMA. Pourtant, pour de nombreux SMA, il est intéressant de mesurer des propriétés globales afin d'évaluer la performance ou les propriétés des solutions obtenues.

### 2.4.2 Mesure de propriétés globales d'un SMA

La mesure de propriétés globales d'un SMA repose sur l'évaluation de l'efficacité du système par rapport à la tâche à accomplir. Les mesures sont faites en prenant en compte les propriétés locales et globales telles que l'objectif de l'agent ou du système et la dynamique de l'environnement. Afin de réaliser cette mesure, plusieurs approches ont été développées en s'inspirant de la biologie (valeur de fitness, ...), de la sociologie (fonction d'utilité, satisfaction, altruisme, ...) ou de la physique (entropie, énergie, ...). Celles-ci (cf. figure 2.7) peuvent être classées en trois catégories :

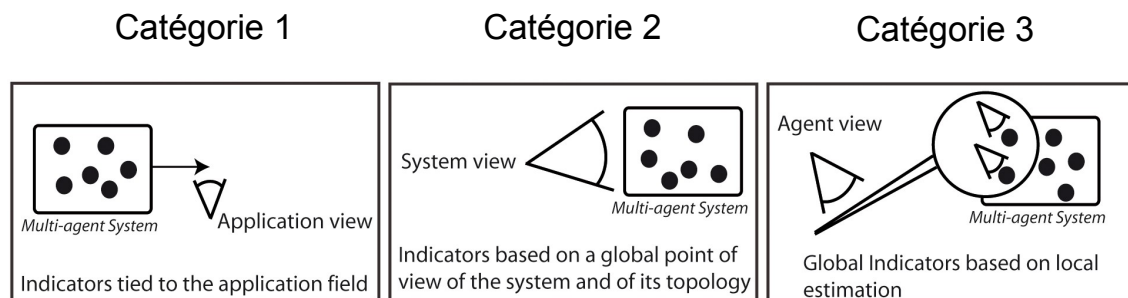


Figure 2.7: Méthodes d'évaluation



1. Evaluation par rapports aux objectifs globaux du système.
  2. Evaluation globale basée sur la mesure des phénomènes globaux.
  3. Evaluation des phénomènes émergents globaux en prenant en compte les mécanismes locaux
- **Evaluation par rapports aux objectifs** : la première catégorie est la plus simple. Cet indicateur s'appuie seulement sur la définition du meilleur résultat que l'on peut obtenir et mesure la différence entre ce résultat optimal, et la solution trouvée [Meignan et al., 2007, Moujahed et al., 2007]. par rapport au domaine d'application et l'évaluation des indicateurs communs. Par exemple, si un système vise à trouver le chemin le plus court dans un graphe, il est facile d'évaluer ses résultats par rapport aux chemins trouvés par d'autres méthodes.
- **Evaluation globale basée sur la topologie (ou la dynamique)** : la deuxième catégorie est la plus répandue dans la littérature. Dans les cas où l'inspiration est physique, l'entropie peut être utilisée pour estimer l'ordre ou l'organisation du système. Il existe plusieurs méthodes pour calculer l'entropie comme l'entropie sociale hiérarchique [Balch, 2000], ou l'entropie dynamique ou statistique [Parunak and Brueckner, 2001]. Toutefois, ce type de mesure a deux inconvénients majeurs. Premièrement, elle dépend de l'historique, en particulier des transformations du système. Elle ne peut pas être considérée comme une fonction d'état. En d'autres termes, deux systèmes peuvent être dans un même état, mais avec deux valeurs différentes d'entropie en fonction de leurs états antérieurs. Deuxièmement, l'entropie est une mesure qui ne prend pas en compte les phénomènes locaux du système.
- **Indicateurs globaux basés sur des phénomènes locaux** : La troisième catégorie d'indicateurs tend à pallier ces inconvénients. Une solution générique est le calcul de l'énergie comme une fonction d'état à la fois sur l'agent et sur les niveaux du système [Gaud et al., 2007]. Cependant, l'énergie n'est pas toujours facilement calculable et doit donc être limitée à des modèles à base de forces ou de champs de potentiel. Enfin, d'autres méthodes intéressantes, sur la base de la physique statistique et la thermodynamique, semblent être de plus en plus utilisées [Contet et al., 2008]. Ces méthodes sont basées sur la fonction de partition  $Z$  dont découlent toutes les fonctions de l'État (fonction de Gibbs, énergie libre, enthalpie, enthalpie libre, etc) des systèmes thermodynamiques. Les méthodes basées sur la fonction  $Z$  sont en mesure de prendre en compte, statistiquement, tous les composants du système afin d'en calculer une valeur représentative. Un potentiel intéressant est l'énergie libre  $A$  qui mesure le travail accessible à partir d'une transformation réversible. L'étude de l'énergie libre permet de déduire certaines propriétés sur l'évolution du système en fonctions de paramètres locaux. La principale difficulté dans l'utilisation des ces méthodes est la restriction des cas d'utilisation. Par exemple, la question : "Est-ce que le nombre d'agents est

suffisamment important pour être statistiquement significatif ?" peut se poser. En fait, d'après Balian [2007] la validité de l'approche statistique ne repose pas sur le nombre d'agent, mais sur le nombre d'états possibles dans le système. Ainsi, si l'environnement est continu, le nombre d'états accessibles est infini, ce qui permet d'utilisation de l'approche statistique.

## 2.5 Conclusion

Les systèmes multi-agents offrent une solution particulièrement intéressante pour la résolution de problèmes complexes. Ils sont applicables à un large panel d'applications. Parmi les modèles classiques, l'approche réactive est l'une des plus pertinentes du point de vue des propriétés (simple, flexible, fiable, adaptable, ...). Ces systèmes sont composés d'agents réactifs, des entités simples dont le comportement est basé sur leurs propres perceptions. La différence entre l'utilisation d'un système multi-agents (SMA) et une méthode classique de résolution de problèmes, réside dans le rôle et l'importance des interactions. De même, l'environnement joue un rôle prépondérant, puisqu'il est l'endroit où le système peut calculer ses nouvelles variables, construire de nouvelles structures et enfin communiquer. En effet, dans le paradigme agent, un seul agent ne peut ni connaître la représentation globale du problème ni calculer la solution globale. En revanche, la solution est obtenue grâce aux mesures des interactions agent-agent et/ou agent-environnement.

Cependant, les systèmes multi-agents souffrent de l'imprédictibilité de l'émergence, de la difficulté de réaliser une mesure des propriétés du système et du manque de méthodes de vérification de son bon fonctionnement.



# LA CONDUITE EN CONVOI

---

**C**E CHAPITRE propose un état de l'art des solutions existantes pour la conduite en convoi ou le platooning.

---

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>44</b>
<b>3.2</b>	<b>Contrôle longitudinal</b>	<b>45</b>
3.2.1	Différents types de contrôles longitudinaux	45
3.2.2	Exemple de contrôle longitudinal	46
<b>3.3</b>	<b>Contrôle longitudinal et latéral</b>	<b>47</b>
3.3.1	Contrôle latéral et longitudinal local	47
3.3.2	Contrôle latéral et longitudinal global	50
3.3.3	Bilan des méthodes pour la conduite en convoi	52
<b>3.4</b>	<b>Phases de transitions</b>	<b>52</b>
<b>3.5</b>	<b>Conclusion</b>	<b>53</b>

---

### 3.1 Introduction

Le platooning se définit par la construction d'un train de véhicules. Nous appellerons ainsi *platoon*, un groupe de véhicules se déplaçant sur une route en coordonnant leurs actions afin de construire une "accroche immatérielle" entre chaque véhicule. Le premier véhicule est appelé *leader* et son rôle est de guider le groupe de véhicule que l'on pourra aussi appeler *train de véhicules*. Chaque véhicule, le leader mis à part, suit le véhicule précédent. La figure 3.1 montre un exemple de platooning avec des véhicules autonomes du projet PATH [Hedrick et al., 1994], évoluant sur une Automated Highway System (AHS).



Figure 3.1: Exemple de platooning avec des véhicules autonomes du projet PATH [Hedrick et al., 1994], évoluant sur une Automated Highway System (AHS)

Le projet CRISTAL<sup>1</sup> a mis en évidence que la non attractivité du transport public pourrait être résolu par une approche basée sur l'utilisation d'une flotte de véhicules électriques conçus pour les villes. Ces véhicules propres offriront des services individuels ou collectifs tel que :

- **Adaptabilité** : des unités réparties dans l'environnement urbain seraient disponibles pour répondre à des demandes ponctuelles d'utilisateurs.
- **Reconfigurabilité** : un nombre variable de véhicules pourraient s'intégrer dans des trains virtuels (figure 3.2), afin de répondre aux variations de la demande des utilisateurs.
- **Autonomie** : les véhicules seront dotés d'intelligence et pourront adopter et suivre différents itinéraires, pour répondre à des besoins ponctuels.

Le point clé de la réalisation de cette solution réside dans le principe de l'accroche immatérielle entre véhicules.

<sup>1</sup>Cellule de Recherche Industrielle en Systèmes de Transport Automatisés Légers



Figure 3.2: Image de synthèse : flotte de véhicules devant le laboratoire Système Et Transport (SET)

D'une manière générale, les approches proposées pour la conduite en convoi se distinguent par le type de contrôle qu'elles mettent en oeuvre :

**Contrôle longitudinal** : la distance et la vitesse relative des voitures sont commandées par un système "de conduite de croisière", tandis que les conducteurs orientent toujours leurs véhicules manuellement. Ce contrôle peut-être comparé à une régulation de vitesse en fonction de celle du précédent.

**Contrôle longitudinal et latéral**, les mouvements latéraux et longitudinaux de chaque véhicule sont commandés de façon autonome relativement à celui ou à ceux qui les précède(nt), formant ainsi un train de véhicules. Le train de véhicules ainsi formé a une composition rigide ne permettant pas d'ajout ou de suppression de véhicules dans le train.

**Phases de transition** : on donne dans ce cas, une autonomie aux véhicules leurs permettant de sortir ou d'entrer du train de véhicules.

## 3.2 Contrôle longitudinal

Le contrôle longitudinal est l'un des premiers principes élaboré pour le suivi de véhicules. Il consiste à contrôler les freins et l'accélération afin de stabiliser la distance entre deux véhicules.

### 3.2.1 Différents types de contrôles longitudinaux

De nombreuses recherches ont été réalisé sur le contrôle longitudinal :

- **Sheikholeslam et Desoer** proposent un contrôle longitudinal en utilisant une méthode de linéarisation exact [Sheikholeslam and Desoer, 1992, 1990, 1993].
- **Ioannou and Xu** contrôlent les freins et l'accélération avec un asservissement par contrôleur PID avec un gain fixe [Ioannou and Xu, 1994].
- **Hendrick et al.** utilisent un mode de contrôle basé sur une méthode non-linéaire [Hedrick et al., 1994, Lu et al., 2001, McMahon et al., 1990, Moskwa and Hedrick, 1990] ainsi que [Fritz, 1999, Gerdes and Hedrick, 2002, Seshagiri and Khalil, 1999] pour le projet OTTO [Gehring and Fritz, 1997].
- **Lee et Kim** propose un contrôle longitudinal par logique floue [Lee and Kim, 2002, Lee and Tomizuka, 2003] ainsi que [Kodagoda et al., 2002].
- **Warnick et Rodriguez** propose une commande longitudinale par contrôle de saturation [Warnick and Rodriguez, 2000].

### 3.2.2 Exemple de contrôle longitudinal

**Hendrick** propose une méthode de contrôle de la distance entre deux véhicules. Le principe du contrôle longitudinal réside en un asservissement non linéaire de la distance  $D$  séparant les véhicules. Le principal intérêt de l'étude de **Hendrick** est la modélisation de tous les organes d'un véhicule réel, i.e. boîte de vitesse automatique, freinage, permettant ainsi d'intégrer le modèle à n'importe quel véhicule.

Ce type de contrôle a été démocratisé pour le grand public. Le contrôle ACC : Adaptive Cruise Control [Zhang et al., 1999] permet de réaliser un contrôle de la distance inter-véhicule proche du contrôle longitudinal <sup>2</sup> :

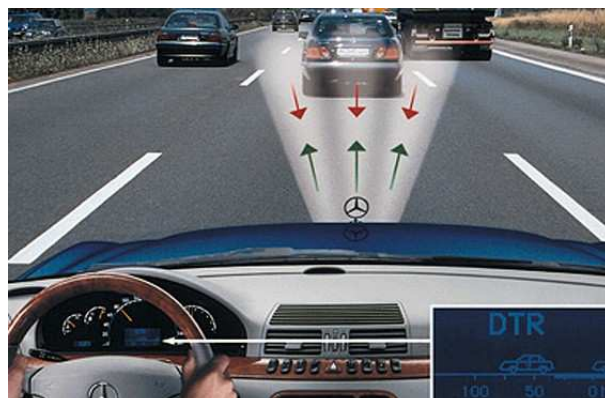


Figure 3.3: Exemple de contrôle ACC Adaptive Cruise Control de Mercedes-Benz

Ainsi, le ACC de Mercedes-Benz permet d'éviter les collisions par rapport à la voiture précédente en contrôlant la distance séparant les véhicules. Cependant, il se désactive automatiquement en cas de freinage d'urgence pour donner la main au conducteur.

<sup>2</sup>[http://reviews.cnet.com/Adaptive\\_cruise\\_control/4660-3424\\_7-6213621.html](http://reviews.cnet.com/Adaptive_cruise_control/4660-3424_7-6213621.html)

### 3.3 Contrôle longitudinal et latéral

Le contrôle longitudinal et latéral correspond au contrôle de la distance ainsi qu'au suivi latéral du véhicule précédent. Pour dresser un état de l'art des différentes solutions, nous allons présenter plusieurs exemples de contrôle latéral et longitudinal. Les exemples seront classés en deux catégories. La première présente des solutions de **contrôle local** et la deuxième montre les **solutions du contrôle global**.

#### 3.3.1 Contrôle latéral et longitudinal local

Le contrôle local correspond au contrôle latéral et longitudinal du véhicule avec la seule perception locale de celui-ci. Cette solution permet d'équiper les véhicules de capteurs leurs permettant de percevoir le véhicule précédent et de ne pas modifier l'infrastructure routière. Cette méthode de contrôle est dite de "proche en proche".

##### 3.3.1.1 Contrôle latéral et longitudinal basé sur un contrôle automatique

**Pascal Daviet, Michel Parent** proposent un contrôle longitudinal et latéral avec un contrôleur PID (Proportionnelle, Intégrale, Dérivée). L'acquisition des valeurs de contrôle est effectuée par des caméras linéaires ou par un capteur télémétrique [Riess, 2000]. Pour le contrôle longitudinal, la loi du suivi de véhicule, proposée dans [Daviet and Parent, 1996, Laugier et al., 1999], consiste à conserver un temps  $T$  constant entre deux véhicules. En intervenant sur ce paramètre, appelé *facteur de Davier-Parent*, il est possible de contrôler la distance à laquelle les véhicules se suivent. Puisqu'il s'agit d'un temps, cela signifie que la distance entre les véhicules varie proportionnellement à leur vitesse relative. La consigne de vitesse est donc proportionnelle à la distance ( $d-d_0$ ) qui reste à parcourir pour atteindre la cible.

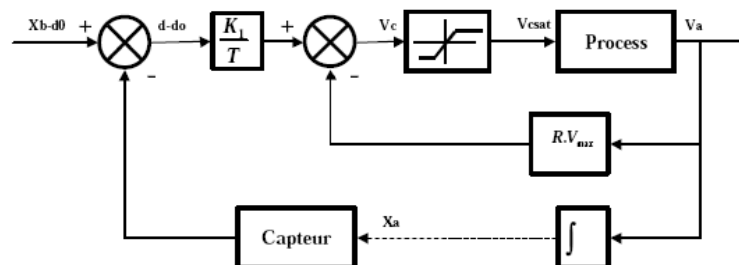


Figure 3.4: Contrôle longitudinal [Daviet and Parent, 1996, Laugier et al., 1999]

La loi de suivi de véhicule latérale consiste, quant à elle, en la conservation d'un angle proche de zéro entre le véhicule précédent et le suivant.



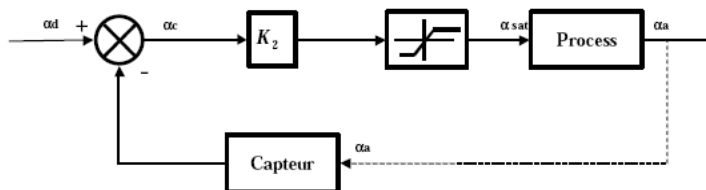


Figure 3.5: Contrôle latéral [Daviet and Parent, 1996, Laugier et al., 1999]

Le contrôle de platooning proposé par **Pascal Daviet**, **Michel Parent** présente des résultats intéressants mais souffre d'un problème d'anticipation des véhicules suiveurs en courbes similaire à celui observé lorsqu'un véhicule tracte une remorque.

### 3.3.1.2 Contrôle latéral et longitudinal à l'aide de phéromones

**Parunak** [Sauter et al., 2002, Van Dyke Parunak et al., 2002] a été un des premiers à travailler sur le contrôle de véhicules avec des phéromones. Ces travaux ont été surtout orientés pour le contrôle aérien par phéromones attractives et répulsives. Les véhicules sont attirés par les phéromones attractives et repoussés par les phéromones répulsives.

Ce contrôle a été réalisé en simulation et donne des résultats probants. Cependant, l'implémentation sur des véhicules réels semble difficile. En effet, il est presque impossible de déposer des marqueurs sur la voie publique simplement. Une solution envisageable serait le partage d'une carte en utilisant des moyens de géolocalisation précis et coûteux.

### 3.3.1.3 Contrôle latéral et longitudinal modélisé par des impédances non amorties

Gehrig and Stein [2001] ont construit une accroche immatérielle en modélisant les véhicules par des particules physiques soumises à des forces.

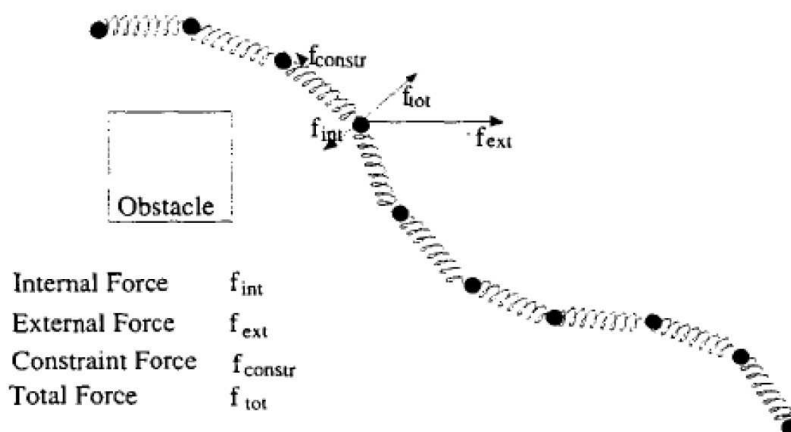


Figure 3.6: Contrôle latéral et longitudinal modélisé par des impédances non amorties

Bilan des forces subies par chaque véhicule :

$$\vec{F}_{totale} = \vec{F}_{int} + \vec{F}_{ext} + \vec{F}_{constr} \quad (3.1)$$

Les particules se déplacent vers leurs nouvelles positions  $\vec{P}_{i,new}$  en fonction des forces subies.

$$\vec{P}_{i,new} = \vec{P}_{i,old} + \alpha \vec{F}_{totale}(\vec{P}_i) \quad (3.2)$$

Cependant ce modèle a certaines limites. En effet, l'accroche immatérielle est modélisée par des ressorts réalisant une instabilité du système puisque les mouvements ne sont pas amortis. Il en résulte que les rayons de courbures doivent être faibles et correspondent davantage, par conséquent, aux autoroutes et non pas au trafic routier urbain.

### Contrôle latéral et longitudinal modélisé par un modèle d'impédance contrôle

**Soo-Yeong Yi et Kil-To Chong** ont construit un modèle d'impédance contrôle afin de modéliser l'accroche immatérielle entre véhicules [Yi and Chong, 2005].

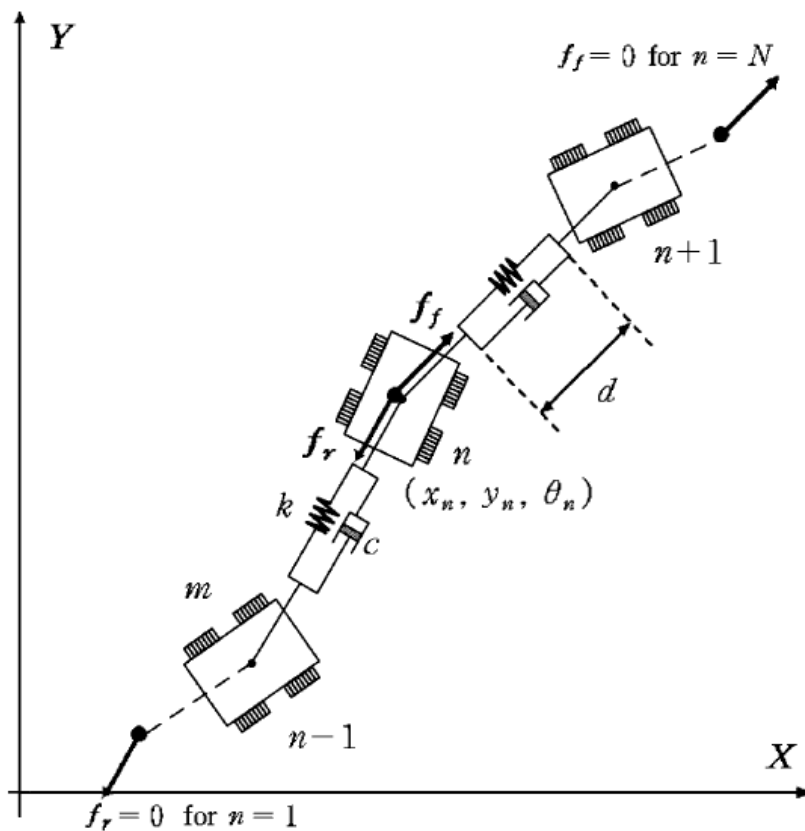


Figure 3.7: Contrôle latéral et longitudinal modélisé par un modèle d'impédance contrôle

Bilan des forces agissant sur le véhicule :

$$\vec{f}_f = k * (\vec{P}_{n+1} - \vec{P}_n - \vec{d}_n) + c * (\dot{\vec{P}}_{n+1} - \dot{\vec{P}}_n) \quad (3.3)$$

$$\vec{f}_r = k * (\vec{P}_n - \vec{P}_{n+1} - \vec{d}_{n-1}) + c * (\dot{\vec{P}}_n - \dot{\vec{P}}_{n-1}) \quad (3.4)$$

Avec  $\vec{P}_n = [x_n, y_n]^t$  le vecteur position de  $\vec{P}_n$  et  $\vec{d}_n$  défini la position entre le véhicule d'indice n et n+1.

Puis, il suffit d'appliquer le principe fondamental de la dynamique <sup>3</sup> :

$$m * \ddot{\vec{P}}_n = \vec{f}_f + \vec{f}_r \quad (3.5)$$

L'accélération peut être ainsi déduite en fonction des forces appliquées.

**Soo-Yeong Yi et Kil-To Chong** ont réussi à prouver la stabilité de leur modèle physique en calculant l'énergie dissipée par le train de véhicules. Cette énergie étant égale à  $m * \dot{\vec{P}}_n$ , il suffit de calculer sa dérivée :

$$\lim_{t \rightarrow \infty} E = 0 \quad (3.6)$$

Donc,

$$\lim_{t \rightarrow \infty} \vec{P} = 0 \quad (3.7)$$

A l'infini, la vitesse relative entre chaque couple de véhicule ne va devenir nulle, i.e. les véhicules vont se suivre à même vitesse et la distance entre eux va converger vers l'allongement à vide du ressort correspondant à la distance de sécurité. Ce qui prouve la stabilité du modèle physique.

Ce contrôle de platooning a été simulé et présente de bons résultats. Cependant, ce contrôle de suivi impose de connaître la distance inter-véhicule entre le véhicule suivant  $d_{i,i+1}$  et précédent  $d_{i-1,i}$ . Cette contrainte impose de disposer de deux dispositifs de capteurs situés à l'avant et à l'arrière du véhicule.

### 3.3.2 Contrôle latéral et longitudinal global

Le contrôle latéral et longitudinal local a été souvent critiqué pour sa mauvaise qualité de fonctionnement. En effet, le contrôle local dans certain cas ne permet pas à un véhicule de suivre la même trajectoire que son précédent (cf. figure 3.8).

<sup>3</sup>Soit un corps de masse m constante, l'accélération subie par un corps dans un référentiel galiléen est proportionnelle à la résultante des forces qu'il subit, et inversement proportionnelle à sa masse m.

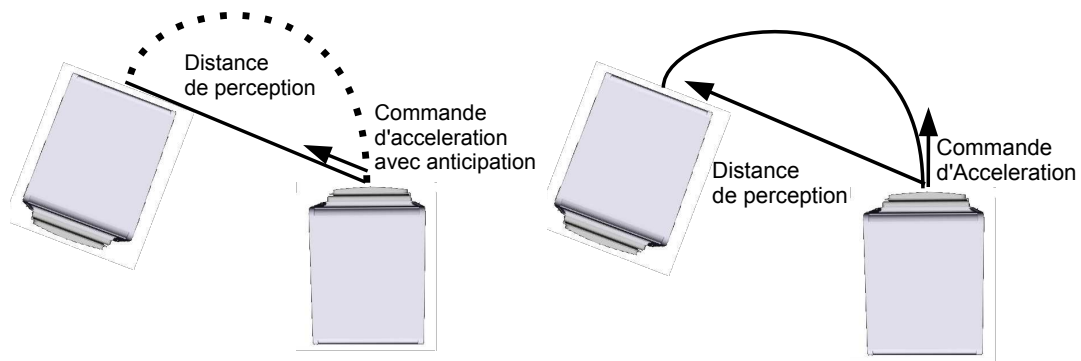


Figure 3.8: Contrôle local VS contrôle global

Comme le présente la figure 3.8, le contrôle local (cf. figure gauche) anticipe les commandes du véhicule dans les courbes ayant un rayon de courbure important. Tandis que le contrôle global permet un suivi précis de la trajectoire puisqu'il possède une représentation de l'environnement dans lequel le train de véhicules évolue.

### Contrôle latéral et longitudinal avec GPS

Le laboratoire LASMEA de Clermont-Ferrand a développé un système de contrôle de platooning [P. Martinet, 2006]. Ce contrôle est basé sur l'utilisation d'un GPS à haute précision dans chaque véhicule. Le véhicule de tête envoie la trajectoire à suivre aux véhicules suivants. Cette approche de platooning marche très bien mais la modification de l'infrastructure est très chère et implique la présence d'une communication fiable entre chaque véhicule.

### Système multi-agents pour la conduite en convoi

Un système multi-agents a été élaboré pour réaliser un contrôle longitudinal et latéral. Un véhicule est assimilé à un agent, lui attribuant un degré d'autonomie. Une approche a été proposée avec des agents intelligents [Bourbakis and Findler, 2001]. Elle consiste à proposer un indice de satisfaction altruisme [Lucidarme et al., 2002] aux agents en fonction de l'espace libre se situant autour d'eux.

Une thèse a été élaborée sur ce principe avec des agents intelligents [Halle and Chaib-draa, 2005]. **Simon Hallé** a construit un système multi-agents basé sur l'algorithme de **Daviet et Parent** [Daviet and Parent, 1996]. La gestion de platooning a été réalisée avec des agents communiquant. Deux types de communications ont été développés : soit le véhicule de "tête" est le maître de la simulation et donne les ordres aux différents agents, soit chaque agent connaît la formation globale du train de véhicules et établit des communications entre véhicules voisins.

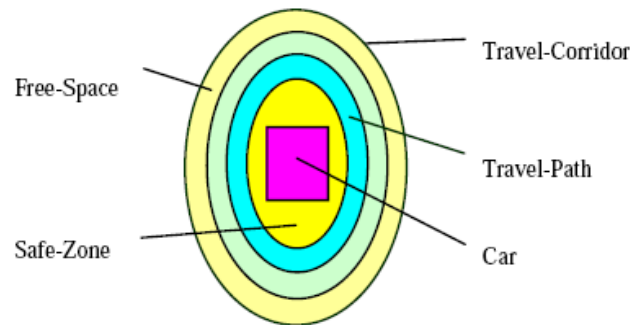


Figure 3.9: Caractérisation des espaces de satisfaction autour des véhicules.

L'algorithme de suivi de véhicule a été simulé et présente de bons résultats mais impose de disposer à tout moment d'une communication entre véhicule.

### 3.3.3 Bilan des méthodes pour la conduite en convoi

Deux grandes familles de contrôle de suivi de véhicule existent dans la littérature. Ces deux approches diffèrent du point de vue du contrôle.

L'approche locale ou "de proche en proche" est la première approche que nous avons présentée. Elle correspond au contrôle dont dispose chaque véhicule pour suivre le véhicule précédent avec uniquement sa perception locale. Ces algorithmes présentent de bons résultats mais souffrent d'une lacune dans le suivi en courbe.

La seconde approche, i.e. le contrôle global, correspond à un contrôle du train et non du véhicule. Chaque véhicule du train dispose d'une géolocalisation précise et d'une communication. Le premier véhicule envoie la trajectoire à suivre aux véhicules suivants. Cette accroche immatérielle présente de très bons résultats mais est soumise à de fortes contraintes matérielles (GPS, communication, ...).

## 3.4 Phases de transitions

Les phases de transitions correspondent à la possibilité donnée aux véhicules de faire des manœuvres de sortie et de rentrée dans le train de véhicules. Actuellement, les études réalisées permettent uniquement de s'insérer et de sortir d'un train de véhicules en ligne droite sur une voie rapide et en mouvement [Franke et al., 1995, Jeon and Choi, 2001, Jeon et al., 2001, Tan et al., 2000, Zhang et al., 1999].

Halle and Chaib-draa [2005] propose ainsi un contrôle longitudinal et latéral autonome avec un système multi-agents composé d'agents intelligents permettant d'ajouter ou d'éjecter un véhicule du train.

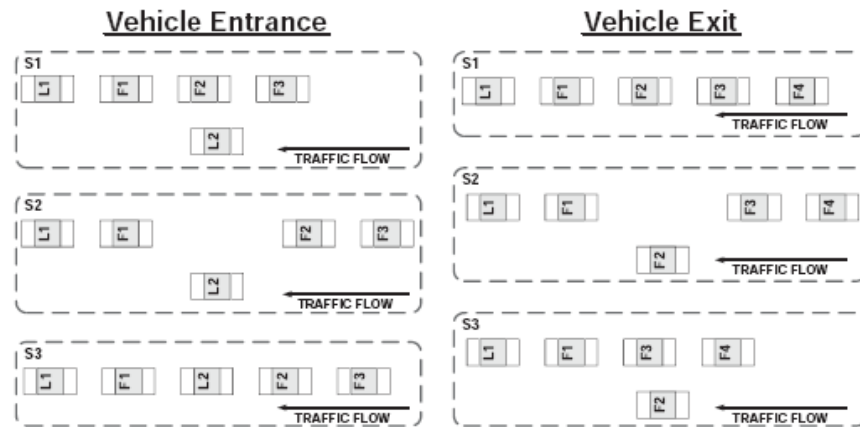


Figure 3.10: Phases de transitions selon Halle and Chaib-draa [2005]

**La sortie d'un véhicule (Split)** se produit dès lors qu'un membre du train de véhicules décide de sortir du peloton. Pour exécuter cette manœuvre, le véhicule souhaitant sortir doit communiquer son intention de quitter le peloton, ainsi le train modifie les distances à l'avant et l'arrière du véhicule sortant. Quand cette nouvelle formation est stable, le véhicule sortant peut changer de voie alors que le reste des véhicules gardent les mêmes distances. Le véhicule sort et l'espace créé par son départ peut être refermé.

**Un véhicule entrant dans le train (Merge)** est l'opposé exact d'une manœuvre d'insertion. Celle-ci exige un peloton constitué d'un unique véhicule, pour communiquer à un autre peloton sa volonté de se joindre. Le train réagit en créant un espace et en communiquant au véhicule entrant la position dynamique de cet espace. Le véhicule modifie sa vitesse pour rejoindre le point de réunion, vérifie s'il est possible d'entrer dans le train et change de voie. Une fois que le véhicule entrant a stabilisé sa distance inter-véhicule, le train est ainsi recomposé avec un véhicule en plus.

### 3.5 Conclusion

L'accroche immatérielle représente un point clé pour offrir un nouveau système de transport publique. Celle-ci consiste en le contrôle latéral et longitudinal de la distance inter-véhicule.

Deux principes sont possibles pour répondre à cette problématique. La solution du contrôle global est la plus performante mais souffre d'une forte demande de modification de l'infrastructure routière et de communication entre chaque véhicule. Tandis que l'approche locale offre une solution simple en vue de sa réalisation mais souffre d'une lacune pour le suivi en courbe.

Dans cette thèse, nous allons proposer un modèle réactif local pour la conduite en convoi

ayant des performances proches d'un contrôle global. On pourra ainsi proposer un algorithme simple et performant de suivi de véhicule sans modification de l'infrastructure routière et sans matériel contraignant au niveau de sa sûreté de fonctionnement comme une communication sans fil, une géolocalisation précise, ...

# MODÈLE MULTI-AGENTS RÉACTIFS POUR LA CONDUITE EN CONVOI

---

**C**E CHAPITRE présente l'approche que nous proposons pour la conduite en convoi ou platooning. Il s'agit d'une approche fondée sur les agents réactifs et un modèle d'interaction inspirée de la physique.

---

## Sommaire

---

<b>4.1</b>	<b>Modèle abstrait de comportement</b>	<b>56</b>
<b>4.2</b>	<b>Modèle d'interaction</b>	<b>57</b>
4.2.1	Mécanisme de perception	57
4.2.2	Construction du modèle physique d'interaction	58
4.2.3	Paramètres du modèle d'interaction	61
<b>4.3</b>	<b>Conclusion</b>	<b>63</b>

---



## 4.1 Modèle abstrait de comportement

Un convoi de véhicules est composé d'un véhicule leader et de véhicules suiveurs. Le leader suit un itinéraire prédéfini soit de façon autonome soit piloté tandis que chaque véhicule suiveur est en accroche immatérielle avec le véhicule qui le précède. De nouveaux véhicules peuvent s'insérer dans le convoi et un véhicule qui en fait partie peut s'en éjecter.

Nous avons décidé de considérer chaque véhicule comme un agent qui règle son comportement au sein du convoi de façon autonome, à partir de sa perception de l'environnement. Chaque agent véhicule réglera son comportement suivant sa situation (hors du train, inséré dans le train, en phase d'insertion ou d'éjection).

Dans le convoi de véhicules, on identifie deux types d'agents : leader (*Head\_Vehicle*) et suiveur (*Follower\_Vehicle*). Le comportement de l'agent *Head\_Vehicle* correspond au comportement du véhicule de tête, permettant à celui-ci de suivre une trajectoire. Le second type d'agent *Follower\_Vehicle* intègre le comportement du véhicule suiveur.

### Modèle de comportement du véhicule suiveur

La figure 4.1 présente le modèle abstrait du comportement du véhicule suiveur. Un véhicule hors du train (cf. état *IDLE* du statechart 9.1) peut s'insérer dès la réception de l'évènement *Vmerge*. Une fois inséré (évènement *end\_merge*) le comportement de l'agent correspond à celui d'un véhicule dans la formation (état *INSERTED*). De la même façon, l'éjection se produit après la réception de l'évènement *Vsplit*.

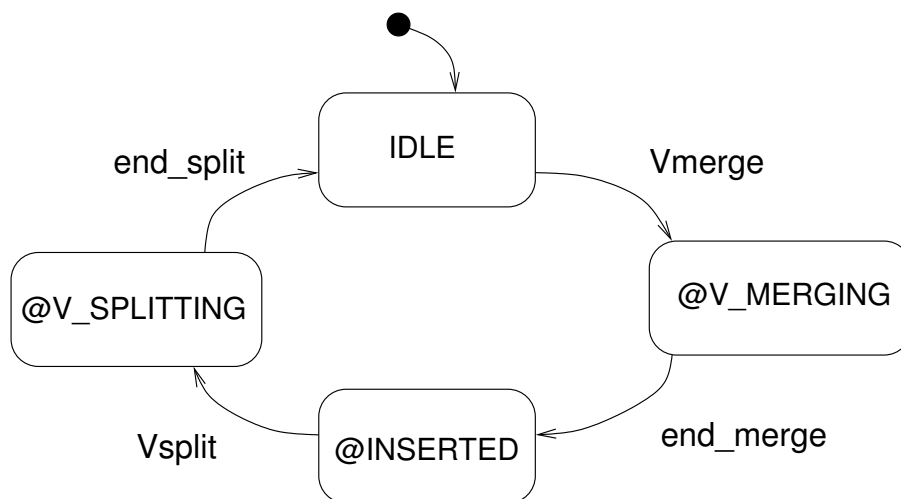


Figure 4.1: Modèle de comportement du véhicule suiveur

Chaque état de la figure 4.1 représente un type de comportement du véhicule, modélisé par un sous-statechart :

- *IDLE* : véhicule hors du train.

- *V\_MERGING* : véhicule en phase d'insertion.
- *INSERTED* : véhicule inséré dans le train.
- *V\_SPLITTING* : véhicule en phase de sortie.

Les comportements correspondant à l'état *INSERTED* sont détaillés dans la spécification présentée au chapitre 9, à partir de la spécification physique des mécanismes d'interaction.

## 4.2 Modèle d'interaction

### 4.2.1 Mécanisme de perception

Les véhicules suiveurs perçoivent le véhicule précédent dans le train ainsi que tous les obstacles de l'environnement qui sont dans sa zone de perception. Cette perception est dite locale. Les capteurs installés sur un véhicule permettent de mesurer la distance le séparant du véhicule précédent.

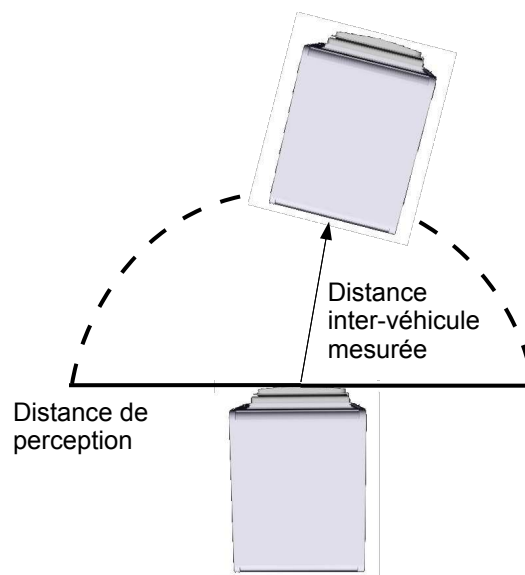


Figure 4.2: Zone de perception des véhicules

La figure 4.2 montre la zone de perception d'un véhicule muni d'un capteur de type télémètre laser. Ce type de capteur émet des rayons laser dans l'infrarouge et capte la lumière réfléchiée par les objets qui se trouvent dans sa portée. La distance entre le capteur et un objet est calculée en mesurant le temps écoulé entre l'émission d'un rayon et la réception de son reflet (cf. chapitre 10.3.2). Ce mécanisme produit les percepts de l'agent : la distance qui le sépare du véhicule précédent et l'angle de courbure.

## 4.2.2 Construction du modèle physique d'interaction

L'accroche immatérielle entre deux véhicules est réalisée grâce à un lien virtuel de type "masse ressort". Ce système correspond au couplage en parallèle d'un ressort et d'un amortisseur (cf. figure 4.3).

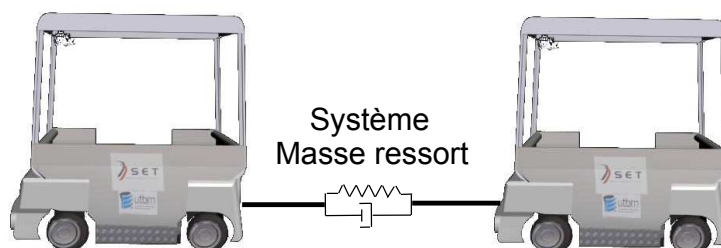


Figure 4.3: Lien virtuel entre véhicules

Chaque véhicule calcule sa consigne de vitesse et de direction à partir du lien virtuel "masse ressort". Cependant, comme le montre la figure 4.4, dans un virage le système "masse ressort" attire le véhicule vers l'intérieur du virage et provoque une erreur dite erreur d'anticipation [Contet et al., 2006].

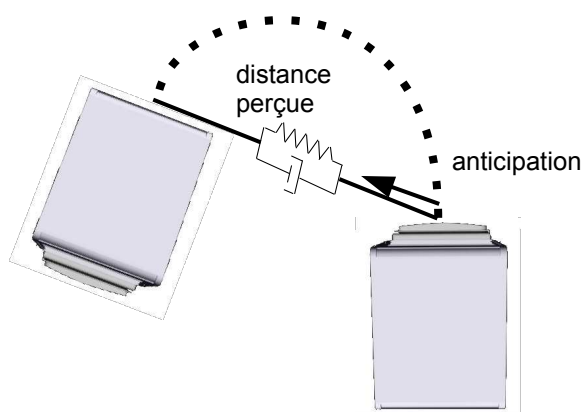


Figure 4.4: Anticipation du contrôle de suivi en courbe

Pour corriger cette erreur, on introduit un mécanisme additionnel qui consiste à courber le système "masse ressort", comme le montre la figure 4.5. Pour cela, le modèle physique du lien virtuel "masse ressort" est complété par un moment de torsion [Contet et al., 2009].

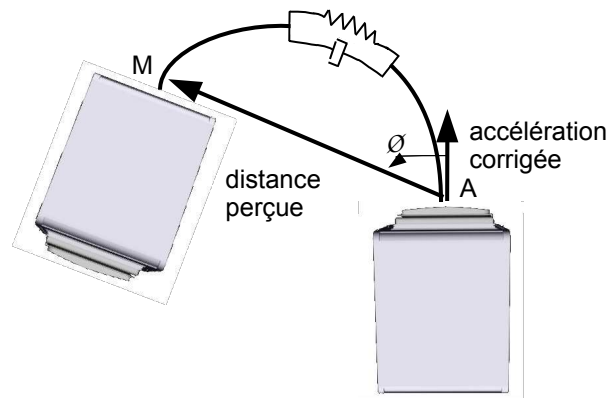


Figure 4.5: Courbure du ressort

Le modèle d'interaction de chaque agent véhicule repose sur deux groupes de forces :

- Les forces “masse-ressort” réalisant le lien mécanique virtuel d’une force ressort ( $\vec{F}_r$ ) et d’une force d’amortissement ( $\vec{F}_a$ ).
- La force de torsion permettant la courbure du système “masse-ressort” ( $\vec{F}_t$ ).

Le système “masse-ressort” est composé d’un ressort de raideur  $k$ , de longueur au repos  $l_v$  et d’un amortisseur de coefficient  $h$ . La courbure du ressort est prise en compte par l’ajout d’une force résultante d’un moment de torsion.

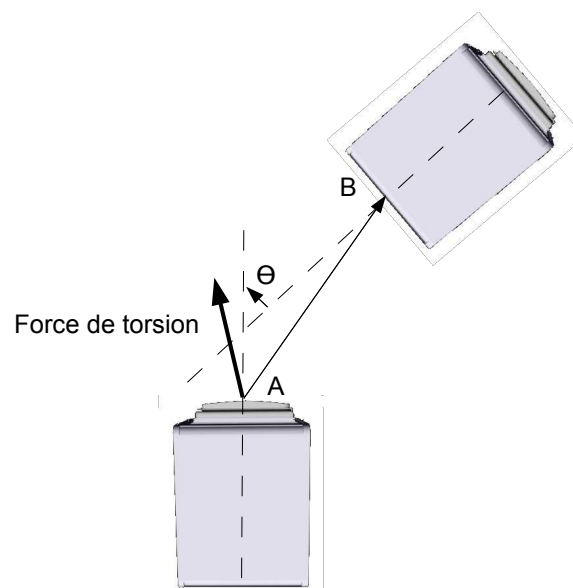


Figure 4.6: Force de torsion

Le moment de torsion  $\vec{M}_A$  correspond à un système “masse-ressort” composé d’un ressort de raideur de torsion  $k_t$  et d’un amortisseur de torsion de coefficient  $h_t$ . Le moment est appliqué en  $A$  et dépend de l’angle de torsion  $\theta$  représentant l’angle entre les axes de symétrie des deux véhicules :

$$\vec{M}_A = \begin{pmatrix} 0 \\ 0 \\ k_t\theta - h_t\dot{\theta} \end{pmatrix} \quad (4.1)$$

Pour décrire les forces du modèle d’interaction, nous utilisons la représentation suivante : la figure 4.7 montre 3 véhicules en convoi. Chaque véhicule  $n$  est représenté par sa position  $X_n$ . La masse des véhicules est notée  $m$  et la longueur du ressort est donnée par  $P_{n,n-1} = \|\vec{X}_{n-1} - \vec{X}_n\| - l_v$ .

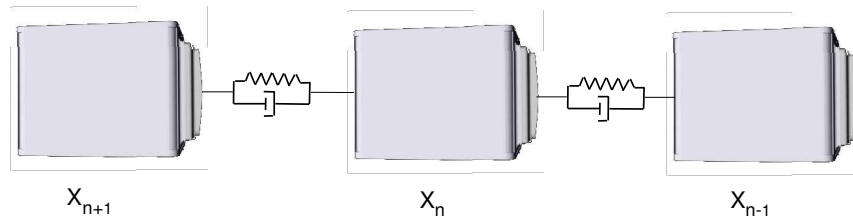


Figure 4.7: Représentation d’un train de véhicules

Chaque agent véhicule  $n$  est soumis aux forces suivantes :

– Force du ressort  $\vec{F}_r^n$  :

$$\vec{F}_r^n = -k P_{n,n-1} \vec{u}_{n,n-1} \quad (4.2)$$

– Force de l’amortisseur  $\vec{F}_a^n$  :

$$\vec{F}_a^n = -h(\dot{\vec{X}}_{n-1} - \dot{\vec{X}}_n) \quad (4.3)$$

– Force de torsion  $\vec{F}_t^n$  :

$$\vec{F}_t^n = \vec{B}A \wedge \vec{M}_A \quad (4.4)$$

A partir des forces simulant l’accroche immatérielle, nous pouvons déduire du principe fondamental de la dynamique, une consigne d’accélération  $\vec{\gamma}^n$  :

$$\begin{cases} m \vec{\gamma}^n = \vec{F}_r^n + \vec{F}_a^n + \vec{F}_t^n \\ m \vec{\gamma}^n = -k(\|\vec{X}_{n-1} - \vec{X}_n\| - l_v)\vec{u}_{n,n-1} \\ -h(\dot{\vec{X}}_{n-1} - \dot{\vec{X}}_n) + -\lambda \dot{\vec{X}}_n + \vec{B}A \wedge \vec{M}_A \end{cases} \quad (4.5)$$

La consigne de vitesse du véhicule  $n$  ( $\|\vec{V}(t)^n\|$ ) et de direction (angle de  $\vec{V}(t)^n$ ) est calculée par une intégration discrète, de pas  $\delta t$ , de l'accélération (cf équation 4.6).

$$\vec{V}(t)^n = \frac{\delta t}{m}(\vec{F}_r^n + \vec{F}_a^n + \vec{F}_t^n) \quad (4.6)$$

En fonction de la distance et de l'angle de courbure entre les deux véhicules  $A_n$  et  $A_{n+1}$ , une force  $\vec{F}$  résultante de la somme de la force du ressort, de l'amortisseur et de la force de torsion est appliquée au véhicule  $A_n$ . L'agent véhicule  $A_n$  élabore alors sa décision (vitesse, direction) en appliquant simplement le principe fondamental de la dynamique (cf. figure 4.8).

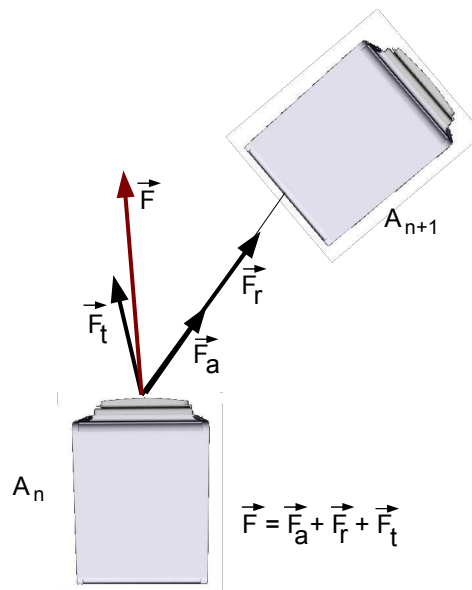


Figure 4.8: Convoi de deux véhicules

Pour compléter le modèle d'interaction, les valeurs des différents paramètres doivent être définies.

### 4.2.3 Paramètres du modèle d'interaction

Cette section expose la démarche pour calculer la valeur des différents paramètres du modèle d'interaction en fonction des caractéristiques d'un véhicule donné. Nous prenons le véhicule  $n$  comme référence (cf. figure 4.7).  $\vec{Y}$  représente ainsi la distance inter-véhicule entre le véhicule  $n$  et  $n - 1$  tandis que  $\dot{\vec{Y}}$  la différence instantanée de vitesse entre les véhicules  $n$  et  $n - 1$ . Le modèle d'interaction introduit six paramètres : la masse  $m$  du véhicule suiveur, la raideur  $k$ , l'amortissement  $h$ , la longueur au repos  $l_v$ , la raideur de torsion  $k_t$  et l'amortissement de torsion  $h_t$  (cf. annexes C.1). On calcule la valeur de chaque paramètre en fonction des caractéristiques dynamiques du véhicule :

- Afin de calculer la valeur des paramètres  $k$  et  $h$ , nous supposons que le train de véhicules évolue en ligne droite ( $\theta = 0$  et  $\dot{\theta} = 0$ ).
- Le paramètre  $k$  représente la raideur du ressort et par conséquent la capacité du véhicule suiveur à changer rapidement la consigne de vitesse pour réguler la distance inter-véhicule. Afin d’obtenir un asservissement robuste, la valeur du paramètre de raideur doit être une fonction de la différence de vitesse  $\vec{Y}$ . Pour obtenir une relation entre la raideur et la vitesse, une étude énergétique simple est réalisée :

$$E_{smr} = E_p + E_c = \frac{1}{2} k (\vec{Y} \cdot \vec{Y}) + \frac{1}{2} m (\dot{\vec{Y}} \cdot \dot{\vec{Y}}) \quad (4.7)$$

où  $E_{smr}$  représente l’énergie du système “masse ressort”,  $E_p$  l’énergie potentielle et  $E_c$  l’énergie cinétique. Cette énergie représente le travail des forces du lien “masse-ressort”. La condition  $\Delta E = 0$  correspond à un équilibre entre l’énergie cinétique et potentielle. Elle traduit une stabilité du système “masse-ressort”. Cette stabilité permet de déduire une valeur pour la raideur du ressort en fonction de la différence de vitesse  $\dot{Y}$  :

$$k = m \frac{\dot{\vec{Y}} \cdot \dot{\vec{Y}}}{\vec{Y} \cdot \vec{Y}} \quad (4.8)$$

- Le deuxième paramètre  $h$  du système “masse-ressort” est déduit d’une étude cinématique. Grâce au principe fondamental de la dynamique, nous établissons l’équation différentielle suivante :

$$\ddot{Y}(t) + 2 \varepsilon \dot{Y}(t) + \omega_0^2 Y(t) = 0 \quad (4.9)$$

Avec :  $2 \varepsilon = \frac{h}{m}$  et  $\omega_0 = \sqrt{\frac{k}{m}}$

En fonction de la valeur du discriminant ( $\Delta = \omega_0^2 (\varepsilon^2 - 1)$ ), on en déduit différent cas d’amortissement : critique, périodique et pseudo périodique . L’amortissement critique a été choisi afin d’obtenir un amortissement rapide, i.e sans dépassement de consigne et non périodique (sans oscillation). L’amortissement critique ( $\Delta = 0 \Rightarrow \varepsilon = 1$ ) permet de déduire une relation entre la raideur  $k$  et l’amortissement  $h$  :

$$\varepsilon = \frac{h}{2 \sqrt{k m}} = 1 \quad (4.10)$$

Connaissant le paramètre  $k$ , l’amortissement est défini par l’équation suivante :

$$h = 2 \sqrt{k m} \quad (4.11)$$

- La masse  $m$  de chaque agent correspond à la masse du véhicule.
- Le dernier paramètre du modèle “masse-ressort” est la longueur au repos  $l_v$ . Ce paramètre correspond à la distance de repos du système “masse-ressort”. Par conséquent, ce paramètre doit être initialisé à la distance inter-véhicule souhaitée.

$$l_v = \text{distance intervehicule souhaitee} \quad (4.12)$$

- Les deux derniers paramètres  $k_t$  et  $h_t$  sont liés au moment de torsion appliqué au système “masse-ressort”. La raideur de torsion  $k_t$  est exprimée aussi grâce à une étude énergétique :

$$E = E_p + E_c = \frac{1}{2} k (\theta^2) + \frac{1}{2} J (\dot{\theta}^2) \quad (4.13)$$

où  $J$  est le moment d’inertie.

La stabilité est obtenue en considérant un système conservatif ( $E = 0$ ), qui exprime la stabilité du système de torsion et permet de déduire la valeur de la raideur de torsion  $k_t$  :

$$k_t = \frac{J \dot{\theta}^2}{\theta^2} \quad (4.14)$$

Le théorème du moment cinétique permet de relier les deux paramètres  $k_t$  et  $h_t$  :

$$\vec{M}_S = \vec{M}_A = \begin{pmatrix} 0 \\ 0 \\ k_t \theta - h_t \dot{\theta} \end{pmatrix} \quad (4.15)$$

avec  $\vec{M}_S$  le moment du système.

Dans notre cas, la force est dite “centrale” (son point d’application coïncide avec le centre du moment) et permet d’écrire la loi suivante :

$$k_t \theta - h_t \dot{\theta} = 0 \quad (4.16)$$

### 4.3 Conclusion

L’accroche immatérielle est réalisée par un lien virtuel de type “masse-ressort” entre chaque véhicule. Ce contrôle est accompli par le modèle d’interaction de chaque agent véhicule suiveur. En fonction de la distance qui le sépare de son prédécesseur et de l’angle de courbure entre eux, chaque véhicule élabore ainsi sa décision (vitesse, angle). Le modèle est basé sur plusieurs paramètres calculés en fonction des propriétés dynamiques du véhicule intelligent (cf. tableau 4.9).



Paramètres	Valeur
Raideur du ressort	$k = \frac{m \dot{Y}^2}{Y^2}$
Amortisseur	$h = 2 \sqrt{k m}$
Raideur de torsion	$k_t = \frac{J \dot{\theta}^2}{\theta^2}$
Amortisseur de torsion	$h_t = \frac{\theta}{k_t \theta}$

Figure 4.9: Paramètres du modèle d'interaction de suivi de véhicule

Le chapitre suivant analyse le modèle d'interaction que nous venons de définir pour d'une part vérifier d'un point de vue purement physique la stabilité du système, évaluer la réponse temporelle et d'autre part formuler les conditions pour la vérification de la sûreté de fonctionnement.

# ANALYSE DU MODÈLE PHYSIQUE D'INTERACTIONS POUR LA CONDUITE EN CONVOI

---

CE CHAPITRE propose, en premier lieu, de réaliser une étude cinématique du modèle d'interaction. Ceci permettra de visualiser la réponse du modèle et de calculer ses limites de fonctionnement. Ces limites serviront à formuler les conditions de sûreté pour la vérification. Puis, des preuves classiques de stabilité inspirées de la physique et de l'automatique permettront de prouver la stabilité du modèle défini pour la conduite en convoi.

---

## Sommaire

---

<b>5.1</b>	<b>Analyse du modèle d'interaction</b>	<b>66</b>
5.1.1	Etude cinématique du modèle d'interaction : cas sans excitation	66
5.1.2	Etude cinématique du modèle d'interaction : cas avec excitation	69
5.1.3	Conditions pour la vérification de la sûreté	70
<b>5.2</b>	<b>Preuves de stabilité du train de véhicules</b>	<b>71</b>
5.2.1	Preuve de la stabilité d'inspiration physique	71
5.2.2	Preuve de la stabilité inspirée de l'automatique	72
<b>5.3</b>	<b>Conclusion</b>	<b>74</b>

---

## 5.1 Analyse du modèle d'interaction

L'analyse du modèle d'interaction (cf. section 4.2) fera abstraction de la force de torsion et se cantonnera au cas d'un mouvement en ligne droite (cf. figure 5.1). Néanmoins, une étude plus approfondie pourra être réalisée, par la suite, afin de prendre en compte cette force dans l'analyse du modèle d'interaction relativement à l'aspect distance inter-véhicule.

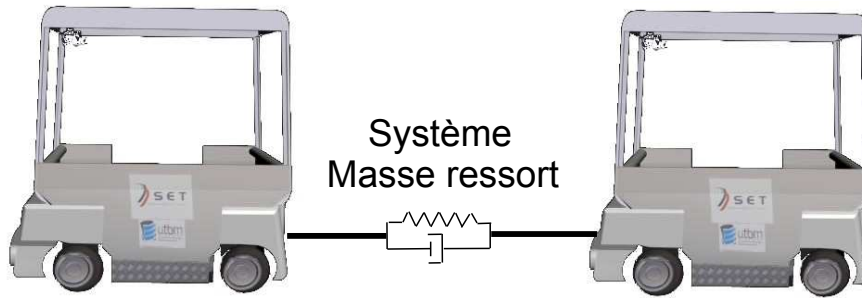


Figure 5.1: Modèle d'interaction dans le cas d'un mouvement rectiligne

### 5.1.1 Etude cinématique du modèle d'interaction : cas sans excitation

A partir du mécanisme “masse-ressort” en appliquant le principe fondamental de la dynamique, on obtient des solutions représentant les réponses possibles du système. La distance de suivi sera notée  $\vec{Y}$  ( $\|\vec{Y}\| = \|\vec{d}\| - l_v$ ) avec  $\vec{d}$  la distance inter-véhicule et  $l_v$  la distance inter-véhicule imposée. Dans ces conditions, la vitesse relative est notée par  $\dot{\vec{Y}}$ . Les forces peuvent être calculées comme suit :

- Force du ressort  $\vec{F}_r$  :

$$\vec{F}_r = -k \vec{Y} \quad (5.1)$$

- Force de l'amortisseur  $\vec{F}_a$  :

$$\vec{F}_a = -h \dot{\vec{Y}} \quad (5.2)$$

D'après le principe fondamental de la dynamique, on définit l'équation différentielle suivante :

$$m * \ddot{\vec{Y}} = -k \vec{Y} - h \dot{\vec{Y}} \quad (5.3)$$

La solution de cette équation est bien connue et s'exprime facilement en adoptant la notation suivante :

$$\ddot{\vec{Y}}(t) + 2 \varepsilon \dot{\vec{Y}}(t) + \omega_0^2 \vec{Y}(t) = \vec{0} \quad (5.4)$$

avec  $2\varepsilon = \frac{h}{m}$ ,  $\omega_0 = \sqrt{\frac{k}{m}}$ . L'équation différentielle est à coefficients constant, la solution dépend de la valeur du coefficient d'amortissement :

$$\Delta = \omega_0^2 (\varepsilon^2 - 1) \quad (5.5)$$

Comme le précise le modèle d'interaction, afin d'obtenir le meilleur suivi, le choix d'un amortissement critique ( $\Delta = 0$ ) a été retenu. Une solution de l'équation différentielle peut être exprimée à partir de cette condition :

$$Y(t) = (A t + B) \exp(-\omega_0 t) \quad (5.6)$$

$Y(t)$  est l'équation de l'évolution de la distance inter-véhicule, avec  $A$  et  $B$  les constantes déduites des conditions initiales :

$$\begin{cases} B = d_i - l_v \\ A = \Delta v_i + \omega_0 B \end{cases} \quad (5.7)$$

avec  $d_i$  la distance inter-véhicule initiale et  $\Delta v_i$  la différence de vitesse initiale entre les véhicules  $n$  et  $n - 1$ . Afin de visualiser l'évolution de la distance inter-véhicule, la valeur des paramètres a été définie d'après le tableau ci-dessous :

Masse	500 kg
Vitesse maximale	8 km/h (2 m/s)
Accélération	0.1 m/s <sup>2</sup>
Décélération	-0.1 m/s <sup>2</sup>
Décélération maximum	-3 m/s <sup>2</sup>
Distance de sécurité ( $d_s$ )	50 cm
Distance maximum de suivi ( $d_{max}$ )	200 cm

Figure 5.2: Paramètres dynamique du véhicule et du modèle d'interaction

Suivant les paramètres définis dans le tableau 5.2, on peut tracer l'évolution de la distance inter-véhicule. Elle permet de visualiser la réponse du système et de calculer ses limites. Pour cela, deux cas ont été choisis représentant les deux situations critiques durant l'évolution du véhicule :

- Distance inter-véhicule égale à la distance de sécurité (distance inter-véhicule permettant de garantir une absence de collision, appelée distance de sécurité et notée  $d_s$ ) : ce cas représente la situation où le véhicule redémarre après un arrêt d'urgence avec une différence de vitesse maximum.
- Distance inter-véhicule supérieur à la distance inter-véhicule souhaitée ( $l_v$ ) : cette situation correspond au cas où le véhicule suiveur s'éloigne avec une différence de vitesse maximum.

Ces situations ont été choisies afin d'illustrer la réaction du système "masse ressort" dans des cas critiques de fonctionnement.

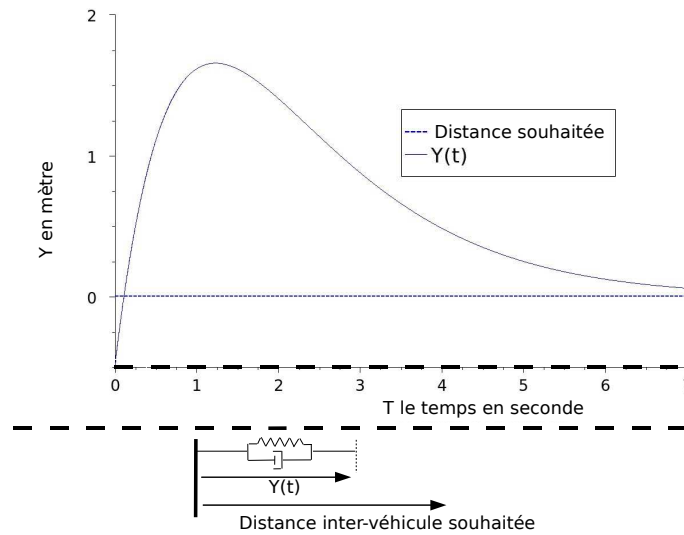


Figure 5.3: Réponse dynamique du système "masse ressort" par rapport à une distance en dessous de la distance inter-véhicule souhaitée

La figure 5.3 présente la réponse du système "masse-ressort" par rapport à une distance inter-véhicule en dessous de la distance souhaitée. Dans ce cas, le système "masse-ressort" rétablit la distance inter-véhicule à zéro ( $Y(t) = 0$  signifie que le distance inter-véhicule est égale à la distance souhaitée :  $l_v$ ).

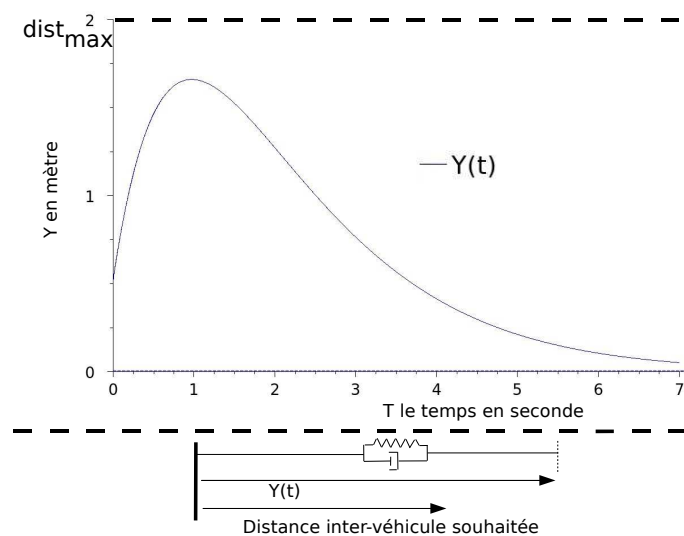


Figure 5.4: Réponse dynamique du système "masse ressort" par rapport à une distance au dessus de la distance inter-véhicule souhaitée

La figure 5.4 présente la réponse du modèle d'interaction lorsque les conditions initiales représentent un dépassement de la distance inter-véhicule de 50% (50 cm au dessus de la distance de sécurité) ainsi qu'une différence de vitesse maximum entre les deux véhicules. Dans ce cas, le contrôle rétablit la distance inter-véhicule souhaitée sans dépassement de la distance maximum de suivi ( $d_{Max}$ ).

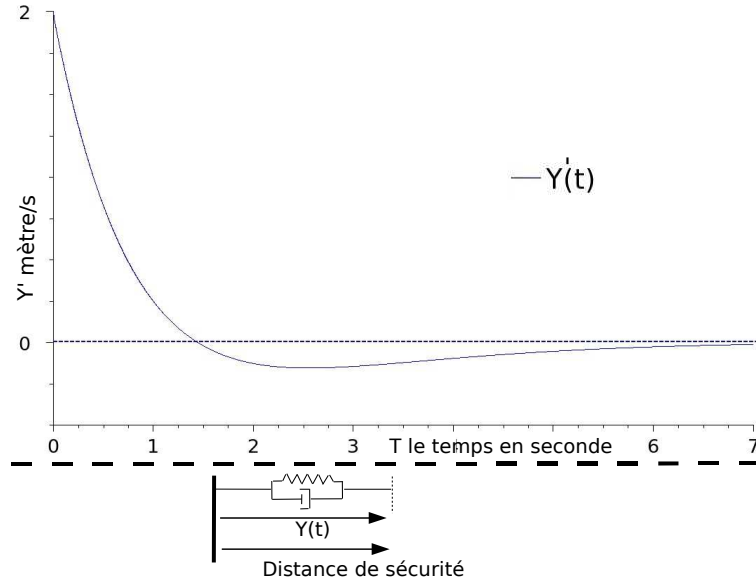


Figure 5.5: Réponse dynamique du système "masse ressort" par rapport à une distance égale à la distance de sécurité

La figure 5.5 montre l'évolution de la différence de vitesse entre les deux véhicules dans le cas où la distance inter-véhicule est égale à la distance de sécurité. Cette figure permet de définir la variation minimale ( $\Delta s_i_{MIN} = -0.25 \text{ m/s}$ ) et maximale de la vitesse inter-véhicule ( $\Delta s_i_{MAX} = 2 \text{ m/s}$ ).

### 5.1.2 Etude cinématique du modèle d'interaction : cas avec excitation

Ce paragraphe propose d'étudier la réaction du modèle d'interaction par rapport à une excitation de type "échelon". Cette excitation consiste à allonger, de façon instantanée, le système "masse-ressort" d'une longueur  $X_a$ . Dans ce cas, l'équation différentielle dictant la réaction du système devient :

$$\ddot{Y}(t) + 2\varepsilon \dot{Y}(t) + \omega_0^2 Y(t) = \omega_0 X_a \quad (5.8)$$

Avec  $2\varepsilon = \frac{h}{m}$ ,  $\omega_0 = \sqrt{\frac{k}{m}}$  et  $X_a$  l'allongement. La solution de cette équation peut être mise sous la forme suivante :

$$Y(t) = (A * t + B) * \exp(-\omega_0 * t) + \omega_0 * X_a \quad (5.9)$$

$Y(t)$  représente l'évolution de la distance inter-véhicule, avec  $A$  et  $B$  des constantes déduites des conditions initiales :

$$\begin{cases} B = d_i - l_v + X_a \\ A = \Delta s_i - \omega_0 B - X_a \end{cases} \quad (5.10)$$

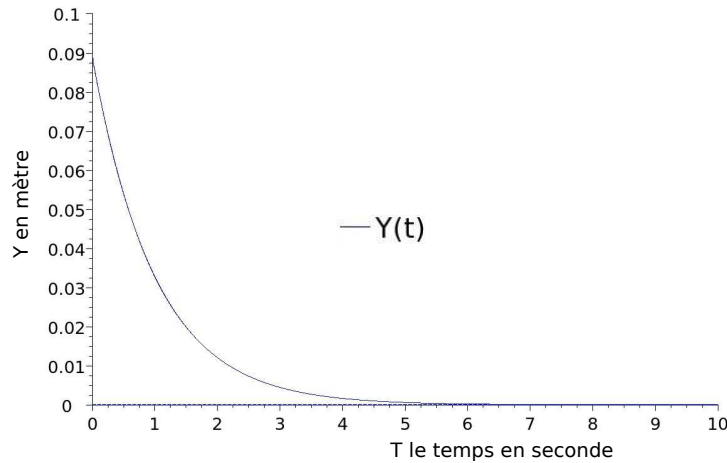


Figure 5.6: Réponse dynamique du système “masse ressort” par rapport à une excitation

La figure 5.1.2 présente la variation de la distance dans le cas d'une excitation (augmentation rapide de la distance inter-véhicule). On visualise la correction effectuée par le modèle d'interaction. On constate la correction rapide de cette distance.

### 5.1.3 Conditions pour la vérification de la sûreté

L'étude cinématique précédente permet de déduire des propriétés du modèle de suivi de véhicule. En effet, les figures 5.3 et 5.4 montrent que dans les deux cas critiques de suivi (distance inter-véhicule égale à la distance de sécurité et à la distance maximal de suivi) que le modèle d'interaction réagit correctement en stabilisant la distance inter-véhicule à la distance inter-véhicule souhaitée ( $l_v$ ). Ces figures permettent de visualiser les limites de l'évolution de la distance inter-véhicule ( $d_s$  et  $d_{Max}$ ) :

$$d_s < Y < d_{Max} \quad (5.11)$$

La figure 5.5 montre la réaction en vitesse relative qui résulte du modèle d'interaction dans un cas critique. Cette variation de la vitesse permet de constater que la réponse du système en vitesse relative est comprise entre la vitesse minimale ( $\Delta_{MIN}$ ) et maximale ( $\Delta_{MAX}$ ). Ceci permet d'exprimer la condition suivante :

$$\Delta_{MIN} < \Delta s_i < \Delta_{MAX} \quad (5.12)$$

L'étude que nous venons de faire porte uniquement sur l'interaction entre deux véhicules. Les deux sections suivantes proposent des preuves d'inspirations physique et automatique sur le train de véhicules (cf. figure 5.7).



Figure 5.7: Etude globale du modèle d'interaction

## 5.2 Preuves de stabilité du train de véhicules

Nous allons faire cette étude suivant deux critères de stabilité. Le premier critère consiste à établir qu'en l'absence de perturbations extérieures le système converge vers un état stable. Le second qu'une perturbation affectant un composant du système ne s'amplifie pas en se propageant vers les autres composants. La première preuve suivra une approche d'inspiration physique tandis que la seconde adoptera une approche automatique.

### 5.2.1 Preuve de la stabilité d'inspiration physique

La preuve consiste à modéliser un train de véhicules comme un système physique. Pour cela, chaque véhicule est considéré comme une particule de masse  $m$  soumise à plusieurs forces. Dans notre cas, chaque véhicule est soumis aux forces du système "masse ressort" qui le sépare du véhicule précédent. Il est important de rappeler que dans notre approche aucun véhicule n'est influencé par le système "masse ressort" du véhicule qui le suit dans le train.

La preuve de stabilité d'un système de particules soumis à plusieurs forces peut être effectuée à partir d'une étude énergétique. L'énergie d'un système est une grandeur représentant l'état du système global. Ainsi, l'étude des variations de l'énergie permet de mesurer l'état d'équilibre du système.

Pour chaque véhicule, l'énergie peut être défini comme la somme des énergies potentielle et cinétique :

$$E = E_p + E_c \quad (5.13)$$

avec

$$\begin{cases} E_p = \frac{1}{2} k (\vec{Y} \cdot \vec{Y}) \\ E_c = \frac{1}{2} m (\dot{\vec{Y}} \cdot \dot{\vec{Y}}) \end{cases} \quad (5.14)$$



Le théorème de *Lyapunov* [Leipholz, 1987, Lyapunov, 1992] établi que le système est stable si les conditions suivantes sont respectées, avec  $V(x, y) = E = E_p + E_c$  :

$$\begin{cases} V(x, y, \theta) \geq 0 & (a) \\ \dot{V}(x, y, \theta) < 0 & (b) \end{cases} \quad (5.15)$$

La condition (a) est respectée puisque  $\|\vec{Y}\|^2 \geq 0$  et  $\|\dot{\vec{Y}}\|^2 \geq 0$ . Pour vérifier la condition (b), il faut exprimer l'équation 4.5 d'une autre façon :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = -h \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} - k \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.16)$$

supposons que :

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \text{ et } \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.17)$$

on peut ainsi réécrire l'équation 4.5 :

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = -\frac{k}{m} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \frac{h}{m} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (5.18)$$

avec

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (5.19)$$

La dérivée de l'énergie peut ainsi s'exprimer :

$$\begin{bmatrix} \frac{\partial E(x,y,\theta)}{\partial x} \\ \frac{\partial E(x,y,\theta)}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial x_1} + \frac{\partial E}{\partial x_2} \\ \frac{\partial E}{\partial x_1} + \frac{\partial E}{\partial x_2} \end{bmatrix} \quad (5.20)$$

d'où

$$\begin{bmatrix} \frac{\partial E(x,y,\theta)}{\partial x} \\ \frac{\partial E(x,y,\theta)}{\partial y} \end{bmatrix} = \begin{bmatrix} -x_2^2 h \\ -y_2^2 h \end{bmatrix} \quad (5.21)$$

La dérivée de l'énergie étant toujours négative puisque  $h$  est positif, les conditions de *Lyapunov* sont vérifiées. Cette étude énergétique [Contet et al., 2006, Gaud et al., 2007] permet d'affirmer que le système tend vers un état stable lorsque le système évolue sans perturbation externe.

### 5.2.2 Preuve de la stabilité inspirée de l'automatique

La preuve de stabilité d'un train de véhicules correspond à un problème classique désigné par "string stability". Ce problème consiste à mesurer l'évolution d'un système composé d'entités liés par des ressorts. Cette problématique a été traité par Garrard and Caudill

[1977]. La stabilité au sens automatique se définit par : *sans aucunes excitations extérieurs du système, l'amplitude d'une erreur décroît lorsqu'elle se propage le long du train de véhicules*. Dans notre cas, la preuve de stabilité est appliquée au contrôle longitudinal (dans le cas d'un mouvement rectiligne). La preuve consiste à exprimer la fonction de transfert du système d'un couple de deux véhicules afin de vérifier [Liang and Peng, 1998, Swaroop and Hedrick, Mar 1996] si l'amplitude est inférieur ou égale à 1.

Pour prendre en compte la réaction dynamique, on suppose que le contrôle du véhicule est réalisé par une boucle de régulation en vitesse du second ordre. La réponse dynamique en accélération peut être ainsi exprimée :  $\dot{\vec{a}}_i = \frac{1}{\tau}(\ddot{Y}_i - \vec{a}_i)$  avec  $\vec{a}_i = \ddot{Y}_i$  l'accélération du véhicule et  $\tau$  la constante de temps du système. La consigne d'accélération de deux véhicules  $i$  et  $i + 1$  correspond à :

$$\begin{cases} m \dot{\vec{a}}_i = \frac{1}{\tau}[-k \vec{Y}_i - h \dot{\vec{Y}}_i - \ddot{Y}_i] \\ m \dot{\vec{a}}_{i-1} = \frac{1}{\tau}[-k \vec{Y}_{i-1} - h \dot{\vec{Y}}_{i-1} - \ddot{Y}_{i-1}] \end{cases} \quad (5.22)$$

On peut déduire la fonction de transfert suivante :

$$\frac{\vec{Y}(s)_i}{\vec{Y}(s)_{i-1}} = \frac{h s + k}{\tau s^3 + s^2 + h s + k} \quad (5.23)$$

La condition de stabilité est exprimée comme suit avec  $s = j\omega$

$$A(j\omega) = \left| \frac{\vec{Y}(j\omega)_i}{\vec{Y}(j\omega)_{i-1}} \right| \leq 1 \quad (5.24)$$

Cette condition est respectée pour :

$$0 \leq \omega \leq 44 \text{ rad/s}^2 \quad (5.25)$$

avec  $k=400$ ,  $h=100$ , on obtient plusieurs fréquences limites de stabilités :

$\tau$	Limites de stabilités	limites fréquentielles
0.01 s	$0 \leq \omega \leq 103 \text{ rad/s}^2$	$f \leq 16 \text{ Hz}$
0.1 s	$0 \leq \omega \leq 44 \text{ rad/s}^2$	$f \leq 7 \text{ Hz}$
1 s	$0 \leq \omega \leq 14 \text{ rad/s}^2$	$f \leq 2 \text{ Hz}$

Figure 5.8: Fréquences limites de stabilité du système

Le tableau 5.8 expose suivant la constante de temps du système ( $\tau$ ) la fréquence limite du système pour assurer une stabilité du train. Suivant la rapidité de réaction du système, la stabilité est assurée si et seulement si la fréquence d'opération du contrôleur respecte ces limites. On peut ainsi dire que la stabilité du train de véhicules est garantie sous certaines conditions.

## 5.3 Conclusion

Nous connaissons maintenant les conditions pour un bon fonctionnement du mécanisme d'interaction "masse ressort". Nous pouvons construire un modèle de spécification d'un agent véhicule pour qu'il se conforme à ces conditions.

Le chapitre suivant propose une solution, au problème d'évitement d'obstacle, à partir d'un modèle multi-agents réactifs inspiré de la physique.

# MODÈLE MULTI-AGENTS RÉACTIFS POUR LA NAVIGATION AUTONOME

---

**T**OUT SYSTÈME de navigation autonome doit comporter une fonction d'évitement d'obstacle : sur tout itinéraire allant d'un point de départ à un point d'arrivé, peuvent se présenter des éléments imprévus qu'il faut essayer de contourner, tout en se dirigeant vers la destination.

On propose ainsi, un mécanisme d'évitement d'obstacle basé sur un système multi-agents réactifs d'inspiration physique. L'organisation émergente du système multi-agents est interprétée et dans le but d'élaborer une décision globale basée sur des indicateurs locaux.

---

## Sommaire

---

<b>6.1</b>	<b>Une approche pour l'évitement d'obstacles</b>	<b>76</b>
6.1.1	Introduction	76
6.1.2	Caractéristiques de l'approche	76
6.1.3	Représentation de l'environnement et attributs des agents	78
<b>6.2</b>	<b>Modèle d'interaction</b>	<b>79</b>
6.2.1	Description du modèle	79
6.2.2	Calcul du comportement de l'agent décisionnel	81
<b>6.3</b>	<b>Evaluation globale pour la prise de décision</b>	<b>82</b>
6.3.1	Calcul du vecteur décision	82
6.3.2	Modulation du vecteur décision à partir de la physique statistique	83

---

## 6.1 Une approche pour l'évitement d'obstacles

### 6.1.1 Introduction

Nous nous intéressons, dans cette section, au comportement d'un seul véhicule se déplaçant le long d'une trajectoire tout en évitant les obstacles. Doter le véhicule d'un tel comportement doit enrichir la conduite en convoi, en permettant au véhicule leader de naviguer de manière complètement autonome le long d'une trajectoire prédéfinie. Par ailleurs, l'apparition d'un obstacle (par exemple, un piéton qui traverse entre deux véhicules du convoi) doit permettre au véhicule de l'éviter tout en poursuivant sa trajectoire.

L'évitement d'obstacle, la planification de mouvement et le suivi de trajectoire sont des problèmes bien connus et étroitement liés. Ils sont classiquement résolus par le calcul d'une trajectoire qui évite les obstacles [Latombe, 1991]. Toutefois, les méthodes standard pour la planification du mouvement ne peuvent être appliquées à un environnement dynamique incertain. Par conséquent, la navigation réactive semble être une solution appropriée. La littérature classe ces approches en trois catégories :

- *Méthodes inspirées de la physique* : dans ce type de modèle, les informations des capteurs sont transformées en lois de commande de direction et de vitesse. Parmi ces méthodes, nous pouvons mentionner celles à base de champs de potentiels comme VFH (Vector Field Histogram) et VFF (Virtual Force Field) [Borenstein et al., 1991] et [Masoud and Masoud, 2000].
- *Méthodes de sélection de commandes de déplacement* : cette approche calcule un ensemble des commandes de déplacement pour en sélectionner une par rapport à des stratégies de navigation [Simmons, 1996]. La stratégie de sélection peut être fondée sur des algorithmes classiques tels que l'architecture HTN (Hierarchical Task Networks) ou des courbes paramétriques de type clothoïdes [Scheuer., 1998].
- *Méthodes de haut niveau* : ces méthodes calculent une description de haut-niveau des informations sensorielles pour obtenir une commande de déplacement. Les diagrammes de navigation [Minguez et al., 2004] appartiennent à ce groupe d'approches.

Nous présentons ici une approche appartenant aux méthodes d'inspiration physique, dont les principales caractéristiques sont présentées par la section suivante.

### 6.1.2 Caractéristiques de l'approche

L'évitement d'obstacles peut être vu comme un problème de décision : quelle vitesse, en direction et magnitude, doit être adoptée par un véhicule, compte-tenu de sa position relativement à un point de destination et compte tenu de la distribution d'obstacles présents dans son environnement perçu. Face à ce problème de décision, nous adoptons une

approche dont les principaux aspects sont :

- Le véhicule est associé à un système multi-agents (cf. figure 6.1), dont les composants actifs sont appelés agents décisionnels (AD)<sup>1</sup>

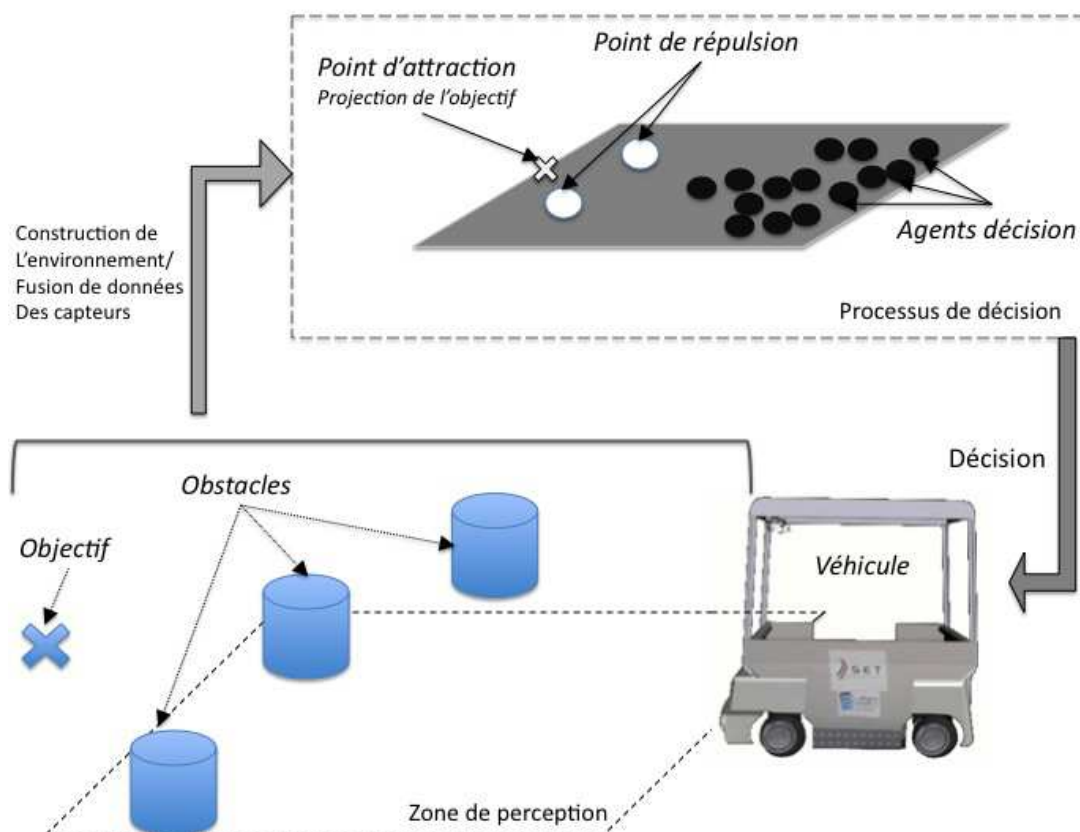


Figure 6.1: Vue d'ensemble de l'approche pour l'évitement d'obstacle

- Les AD évoluent dans un environnement virtuel qui est construit à partir de la perception de l'environnement matériel par le véhicule [Simonin and Gechter, 2006]. L'environnement virtuel contient une représentation des obstacles perçus et du point de destination.
- Les AD se déplacent dans l'environnement virtuel, en fonction de la composition de plusieurs champs de force produits par les éléments présents : les obstacles, le point de destination, les AD eux mêmes.
- L'environnement virtuel change en fonction du déplacement du véhicule et de sa zone de perception. Ces changements provoquent une reconfiguration des champs de force

<sup>1</sup>Il s'agit ici d'agents réactifs. Par conséquent, il ne faut pas donner au mot agent décisionnel le même sens qu'on lui donnerait dans le contexte des SAM dits cognitifs. Ce qui persiste, cependant, est l'idée que, même sans connaissances explicites, de très simples agents réactifs peuvent, collectivement, faire émerger une décision.

et, partant, une évolution de la distribution des AD.

- Les champs de force ont été définis de telle façon que la distribution des AD dans l’environnement virtuel à un instant donné soit sensiblement dépendante des positions relatives des obstacles et du point de destination. Par conséquent, de cette distribution il sera en principe possible d’extraire des informations permettant d’élaborer une décision : la meilleure vitesse pour le véhicule, en direction et magnitude.
- Il s’agira donc d’évaluer des indicateurs globaux à partir de l’ensemble de données locales constitué par la position et la vitesse de chaque AD dans l’environnement virtuel. Pour la direction du vecteur vitesse nous adoptons un indicateur globale classique : le centre de masse de l’ensemble des AD complété par la vitesse moyenne des AD. Pour la magnitude du vecteur vitesse nous adoptons une approche inspirée de la physique statistique, qui sera détaillée par la suite.

### 6.1.3 Représentation de l’environnement et attributs des agents

Dans l’approche proposée ici, l’environnement est défini à partir de la perception des capteurs (cf. figure 6.2). Les limites de l’environnement des AD correspondent aux limites de perception des capteurs. Les interactions sont définies par le modèle physique suivant : les agents décisionnels sont repoussés par les autres AD et par les obstacles. Nous considérons que les limites de l’environnement construit font partie des obstacles. En revanche, ils sont attirés par l’objectif. Il est également intégré dans l’environnement, soit comme point d’attraction s’il est dans le champ de perception du véhicule soit comme une direction d’attraction si les capteurs ne le perçoivent pas.

Chaque obstacle détecté par un ou plusieurs capteurs est transformé en un point de répulsion dans l’environnement des AD :

- *Position 2D* : calculée à partir des informations des capteurs.
- *Coefficient de répulsion* : noté  $R_o$ , dépend de la taille de l’objet.

L’objectif est transformé en point d’attraction avec les propriétés suivantes :

- *Position 2D* : correspond à la position du point d’attraction s’il est dans le champ de perception du véhicule ou une direction d’attraction si les capteurs ne le perçoivent pas.
- *Coefficient d’attraction* : noté  $A_w$  et proportionnelle à la distance entre le véhicule et l’objectif.

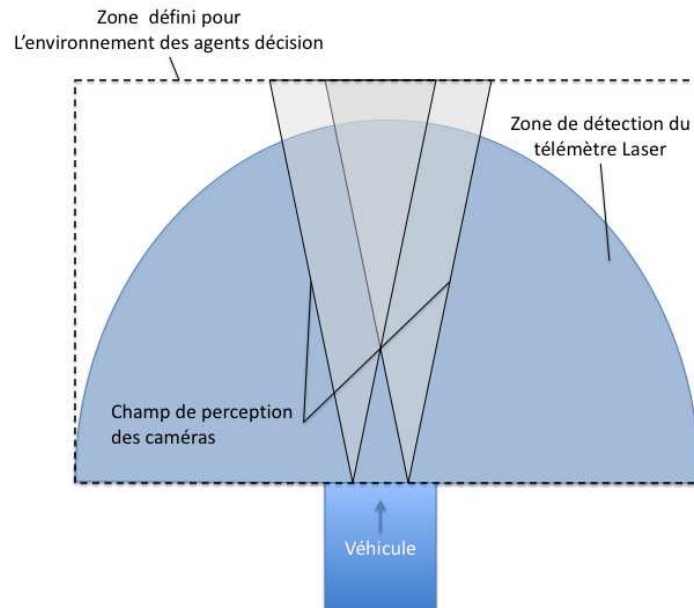


Figure 6.2: Zones de perception des agents décisionnels

Dans ce contexte, nous avons considéré les agents comme des particules de faible masse, dont le comportement tend à atteindre le minimum d'un champ de potentiel. Ce potentiel est créé à partir de l'attraction/répulsion des points de l'environnement et des agents décisionnels. Les AD détiennent les attributs suivants :

Attribut	Description
$m$	masse des agents décisionnels
$A_i = (p_x, p_y)_i$	position de l'agent $i$ en $x$ et $y$
$V_i = (v_x, v_y)_i$	vitesse de l'agent $i$ en $x$ et $y$
$\gamma_i = (a_x, a_y)_i$	accélération de l'agent $i$ en $x$ et $y$
$P_R$	rayon de perception

Table 6.1: Attributs des agents décisionnels

## 6.2 Modèle d'interaction

### 6.2.1 Description du modèle

Trois types d'interaction sont définies : (i) interaction entre agents (ii) interaction entre agents et obstacles (points de répulsion) (iii) interaction entre agents et points de passages (points d'attraction).

- *Interaction Agent-Agent* : nous les considérons comme une simple force de répulsion de type Newtonienne en  $1/d^2$ , de coefficient  $R_A$ . Son expression est la suivante, en



considérant deux agents décisionnels  $i$  et  $j$  situés respectivement à la position  $A_i$  et  $A_j$  :

$$\vec{R}_{ij} = R_A \cdot m \cdot m \frac{A_i \vec{A}_j}{\|A_i \vec{A}_j\|^3} \quad (6.1)$$

pour chaque agent décisionnel  $A_j$  tel que  $\|A_i \vec{A}_j\| < P_R$ . Dans notre cas, puisque la population est homogène, ces coefficients sont identiques. La résultante de toutes les forces induites par les autres agents peuvent être calculée par les équations :

$$\left\{ \begin{array}{l} R_{X_j} = \sum_i \left( R_A \cdot m \cdot m \frac{(x_j - x_i)}{((y_j - y_i)^2 + (x_j - x_i)^2)^{\frac{3}{2}}} \right) \\ R_{Y_j} = \sum_i \left( R_A \cdot m \cdot m \frac{(y_j - y_i)}{((y_j - y_i)^2 + (x_j - x_i)^2)^{\frac{3}{2}}} \right) \end{array} \right. \quad (6.2)$$

avec  $i$  tel que  $i \neq j$  et  $d_{ij} < P_R$

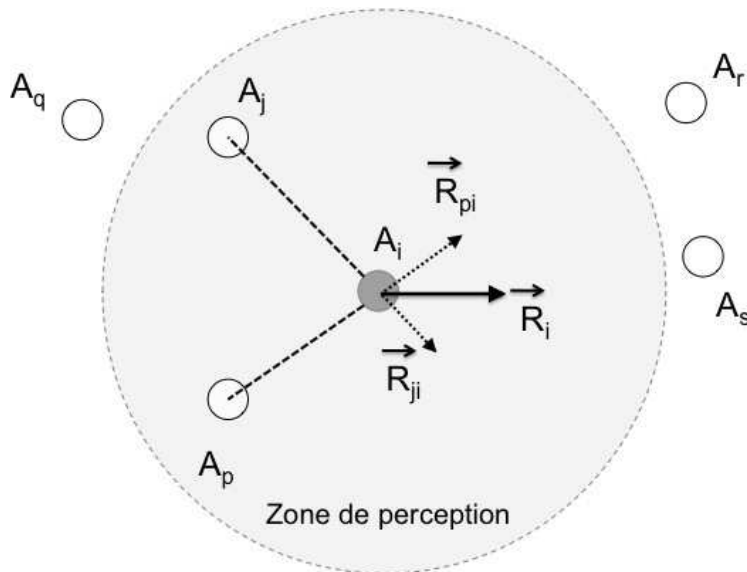


Figure 6.3: Comportement de répulsion Agent-Agent

La figure 6.3 montre la répulsion provoquée par le comportement des agents  $A_j$  (force  $\vec{R}_{ji}$ ) et  $A_p$  (force  $\vec{R}_{pi}$ ) sur l'agent  $A_i$ . La répulsion globale est donnée par la force notée  $\vec{R}_i$ . Les agents  $A_j$  et  $A_p$  sont dans la perception de l'agent  $A_i$ . Les agents  $A_q$ ,  $A_r$  et  $A_s$  sont hors de la zone perception et n'ont donc pas d'influence sur l'agent  $A_i$ .

- *Interaction Agent-Obstacle* : est semblable à celle définie pour l'interaction Agent-Agent. L'expression suivante correspond à l'interaction entre l'agent  $i$  et l'obstacle

$o$  :

$$\vec{R}_{io} = R_o \cdot m \cdot m \frac{\vec{O}\vec{A}_i}{\|\vec{O}\vec{A}_i\|^3} \quad (6.3)$$

Pour chaque obstacle  $O$  tel que  $\|\vec{A}_i\vec{O}\| < P_R$ . La résultante de toutes les forces induites par les obstacles peuvent être calculée par les équations :

$$\begin{cases} R_{X_{io}} = \sum_o \left( R_o \cdot m \cdot m \frac{(x_i - x_o)}{((y_i - y_o)^2 + (x_i - x_o)^2)^{\frac{3}{2}}} \right) \\ R_{Y_{io}} = \sum_o \left( R_o \cdot m \cdot m \frac{(y_i - y_o)}{((y_i - y_o)^2 + (x_i - x_o)^2)^{\frac{3}{2}}} \right) \end{cases} \quad (6.4)$$

avec  $o$  tel que  $d_{io} < P_R$

- *Interaction Agent-Objectif* : nous l'avons définie par une simple attraction linéaire :

$$\vec{A}_{iw} = A_w \cdot m_i \cdot \vec{A}_i\vec{W} \quad (6.5)$$

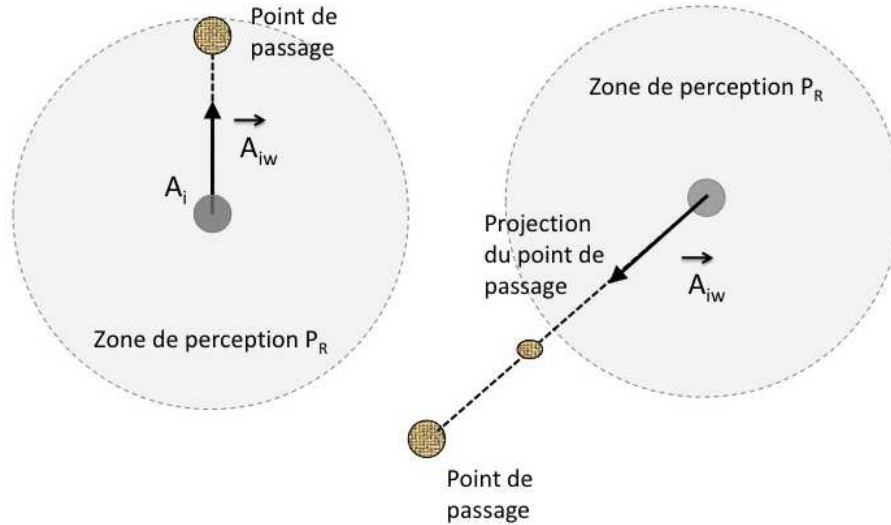


Figure 6.4: Comportement d'attraction Agent-point de passage (Gauche) point de passage dans la zone de perception, (Droite) point de passage en dehors de la zone de perception

## 6.2.2 Calcul du comportement de l'agent décisionnel

La position, la vitesse et l'accélération de chaque agent sont calculées pour chaque pas de temps  $\delta t$ . En appliquant le principe fondamental de la dynamique, l'accélération de chaque agent peut être calculée à partir des forces (cf. équation 6.6). Ici,  $\vec{\gamma}_i$  représente l'accélération,  $m$  la masse de l'agent, et  $\vec{F}_i$  la somme des forces qui s'exerce sur l'agent  $i$ .

$$\vec{\gamma}_i = \frac{1}{m} \vec{F}_i \quad (6.6)$$

Par double intégration, l'équation suivante est obtenue :

$$X_i^{\vec{}}(t) = \vec{X}_i(t-1) + \left( \vec{V}_i(t-1)\delta t + \frac{(\delta t)^2}{2m} \left( \vec{R}_{io} + \vec{R}_i + \vec{A}_{iw} \right) \right) \quad (6.7)$$

avec la position de l'agent  $i$   $X_i^{\vec{}}(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}$

## 6.3 Evaluation globale pour la prise de décision

### 6.3.1 Calcul du vecteur décision

Le calcul du vecteur décision est calculé à partir de la position et la vitesse moyenne des agents. La position moyenne donnera une composante du vecteur décision représentant une direction d'évitement. Le composant lié à la vitesse moyenne des agents permet de prendre en compte le mouvement d'évitement déjà entrepris par les agents décisionnels.

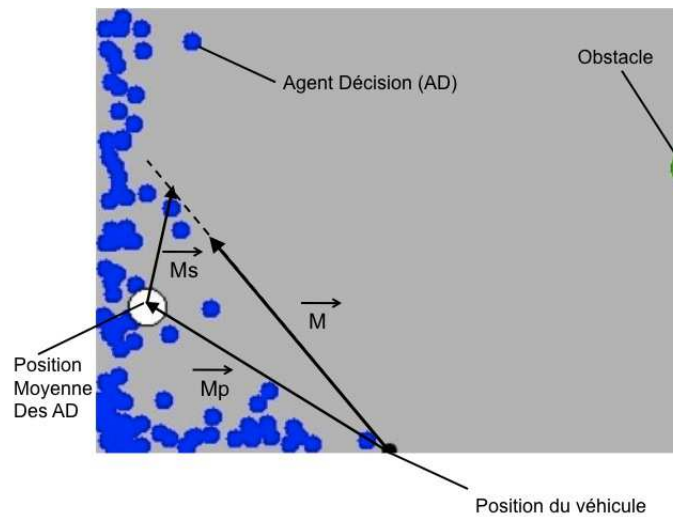


Figure 6.5: Calcul du vecteur décision  $\vec{M}$

Cette décision est représentée par le vecteur  $\vec{M}$  dont on déduit de l'orientation un angle et de sa norme une vitesse à appliquer sur le véhicule. L'angle du vecteur vitesse est ensuite modifié en fonction de la valeur de la vitesse moyenne. Cette modification produit le vecteur  $\vec{M}$  qui vient de la combinaison du vecteur  $\vec{M}_p$  et de la vitesse moyenne  $\vec{M}_s$ . Ensuite, la norme du vecteur de combinaison  $\vec{M}$  est fixé en fonction de la norme du vecteur vitesse

moyenne (cf. figure 6.5), comme le décrit l'équation suivante :

$$\vec{M} = \left( \vec{M}_p + \frac{\vec{M}_s}{\sqrt{n}} \right) \frac{\|\vec{M}_p\|}{\left\| \left( \vec{M}_p + \frac{\vec{M}_s}{\sqrt{n}} \right) \right\|} \quad (6.8)$$

avec  $n$  le nombre d'agents décisionnels.

### 6.3.2 Modulation du vecteur décision à partir de la physique statistique

Dans ce paragraphe, nous proposons un mécanisme de modulation de la magnitude du vecteur décision, en fonction de la contrainte imposée par la configuration des obstacles. Par exemple, dans le cas d'obstacles disposés en "entonnoir", i.e se situant de part et d'autre du véhicule, la vitesse de ce dernier devrait diminuer dans le but d'éviter toute éventuelle collision.

Pour effectuer cette modulation, nous proposons d'évaluer la répartition des agents décisionnels et en particulier la contrainte produite par les champs de répulsions des obstacles. Pour effectuer cette mesure, nous utilisons la physique statistique. Une introduction aux concepts de la physique statistique est présentée dans l'annexe A. Parmi les indicateurs présentés en annexe, nous choisissons l'énergie libre de Helmholtz  $A$ . La première étape consiste à exprimer l'énergie du système :

- *Energie cinétique* : Dans l'équation suivante, l'agent  $i$  est représentée par sa masse  $m$  et sa vitesse  $\vec{V}_i$ .

$$E_c = \frac{1}{2} m \vec{V}_i \cdot \vec{V}_i \quad (6.9)$$

- *Energie potentielle* : Pour l'agent  $i$  elle est calculée en utilisant l'expression de l'énergie  $U$  (cf. équation 6.10) où  $\delta W$  représente le travail du système et  $\delta Q$  le flux de chaleur (ici,  $\delta Q = 0$  puisqu'il n'y a pas de dissipation de chaleur).

$$U = \delta W + \delta Q \quad (6.10)$$

Le travail du système  $\delta W$  est calculé en considérant que la force est conservative (cf. équation 6.11) avec  $\vec{d}u$  le vecteur directeur unité de la vitesse de l'agent.

$$E_p = \delta W = \vec{F}_{totale} \cdot \vec{d}u \quad (6.11)$$

L'énergie de l'agent correspond à la somme de l'énergie cinétique et de l'énergie potentiel ( $E = E_c + E_p$ ). Maintenant que l'énergie est connue pour chaque agent, l'énergie libre  $A$  peut être déduite :

$$\begin{cases} A(T, V, N_i) = -\ln(Z) \\ Z = e^{-\beta E_i} \end{cases} \quad (6.12)$$

Avec  $T$  la température,  $V$  le volume,  $N_i$  le nombre d'agents et  $E_i = E_c + E_p$ . Le volume correspond à la zone de perception du véhicule et la température à celle de la simulation. Le paramètre  $\beta$  est un facteur d'échelle permettant d'adapter la grandeur de l'énergie à notre système.

Le vecteur décision  $\vec{M}$  calculé précédemment est modulé en fonction du potentiel thermodynamique  $A$  pour obtenir le vecteur de décision final  $\vec{D}$  :

$$\vec{D} = (1 + \kappa.A) . \vec{M} \quad (6.13)$$

avec  $\kappa$  un facteur de prépondérance. Ce vecteur est ensuite appliqué comme une loi de commande du véhicule.

---

DEUXIÈME PARTIE

---

# **Spécification formelle et vérification**

---



# LANGAGES FORMELS

---

**C**E CHAPITRE présente un cadre conceptuel pour les approches formelles. Ensuite, quelques langages formels sont brièvement décrits. Le chapitre se termine par une comparaison de ces langages et la justification de notre choix.

---

## Sommaire

---

<b>7.1</b>	<b>Introduction</b>	<b>88</b>
<b>7.2</b>	<b>Systèmes formels, vérification, preuve, model-checking</b>	<b>88</b>
<b>7.3</b>	<b>Quelques langages formels</b>	<b>90</b>
7.3.1	Z et Object-Z	90
7.3.2	La méthode B	92
7.3.3	Les langages multi formalismes : <i>OZS</i>	95
<b>7.4</b>	<b>Conclusion</b>	<b>97</b>

---



## 7.1 Introduction

Les approches formelles en génie logiciel sont arrivées à un certain degré de maturité industrielle mais ne peuvent pas être considérées comme des approches universellement praticables, utilisables à un coût raisonnable pour toutes les classes d'applications. Elles ont trouvé leur champs d'utilisation dans un domaine du logiciel : celui des applications enfouies dites critiques, car soumises à des propriétés impératives de sûreté de fonctionnement. Ce sont des applications dont un défaut de conception pourrait se traduire par une faille de fonctionnement, avec des pertes humaines et/ou matérielles. Les approches formelles sont incontournables dans ce domaine, car elles donnent la possibilité de prouver, au sens d'une preuve mathématique, que le modèle formel d'une application satisfait en toute circonstance une propriété donnée.

Les champs d'application pour les systèmes multi-agents réactifs que nous présentons dans ces travaux incluent des applications critiques. Un exemple en est le problème même qui constitue le cas d'application présenté ici, le platooning. Le rôle des approches formelles dans nos travaux trouve là sa pleine justification. Dans cette perspective, le présent chapitre décrit des aspects du domaine formel, dans le but d'introduire les notions appliquées par la suite et de justifier certains choix. Nous concentrons notre attention au cours de ce chapitre, sur les aspects des approches et formalismes considérés qui sont liées à la possibilité de vérifier la validité de propriétés dites de sûreté.

## 7.2 Systèmes formels, vérification, preuve, model-checking

Les approches formelles font appel à un système formel, en tant que support de modélisation. Tout système formel comprend trois éléments de base  $(\mathcal{L}, \mathcal{S}, f_s)$ , où  $\mathcal{L}$  est un langage, défini en donnant sa syntaxe et utilisé pour construire des modèles formels de spécification.  $\mathcal{S}$  est un domaine sémantique, une structure mathématique constituée d'ensembles, fonctions, relations, opérateurs, ... La fonction  $f_s$ , dite fonction sémantique, établit une correspondance entre une entité syntaxique et une entité sémantique. En général, on définit  $f_s$  de façon inductive, à partir de la définition des règles syntaxiques de  $\mathcal{L}$ .

Aux trois éléments de base, on ajoute une troisième composante,  $\mathcal{T}$ , qui est une théorie du premier ordre (TPO), définie sur les entités du domaine sémantique, éventuellement augmenté d'autres domaines, si nécessaire. Une TPO est une structure de la logique formelle, qui comprend un ensemble de prédicats qui s'appliquent aux entités du domaine sémantique et un langage d'expressions logiques, défini à partir des prédicats, à l'aide d'opérateurs logiques classiques et/ou modaux. La TPO peut inclure également un système de déduction, constitué d'un ensemble d'axiomes et d'un ensemble de règles de déduction. Le rôle de

la TPO sera d'exprimer des propriétés relatives aux entités sémantiques. La présence d'un système de déduction introduit la possibilité de prouver les dites propriétés, relativement à des entités sémantiques. Il est à signaler que pour certains systèmes formels, le domaine sémantique et la TPO se confondent. C'est le cas des formalismes basées sur la logique du premier ordre et la théorie des ensembles.

La TPO  $T$  associée à un système formel  $(\mathcal{L}, \mathcal{S}, f_s)$  permet de formuler des propriétés relatives aux entités sémantiques. Une formule logique  $P$  peut être évaluée à partir d'une entité sémantique  $\Sigma$  : cette dernière peut fournir des valeurs aux variables libres de  $P$ . La façon dont  $\Sigma$  fournit des valeurs permettant d'évaluer la formule  $P$  dépend de la nature de l'entité sémantique  $\Sigma$ . Un cas fréquent est le suivant : l'entité sémantique  $\Sigma$  est un système de transition qui produit potentiellement un ensemble dit le *comportement* de  $\Sigma$ , noté  $\mathcal{C}(\Sigma)$ . Les éléments de  $\mathcal{C}(\Sigma)$  sont des chaînes : des suites infinies d'états du système  $\Sigma$ . La propriété  $P$  peut être évaluée relativement à chacune de ces chaînes. Dans ce cas, la TPO associée appartient sûrement au domaine des logiques temporelles, qui sont les plus aptes à exprimer des propriétés relatives à des chaînes.

Si  $\sigma \in \mathcal{C}(\Sigma)$ , nous notons  $\sigma[P]$  l'évaluation de la propriété  $P$  par  $\sigma$  et nous notons  $\sigma \models P$  si  $\sigma[P] = \text{vrai}$ . Dans ces conditions, soit  $\Sigma$  une entité sémantique de comportement  $\mathcal{C}(\Sigma)$ , et  $P$  une propriété, nous dirons que  $P$  est valide relativement à  $\Sigma$  si, quelle que soit  $\sigma \in \mathcal{C}(\Sigma)$ ,  $\sigma \models P$ . Dans ce cas on note :  $\Sigma \models P$ .

La vérification de la validité est une forme de test qui s'applique à  $\Sigma$  et à  $P$  et qui répond par oui ou par non à la question : *P est-elle valide relativement à  $\Sigma$  ?* Suivant la mise en œuvre de l'algorithme de vérification, on parlera de preuve (theorem proving) ou de model-checking. La preuve repose sur la présence d'un système de déduction associé à la TPO du système formel. L'entité sémantique  $\Sigma$  est représentée par un ensemble de formules  $\mathcal{H}(\Sigma)$  qui jouent le rôle d'hypothèses. Dans ce cas, on montre  $\Sigma \models P$  en montrant, par une preuve de  $P$  sous les hypothèses  $\mathcal{H}(\Sigma)$  (ce qui est noté  $\mathcal{H}(\Sigma) \vdash P$ ).

Le model-checking est basé sur les procédures de décision de la validité ou de la satisfiabilité, c'est à dire la recherche de modèles, au sens logique du terme, par exploration de l'espace d'état. Si nous considérons, par exemple, le cas des systèmes de transition, les modèles de  $P$  sont les chaînes  $\sigma$  telles que  $\sigma \models P$ . En général, les algorithmes de décision les plus efficaces sont ceux qui décident de la satisfiabilité (appelés SAT solvers). Dans ce cas, on cherchera à vérifier que  $\Sigma \models P$  en cherchant à satisfaire  $\neg P$ . Si l'on trouve une chaîne  $\sigma \in \mathcal{C}(\Sigma)$  telle que  $\sigma \models \neg P$ , alors  $\sigma$  est un contre-exemple de la propriété étudiée ( $P$ ). Si aucun contre-exemple n'est trouvé, on conclut que  $\Sigma \models P$ .

Pour finir cette section, quelques observations relatives au type de propriétés soumises à vérification. Les deux principales classes en sont les propriétés de sûreté et les propriétés de vivacité. Elles s'expriment facilement à l'aide de deux opérateurs de la logique temporelle linéaire :  $\diamond$  (éventuellement) pour la vivacité et  $\square$  (nécessairement), pour la sûreté.

Si  $P$  est une formule logique d'état (c'est à dire sans opérateur temporel) exprimant une propriété de sûreté, alors il s'agira d'établir la validité de la formule  $\square P$ , ce qui veut dire " $P$  est satisfaite par tout état du système visité lors de toute évolution possible". En d'autres termes,  $P$  est un invariant du système. Les méthodes de vérifications basées sur le model-checking sont bien adaptées à la décision de la satisfiabilité et de la validité des formules de la logique temporelle, dont l'invariance. Les méthodes basées sur la preuve le sont moins en général, sauf justement pour les propriétés d'invariance qui sont en général vérifiables par des techniques de preuve inductive.

Nous allons présenter quelques langages formels qui s'inscrivent dans le cadre conceptuel que nous venons de présenter et qui, par conséquent, sont adaptés à une démarche de vérification par la preuve ou le model checking.

## 7.3 Quelques langages formels

### 7.3.1 Z et Object-Z

Il s'agit de langages de spécification formelle dont la notation s'inspire de la formalisation de la théorie des ensembles et relations et de la logique du premier ordre.  $Z$  est un langage **fortement typé** [Spivey, 1988] : tout composant d'une spécification  $Z$  possède un type, et donc peut être associé à un ensemble de valeurs.  $Z$  se présente sous la forme d'un langage et d'un calcul des *schémas*. Le schéma, unité de spécification de  $Z$ , encapsule une partie déclaration et une partie contrainte, comme le montre l'exemple suivant :

<i>Compteurs</i>
$x : \mathbb{N}$
$y : \mathbb{N}$
$x \leq 32$
$y < x$

Ce schéma peut être facilement interprété comme la définition prédicative d'un ensemble,

$$\{(x, y) \in \mathbb{N}^2 \mid x \leq 32 \wedge y < x\}$$

qui est justement le type du schéma *Compteurs*. Le prédicat  $x \leq 32 \wedge y < x$  est la partie contrainte du schéma. L'exemple présente un schéma d'état, où des variables sont déclarées

et contraintes. La contrainte d'un schéma d'état est un invariant du modèle. Ainsi, les propriétés de sûreté qui pourraient être associées au modèle de spécification  $Z$  apparaîtraient comme des contraintes dans des schémas d'état. En plus des schémas d'état d'autres types de schémas sont présents : D'opération, de définition (dits axiomes), ces derniers peuvent être de différents sous types : ouverts, constants, génériques. Les opérations sont définies par des schémas spécifiques. Elles décrivent des changements d'état, c'est à dire des changements de valeur affectant une ou plusieurs variables. Ainsi, par exemple, le schéma :

$\begin{array}{l} \text{Inc}_y \\ \Delta y \\ \text{data?} : \mathbb{N} \end{array}$
$\begin{array}{l} y + \text{data?} < x \\ y' = y + \text{data?} \end{array}$

présente une opération  $\text{Inc}_y$  qui change la valeur de  $y$ . Le paramètre d'entrée  $\text{data?}$  contient l'incrément de cette valeur. La déclaration  $\Delta y$  revient à une double déclaration :  $y, y' : \mathbb{N}$ . L'opération est décrite comme un prédicat mettant en relation la valeur de  $y$  avant l'opération (notée  $y$ ) et la valeur après opération (notée  $y'$ ). La contrainte  $y + \text{data?} < x$  peut être vue comme une précondition destinée à préserver l'invariant  $x \leq 32 \wedge y < x$ . La définition prédictive du type associé à ce schéma est :

$$\{(y, y') \in \mathbb{N}^2 \mid \exists \text{data?} \in \mathbb{N} \bullet y + \text{data?} < x \wedge y' = y + \text{data?}\}$$

qui représente l'ensemble de couples  $(y, y')$  qui peuvent être des valeurs de  $y$  dans un couple d'états (avant, après) relativement à l'opération  $\text{Inc}_y$ .

Le principal intérêt de  $Z$  réside dans sa richesse expressive, donnée par un large éventail de constructeurs de types : ensemble, produits cartésiens, fonctions et relations, séquences, types récursifs (libres), ... A chaque constructeur sont associées des opérations. En arrière-plan, une théorie constituée d'axiomes et règles de déduction prépare le terrain pour la vérification par la preuve. Le langage  $Z$  constitue donc un exemple de formalisme qui se confond avec sa TPO. D'autre part, la fonction sémantique est pratiquement la fonction identité, dans la mesure où tout schéma correspond presque directement à un prédicat, qui représente un ensemble de valeurs, et ce dernier constitue l'entité sémantique associée.

Le formalisme *Object* –  $Z$  [Duke and Rose, 2000, Smith, 2000] donne un objet à  $Z$  en permettant d'encapsuler l'ensemble des schémas qui constituent le modèle de spécification d'un système, dans un schéma de classe. Toutes les variables d'état sont déclarées et contraintes dans un schéma anonyme dit schéma d'état. Les contraintes de ce schéma sont les invariants d'état de la classe. Une opération standard d'initialisation décrit l'ensemble

d'états initiaux possibles pour la classe. En outre, comme correspond tout formalisme de type objet, *Object - Z* propose des mécanismes formels pour l'agrégation et l'héritage.

Si  $Z$  n'inclut aucune construction pour décrire le comportement d'un modèle, *Object - Z* peut inclure, dans la partie contrainte du schéma de classe, une formule de la logique temporelle linéaire, l'invariant temporel de la classe<sup>1</sup> qui exprime d'éventuelles restrictions sur l'enchaînement des opérations de la classe au cours du temps. Cet invariant dit temporel ne doit pas être confondu avec l'invariant d'état lequel, comme nous l'avons dit, est un prédicat sans opérateur temporel, qui apparaît dans la partie contraintes du schéma d'état.

$Z$  et *Object - Z*, dont les notations se basent sur la formalisation de la logique du premier ordre, sont adaptés à la vérification par la preuve. En particulier, les propriétés de sûreté de fonctionnement, qui, comme nous l'avons dit, s'intègrent facilement sous la forme d'invariants d'état, peuvent être prouvés inductivement :

1. prouver que l'ensemble d'états initiaux satisfait l'invariant
2. pour chaque opération  $Op$ , prouver que si l'état immédiatement avant  $Op$  satisfait l'invariant, alors l'état immédiatement après  $Op$  le satisfait aussi.

Malgré la place faite à la logique, aucun environnement d'aide à la preuve pour  $Z$  ou *Object - Z* n'est arrivé à une maturité considérable.

### 7.3.2 La méthode B

La méthode B [Abrial, 1996], née d'une évolution du langage  $Z$ , appartient, comme ce dernier, à la famille des formalismes basés sur la notation formelle de la logique et de la théorie des ensembles et dès lors, se confond avec sa TPO et inclut un système de déduction.

L'évolution par rapport à  $Z$  consiste particulièrement en l'adoption des substitutions généralisées comme formalisme de description des transformations opérées sur les données. Une substitution est un transformateur de prédicats. Si  $P$  est un prédicat et  $S_{op}$  la substitution qui décrit l'opération  $op$ , alors on note  $[S_{op}]P$  le prédicat qui résulte de l'application de  $S_{op}$  à  $P$ . Il a été prouvé que le prédicat  $[S_{op}]P$  est **la plus faible pré condition** que doit satisfaire un état du système auquel on applique l'opération  $op$ , pour que l'état qui en résulte satisfasse  $P$ . La preuve de propriétés relatives aux opérations d'un système s'inscrit ainsi dans une stratégie de déduction basée sur le concept de plus faible pré condition, ce qui en augmente l'efficacité et le degré d'automaticité (travail sans intervention de l'utilisateur).

L'autre évolution par rapport à  $Z$  consiste en l'introduction d'une unité de spécification bien structurée, la **machine abstraite**. Mais  $Z$  avait aussi marqué, avec *Object - Z*, une évolution dans le sens d'une structuration. Au cœur de la méthode B il y a un langage,

<sup>1</sup>Ce terme constitue à notre avis, un bel oxymore, car comment être à la fois invariant et temporel ?

l'Abstract Machine Notation (AMN). La méthode B inclut aussi un mécanisme de raffinement permettant de concevoir en allant de l'abstrait vers le concret, pour obtenir in-fine une spécification dite concrète, proche d'un programme. D'autre part, la méthode B possède plusieurs mécanismes de structuration qui favorisent la modularité et la réutilisation. Un modèle B est composé d'un ensemble de machines abstraites liées mutuellement, suivant un certain nombre de relations. Une machine abstraite est constituée d'une série de sections. Une section peut contenir soit une liste d'identificateurs (sections *SETS*, *CONSTANTS*, *VARIABLES*), soit un prédicat (sections *CONSTRAINTS*, *PROPERTIES*, *INVARIANT*), soit des substitutions (sections *INITIALISATION*, *OPERATIONS*).

Considérons le simple exemple suivant, ne contenant pas toutes les sections mentionnées. La machine *Portes* qui, étant donné un ensemble de portes, maintient l'information relative à l'état (ouverte ou fermée) de chacune d'entre elles :

```

MACHINE Portes
SETS PORT; POSIT = {ouverte, fermee}
VARIABLES position
INVARIANT position ∈ PORT → POSIT
INITIALISATION position := PORT × {fermee}
OPERATIONS
  ouvrir (dd) ≐
    PRE dd ∈ PORT THEN position(dd) := ouverte END
  fermer (dd) ≐
    PRE dd ∈ PORT THEN position(dd) := fermee END
END

```

La section *SETS* introduit les noms de deux ensembles, *PORT* et *POSIT*. Ce dernier étant un ensemble contenant deux éléments seulement, sa définition in-extenso peut être donnée d'emblée, en donnant un nom à chacun des deux éléments. La section *VARIABLES* introduit le nom d'une variable. La section *INVARIANT* donne le type de la variable : il s'agit d'une fonction totale entre les deux ensembles introduits auparavant. La déclaration du type dans la section *INVARIANTS* garantit l'une des propriétés de base pour la consistance d'une application : le type d'une variable doit être préservé, toute valeur affectée à la variable doit appartenir à son type. La section *INVARIANTS* peut servir également à l'expression d'autres contraintes invariantes, telles que les propriétés de sûreté.

La méthode B introduit d'emblée la notion de consistance d'une machine abstraite. A chaque section contenant un prédicat ou des transformateurs de prédicats sont associées des obligations de preuve : des prédicats que la machine abstraite doit satisfaire. Une machine abstraite est consistante si elle satisfait toutes les obligations de preuve qu'elle produit. Considérons l'exemple précédent, où la section invariant produit quatre obligations de preuve :

$$\exists p \bullet p \in PORT \rightarrow POSIT \quad (7.1)$$

$$[position := PORT \times \{fermee\}](position \in PORT \rightarrow POSIT) \quad (7.2)$$

$$(position \in PORT \rightarrow POSIT) \wedge (dd \in PORT) \Rightarrow [position(dd) := ouverte](position \in PORT \rightarrow POSIT) \quad (7.3)$$

$$(position \in PORT \rightarrow POSIT) \wedge (dd \in PORT) \Rightarrow [position(dd) := fermee](position \in PORT \rightarrow POSIT) \quad (7.4)$$

L'obligation (1) dit que l'invariant peut être satisfait. Comme cette obligation ne contient pas de variable libre, elle doit s'évaluer à vrai. Pour cela il suffit que les ensembles *PORT* et *POSIT* ne soient pas vides. L'obligation (2) dit que l'invariant est établi par l'opération d'initialisation. En effet, la substitution appliquée à l'invariant donne la plus faible pré condition sur l'état avant initialisation, pour que l'invariant soit satisfait après. Or le prédicat (2) s'évalue à vrai (car la substitution produit le prédicat  $PORT \times \{fermee\} \in PORT \rightarrow POSIT$ , qui est vrai). Cela veut dire que l'initialisation établit l'invariant, quel que soit l'état la précédant. Les obligations (3) et (4) dérivent des deux opérations *ouvrir* et *fermer* et sont satisfaites si, en supposant que l'état avant opération satisfait l'invariant, l'état après opération le satisfera aussi (les opérations préservent l'invariant). En fait, ce qui est orchestré n'est pas autre chose qu'une preuve inductive de l'invariant : l'initialisation l'établit et chaque opération le préserve.

La méthode B est, parmi les approches formelles, celle qui a atteint le plus haut niveau de maturité technologique [Habrias and Griech, 1997]. En effet, deux environnements de niveau industriel sont disponibles, *Atelier-B*<sup>2</sup> et *B-core*<sup>3</sup>, qui offrent l'ensemble d'outils nécessaires au déroulement de projets d'envergure : gestionnaires de projet, éditeurs, outils de preuve et d'animation, générateur d'obligations, générateurs de code. La preuve est orientée vers l'invariance, qui englobe les propriétés de sûreté, qui nous intéresse ici. Du côté des limitations, nous pouvons mentionner la prise en compte relativement basique des aspects comportementaux et l'absence de types prédéfinis correspondant aux valeurs numériques réelles. En ce qui concerne le premier aspect, lié au comportement, il existe

<sup>2</sup><http://www.atelierb.eu/>

<sup>3</sup><http://www.b-core.com/>

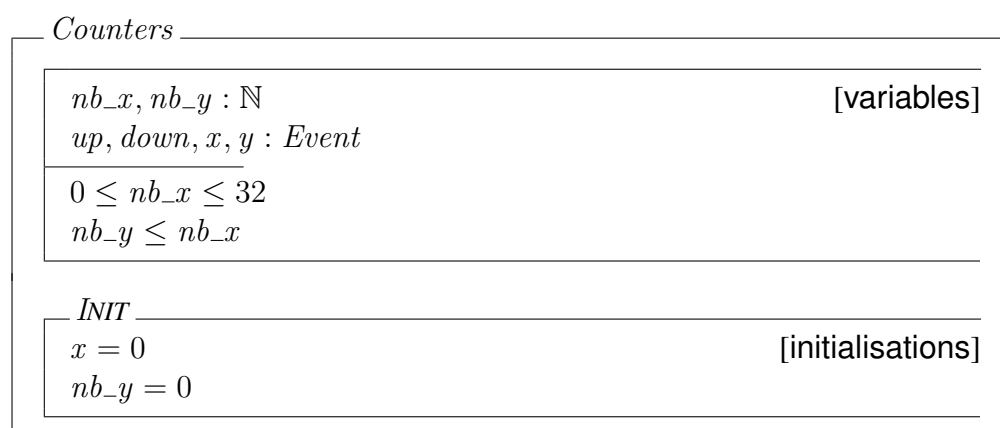
désormais une variante de la méthode  $B$ , appelée *EventB*<sup>4</sup> qui inclut la notion d'événement sous la forme d'un type particulier d'opération. Il est, dès lors en principe possible de modéliser des comportements par des machines abstraites qui représentent des automates d'état fini "plats". La deuxième limitation, l'inexistence de types numériques réels, prend une importance non négligeable lorsqu'il s'agit de modéliser des systèmes qui comportent des aspects physiques ou inspirés par la physique [Colin et al., 2008, 2009, Lanoix, 2008, Scheuer et al., 2008].

### 7.3.3 Les langages multi formalismes : *OZS*

L'un de nos objectifs est la spécification et la vérification de systèmes multi-agents réactifs (SMAR). Ce type de système présente des aspects aussi bien structuraux que comportementaux qui peuvent être assez complexes. D'autre part, les systèmes multi-agents étant des systèmes essentiellement évolutifs, les modèles du comportement revêtent un intérêt primordial. Les formalismes que nous avons présentés jusqu'à maintenant ne prennent en charge que l'aspect structurel, au sens de la structure de données, et l'aspect fonctionnel. Ils font l'impasse sur l'aspect comportement. Dans cette section nous allons présenter un formalisme de spécification où chacun des domaines, structurel et comportemental, occupe la place qui lui est propre. Ce formalisme, appelé *OZS*, résulte de l'intégration syntaxique et sémantique des statecharts et de *Object - Z* [Gruet et al., 2004].

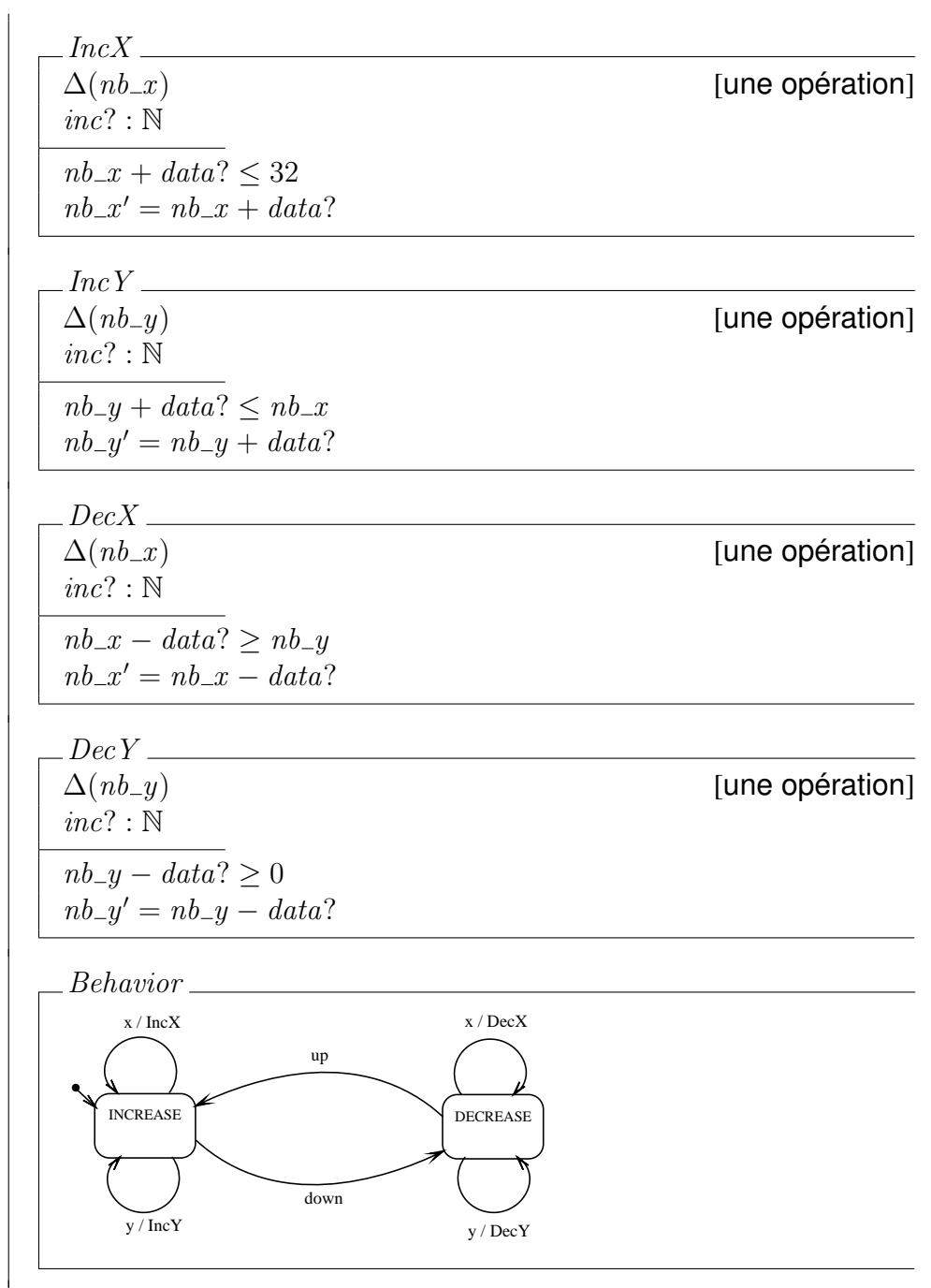
Pour *OZS*, l'intégration syntaxique se fait par la présence d'un schéma de comportement appelé *Behavior*, qui a accès aux variables de la classe. L'intégration sémantique se fait par adoption d'un domaine sémantique commun ou unique, les systèmes de transitions.

Pour illustrer les principales caractéristiques syntaxiques de *OZS* considérons un exemple. La classe *Counters* ci-dessous.



<sup>4</sup><http://www.event-b.org/>





Cette classe contient deux variables,  $nb\_x$  et  $nb\_y$ , dont la valeur est contrainte par un invariant :  $(0 \leq nb\_x \leq 32) \wedge (nb\_y \leq nb\_x)$ . L'événement  $x$  déclenche un changement de valeur de la variable  $nb\_x$  et l'événement  $y$  déclenche un changement de valeur de la variable  $nb\_y$ . Ce changement de valeur est soit un incrément (fonctions  $IncX$  et  $IncY$ ), soit un décrétement (fonctions  $DecX$  et  $DecY$ ), suivant l'état actif du statechart *Behavior* (*INCREASE* ou *DECREASE*). La valeur de l'incrément/décrétement est donnée par la variable d'entrée  $inc?$ . L'évènement  $up$  instaure l'état *INCREASE* et l'évènement  $down$  instaure l'état *DECREASE*.

Le formalisme *OZS* ne possède pas d'outil de vérification associé. Les vérifications peuvent se faire par l'intermédiaire de l'entité sémantique associée à tout modèle *OZS* : un système de transitions. De nombreux outils de vérification, basés principalement sur le model-checking, ont été proposés qui opèrent sur des modèles de type système de transitions. En particulier, nous nous sommes intéressés à la boîte à outils SAL, que nous présentons par la suite. SAL inclut un langage pour l'écriture de modèles de type système de transition. La sémantique d'*OZS*, à quelques détails près, permet de traduire facilement un modèle *OZS* vers un système de transitions SAL.

## 7.4 Conclusion

Les SMAR apparaissent comme une approche prometteuse pour des applications réactives présentant un caractère naturellement distribué. De telles applications comportent au moins deux caractéristiques :

- les aspects comportementaux sont très importants car ils spécifient une dynamique d'évolution conduisant à l'émergence des résultats espérés.
- lorsque l'application inclut des agents matériels situés dans un environnement physique, elle peut être soumise à la satisfaction de propriétés de sûreté de fonctionnement.

L'acceptation des approches SMAR dans de telles classes d'applications passe par un objectif : l'adoption de méthodes formelles adaptées et la disponibilité d'outils appropriés de vérification de propriétés de sûreté. Nous allons adopter ici ce critère de sélection, pour justifier le bien fondé du choix que nous avons fait : l'adoption de *OZS* en tant que formalisme de spécification et de modélisation et de SAL en tant qu'outil de vérification. Revenons donc sur les langages formels présentés ci-avant et considérons-les à la lumière de l'attention qu'ils accordent aux aspects comportementaux. D'autres aspects secondaires pourront renforcer le choix que nous avons fait.

**Z et Object-Z** : Malgré la place faite à la logique, aucun environnement d'aide à la preuve pour *Z* ou *Object - Z* n'est arrivé à une maturité considérable. Ce manque est l'une des raisons pour lesquelles nous n'avons adopté ni *Z* ni *Object - Z*, tel que, comme langage formel. L'autre raison, peut être la plus importante, est l'absence ou l'insuffisance des moyens de représentation des aspects comportementaux. La présence d'un invariant temporel dans une classe *Object - Z* apparaît comme un moyen peu adapté pour l'expression de ces aspects, particulièrement dans une perspective de conception et de mise en œuvre.

**La méthode B** : La maturité technique de ce langage/méthode et des outils associés, les placent parmi les plus aptes a-priori dans la perspective d'une approche formelle pour nos travaux. D'autant plus que la preuve de propriétés de sûreté se fait en B de façon naturelle, par technique inductive de preuve des invariants. Cependant, après quelques expériences d'utilisation, nous avons abandonné cette voie, pour les raisons que nous indiquons maintenant. Comme nous l'avons signalé, le langage B ne possède pas en forme native, une représentation ou formalisation des nombres réels. Dès lors, elle peut être appliquée à la modélisation de systèmes qui incluent des variables de ce type, au prix d'une discrétisation des dites variables et une simplification des modèles. Dans ces conditions, la confiance que l'on puisse attribuer aux résultats de la vérification est en relation inverse avec la distance entre le modèle et la réalité. Une autre raison, mais secondaire, a été la haute technicité de la preuve des invariants, en particulier du prouveur interactif. Les connaissances requises pour de telles interventions peuvent sans doute être acquises, mais nous avons préféré investir nos efforts de réflexion autour de la vérification avec OZS comme langage de spécification et SAL comme outil de vérification et dans une autre problématique, qui est celle de la vérification compositionnelle.

**OZS et SAL** : L'outillage formel que nous avons adopté est celui constitué du langage bi-formalisme OZS et de SAL (cf. chapitre 8.3). Nous avons ainsi combiné les facilités de OZS en tant que langage pour la construction des spécifications formelles, avec le richesse de la boîte à outils SAL en termes de model-checkers. SAL inclut lui même un langage orienté vers les systèmes de transition, nous avons préféré conserver OZS parce qu'il produit des modèles plus lisibles et qu'il s'adapte bien à la démarche que nous avons adoptée, qui sera décrite au chapitre 8. D'autre part, la traduction d'OZS vers SAL ne présente pas, comme nous l'avons dit plus haut, de difficulté majeure. SAL contient un model-checker qui est capable de traiter des modèles qui contiennent des variables réelles. D'autre part, ce model-checker, comme nous le verrons, propose des options de fonctionnement qui facilitent la mise en pratique de la procédure de vérification compositionnelle que nous proposons. En particulier, la possibilité d'exécuter le model-checking en lui fournissant une hypothèse complémentaire qui encadre l'exploration de l'espace d'état est, comme il sera expliqué par la suite, une caractéristique essentielle pour notre approche de vérification.

# UNE APPROCHE DE VÉRIFICATION

---

**C**E CHAPITRE présente le cadre formel permettant de spécifier nos modèles multi-agents et de vérifier leurs propriétés.

---

## Sommaire

---

<b>8.1</b>	<b>Introduction</b>	<b>100</b>
<b>8.2</b>	<b>Les difficultés de la vérification et comment les contourner</b>	<b>100</b>
<b>8.3</b>	<b>La boîte à outils SAL</b>	<b>102</b>
<b>8.4</b>	<b>De <i>OZS</i> vers SAL</b>	<b>103</b>
8.4.1	Méthode de traduction	103
8.4.2	Structure du système de transition associé à une classe <i>OZS</i>	104
8.4.3	La méthode de vérification	107
8.4.4	Vérification par model-checking	109

---

## 8.1 Introduction

La méthode de vérification proposée ici commence par la traduction des modèles de spécification du système, écrit à l'aide de *OZS*, vers un système de transition écrit suivant le langage de modélisation de SAL. Les règles de base pour la traduction de *OZS*, dont le domaine sémantique sont les systèmes de transition, ont été formulées dans [Gruet et al., 2004]. Nous nous sommes basés sur ces règles, mais nous avons simplifié les systèmes de transition résultants, en tenant compte de la structure de la classe *OZS* résultat de la construction du modèle de spécification de la classe d'agents. La simplification tient également compte de certains aspects relatifs à la synchronisation entre comportements associés aux différents rôles d'un agent.

Le paragraphe 8.4.1 présente cet aspect de la traduction *OZS* vers SAL. Nous présentons ensuite la procédure de vérification qui a été appliquée. Elle a servi à vérifier une propriété de sûreté, qui s'exprime aisément à l'aide du langage des expressions logiques de SAL, qui englobe un fragment significatif de la logique temporelle. Nous avons utilisé le bounded-model-checker (bmc) de SAL, qui est requis dès lors que le modèle de système à vérifier contient des variables réelles. Au paragraphe 8.4.4 nous présentons les concepts les plus importants relativement à cet aspect.

L'objectif final étant de vérifier la validité d'une propriété de sûreté relativement à un modèle formel, il est important de préciser la procédure de vérification adoptée. Celle-ci doit tenir compte des difficultés de la vérification face à la complexité du système modélisé. La section 8.2 décrit ces difficultés et introduit des pistes de contournement, dont celle retenue, la vérification compositionnelle.

Ainsi, la procédure de vérification repose sur une règle de vérification compositionnelle qui permet de palier au problème de la complexité combinatoire des systèmes vérifiés. Le paragraphe 8.4.3 décrit la règle de vérification et illustre son utilisation dans le cadre de SAL.

## 8.2 Les difficultés de la vérification et comment les contourner

Les approches de vérification ont leurs limitations, que nous allons présenter de façon très succincte. Des résultats théoriques relatifs à la calculabilité de la preuve en logique, indiquent que le "prouveur universel" capable de construire toutes les preuves possibles sans intervention humaine n'est pas à la portée des machines englobées dans l'hypothèse de Church-Turing. Par conséquent, la preuve ne peut être que semi-automatique : elle nécessite souvent l'aide d'un opérateur humain, à condition que celui-ci ait des idées susceptibles de faire avancer la démonstration. Cela n'empêche pas la puissance des algorithmes de preuve d'augmenter régulièrement. Mais on est forcé de constater que les prouveurs aujourd'hui

disponibles au sein des environnements de conception formelle pour le logiciel ne sont pas totalement automatiques. A partir d'un niveau de complexité assez bas, la preuve des propriétés relatives à un logiciel nécessite une participation humaine. La haute technicité de telles interventions est l'une des causes qui empêchent les approches formelles d'être considérés comme méthode universelle pour la conception du logiciel. Un autre inconvénient de la preuve est qu'un échec dans la démonstration ne donne pas facilement, en général, des indications sur le défaut sous-jacent de conception qui le provoque.

Les limitations du model-checking [Grumberg and Long, 1991] viennent de la nature exploratoire des algorithmes qui le mettent en œuvre. En effet, ceux-ci reposent sur un parcours de l'espace d'états du système modélisé. Ce parcours est potentiellement exhaustif, dans le cas où la propriété est valide (car dans ce cas l'exploration ne sera pas arrêtée par la découverte d'un contre-exemple). Pour peu que le modèle incorpore des variables d'état à valeur prise dans des ensembles non finis, le nombre d'états possibles devient infini, et le model-checking peut, dans ces conditions, ne pas se terminer.

Sans arriver au cas d'un espace d'états infini, des systèmes dont le vecteur d'état global possède un nombre important de variables à valeur finie peuvent cependant posséder un espace d'états de taille non-traitable. C'est la situation dite d'explosion combinatoire. La puissance des algorithmes n'a pas cessé d'augmenter au fil des recherches (" $2^{100}$  states and beyond" dit une phrase consacrée dans ce domaine). Malheureusement, des cas d'explosion combinatoire se présentent souvent dans la pratique réelle de la vérification.

Y a-t-il un remède aux problèmes mentionnés, dans le cas du model-checking ? Nous devons distinguer, pour bien répondre à cette question, les deux cas présentés plus haut :

1. Espace d'état infini : on évite la non-terminaison de l'exploration en limitant sa profondeur (profondeur bornée). Si aucun contre-exemple n'est détecté, dans les limites de cette exploration, dans certains cas il est possible de procéder à une preuve de nature inductive qui permet d'établir la validité de la propriété.
2. Espace d'état fini mais combinatoirement complexe : deux approches pour contourner cette difficulté ont fait l'objet de recherches :
  - Les approches basées sur l'abstraction : l'idée est de simplifier l'état global du système en faisant abstraction des valeurs des variables qui n'ont pas d'influence sur la satisfiabilité de la propriété étudiée. On peut par exemple constater que seul le signe d'une variable numérique peut influencer sur la satisfiabilité. L'application de cette technique passe par l'identification d'une fonction d'abstraction qui attribue

un état abstrait à chaque état concret. Il est nécessaire d'établir que la dite fonction préserve la validité de la propriété étudiée : si elle est valide relativement au système abstrait, elle sera valide relativement au système concret. Ceci constitue la difficulté principale pour l'application de la méthode.

- Les approches compositionnelles [Abadi and Lamport, 1993, Chang, 1993, Jonsson, 1994, Winter and Smith, 2003] : les systèmes modélisés présentent, dans tous les cas non triviaux, une structure qui les organise en hiérarchie de modules ou composantes. L'état global résulte de la composition des états locaux des composants, dont la taille est inférieure. L'idée dans ce cas est de vérifier une série de propriétés auxiliaires locales aux composants, et d'appliquer ensuite une règle de composition qui permet de déduire la propriété globale à partir des propriétés locales. Lors de la vérification de chaque propriété locale, le reste du système doit être pris en compte, car il interagit avec le composant sur lequel la vérification locale est effectuée. En général, on fait intervenir une propriété ou hypothèse qui est supposée "représenter" le reste du système. La validité de cette hypothèse doit être prouvée.

### 8.3 La boîte à outils SAL

Un modèle SAL est encapsulé par un *contexte* qui contient des définitions de constantes, de types et de fonctions. Un contexte contient également un ensemble de *modules*. Chaque module est un système de transition dont l'état est défini par un ensemble de variables locales, globales, d'entrée et de sortie. Les modules d'un contexte peuvent être composés de façon synchrone ou asynchrone. Une composition synchrone :  $M_1 || M_2$  est un module dont les transitions sont des compositions de transitions des modules  $M_1$  et  $M_2$  dont les gardes sont compatibles (c'est à dire, des transitions qui peuvent se déclencher simultanément). Une composition asynchrone :  $M_1 [ ] M_2$  est un module dont le comportement résulte d'un entrelacement de transitions de  $M_1$  avec des transitions de  $M_2$ .

La traduction de *OZS* vers SAL se base sur la sémantique de *OZS* présentée dans [Gruet et al., 2004]. En général, la traduction des constantes, types et fonctions suit une approche semblable à celle présentée dans [Smith and Wildman, 2005]. Une classe *OZS C* correspond à un contexte SAL. Chaque composante orthogonale du statechart  $C.Behavior$ , comportement de la classe, contribue avec un module SAL<sup>1</sup>. Un module additionnel sert à composer les modules orthogonaux. La composition synchrone correspond à la sémantique

<sup>1</sup>En générale, le comportement de la classe  $C$  résulte de la composition AND des statecharts comportement des sous classes  $C_1, \dots, C_n$  de  $C$ . Les  $C_i.Behavior$  sont les composantes orthogonales de  $C.Behavior$

de *OZS*, à quelques détails près qui sont pris en compte sans grande difficulté.

SAL inclut tout un éventail de model-checkers qui, étant donnés un modèle  $\mathcal{M}$  et une formule  $F$  de la logique temporelle, sont capables de décider de la validité de  $F$  relativement à  $\mathcal{M}$ . En cas de réfutation, un contre-exemple est en général présenté.

## 8.4 De OZS vers SAL

### 8.4.1 Méthode de traduction

Nous allons présenter le principe de traduction qui a permis de construire le système de transitions SAL, associé à la classe *OZS* qui modélise une classe d'agents. Pour illustrer ce principe, nous considérons une structure *OZS* d'agent proche de cela que nous avons définie dans le cadre de l'application. Ainsi, la figure 8.1 présente la structure de la classe *Agent*. Cette classe *OZS* résulte de l'agrégation des rôles *Perception*, *Action* et *DynamicModel*, modélisés à leur tour par des classes *OZS*.

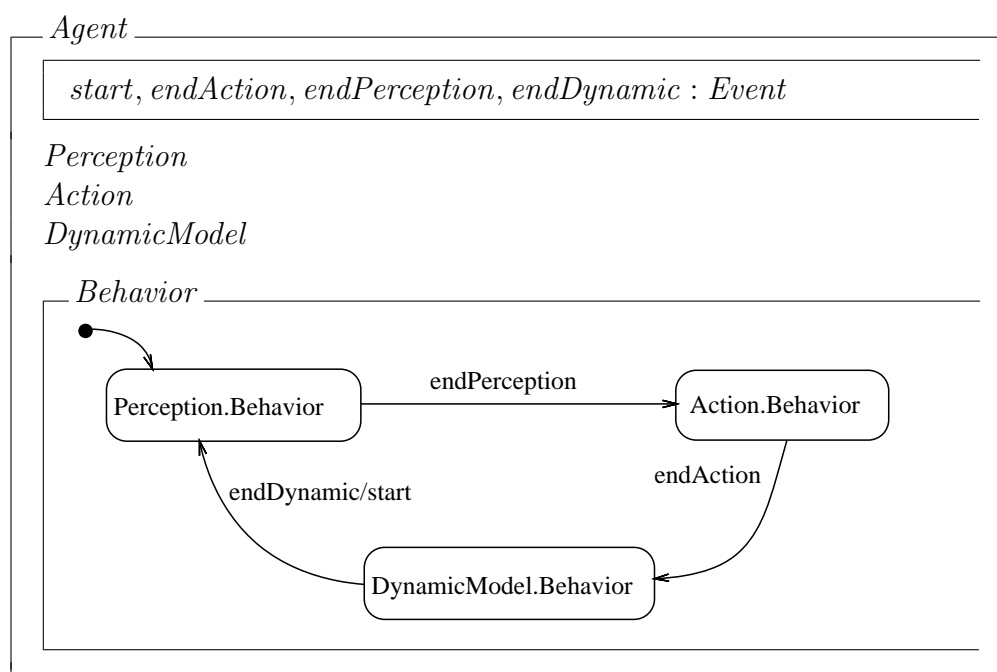


Figure 8.1: Classe *OZS* d'un agent réactif

Le modèle SAL comportera alors le contexte SAL *Agent*, qui contiendra la transcription de toutes les définitions globales de la classe *OZS Agent* (constantes, types, fonctions, variables d'état). Le contexte *Agent* contiendra en outre les modules SAL *Perception*, *Action* et *DynamicModel*. Chaque module est obtenu à partir de la sous-classe *OZS* correspondante, en appliquant partiellement les règles sémantiques définies par Gruer et al. [2004].



La sémantique mentionnée a été appliquée partiellement car, nous avons procédé à des simplifications qui tiennent compte de deux constats :

- le comportement de la classe *Agent* est la composition séquentielle des comportements de chaque rôle (perception, action, modèle dynamique). Ainsi, il n’y aura pas de situation où une transition du modèle SAL résultera de la composition de plusieurs transitions de statechart<sup>2</sup>. Ce constat permet de définir le comportement du contexte SAL comme la composition asynchrone des modules composants.
- Le modèle ne comporte pas d’événement externe. Les seuls événements sont internes et servent à la transition entre rôles : chaque rôle en fin d’opération émet un événement qui provoque l’activation du rôle suivant. Ceci a permis de simplifier la modélisation des événements, en les représentant par des variables booléennes.

Dans ces conditions, le modèle SAL sera construit à partir des éléments constitutifs de la classe *OZS* de départ, comme le montre la section suivante.

```

Agent : CONTEXT =
BEGIN
Définition des constantes
Définition des fonctions
Définition des types
Définition des variables
Définition des modules
END ;

```

Figure 8.2: Contexte SAL associé à la classe *Agent*

#### 8.4.2 Structure du système de transition associé à une classe *OZS*

Le modèle SAL de la classe *OZS Agent* sera donc encapsulé par un contexte dont la structure est présentée par la figure 8.2. Le contexte inclue un certain nombre de sections. Le contenu de chacune des sections dérive de façon assez directe des éléments constitutifs de la classe *OZS* de départ.

**Les définitions de constantes** transcrivent les définitions faites dans la classe *OZS* par l’intermédiaire de schémas ouverts, dits schémas d’axiomes :

<sup>2</sup>Il pourra y avoir des comportements de rôle qui comportent des composantes concurrentielles, et dans ce cas, il sera possible de traduire ce composant en appliquant la sémantique de [Gruet et al., 2004]

$SafetyDistance : \mathbb{R}$ $Delay : \mathbb{R}$
$SafetyDistance = \dots$ $Delay = \dots$

La transcription est directe :

```
%% Constants definition
SafetyDistance : REAL = ... ;
Delay : REAL = ... ;
...
```

**Les définitions de fonctions** sont semblables, il s'agit de traduire un schéma d'axiome vers une définition de fonction SAL :

$ComputeDistance : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$
$\forall speed, deltaSpeed : \mathbb{R} \bullet$ $ComputeDistance(speed, deltaSpeed) =$ $(speed + deltaSpeed) * Delay$

La correspondance est, ici aussi, directe :

```
%% définition de fonctions
ComputeDistance (speed : REAL, deltaSpeed : REAL) : REAL =
(speed + deltaSpeed)*Delay ;
...
```

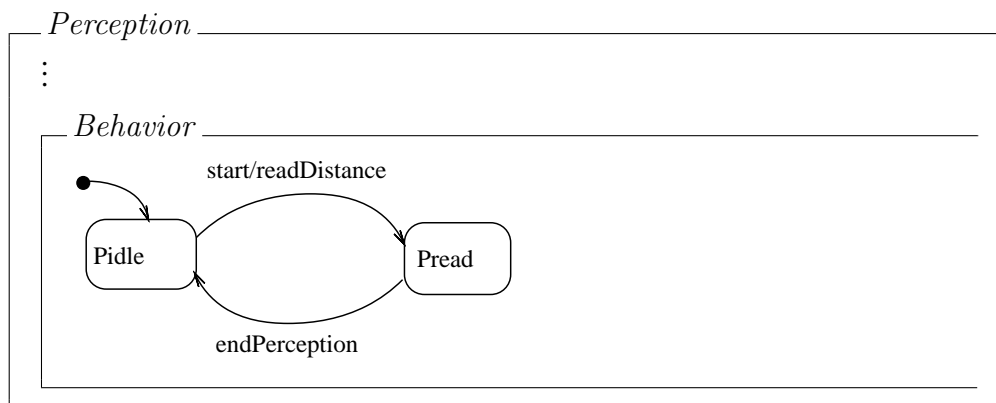
**Les définitions de types** sont liées principalement à la définition des modules. Rappelons qu'un module est un système de transition qui traduit un statechart. Ce dernier est le modèle de comportement d'un rôle de l'agent. Pour chaque rôle il y aura une définition de type sous la forme d'un ensemble défini en extension, dont les éléments sont symbolisés par les noms des états du statechart<sup>3</sup>. L'exemple suivant illustre cette traduction. Étant donné le schéma de comportement suivant, du rôle *Perception* :

la définition de type suivante sera incluse dans le contexte *Agent* :

```
%% définition de types
PerceptionState : TYPE = {Pidle, Pread} ;
...
```

**Les déclarations de variables globales** incluent toutes les variables globales de la classe *OZS* qui peuvent être transcrites de façon directe. A ces variables il faut ajouter une variable booléenne pour chaque événement interne. Ainsi, la classe de la figure 8.1 produit les variables d'événement :

<sup>3</sup>Ici aussi nous avons simplifié relativement à la règle générale, qui exigerait la définition d'un type configuration, constitué de sous ensembles d'états. Tel serait le cas si les comportements de rôle comportaient des sous-comportements concurrents.

Figure 8.3: Comportement du rôle *Perception*

```

%% déclaration de variables
...
start, endAction, endPerception, endDynamic : BOOL;
...

```

**La définition d'un module** se base principalement sur la traduction d'un statechart correspondant à un rôle de l'agent. La syntaxe SAL permet une traduction simple, en particulier dans le cas où le comportement du rôle ne contient pas de sous-comportements concurrents. Ainsi, le comportement de la figure 8.3 produit un module dont la structure sera comme ceci :

```

Perception : MODULE =
BEGIN
...
LOCAL state : PerceptionState = Pidle;
...
TRANSITION
[
state = Pidle AND start -- >
%% opération readDistance
...
start' = FALSE
state' = Pwait
[]
state = Pread -- >
endPerception' = TRUE
state' = Pidle
]
END;

```

Chaque transition du statechart donne une commande gardée du module, sous la forme  $\langle \text{garde} \rangle \text{ -- } \langle \text{instruction} \rangle$ . La garde  $\text{state} = \text{Pidle AND start}$  veut dire que l'état *Pidle* est actif et que l'événement interne *start* est présent. La règle pour la gestion des événements internes, qui sont des variables booléennes, est la suivante : la transition

qui "émet" l'événement met la variable à **vrai**, la transition qui "reçoit" l'événement met la variable à **faux**.

**Le module principal** est construit par la composition asynchrone des modules *Perception*, *Action* et *DynamicModel* :

```
Agent : MODULE =
  Perception [] Action [] DynamicModel
END;
```

### 8.4.3 La méthode de vérification

#### Une règle de vérification compositionnelle

La règle de vérification s'applique aux instances  $M_i$  du modèle du système. L'instance  $M_i$  résulte de  $M_{i-1}$  par l'addition d'un nouveau composant  $C_i$  (noté  $M_i = C_i \parallel M_{i-1}$ ). Généralement,  $C_i$  et  $M_{i-1}$  interagissent par l'intermédiaire d'un mécanisme de synchronisation ou de communication. Les systèmes composés de dispositifs physiques situés, peuvent comporter des interactions par l'intermédiaire de perceptions, sans aucun mécanisme explicite de communication ou synchronisation.

Le processus de vérification démarre avec une unique instance de composant  $M_0 = C_0$ . Les propriétés à vérifier à chaque pas de vérification sont :

- la propriété spécifique de sûreté, notée  $S_i$ , qui exprime une relation entre le composant  $C_i$  et l'instance  $M_{i-1}$  du système.
- une propriété auxiliaire  $T_i$ , relative au système  $M_i$ .

L'idée principale mise en pratique dans une méthode itérative de vérification compositionnelle est de simplifier la vérification de  $S_i$  en remplaçant  $M_{i-1}$  par la propriété auxiliaire  $T_{i-1}$ , introduite en tant qu'hypothèse. Le processus itératif s'arrête lorsque, pour un certain  $k$  l'instance  $M_k$  ne satisfait plus la propriété auxiliaire  $T_k$ .

Étant données les propriétés  $P_1, \dots, P_n$ , nous notons  $M \models P_1, \dots, P_n$  lorsque les propriétés  $P_i$  sont valides relativement au modèle  $M$ . Les propriétés sont évaluées sur des séquences infinies  $\sigma$  d'états, qui représentent des évolutions causales et progressistes du système. Notons  $\mathcal{B}(M)$  l'ensemble de séquences produites par  $M$  et  $Sat(P)$  l'ensemble de séquences qui satisfont la propriété  $P$ . Alors,  $M \models P$  si et seulement si  $\mathcal{B}(M) \subseteq Sat(P)$ . Nous notons  $M, P_1 \models P_2$  si la propriété  $P_2$  est satisfaite par le modèle  $M$  sous l'hypothèse  $P_1$  (dont la signification est  $M \models P_1 \Rightarrow P_2$ ).

La méthode de vérification compositionnelle applique la règle suivante pour  $i > 0$  :

$$R_c : \frac{C_i, T_i \models S_i}{C_i \parallel M_{i-1} \models S_i}$$

Une condition suffisante de consistance pour cette règle est que  $\mathcal{B}(C_i \parallel M_{i-1}) \subseteq \text{Sat}(T_i)$ , ce qui signifie que l'addition du composant  $C_i$  n'empêche  $M_{i-1}$  de satisfaire  $T_i$ . Le processus de vérification peut être décrit comme ceci :

$$\left\{ \begin{array}{l} \text{verify } C_0 \models T_0 \\ \text{if } C_i, T_i \models S_i \text{ then verify that } C_i, T_i \models T_{i+1} (i = 1, 2, \dots) \end{array} \right.$$

La formulation des propriétés auxiliaires  $T_i$  dépend des caractéristiques du problème. En générale, on peut dire que  $T_i$  conditionne la satisfaction de  $S_i$ , éventuellement d'une façon indirecte. De plus, un ensemble de paramètres peut éventuellement intervenir dans la formulation de la propriété  $T_i$ . Donner des valeurs à ces paramètres est un aspect important de l'approche. Aussi bien la formulation de  $T_i$  que la valuation de ses paramètres sont facilités par l'inspiration physique du modèle d'interaction, comme le montrera l'étude de cas de l'application.

### Expression des propriétés

Deux sortes de propriétés doivent être exprimées, d'une part, la propriété de sûreté qu'on souhaite vérifier et d'autre part, la propriété auxiliaire  $T_i$  qui sera utilisé dans le processus de vérification compositionnelle.

- **Propriété de sûreté** : Celle-ci représente la propriété à vérifier pour valider la sûreté de fonctionnement du système. Le langage SAL inclut la clause *THEOREM* qui permet d'énoncer qu'une propriété est valide par rapport à un système modélisé par un contexte. Ainsi,

$\text{SafetyCondition} : \text{THEOREM } Agent \mid - G(P_s);$

exprime que la formule  $G(P_s)$  est valide relativement au système *Agent*, sachant que  $P_s$  est une formule qui exprime la propriété de sûreté. Le model-checking va établir si la clause *THEOREM* est vérifié ou non.

- **Propriétés auxiliaires** : La propriété auxiliaire  $T_i$  permet de remplacer une instance de  $M_i$  par  $T_i$ , dans la règle de déduction  $R_c$ . Pour déterminer cette propriété, on utilise les conditions de sûreté, obtenues lors de l'analyse du modèle physique d'interaction. Le langage SAL inclut la clause *LEMMA* permettant d'énoncer une hypothèse par rapport à un système donné :

*AuxiliaryProperty* : *LEMMA Agent* |  $\neg G(T_i)$ ;

Cette clause SAL exprime que la formule  $G(T_i)$  est une hypothèse valide pour le contexte *Agent*.

## 8.4.4 Vérification par model-checking

### Le model-checking borné

Parmi les outils inclus dans SAL, le bounded model checker (*sal\_inf\_bmc*) offre la possibilité de vérifier des modèles qui incluent des variables à valeur réelle. Il s'agit donc d'un model checker adapté aux systèmes à espace d'état infini. La profondeur d'exploration est bornée pour éviter la non terminaison de la recherche. Le bounded model checker de SAL effectue une vérification par réfutation, i.e. cherche un contre-exemple. Dans certains cas la recherche est infructueuse : *sal\_inf\_bmc* ne trouve pas de contre exemple dans la profondeur d'exploration choisie. Ceci ne suffit pas à prouver la validité de la propriété vérifiée : un contre-exemple pourrait être trouvé si on explorait jusqu'à une plus grande profondeur.

Pour sortir de cette impasse, *sal\_inf\_bmc* dispose d'une fonctionnalité basée sur la méthode dite de la k-induction [De Moura et al., 2003]. Pour cela, *sal\_inf\_bmc* est exécuté avec l'option  $--induction$ . Dans ce cas, en absence de contre-exemple *sal\_inf\_bmc* peut signaler l'un des deux situations suivantes :

- l'induction réussit, et dans ce cas la validité est établie.
- l'induction a échoué, et dans ce cas il est possible de relancer le model-checking en augmentant la profondeur d'exploration.

Augmenter la profondeur d'exploration peut se traduire par une augmentation très forte de la durée d'exploration. En outre, il y a des cas où l'algorithme d'induction échoue, quelle que soit la profondeur d'exploration. Nous n'avons pas été confrontés à cette situation, dans le cas où nous avons utilisé la méthode compositionnelle, ce qui représente un atout important de celle-ci.

### Processus de vérification

Le processus de vérification est décrit par la récurrence suivante :

$$\left\{ \begin{array}{l} \mathbf{verify} \ C_0 \models T_0 \\ \mathbf{if} \ C_i, T_i \models S_i \ \mathbf{then} \ \mathbf{verify} \ \mathbf{that} \ C_i, T_i \models T_{i+1} \ (i = 1, 2, \dots) \end{array} \right.$$

La figure 8.4 illustre l'application de la récurrence.

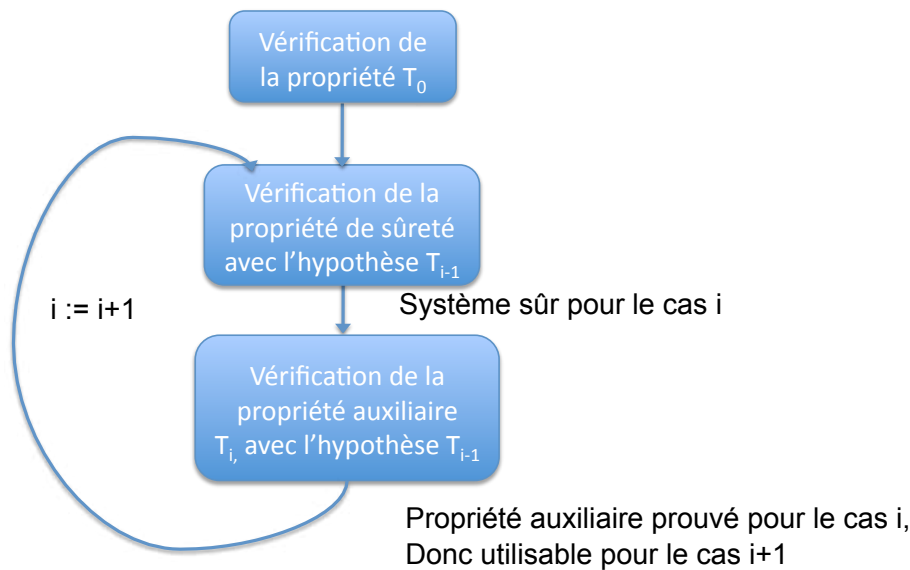


Figure 8.4: Application de la règle de déduction

La vérification compositionnelle débute par le model-checking de  $T_0$ , pour cela, nous appelons le *sal-inf-bmc* sur le système *Agent* :

```
sal-inf-bmc -induction Agent AuxiliaryProperty
```

Dans le cas où le "model checker" ne trouve pas de contre exemple à la propriété auxiliaire, nous pouvons vérifier la sûreté de fonctionnement du système à chaque pas  $i$  :

```
sal-inf-bmc -induction -l AuxiliaryProperty Agent SafetyCondition
```

Ensuite, nous vérifions la propriété auxiliaire pour le pas suivant, pour cela, nous utilisons la propriété auxiliaire *AuxiliaryPropertyPre* correspondant à la propriété  $T_{i-1}$  et la propriété auxiliaire *AuxiliaryPropertyPost* correspondant à la propriété  $T_i$  :

```
AuxiliaryPropertyPre : LEMMA Agent | - G (T_{i-1});
AuxiliaryPropertyPost : LEMMA Agent | - G (T_i);
```

Nous pouvons maintenant vérifier, la propriété auxiliaire  $T_i$  avec l'hypothèse  $T_{i-1}$  :

```
sal-inf-bmc -induction -l AuxiliaryPropertyPre Agent AuxiliaryPropertyPost
```

Pour vérifier la sûreté du système  $M_{i+1}$ , nous redéfinissons la propriété auxiliaire en remplaçant *AuxiliaryProperty* par *AuxiliaryPropertyPost*.

Nous appliquons cette procédure itérative tant que nous n'avons pas effectué la vérification avec le nombre souhaité de composants.

# SPÉCIFICATION ET VÉRIFICATION DE LA CONDUITE EN CONVOI

---

**C**E CHAPITRE propose un modèle de spécification du train de véhicules afin de réaliser la vérification de ses propriétés de sûreté. La vérification du train est réalisée en appliquant l'approche de vérification compositionnelle.

---

## Sommaire

---

<b>9.1</b>	<b>Introduction</b>	<b>112</b>
<b>9.2</b>	<b>Modèle de spécification</b>	<b>112</b>
9.2.1	Modèle de spécification du véhicule	113
9.2.2	Modèle de spécification de la perception	113
9.2.3	Modèle de spécification du contrôle	114
9.2.4	Spécification du modèle dynamique	116
<b>9.3</b>	<b>Traduction d'OZS vers SAL</b>	<b>117</b>
<b>9.4</b>	<b>Vérification de la sûreté de fonctionnement d'un train de véhicules</b>	<b>120</b>
9.4.1	Propriété à vérifier	120
9.4.2	Mise en pratique de la vérification compositionnelle	120
9.4.3	Etude comparative des approches de vérification	122
<b>9.5</b>	<b>Conclusion</b>	<b>123</b>

---



## 9.1 Introduction

Le modèle abstrait de comportement a été défini conformément à la figure 9.1. Le mécanisme physique que nous avons analysé au chapitre 5 correspond à l'évolution du train suivant un mouvement rectiligne. Cela correspond au comportement spécifié par l'état *INSERTED*. Nous avons développé le modèle de spécification, sous la forme d'une classe *OZS*, correspondant à ce mode d'opération de l'agent véhicule. Cela consistera d'une part à spécifier un statechart qui raffine l'état *INSERTED* de la figure 9.1 et d'autre part à définir les opérations de la classe *OZS* à partir du modèle d'interaction.

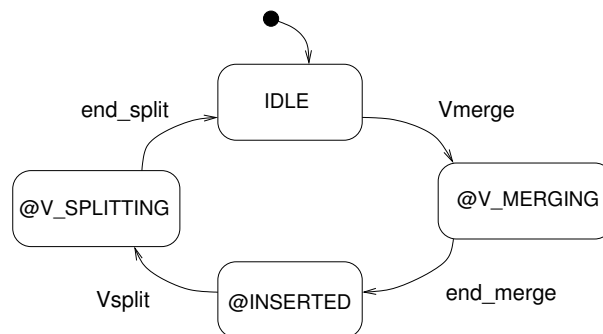


Figure 9.1: Modèle de comportement du véhicule suiveur

## 9.2 Modèle de spécification

Le statechart qui raffine l'état *INSERTED* modélise, à la base, le cycle perception-action de l'agent. Il s'agit des états *Perception* et *VehicleControl* de la figure 9.2. A ces deux activités, nous ajoutons une troisième, représenté par l'état *VehicleModel*, qui modélise la réaction du véhicule en tant que dispositif matériel lorsque les actions de l'agent s'exercent sur celui-ci. La réaction de l'entité matérielle doit être prise en compte pour procéder à la vérification. Cette façon de procéder est courante : la terminologie de la vérification de systèmes réactifs utilise le mot *témoin* (witness) [Michel, 2007] pour désigner ce type de comportement ajouté.

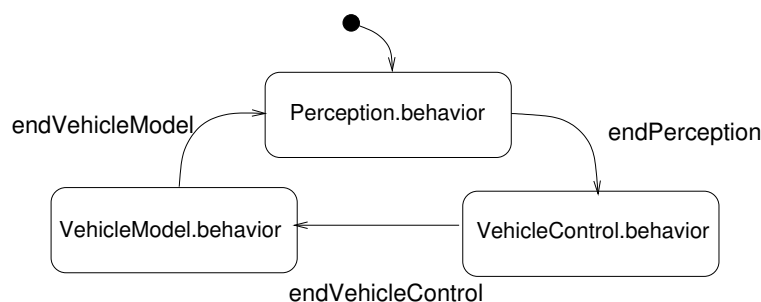


Figure 9.2: Sous statechart associé à l'état *INSERTED*

Chacun de ces trois états correspond donc à une activité :

- *Perception* : mesure de la distance inter-véhicule dans la zone de perception.
- *VehicleControl* : calcul de la nouvelle consigne de vitesse à partir du modèle d'interaction simplifié.
- *VehicleModel* : calcul de la réaction en fonction de ses caractéristiques dynamiques et de la consigne de vitesse.

La spécification formelle consiste à définir une classe *OZS* correspondant au mode de fonctionnement de l'agent que nous venons de décrire.

### 9.2.1 Modèle de spécification du véhicule

L'agent véhicule, dans le mode de fonctionnement inséré, est spécifié par la classe principale *InsertedVehicle*. Cette classe résulte de l'agrégation de trois sous classes : *Perception*, *VehicleControl* et *VehicleModel*.

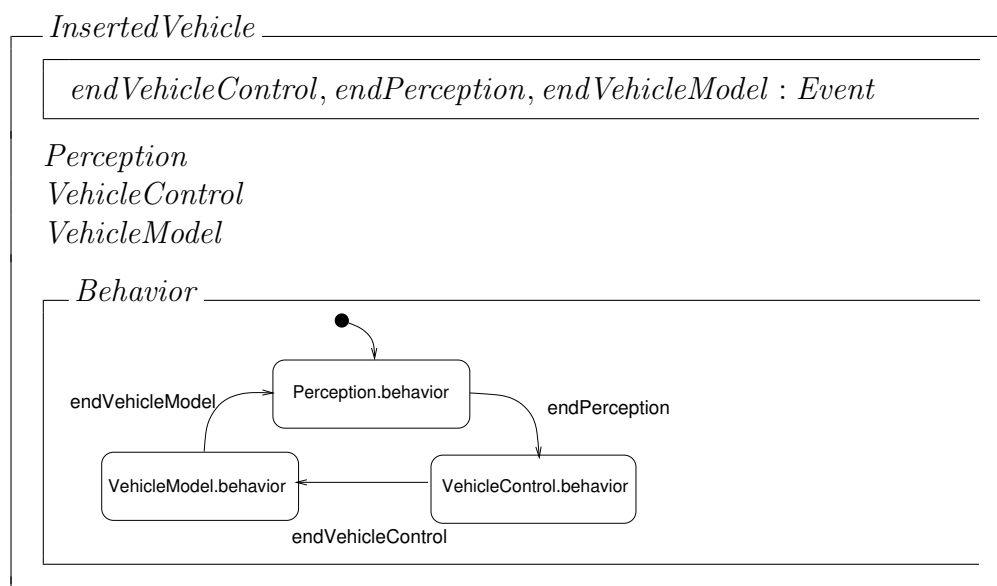


Figure 9.3: Structure de la classe *InsertedVehicle*

### 9.2.2 Modèle de spécification de la perception

La classe *Perception* contient essentiellement la fonction *computeMeasuredDistance* qui détermine pour chaque cycle d'opération la distance entre le véhicule et son prédécesseur. La fonction *computeMeasuredDistance* calcule cette distance à partir de la vitesse du véhicule et de celle du précédent. La vitesse du véhicule est déterminé par le modèle physique (classe *VehicleModel*). La vitesse du précédent est contrainte par les constantes *DeltaSpeedMin* et *DeltaSpeedMax*. La valeur de cette vitesse n'est pas calculée de façon précise. Les variations du véhicule précédent sont hors du contrôle de l'agent et ces

variations donneront lieu à l'ensemble d'états globaux exploré lors du modèle checking (vérification du système multi-agents).

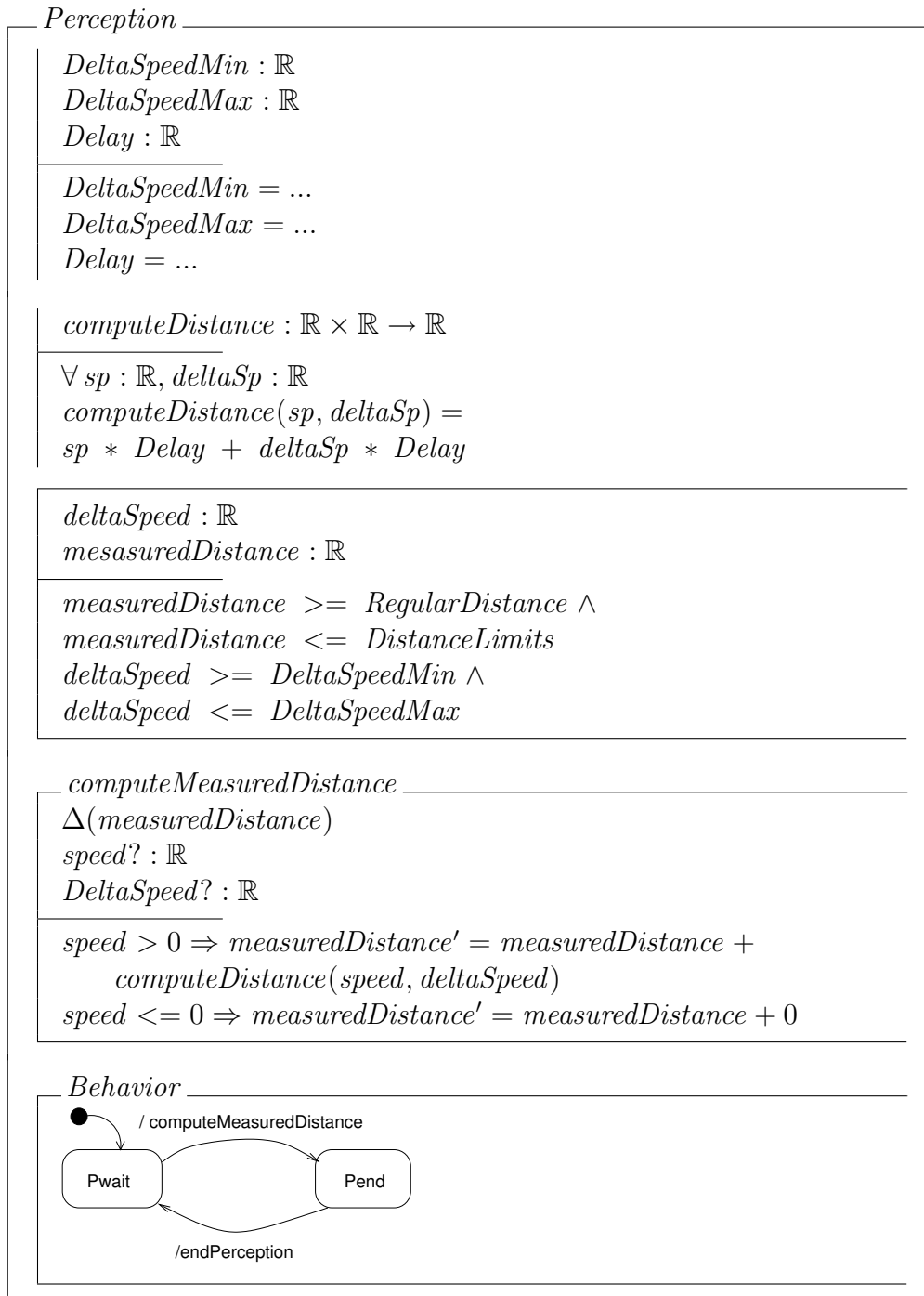


Figure 9.4: Modèle de perception du véhicule

### 9.2.3 Modèle de spécification du contrôle

La class *VehicleControl* spécifie l'activité de l'agent qui consiste à calculer les actions à partir des percepts produits par l'activité de perception. Dans le cas de l'agent véhicule,

l'action consiste à produire une consigne d'accélération qui sera appliquée aux organes physiques du véhicule.

Les fonctions permettant de calculer l'action (accélération) à partir des percepts (distance entre le véhicule et son prédécesseur) sont définies sur la base du modèle physique d'interaction (cf. chapitre 4). Nous rappelons donc que le modèle d'interaction, dans le cas choisi (mouvement rectiligne), se résume à deux forces :

Force du ressort	$F_r = -k(d - l_v)$
Force de l'amortisseur	$F_a = -h \frac{d}{\Delta t}$

Avec  $k$ ,  $h$  les paramètres de raideur et d'amortissement,  $d$  la distance inter-véhicule,  $l_v$  la distance souhaitée entre les véhicules (noté *RegularDistance*) et  $\Delta t$  le pas de discrétisation. La fonction *computeAcc* de class *VehicleControl* est défini à partir de ces forces.

Le calcul de la consigne d'accélération est fait suivant deux cas :

- Distance inférieure à la distance de sécurité : le véhicule provoque un arrêt d'urgence en appelant la fonction *safetyStop*.
- Distance supérieure à la distance de sécurité : le véhicule calcule la consigne en fonction du modèle d'interaction, en appelant la fonction *computeAcceleration*.

*VehicleControl*

$K : \mathbb{R}$   
 $H : \mathbb{R}$   
 $RegularDistance : \mathbb{R}$   
 $Delay : \mathbb{R}$   
 $Mass : \mathbb{R}$   
 $SafetyAcceleration : \mathbb{R}$

$K = \dots$   
 $H = \dots$   
 $RegularDistance = \dots$   
 $Delay = \dots$   
 $Mass = \dots$   
 $SafetyAcceleration = \dots$

$computeAcc : \mathbb{R} \rightarrow \mathbb{R}$

$\forall distance : \mathbb{R} \bullet computeAcc(distance) =$   
 $((K * (distance - RegularDistance) - (H * distance / Delay)) / Mass$

$acceleration : \mathbb{R}$

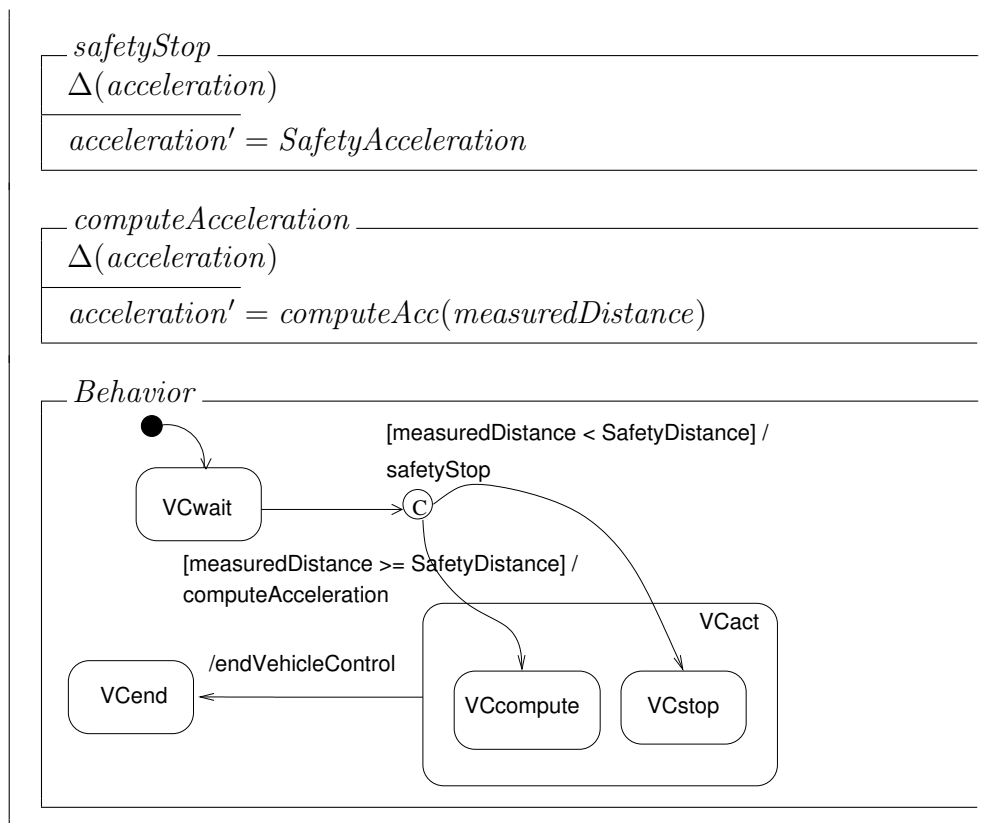
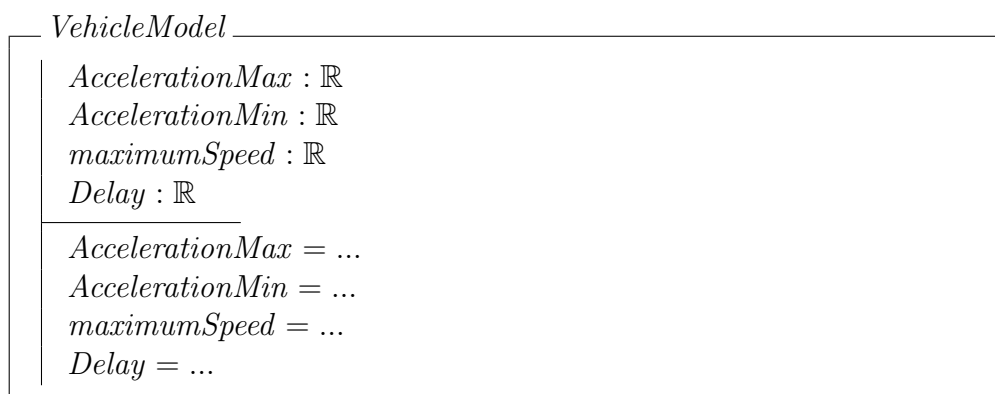


Figure 9.5: Modèle de contrôle du véhicule

### 9.2.4 Spécification du modèle dynamique

La classe *VehicleModel* définit le calcul de la réaction du véhicule en fonction de la consigne d'accélération calculée par la classe *VehicleControl*. La vitesse résultante est calculée par l'opération *computeSpeed*, par rapport aux paramètres dynamiques du véhicule, grâce à la fonction *ComputeCurrentSpeed* qui calcule la vitesse courante en fonction de l'accélération et de la décélération maximale du véhicule. L'opération *computeSpeedLimits* limite la vitesse résultante grâce à la fonction *computeLimits* qui borne la vitesse en tenant compte de la vitesse maximale du véhicule.



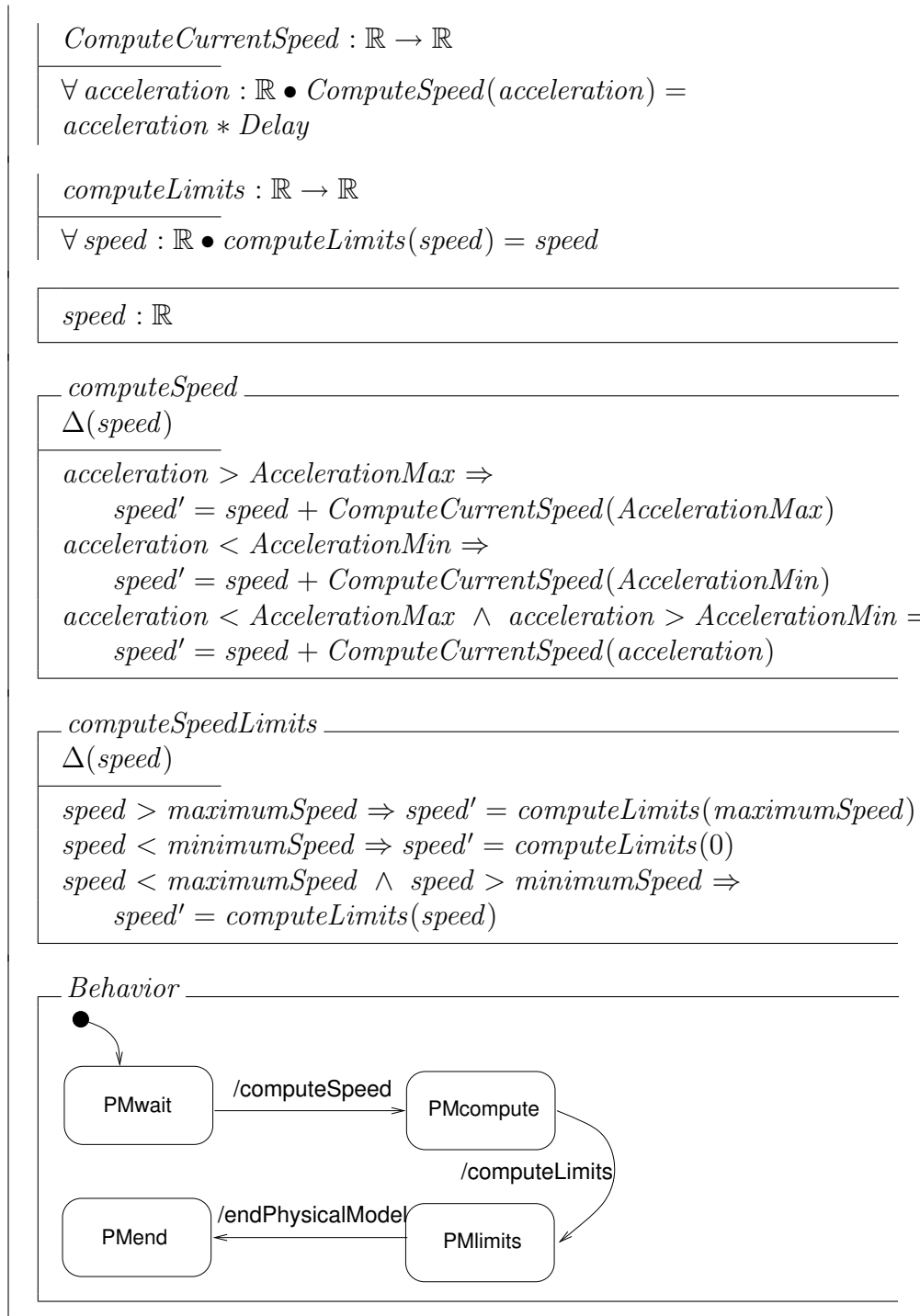


Figure 9.6: Modèle dynamique du véhicule

### 9.3 Traduction d'OZS vers SAL

Avant vérification, le modèle de spécification *OZS* est traduit sous la forme d'un système de transition SAL, conformément à la sémantique du langage *OZS*. Le modèle de vérification SAL est réalisé selon l'approche de transcription définie dans le chapitre 7.

A la classe *OZS InsertedVehicle* correspond le contexte *Vehicle*. De la même manière, les modules SAL *Perception*, *VehicleControl* et *VehicleModel*, correspondent respectivement aux classes *OZS Perception*, *VehicleControl* et *VehicleModel*. Le module *InsertedVehicle* correspond à la classe principale *OZS* et intègre, par composition asynchrone, les sous modules.

```

Vehicle : CONTEXT =
BEGIN
%% Constants definition
%% Types definition
%% Functions definition
VehicleModel : MODULE = ...
VehicleControl : MODULE = ...
Perception : MODULE = ...
InsertedVehicle : MODULE = VehicleModel [] VehicleControl [] Perception ;
END ;

```

Le contexte *Vehicle* inclut les définitions des constantes, des types et des fonctions utilisés par les différents modules :

- **Définition des constantes** : cette partie du contexte permet de définir les constantes du modèles dynamique (masse, accélération et vitesse maximum, ... ) et les constantes du modèle d'interaction (distance de suivi souhaitée, ... ).

```

%% Constants definition
SafetyDistance : REAL = ... ;
AccelerationMin : REAL = ... ;
AccelerationMax : REAL = ... ;
Mass : REAL = ... ;
RegularDistance : REAL = ... ;
.....

```

- **Définition des types** : Afin de réaliser la transcription des statecharts de comportement de chaque sous module, le contexte *Vehicle* défini trois types correspondant respectivement aux états de chaque statechart défini dans le modèle de comportement des classes OZS.

```

%% Types definition
PerceptionState : TYPE = Pwait,Pend ;
VehicleControlState : TYPE = VCwait,VCcompute,VCstop, VCend ;
VehicleModelState : TYPE = PMwait,PMcompute,PMLimits,PMend ;

```

- **Définition des fonctions** : le contexte SAL défini pour chaque fonction des classes OZS, une fonction SAL permettant de réaliser les calculs de comportement des modules SAL. Par exemple, la fonction *computeAcceleration* permet de calculer la nouvelle consigne d'accélération par rapport à la distance inter-véhicule mesurée

(*\_distance*) :

```
%% Functions definition
computeAcceleration(_distance : REAL) : REAL = (K*(_distance-RegularDistance) - (H*_distance)/Delay)/Mass ;
....
```

- **Définition des modules** : chaque classe OZS correspond à un module SAL. Par exemple, le module *VehicleControl* contient un système de transitions obtenues à partir du statechart *VehicleControl.behavior*. La valeur de la variable locale *vcState* correspondra à l'état actif du statechart. Les variables des modules sont définies soit comme des variables d'entrée (*INPUT*), soit comme des variables de sortie (*OUTPUT*). Les variables communes entre les différents modules sont déclarées *GLOBAL* et leur valeur est accessible à tous les modules en lecture et écriture.

```
VehicleControl : MODULE =
BEGIN
  LOCAL vcState : VehicleControlState
  GLOBAL endAction, endPerception, endPhycalModel : BOOLEAN
  INPUT measuredDistance : REAL
  OUTPUT acceleration : REAL
  INITIALIZATION acceleration = 0 ; vehState = VCwait
  TRANSITION
  [
    vcState = VCwait AND endPerception AND
    measuredDistance < SafetyDistance -- >
    acceleration' = SafetyAcceleration ;
    vcState' = VCact
    endPerception' = FALSE
  ]
  [
    vcState = VCwait AND endPerception AND
    measuredDistance >= SafetyDistance -- >
    acceleration' = computeAcceleration ?(measuredDistance) ;
    vcState' = VCact
    endPerception' = FALSE
  ]
  [
    vcState = VCact -- >
    endVehicleControl' = TRUE ;
    vcState' = VCend
  ]
END ;
```



## 9.4 Vérification de la sûreté de fonctionnement d'un train de véhicules

### 9.4.1 Propriété à vérifier

Nous souhaitons vérifier que le système tel qu'il est spécifié par le modèle formel, satisfait la propriété suivante :

Etant donné un train de  $n$  véhicules ( $n = 2,3,4 \dots$ ), le  $n$ -ième véhicule ne rentre pas en collision avec le véhicule qui le précède.

La difficulté de cet objectif réside dans le fait que même pour des valeurs de  $n$  relativement petite ( $n \geq 4$ ), la vérification du système dans sa globalité est empêchée par l'explosion combinatoire de l'espace d'état. Le paragraphe 9.4.3 présentera ce phénomène par des données chiffrées.

D'autre part, des travaux de vérification de la conduite en convoi basé sur des modèles globaux (sans vérification compositionnelle) ont été effectués récemment [Colin et al., 2008, 2009, Lanoix, 2008, Scheuer et al., 2008]. Ces travaux se basent sur une spécification formelle à l'aide de la méthode  $B$  et évitent l'explosion combinatoire en utilisant la preuve comme méthode de vérification. Le prix payé est l'absence d'une représentation pour les nombres réels qui oblige à une discrétisation des variables.

L'approche de vérification compositionnelle que nous avons présentée au chapitre 8 permet de contourner cette difficulté : le train de  $n$  véhicules est remplacé par une propriété auxiliaire et la vérification se fait sur l'espace d'état produit par le modèle d'un unique véhicule. Dans ces conditions, indépendamment du nombre de véhicules, la propriété de sûreté exprimant l'absence de collision peut toujours être formulée, conformément à la syntaxe de SAL, de la façon suivante :

$SafetyProperty : THEOREM Vehicle \mid \neg G(measuredDistance \geq SafetyDistance);$

Cette propriété correspond à la condition suivante : distance inter-véhicule ( $measuredDistance$ ) plus grande que la distance de sécurité ( $SafetyDistance$ ). La notation  $G(P)$  exprime l'invariance de la propriété  $P$  pour tous les états du système.

### 9.4.2 Mise en pratique de la vérification compositionnelle

Nous rappelons que la vérification compositionnelle se base sur la règle de déduction suivante pour  $i > 0$  :

$$R_c : \frac{C_i, T_i \models S_i}{C_i \parallel M_{i-1} \models S_i} \quad (9.1)$$

où  $C_i$  est le modèle du composant  $i$ ,  $T_i$  est une propriété auxiliaire relative au système  $M_{i-1}$ , constitué des composants  $C_1, \dots, C_{i-1}$  et  $S_i$  la propriété de sûreté relative au système  $C_i \parallel M_{i-1}$  constitué à l'ajout du  $C_i$  au système  $M_{i-1}$ . A partir de cette règle, un processus de vérification compositionnelle peut être mis en place :

$$\left\{ \begin{array}{l} \text{verify } C_0 \models T_0 \\ \text{if } C_i, T_i \models S_i \text{ then verify that } C_i, T_i \models T_{i+1} \text{ (} i = 1, 2, \dots \text{)} \end{array} \right. \quad (9.2)$$

Initialement, on montre qu'un unique composant  $C_0$  satisfait la propriété  $T_0$ . Ensuite, de façon itérative, on utilise la règle  $R_c$ , pour montrer que la propriété de sûreté est satisfaite par  $C_i$ , sous l'hypothèse  $T_i$ . Puis, pour procéder à une nouvelle itération, nous vérifions que l'ajout du comportement  $C_i$  sous l'hypothèse  $T_i$ , satisfait une nouvelle propriété auxiliaire  $T_{i+1}$  qui sera utilisée pour la prochaine itération.

Nous allons appliquer cette procédure à la vérification de la propriété *SafetyProperty* relativement au train de véhicules. Ceci implique que les composants  $C_i$  sont identiques, puisqu'ils modélisent le comportement d'un agent véhicule. Dans notre cas, la propriété auxiliaire  $T_i$  a été choisi pour exprimer les limites de la variation de vitesse du véhicule  $i$ , donné par l'équation :

$$T_i \equiv \Delta SpeedMin_i \leq \Delta Speed_i \leq \Delta SpeedMax_i \quad (9.3)$$

Les valeurs de  $\Delta SpeedMin_0$  et de  $\Delta SpeedMax_0$ , correspondant au cas  $T_0$ , sont données par l'analyse du modèle physique d'interaction conformément à l'équation 5.12. En effet, cette analyse, établit la condition initiale  $C_0 \models T_0$  ( $\Delta SpeedMin_0 = -0.1$  et  $\Delta SpeedMax_0 = 0.13$ ).

Pour exprimer les propriétés auxiliaires  $T_i$  et  $T_{i+1}$ , de nouvelles constantes sont déclarées dans le contexte *Vehicle* :

```
%% Constants definition
DeltaSpeedMinPre : REAL = ... ;
DeltaSpeedMaxPre : REAL = ... ;
DeltaSpeedMinPost : REAL = ... ;
DeltaSpeedMaxPost : REAL = ... ;
%% Global variable declaration
deltaSpeed : REAL ;
```

où les valeurs des constantes  $\Delta SpeedMinPre$ ,  $\Delta SpeedMaxPre$  sont utilisées dans la formulation de  $T_i$  et  $\Delta SpeedMinPost$ ,  $\Delta SpeedMaxPost$  dans la formulation de  $T_{i+1}$  :

```

AuxiliaryPropertyPre : THEOREM Vehicle | - G (deltaSpeed >= DeltaSpeedMinPre AND deltaSpeed <=
DeltaSpeedMaxPre);
AuxiliaryPropertyPost : THEOREM Vehicle | - G (deltaSpeed >= DeltaSpeedMinPost AND deltaSpeed <=
DeltaSpeedMaxPost);

```

On commence le processus de vérification en effectuant le model checking de la propriété de sûreté :

```

sal-inf-bmc -depth=50 -induction -l AuxiliaryPropertyPre Vehicle Safetyproperty

```

L'option `-l` permet d'inclure la propriété auxiliaire *AuxiliaryPropertyPre*, comme hypothèse de vérification.

Ensuite, on vérifie les propriétés auxiliaires qui vont servir pour le cas suivant (*AuxiliaryPropertyPost*) :

```

sal-inf-bmc -depth=50 -verbose=5 -induction -l AuxiliaryPropertyPre Vehicle AuxiliaryPropertyPost

```

A partir de cette vérification, on transfère la valeur des constantes *DeltaSpeedMinPost* et *DeltaSpeedMaxPost* à *DeltaSpeedMinPre* et *DeltaSpeedMaxPre*.

Les itérations s'arrêtent soit parce qu'on arrive au nombre de véhicule souhaité, soit parce qu'un contre exemple est trouvé par le modèle checker. Dans le deuxième cas, on peut être amené à augmenter la distance de sécurité et à revérifier.

### 9.4.3 Etude comparative des approches de vérification

Nous avons voulu établir la pertinence de la vérification compositionnelle en effectuant une comparaison avec la vérification non compositionnelle. Cette dernière consiste à vérifier un modèle qui résulte de la composition d'un nombre  $n$  d'agents véhicule.

#### Vérification non compositionnelle

Le modèle de vérification est obtenu en indexant le modèle d'un agent. Ainsi, le comportement du système peut être modéliser par :

```

SMAVehicle : CONTEXT =
BEGIN
SMAVehicleBehavior : MODULE = Vehicle(1) [] ... [] Vehicle(n);
END;

```

Le tableau 9.7 présente le nombre de nœuds explorés en fonction du nombre de véhicules.

Nombre de véhicules	Nombres de nœuds explorés
2	34912
3	70242
4	Explosion des cas
5	Explosion des cas

Figure 9.7: Tableau de la relation entre le nombre de véhicules et nombre de nœuds parcourus lors d'une vérification non compositionnelle

Nous constatons qu'à partir de quatre véhicules, le model-checking semble ne pas se terminer. Le model-checker s'est exécuté pendant cinq jours complets sans s'arrêter lors de cette expérience.

### Vérification compositionnelle

La vérification compositionnelle se base sur l'approche définie dans le chapitre précédent. Le tableau 9.8 présente le nombre de nœuds explorés pour vérifier la propriété de sûreté.

Nombre de véhicules	Nombres de nœuds explorés
2	34912
3	34273
4	34589
5	34380

Figure 9.8: Tableau de la relation entre le nombre de véhicules et nombre de nœuds parcourus lors d'une vérification compositionnelle

On peut constater que le nombre de nœuds visités est relativement constant. Ceci s'explique par le fait qu'à chaque cas de vérification, le model checker explore le même modèle de comportement, celui d'un agent véhicule. Le reste du système est pris en compte par la propriété auxiliaire.

## 9.5 Conclusion

Ce chapitre présente la vérification d'une propriété de sûreté : absence de collision entre véhicules. Nous avons vérifié le comportement d'un train, allant jusqu'à 10 véhicules. Pour une distance de sécurité donnée, nous avons pu vérifier la propriété de sûreté pour un train de 10 véhicules. Ensuite, nous avons augmenté la distance de sécurité et le model checker a donné un contre exemple pour un train de 4 véhicules. Par exemple, dans le cas d'un train de 4 véhicules avec une distance de sécurité supérieure à un mètre et une distance de suivi

de 180 cm, des contre exemples sont trouvés par le model checker.

Ces expériences montrent la pertinence de l'approche de vérification en tant qu'outil de dimensionnement : des scénarii de fonctionnement peuvent être défini en donnant des valeurs aux multiples paramètres du modèle pour déterminer les meilleures configurations qui satisfont la propriété de sûreté.

---

TROISIÈME PARTIE

---

# **Simulation et expérimentation**

---



# LES PLATE-FORMES DE SIMULATION ET D'EXPÉRIMENTATION

---

CE chapitre présente les plates-formes utilisées pour tester le fonctionnement des véhicules et leurs capacités d'évoluer au sein du convoi. La première plate-forme est un simulateur 3D qui intègre un moteur physique, lequel permet d'exécuter des scénarii très proches de la réalité. La deuxième plate-forme est constituée de véhicules dont le système de commande et l'intelligence embarquée ont été conçu au laboratoire SeT.

---

## Sommaire

---

<b>10.1 Introduction</b>	<b>128</b>
<b>10.2 Plate-forme de simulation 3D</b>	<b>128</b>
10.2.1 Présentation de la plate-forme	128
10.2.2 Architecture du simulateur	129
<b>10.3 Plate-forme d'expérimentation</b>	<b>131</b>
10.3.1 Présentation de la plate-forme	131
10.3.2 Capteurs	133
<b>10.4 Protocole de simulation et d'expérimentation</b>	<b>134</b>
10.4.1 Protocole de simulation	135
10.4.2 Protocole d'expérimentation	136

---



## 10.1 Introduction

L'expérimentation joue un rôle complémentaire par rapport à la vérification. D'une part la vérification repose sur des modèles qui comportent des abstractions : certains aspects de la réalité physique ne sont pas pris en compte. D'autre part, la vérification a été faite par rapport à un cas particulier de l'évolution : train de véhicules se déplaçant en ligne droite. Les expérimentations permettront de s'approcher de la réalité matérielle mais aussi de tester un éventail plus large de situations (trajectoire arbitraire, présence d'obstacle, ...). Nous envisageons des expérimentations sur deux types de plate-formes :

- **Simulateur 3D temps réel** : la première plate-forme est un simulateur 3D en temps réel permettant d'exécuter des scénarii de tests proches de la réalité, tenant en compte de la dynamique de chaque véhicule, des contacts roues/sols, ...
- **Véhicule électriques** : ces véhicules ont été développés au laboratoire SeT sur la base de voitures électriques GEM CAR. Celles-ci ont été dotées de systèmes de commande et d'une infrastructure d'informatique embarquée qui permet l'implantation d'algorithmes pour la navigation multi-véhicules.

Nous avons défini un protocole d'expérimentation permettant d'exécuter les mêmes scénarii en simulation et en expérimentation sur véhicule, et de comparer les résultats obtenus.

## 10.2 Plate-forme de simulation 3D

### 10.2.1 Présentation de la plate-forme

L'outil de simulation 3D a été développé dans le cadre du projet CRISTAL<sup>1</sup>. Cet outil a été entièrement réalisé au sein du laboratoire SET. Le simulateur CRISTAL est un environnement de simulation 3D adapté à la fois à la visualisation et à la simulation de véhicules en convoi.

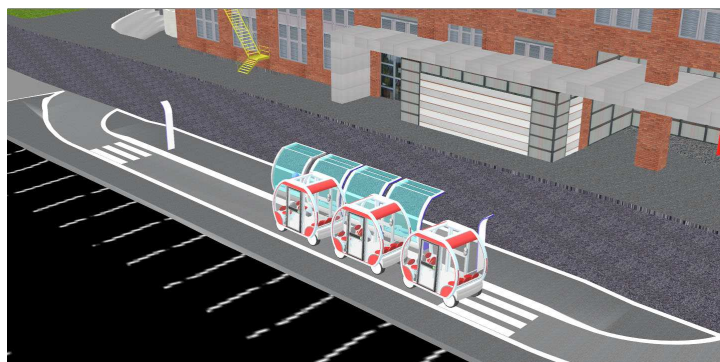


Figure 10.1: Simulation 3D temps réelle d'un train de véhicules en station

<sup>1</sup><http://www.projet-cristal.net/>

Le premier but de cet environnement est de simuler et d'étudier de façon aussi fine que possible le fonctionnement de la solution de platooning proposée. Il s'agit également d'un outil de communication permettant de visualiser l'évolution d'un train de véhicules en milieu urbain dense (cf. figure 10.1).

### 10.2.2 Architecture du simulateur

Pour que le simulateur permette d'obtenir des résultats pertinents son architecture doit comporter des éléments tels que : la perception de l'environnement, la fusion des informations perçues, le calcul de la loi de commande et la réaction du véhicule face à celle-ci. En particulier, il faut simuler de manière précise le fonctionnement des capteurs, le comportement dynamique des véhicules en fonction de leurs caractéristiques techniques et le contrôle du platooning.

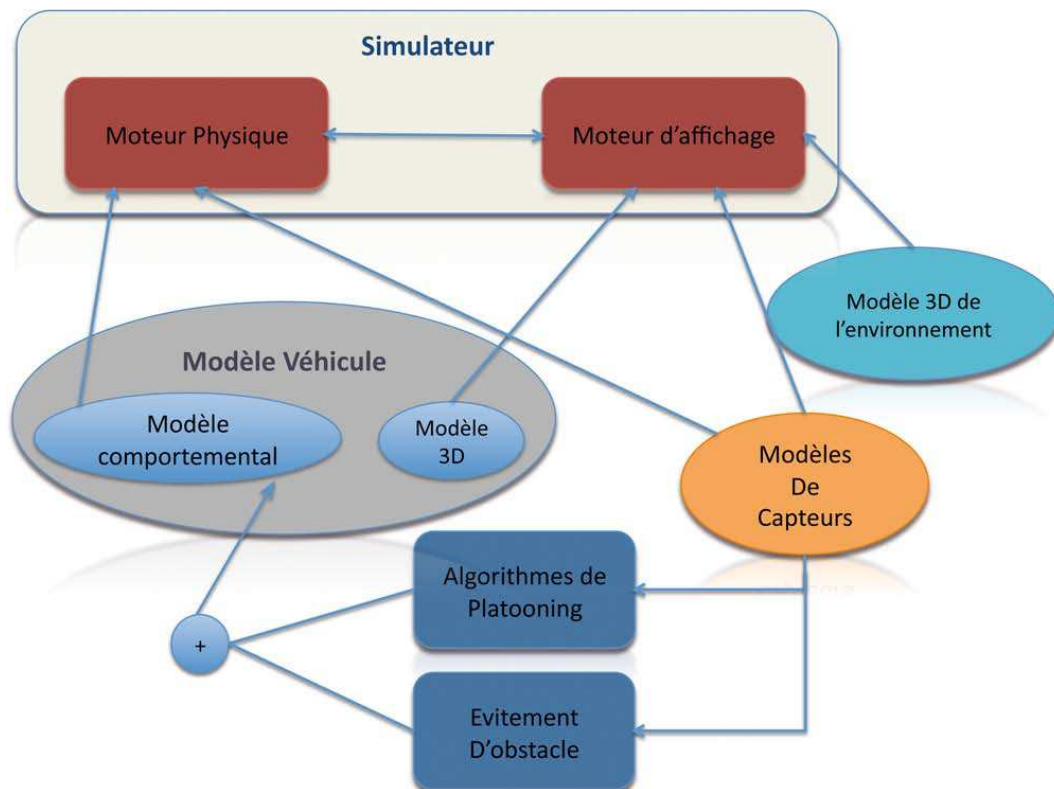


Figure 10.2: Architecture du simulateur 3D temps réel

Par conséquent, nous avons adopté une architecture spécifique (cf. figure 10.2), s'articule autour de deux éléments principaux :

**Moteur d'affichage** : permet d'importer et d'afficher les modèles 3D de l'environnement, les véhicules et leurs composants (roues, châssis, ...). Ce moteur est esclave du moteur physique i.e. attend les déplacements des objets 3D.

**Moteur physique** : basé sur l'outil *PhysX*<sup>2</sup>, permettant de simuler les interactions entre les différents constituants (cf. figure 10.3) d'un véhicule (roues, amortisseurs, châssis, ...) et d'une scène (nature du sol, ...) à partir de leurs propriétés physiques (coefficient de frottement, raideur des ressorts, ...) et de la nature de leurs liaisons mécaniques (pivot, rotule, ...). La réalisation du moteur physique a été réalisée au cours de cette thèse en portant une attention particulière à la modélisation la plus fine possible du comportement du véhicule.

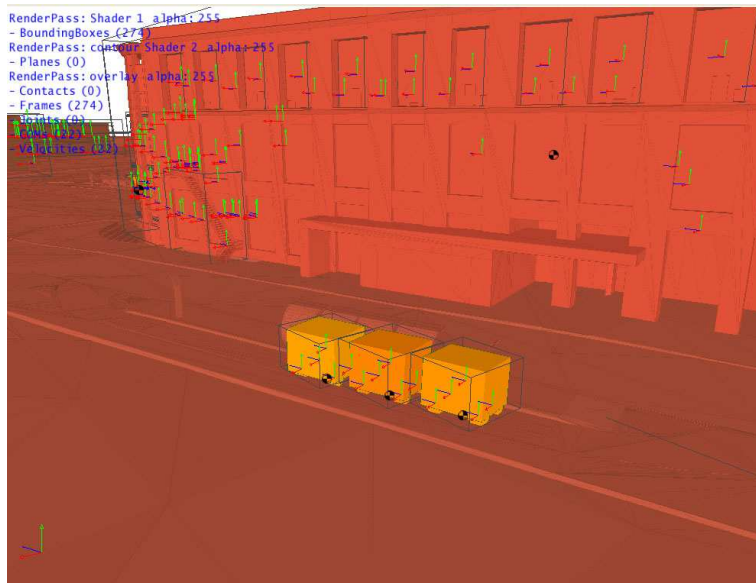


Figure 10.3: Modèle physique d'un train de véhicules en station

Ces deux moteurs communiquent par un protocole spécifique permettant une utilisation sur plusieurs machines, améliorant ainsi la rapidité de la simulation temps réel.

Les deux principaux éléments utilisent les sous composants suivants :

**Modèle du véhicule** : Le véhicule est représenté dans le simulateur par deux modèles : le modèle 3D et le modèle physique. Le modèle 3D correspond à la représentation visuelle du véhicule et à son apparence dans le monde virtuel. Le modèle physique correspond à une description structurale et hiérarchique de tous les éléments mécaniques constituant le véhicule. La figure 10.4 représente la modélisation *PhysX* d'un véhicule. Cette modélisation correspond à la composition d'un châssis (forme rectangulaire) et quatre amortisseurs roues. Ces systèmes simulent la liaison mécanique d'un amortisseur reliant une roue au châssis. Cette roue est modélisée à l'aide d'un modèle de Dugoff et al. [1970] permettant une représentation, la plus fine possible de l'adhérence, de l'amortisseur, ... des véhicules.

<sup>2</sup>[http://www.nvidia.fr/object/physx\\_new\\_fr.html](http://www.nvidia.fr/object/physx_new_fr.html)

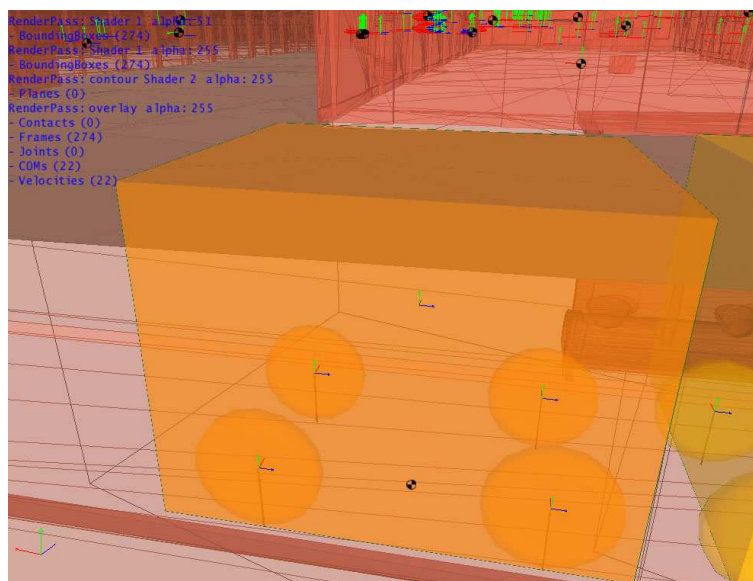


Figure 10.4: Modèle *PhysX* d'un véhicule

**Modèle des capteurs** : Les capteurs sont simulés, selon leurs propriétés intrinsèques, à partir d'observations issues soit du monde 3D soit du monde physique. Par exemple, les capteurs de type télémètres lasers prennent la scène 3D comme source d'information. Dans notre simulation, les véhicules sont équipés de ce type de capteurs pour percevoir l'environnement et en particulier la distance les séparant du véhicule précédent.

**Fonction de platooning** : Dans cette architecture, la fonction de platooning prend comme paramètres les informations des capteurs et renvoie une consigne de distance et de vitesse au véhicule simulé. L'algorithme de platooning ainsi codé est exactement le même que celui exécuté sur le véhicule réel. En effet, les capteurs et la loi de commande du véhicule simulés étant les mêmes que sur le véhicule réel, l'algorithme est identique pour les deux plates-formes.

**Fonction d'évitement d'obstacle** : Cette fonction peut-être activée durant la simulation d'un train de véhicules. Elle permet, en fonction des obstacles perçus par le véhicule, de préempter la commande manuelle du conducteur pour éviter des obstacles.

## 10.3 Plate-forme d'expérimentation

### 10.3.1 Présentation de la plate-forme

Le laboratoire SeT dispose de deux véhicules électriques de type GEM (cf. figure 10.5). Ces véhicules ont été automatisés et peuvent être contrôlés par un système informatique embarqué.



Figure 10.5: Véhicules intelligents du laboratoire SET

L'asservissement est réalisé à l'aide d'actionneurs qui commandent le freinage, la direction, et la vitesse du véhicule.

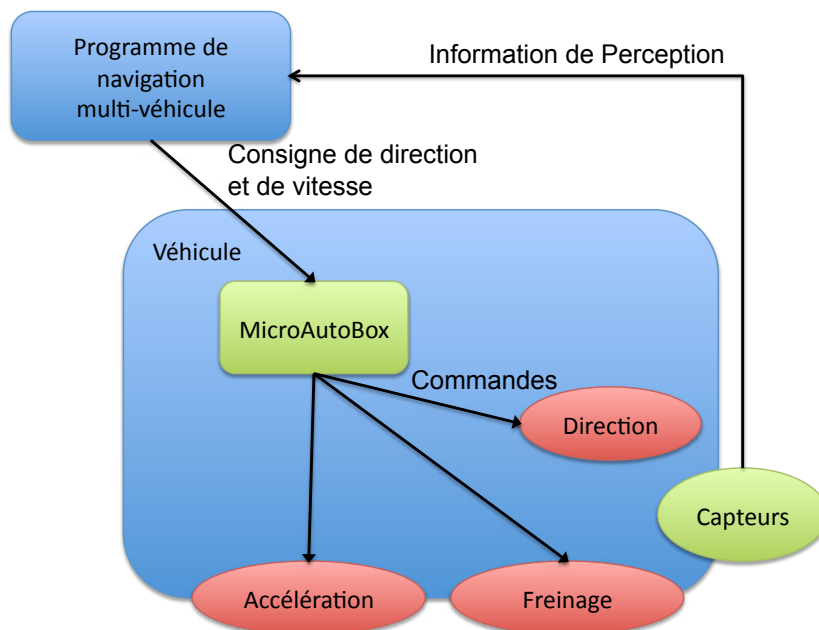


Figure 10.6: Architecture du système embarqué du véhicule

La figure 10.6 présente le système embarqué du véhicule. Un ordinateur embarqué envoie les consignes de direction et de vitesse à la carte de contrôle temps réel *MicroAutoBox*<sup>3</sup> qui commande les actionneurs. La carte de contrôle prend en charge l'asservissement de la vitesse du véhicule.

<sup>3</sup><http://www.dspace.de/ww/fr/fra/home/products/hw/micautob.cfm>

### 10.3.2 Capteurs

Le véhicule perçoit son environnement à l'aide de capteurs et utilise ces perceptions pour calculer les nouvelles consignes de direction et de vitesse (cf. figure 10.7).



Figure 10.7: Capteurs installés sur le véhicule du laboratoire SET

Les véhicules disposent de deux types de capteurs :

**Télémètres lasers** : Deux télémètres lasers sont installés sur le devant du véhicule (cf. figure 10.7 et 10.8). Le premier (télémètre noir : IBEO, cf. annexes D.1.2) est un capteur de dernière génération spécialement adapté aux véhicules et possédant une ouverture faible (de 60 à 100°) avec une fréquence rapide (25 Hz). Le deuxième (télémètre gris : LMS 211, cf. annexes D.1.1), est un capteur industriel permettant une perception très large (180°). Nous avons choisi d'installer ces deux types de capteurs permettant une perception complète et redondante suivant leurs propriétés.

**GPS** : La localisation du véhicule est réalisée à l'aide d'un GPS de type RTK <sup>4</sup> (Real Time Kinematic). Ce type de GPS permet une localisation centimétrique dans la zone de couverture radio de la base.

<sup>4</sup>Proflex 500 <http://www.promagellangps.com/fr/products/product.asp?PRODID=1298>

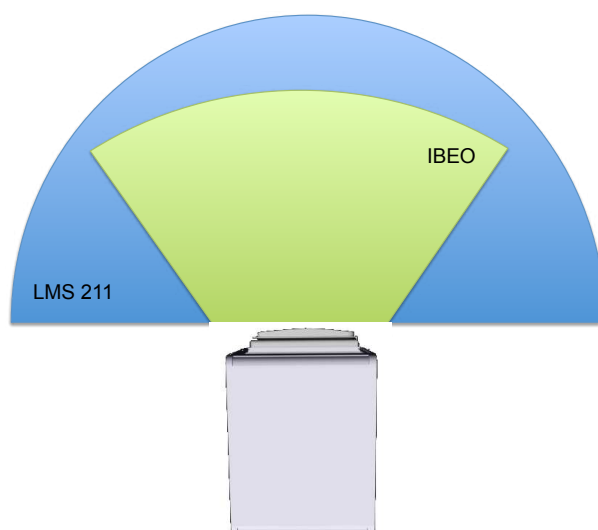


Figure 10.8: Zone de perception du véhicule

La perception de la distance entre véhicules et de l'angle se fait par télémétrie laser. Pour améliorer cette mesure, nous avons installé une balise réfléchissante à l'arrière du véhicule suivi (cf. figure 10.9).



Figure 10.9: Balise réfléchissante

Cette balise est composée de trois dossards. Celui du milieu permet de mesurer la distance entre véhicules. Les dossards de gauche et de droite permettent de calculer l'angle  $\theta$  (cf. 4.6).

## 10.4 Protocole de simulation et d'expérimentation

Les expérimentations ont été réalisées sur le site du Technopôle<sup>5</sup> de la ville de Belfort. Les simulations ont été réalisées sur un modèle 3D du même site.

<sup>5</sup><http://www.mairie-belfort.com/economie/technopole.html>



Figure 10.10: Scénario de simulation 3D et d'expérimentation réelle

La figure 10.10 présente le circuit (courbe en blanc) utilisé pour la simulation et l'expérimentation du train de véhicules. Il a été sélectionné puisqu'il permet de faire évoluer le train sur une longue distance dans un environnement urbain, sur une trajectoire qui comporte différents rayons de courbures.

### 10.4.1 Protocole de simulation

Nous nous sommes fixé comme objectif de faire en sorte que les résultats de la simulation soient comparables avec les résultats de l'expérimentation. Ainsi, la perception de chaque véhicule est effectuée grâce à la simulation d'un télémètre laser ayant les mêmes caractéristiques que le capteur IBEO (cf. annexes D.1.2). De plus, nous utilisons le même paramétrage représenté par les valeurs suivantes :

Masse	500 kg
Vitesse maximale	8 km/h (2 m/s)
Accélération	0.1 m/s <sup>2</sup>
Décélération	-0.1 m/s <sup>2</sup>
Décélération maximum	-3 m/s <sup>2</sup>
Distance de sécurité	50 cm
RegularDistance	180 cm

Figure 10.11: Paramètres dynamiques du véhicule et du modèle de contrôle de suivi



## 10.4.2 Protocole d'expérimentation

L'algorithme implémenté pour la simulation a été repris intégralement et dans le même langage pour faciliter la comparaison entre les résultats de la simulation et de l'expérimentation. Cependant des changements ont été effectués pour adapter l'algorithme au véhicule :

- **Paramètres variables** : Durant les expérimentations, nous nous sommes aperçus que la variation des paramètres du modèle d'interaction n'était pas assez importante. En particulier, une raideur  $k$  plus importante du ressort contribue à un meilleur démarrage du véhicule. Cela nous a conduit à ajouter une valeur minimum  $k'$  pour ce paramètre :

$$k = k' + \frac{m * \dot{Y}^2}{Y^2} \quad (10.1)$$

avec  $k'$  la valeur de  $k$  calculée pour obtenir au moins  $2 \text{ km/h}$  lors du démarrage. Soit  $k' = 400$ .

- **Augmentation de la distance de suivi** : les expérimentations ont été réalisées avec une distance de suivi plus importante afin de ne pas risquer une collision pouvant entraîner des dommages irréversibles sur les véhicules. Nous avons ainsi fixé la distance de suivi souhaitable à 4 m et la distance de sécurité à 1.5 m.

A partir de ces légers changements permettant une adaptation au véhicule réel, nous pouvons réaliser des simulations et expérimentations comparables.

# SIMULATION ET EXPÉRIMENTATION

---

CE chapitre présente les résultats de la simulation et de l'expérimentation de plusieurs scénarios de conduite en convoi et d'évitement d'obstacle.

---

## Sommaire

---

<b>11.1 Introduction</b>	<b>138</b>
<b>11.2 Evaluation de la distance inter-véhicule</b>	<b>138</b>
11.2.1 Simulation : évaluation de la distance inter-véhicule	139
11.2.2 Expérimentation : évaluation de la distance inter-véhicule	141
11.2.3 Bilan : évaluation de la distance inter-véhicule	143
<b>11.3 Evaluation de l'erreur latérale</b>	<b>143</b>
11.3.1 Simulation : évaluation de l'erreur latérale	143
11.3.2 Expérimentation : évaluation de l'erreur latérale	145
11.3.3 Bilan : évaluation de l'erreur latérale	146
<b>11.4 Evitement d'obstacle simple</b>	<b>147</b>
11.4.1 Simulation : évitement obstacle simple	148
11.4.2 Expérimentation : évitement obstacle simple	149
<b>11.5 Evitement d'obstacle multiple</b>	<b>150</b>
11.5.1 Simulation : obstacles multiples	150
11.5.2 Expérimentation : obstacles multiples	151
<b>11.6 Conclusion</b>	<b>152</b>

---

## 11.1 Introduction

Ce chapitre présente l'ensemble des tests effectués aussi bien en simulation et en expérimentation pour évaluer la qualité de fonctionnement de la navigation multi-véhicules, suivant notre approche par les systèmes multi-agents réactifs. Les cas suivants ont été abordés :

- évaluation de la distance inter-véhicule : mesure de la distance séparant les véhicules deux à deux, par rapport à la distance de suivi souhaitée, pendant l'évolution du train de véhicules (cf. figure 11.1).
- évaluation de l'écart latéral : mesure de la distance entre les trajectoires du centre géométrique d'un véhicule par rapport à la même trajectoire de son prédécesseur (cf. figure 11.2). On sélectionne des points sur la trajectoire du premier véhicule, on trace les normales de ces points et on mesure la distance entre le point choisi et le point d'intersection avec la trajectoire du prédécesseur.
- test du système d'évitement d'obstacle : présentation de la trajectoire prise par le véhicule par rapport à la position des obstacles se situant dans son environnement.

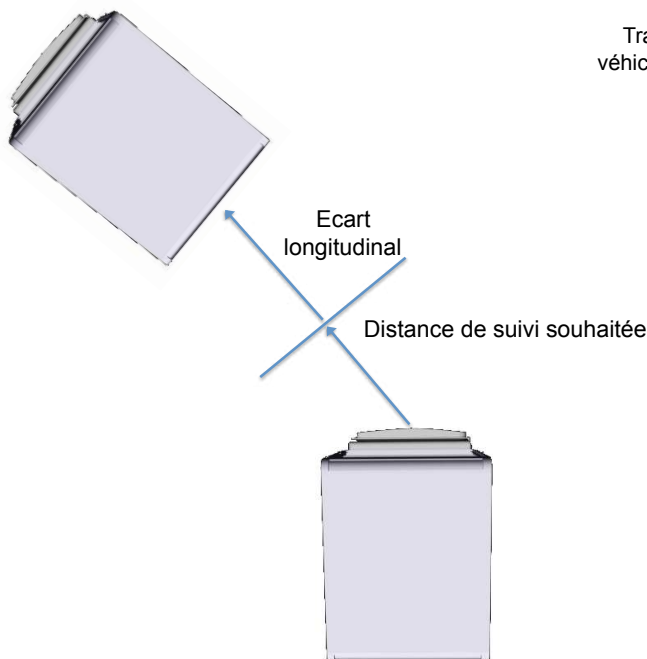


Figure 11.1: Evaluation de l'écart longitudinal

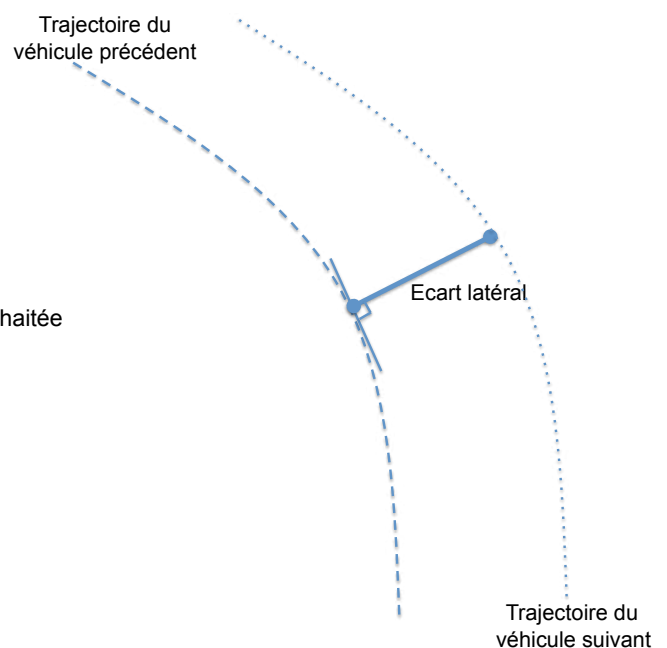


Figure 11.2: Evaluation de l'écart latéral

## 11.2 Evaluation de la distance inter-véhicule

La section suivante présente les résultats de l'évaluation de la distance inter-véhicule dans un train de quatre véhicules en simulation et dans un train de deux véhicules en expé-

rimentation.

Pour mesurer les variations de cette distance, nous faisons évoluer le train de véhicules dans des cas critiques de fonctionnement tels que le démarrage, le freinage d'urgence et le changement rapide de la vitesse du premier véhicule.

### 11.2.1 Simulation : évaluation de la distance inter-véhicule

La figure 11.3 présente les variations de la distance entre les véhicules par rapport à des changements rapides de la vitesse du premier véhicule.

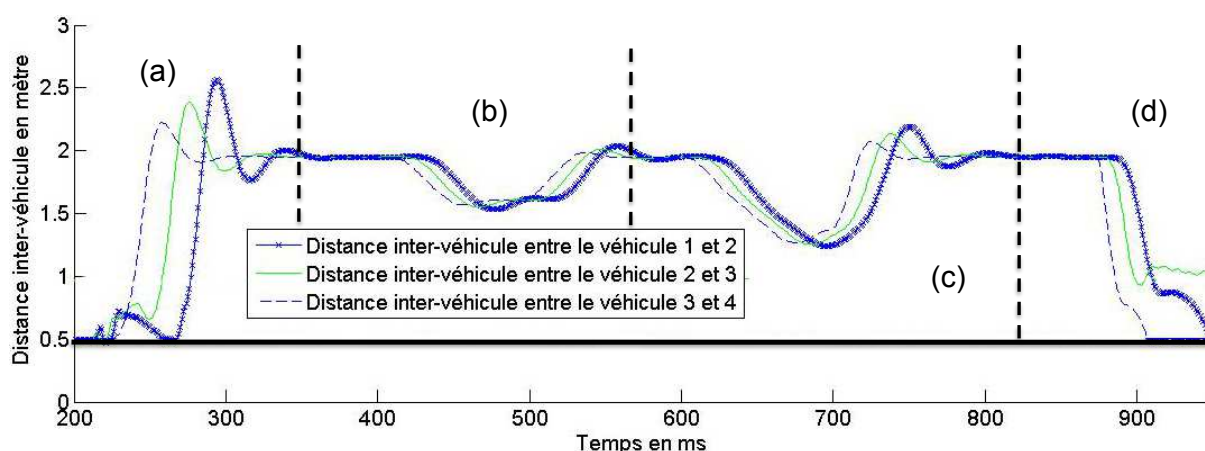


Figure 11.3: Visualisation de la distance inter-véhicule dans un train de 4 véhicules

#### Démarrage du train de véhicules

La figure 11.3 zone (a) présente le cas du démarrage rapide. Le premier véhicule accélère jusqu'à sa vitesse maximum ( $8 \text{ km/h}$ ). Les distances inter-véhicules augmentent et on constate un dépassement de 30 % par rapport à la distance de suivi souhaitée. Ensuite, les distances inter-véhicule se stabilisent à la valeur de la distance de suivi souhaitée. Cette variation est due à la réponse dynamique des véhicules, et en particulier du premier véhicule suiveur.

Pour visualiser l'influence de la vitesse sur les variations de la distance inter-véhicule, nous avons réalisé plusieurs simulations avec des vitesses maximum différentes. Nous appellerons erreur d'écart la différence entre l'écart mesuré et l'écart souhaité. Nous souhaitons évaluer l'erreur maximale et l'erreur moyenne (cf. figure 11.4). La simulation consiste à réaliser un démarrage rapide à la vitesse souhaitée et à visualiser les distances entre les véhicules.

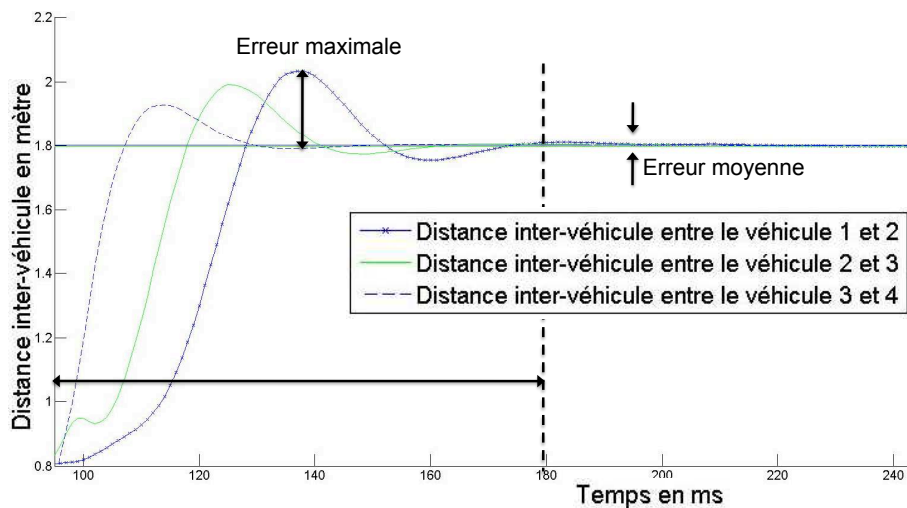


Figure 11.4: Mesure de l'erreur longitudinale

La figure 11.4 présente, par exemple, la variation des distances inter-véhicules entre les véhicules lors du démarrage du train de véhicules, réalisée à la vitesse maximale de 2  $km/h$ .

Vitesse km/h	Erreur moyenne cm	Erreur maximale cm	Unité de temps ms
2	2	20	90
4	8	40	130
8	20	150	250

Figure 11.5: Mesure de l'erreur longitudinale en fonction de la vitesse du véhicule précédent

Le tableau 11.5 présente l'erreur longitudinale que nous obtenons avec un train de quatre véhicules. Dans le cas d'un fonctionnement normal, par exemple, i.e. 8  $km/h$ , nous avons un dépassement de 4 % de la distance de suivi souhaitée pour un temps de 250  $ms$  avant d'obtenir une stabilité de la distance dans le train.

### Changement rapide de la vitesse du premier véhicule

Le second cas de fonctionnement critique est représenté par des variations en “dent de scie” de la vitesse du premier véhicule (cf. figure 11.3). Les zones (b) et (c) représentent deux cas de variation rapide de vitesse. Le premier cas (b) présente une diminution de 30% puis une augmentation jusqu'à la vitesse maximale tandis que le cas (c) présente une diminution de 70%. Dans les deux cas, la distance inter-véhicule diminue puis se stabilise

à la valeur de la distance de suivi souhaitée tout en restant toujours au dessus de la distance de sécurité (cf. tableau 11.6).

### Freinage d'urgence

La figure 11.3 cas (d) présente le cas le plus critique : freinage d'urgence. Le premier véhicule freine avec une décélération maximale. La figure montre que la valeur de la distance inter-véhicule reste toujours au dessus de la valeur de la distance de sécurité (cf. tableau 11.6).

Changement de vitesse en % par rapport à la vitesse de 8 km/h	Variation de la distance inter-véhicule en cm par rapport à la distance de suivi (180 cm)
30 %	40
70 %	60
100 %	110

Figure 11.6: Changement rapide de vitesse du premier véhicule en simulation

### 11.2.2 Expérimentation : évaluation de la distance inter-véhicule

La figure 11.7 montre l'évolution de la distance entre les deux véhicules électriques au cours de l'expérimentation.

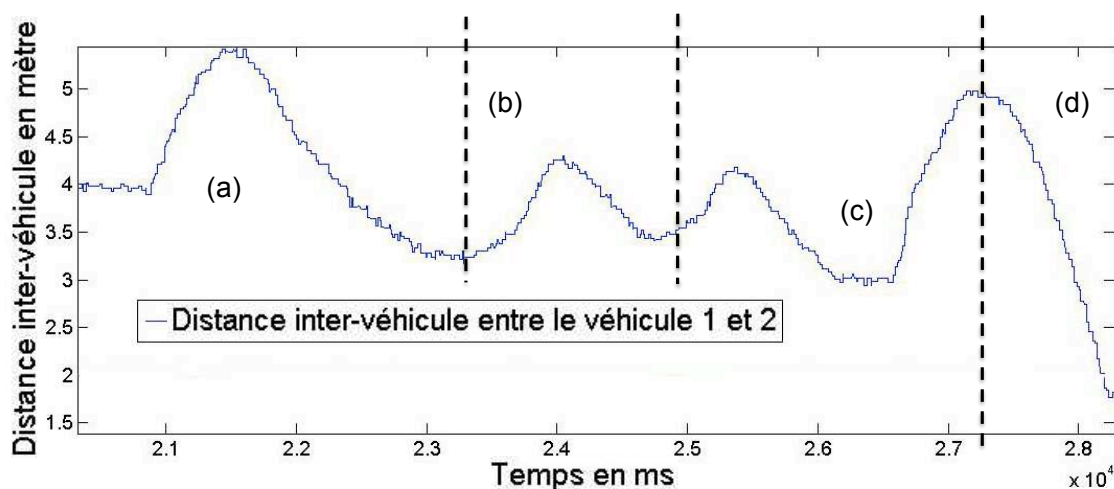


Figure 11.7: Visualisation de la distance inter-véhicule sur un train constitué de 2 véhicules

## Démarrage du train de véhicules

La figure 11.7 illustre de la même façon que la figure 11.3, la variation de la distance inter-véhicule par rapport à des changements rapides de la vitesse. La zone (a) présente un démarrage rapide du train de véhicules. Le premier véhicule accélère jusqu'à sa vitesse maximum ( $8 \text{ km/h}$ ). La valeur de la distance inter-véhicules augmente et on constate un dépassement de 30 % par rapport à la distance de suivi souhaitée, fixé à 4 m pour les expérimentations. Ensuite, la distance entre les véhicules se stabilise à la valeur de la distance de suivi souhaitée. Cette variation est due à la réponse dynamique du véhicule, et en particulier au temps de 500 ms entre l'envoi de la commande de vitesse et l'action du moteur électrique.

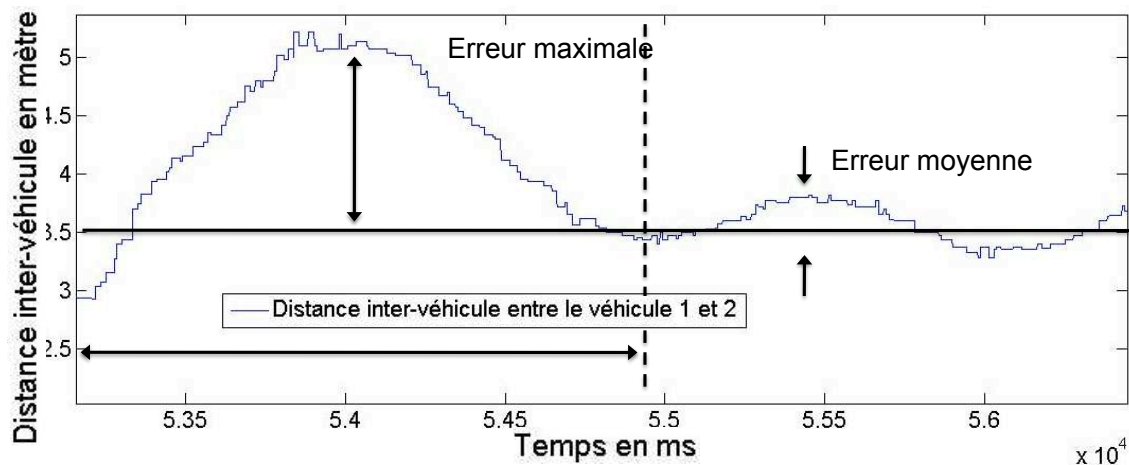


Figure 11.8: Mesure de l'erreur d'écart

La figure 11.8 présente l'erreur d'écart pour un démarrage à la vitesse de  $8 \text{ km/h}$ . On visualise une variation maximale de 150 cm par rapport à la distance de suivi souhaitée.

## Changement rapide de la vitesse du premier véhicule

Les zones (b) et (c) (cf. figure 11.7) représentent deux cas de variation rapide de la vitesse. Le premier cas (b) montre une diminution de 30% puis une augmentation jusqu'à la vitesse maximale, de la vitesse du premier véhicule tandis que le cas (c) présente une diminution de 70%. Dans les deux cas, la distance inter-véhicule diminue et se stabilise ensuite à la distance de suivi souhaitée tout en restant au dessus de la distance de sécurité.

## Décélération

La figure 11.7 cas (d) montre le cas où le véhicule de tête réalise une décélération. Nous ne pouvons pas, pour le moment, réaliser un freinage d'urgence avec une décélération maximale puisque nous avons un temps de latence trop important. On montre ainsi que la distance entre les véhicules reste au-dessus de la distance de sécurité, fixé à 1.5 m.

Changement de vitesse en % par rapport à la vitesse de 8 km/h	Variation de la distance inter-véhicule en cm par rapport à la distance de suivi (400 cm)
30 %	50
70 %	80
100 %	220

Figure 11.9: Changement rapide de la vitesse du premier véhicule en expérimentation

### 11.2.3 Bilan : évaluation de la distance inter-véhicule

Le tableau 11.10 dresse un bilan de la variation de la distance entre les véhicules d'un train. Nous pouvons observer que malgré des variations très brutales de la vitesse du premier véhicule du train, la valeur de la distance entre véhicules reste au-dessus de la valeur de la distance de sécurité.

Cas critique	distance entre véhicule
Simulation	
Démarrage rapide de 0 à la vitesse maximale	Augmentation de 30% de la distance
Variation de la vitesse entre 30 et 70 %	Variation de la distance entre 20 et 50 %
Freinage de la vitesse maximum à 0	Respect de la distance de sécurité
Expérimentation	
Démarrage rapide de 0 à la vitesse maximale	Augmentation de 55% de la distance
Variation de la vitesse entre 30 et 70 %	Variation de la distance entre 30 et 50 %
Freinage de la vitesse maximum à 0	Respect de la distance de sécurité

Figure 11.10: Bilan de l'étude des cas critiques

## 11.3 Evaluation de l'erreur latérale

Les mesures ont été faites conformément à la définition présentée au paragraphe 11.1.

### 11.3.1 Simulation : évaluation de l'erreur latérale

La simulation a été réalisée avec un train de quatre véhicules.



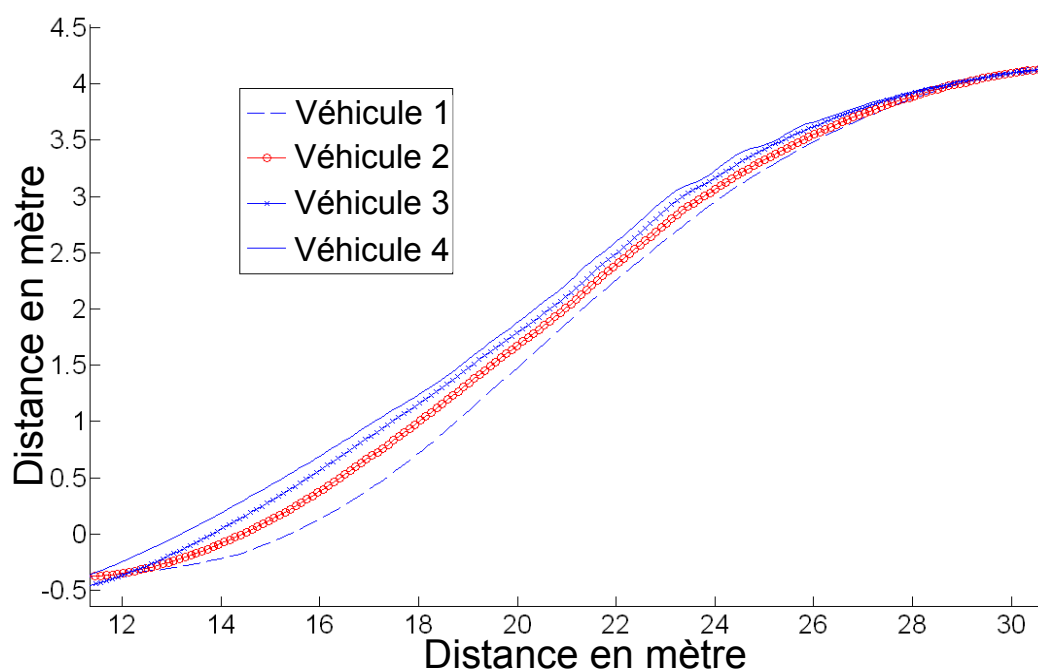


Figure 11.11: Erreur latérale lors de la sortie d'une station

La figure 11.11 présente les traces des véhicules permettant de visualiser l'erreur latérale entre chaque véhicule. La simulation a été réalisée durant la sortie d'une station. Dans ce cas, l'erreur n'est pas supérieure à la largeur d'un pneu (i.e. 20 cm).

Rotation des roues (degré)	Rayon de courbure	erreur moyenne	erreur maximale
5.73	18 m	12 cm	14 cm
11.46	9 m	30 cm	40 cm
17.2	6 m	50 cm	65 cm
22.9	4.5 m	55 cm	67 cm
28.65	3.6 m	67 cm	75 cm

Figure 11.12: Simulation : relation entre la rotation des roues (rayon de courbure) et de l'erreur latérale

Afin de visualiser l'erreur maximale lors de l'évolution du train en courbe, nous avons tracé l'erreur latérale en fonction de la rotation des roues du véhicule suiveur (cf. figure 11.13).

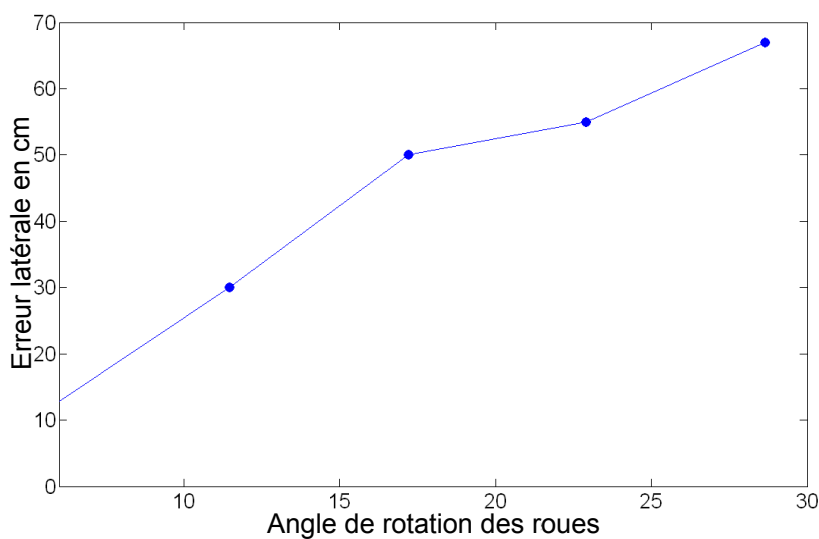


Figure 11.13: Relation entre la rotation de la roue et l'erreur latérale

Le tableau 11.12 montre l'erreur latérale maximale et moyenne en fonction du rayon de courbure du véhicule précédent. On note que, dans le cas d'un fonctionnement normal (rotation des roues < 10 degrés), nous avons une erreur latérale inférieure à 20 cm.

### 11.3.2 Expérimentation : évaluation de l'erreur latérale

L'expérimentation a été effectuée avec les deux véhicules électriques. La mesure de l'écart a été réalisée grâce aux GPS RTK installés sur les véhicules permettant de connaître leurs positions au centimètre près.

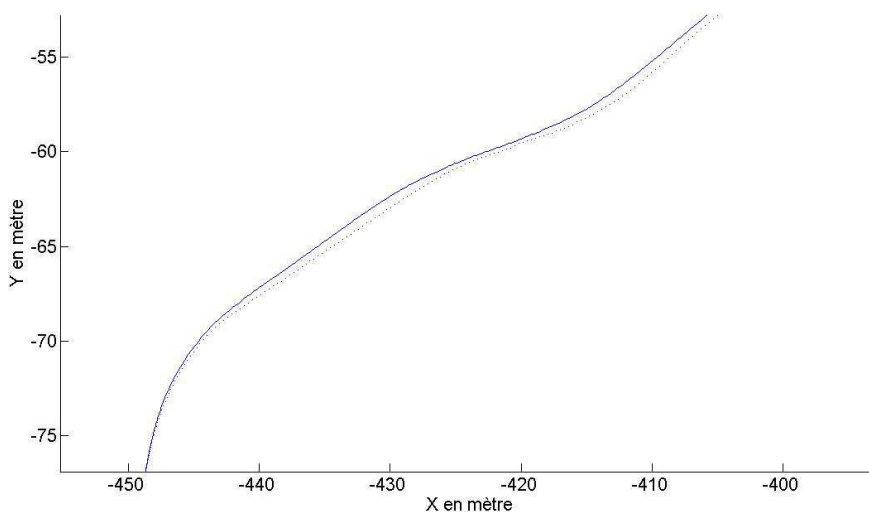


Figure 11.14: Erreur latérale lors de la sortie d'une station

La figure 11.14 présente les trajectoires des véhicules et permet de visualiser l'erreur latérale.

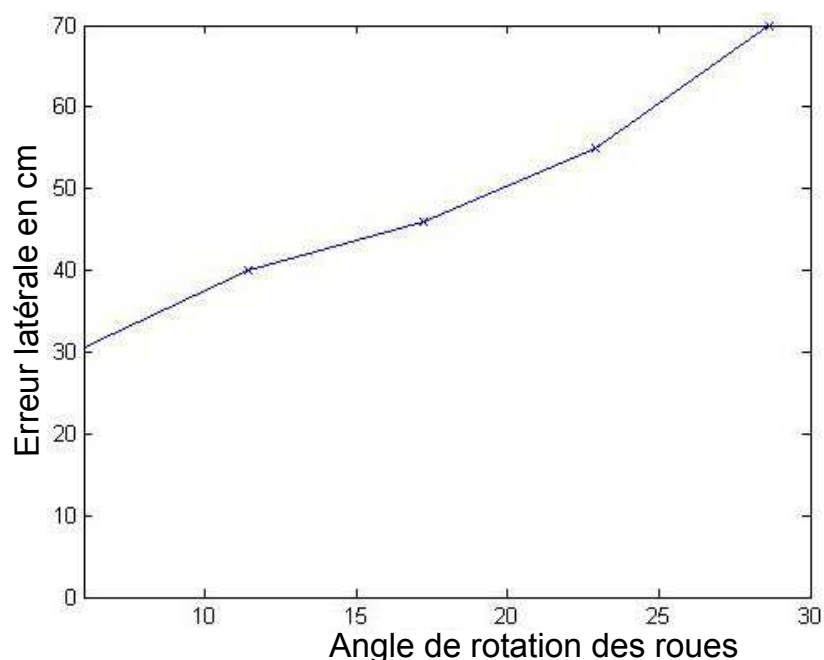


Figure 11.15: Relation entre la rotation de la roue et l'erreur latérale

Afin de visualiser l'erreur maximale lors de l'évolution du train en courbe, nous avons tracé l'erreur latérale en fonction de la rotation des roues du véhicule suiveur à partir de la valeur moyenne de trois parcours identiques (cf. figure 11.15).

Rotation des roues	Rayon de courbure	erreur moyenne	erreur maximale
5.73°	18 m	30 cm	45 cm
11.46°	9 m	40 cm	52 cm
17.2°	6 m	46 cm	56 cm
22.9°	4.5 m	55 cm	70 cm
28.65°	3.6 m	70 cm	90 cm

Figure 11.16: Expérimentation : relation entre la rotation des roues (rayon de courbure) et de l'erreur latérale

### 11.3.3 Bilan : évaluation de l'erreur latérale

Le tableau 11.16 présente l'erreur latérale maximale et moyenne en fonction du rayon de courbure du véhicule précédent. Les résultats exposés dans le tableau 11.16 montrent que l'erreur de suivi du véhicule en simulation est proche de l'expérimentation. En effet, les résultats présentés dans le tableau 11.12 correspondent à l'erreur moyenne entre chaque

véhicule du train, si on compare avec l'expérimentation, on retrouve ainsi les mêmes ordres de valeurs.

Cependant, la définition communément adoptée pour l'erreur latérale est la mesure de l'écart entre les traces de roues du véhicule par rapport à son prédécesseur. Cette mesure ne correspond pas exactement à la mesure de l'écart latéral entre les centres géométriques de véhicules que nous avons effectué, mais elle est difficile à réaliser. Il serait intéressant de trouver une relation entre les deux façons de mesurer l'écart latéral. Cela permettrait d'évaluer la qualité de suivi de trajectoire en appliquant le critère dit du suivi "mono trace". On considère un suivi comme étant "mono trace" lorsque l'écart entre la trace des roues est inférieur à la largeur typique d'un pneumatique (20 cm).

## 11.4 Evitement d'obstacle simple

Nous avons considéré deux cas, d'une part la présence d'un unique obstacle et d'autre part la présence d'un obstacle de chaque côté de la trajectoire. Cette section présente le premier cas, le deuxième étant traité dans la section suivante.

La première expérience consiste à évaluer la trajectoire prise par le véhicule, dans le cas de la présence d'un unique obstacle. La figure 11.17 illustre la configuration adoptée pour l'expérimentation de l'évitement d'obstacle simple, i.e. un unique objet sur la chaussée.



Figure 11.17: Configuration d'obstacle simple

### 11.4.1 Simulation : évitement obstacle simple

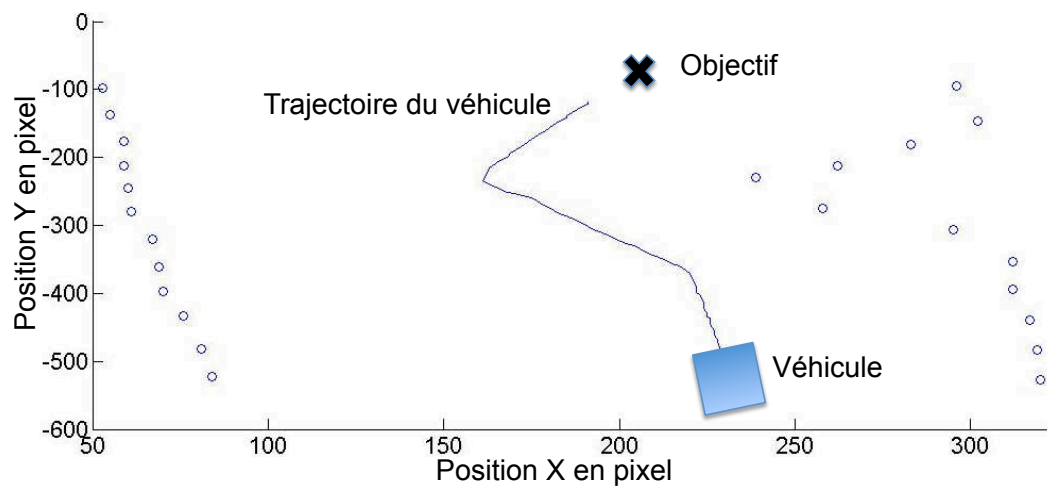


Figure 11.18: Evitement d'obstacle simple

La figure 11.18 présente la trajectoire prise par le véhicule dans le cas où un obstacle se situe sur un côté de la chaussée. L'objectif à atteindre est placé au dessus tandis que la position initiale du véhicule est fixée en bas.

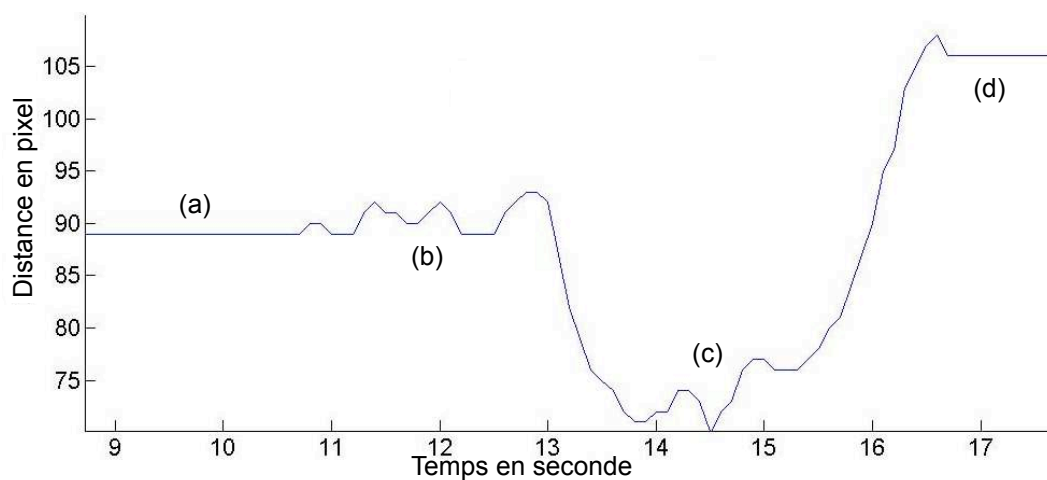


Figure 11.19: Distance de l'obstacle le plus proche avec une configuration d'obstacle simple

La figure 11.19 montre la distance minimale entre l'obstacle et le véhicule au cours du déplacement. La première valeur (notée (b)) correspond à la première apparition de l'obstacle dans l'environnement de perception des agents décisionnels. Puis, la distance

véhicule-obstacle tend vers un minimum (c). Si aucun nouvel obstacle n'est détecté, la distance augmente (d).

### 11.4.2 Expérimentation : évitement obstacle simple

La figure 11.20 illustre l'expérimentation du même cas d'évitement d'obstacle traité précédemment en simulation.

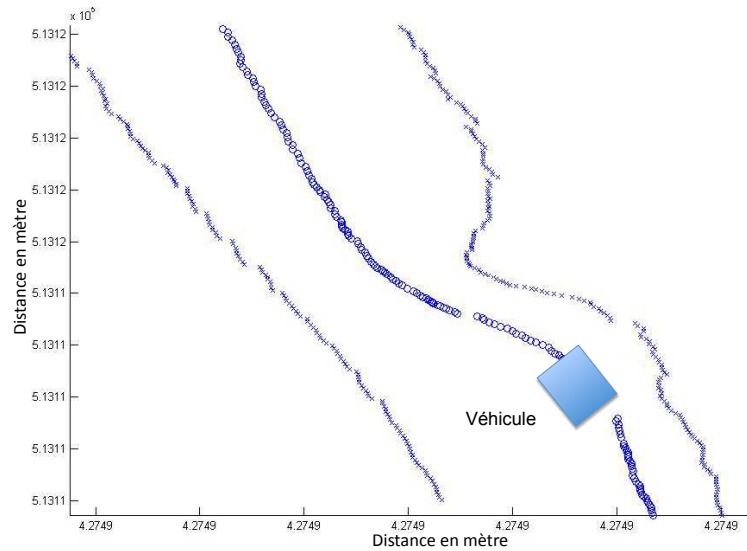


Figure 11.20: Trajectoire du véhicule avec une configuration d'obstacle simple

La figure 11.21 permet de constater la similitude entre les résultats obtenus par simulation et expérimentation.

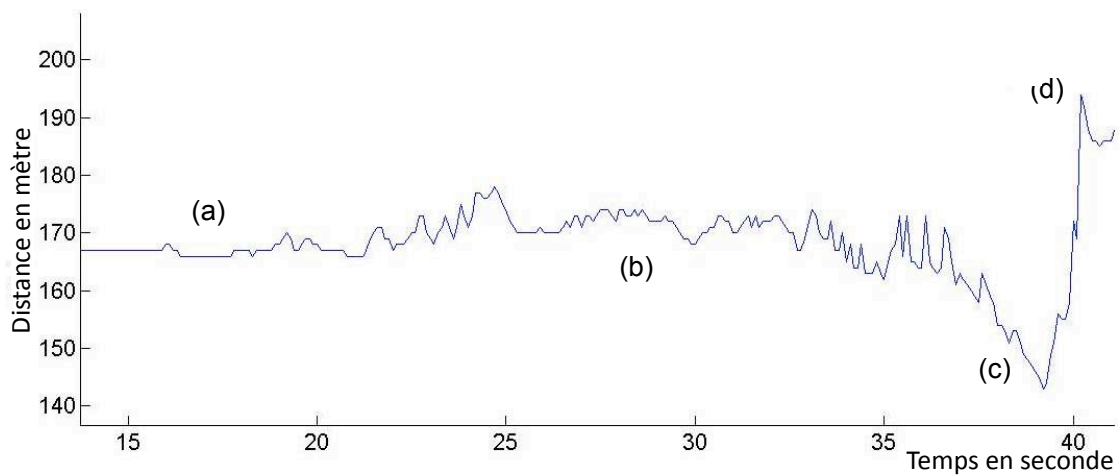


Figure 11.21: Distance de l'obstacle le plus proche avec une configuration d'obstacle simple

## 11.5 Evitement d'obstacle multiple

L'objectif de cette expérience est d'observer l'évolution de la vitesse du véhicule, en fonction des contraintes générées par la présence d'un obstacle de chaque côté de la trajectoire. La figure (cf. figure 11.22) illustre la configuration adoptée où l'obstacle à droite est construit par des boîtes en carton et l'obstacle à gauche est la haie.



Figure 11.22: Evitement d'obstacles multiples

### 11.5.1 Simulation : obstacles multiples

La figure 11.23 présente la trajectoire prise par le véhicule lors de la simulation.

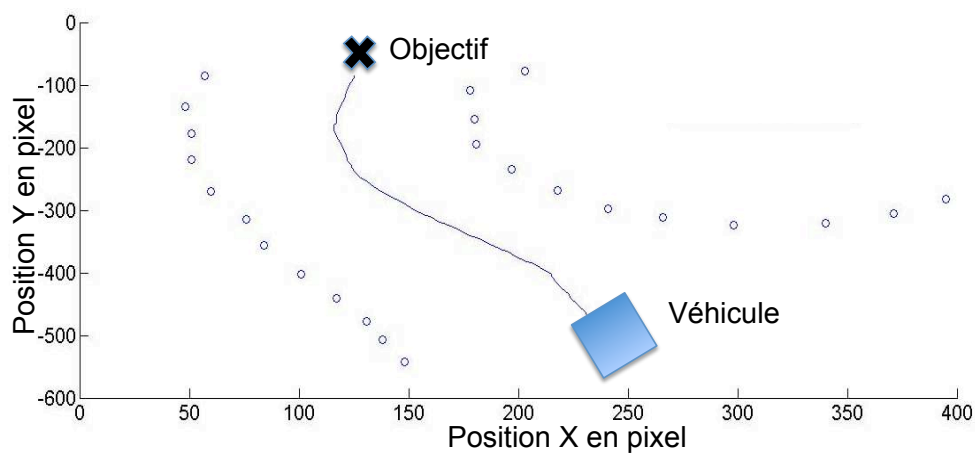


Figure 11.23: Trajectoire du véhicule lors de la simulation

Nous nous apercevons que le véhicule suit une trajectoire se situant à égale distance entre les deux obstacles.

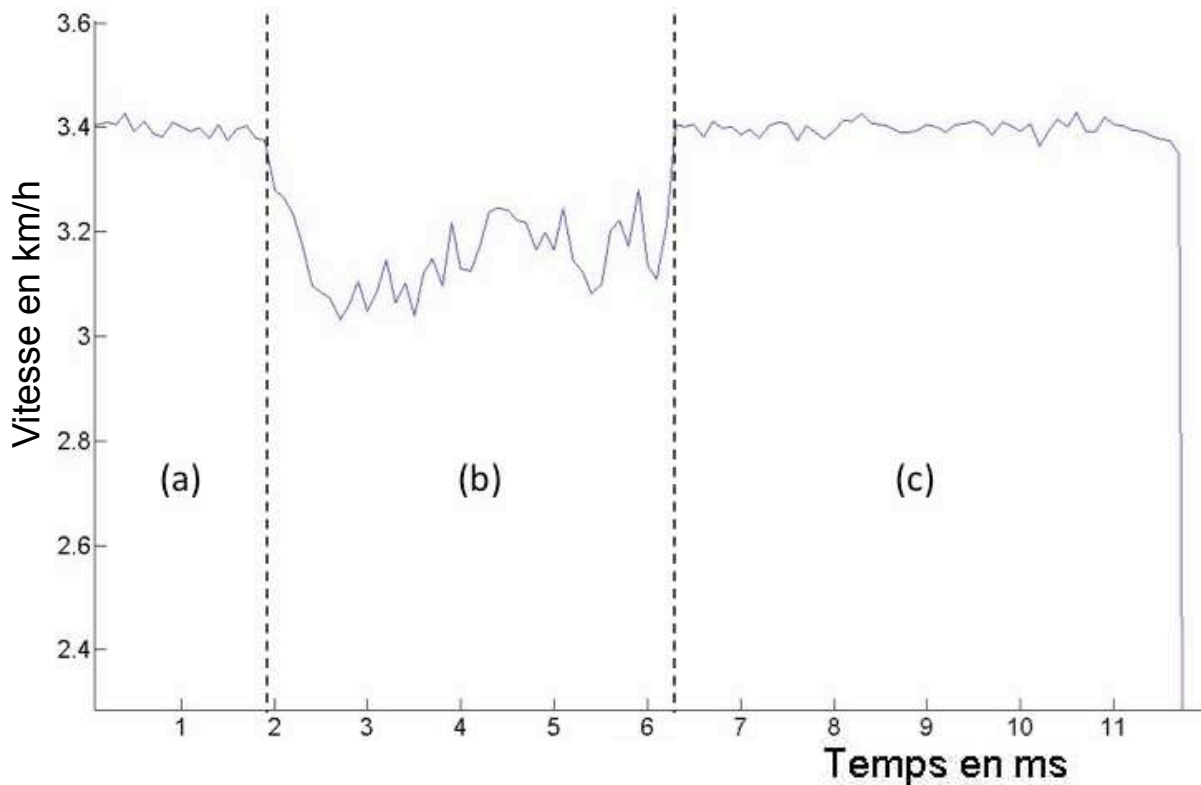


Figure 11.24: Vitesse du véhicule

La figure 11.24 présente l'évolution de la vitesse du véhicule durant la simulation. La première partie (notée (a)) de cette courbe correspond à la trajectoire avant la détection des obstacles. Dans cette partie, le véhicule atteint sa vitesse maximale. Dès que les obstacles sont détectés dans l'environnement des agents décisionnels (partie notée (b)), la vitesse du véhicule commence à diminuer en raison du facteur de correction statistique (cf. chapitre 6). Lorsque le véhicule franchit la ligne d'obstacle (partie notée (c)), la vitesse augmente jusqu'à sa valeur maximale.

### 11.5.2 Expérimentation : obstacles multiples

Nous avons réalisé le même cas de figure que pour la simulation précédente. La figure 11.25 présente la trajectoire du véhicule au cours de l'expérimentation.



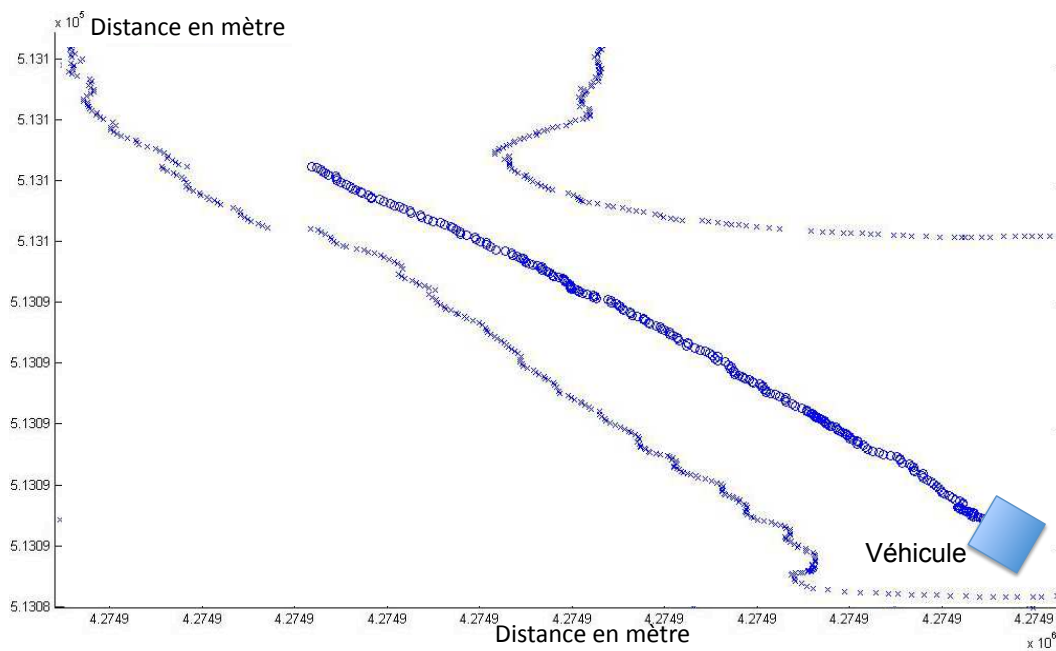


Figure 11.25: Trajectoire du véhicule lors l'expérimentation

L'évolution de la vitesse est décrite par la figure 11.26. Nous constatons le même type d'évolution que la simulation. Le véhicule diminue sa vitesse en réponse aux nombreuses contraintes, puis accélère une fois l'obstacle franchi.

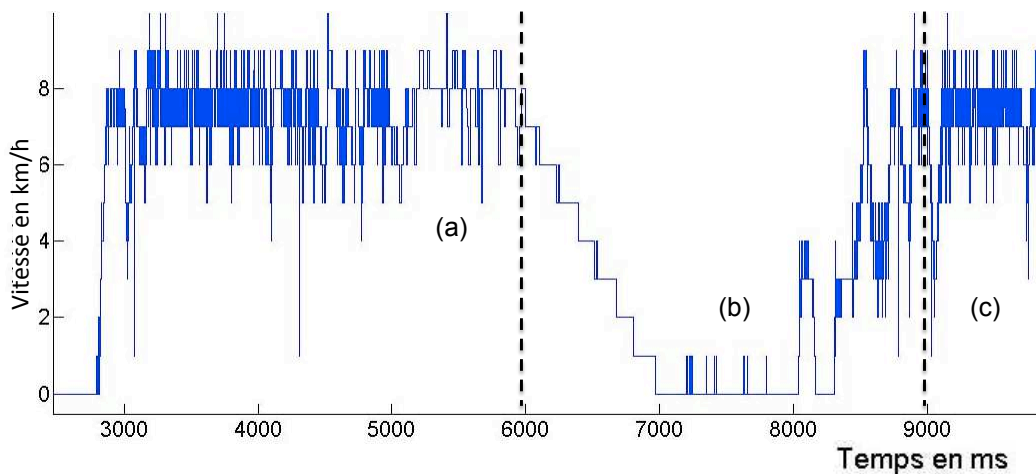


Figure 11.26: Vitesse du véhicule lors de l'expérimentation

## 11.6 Conclusion

Dans le but de tester les algorithmes développés pour la navigation multi-véhicules, nous avons réalisé des simulations et expérimentations sur la base de scénarii de fonction-

nement normaux et critiques. Pour cela, nous avons utilisé deux plate-formes, un simulateur en 3D et des véhicules électriques.

Nous avons ainsi montré que le suivi de véhicule dans les cas traités, aussi bien en simulation qu'en expérimentation, est conforme à la propriété de sûreté : la distance entre véhicule est toujours supérieure à la distance de sécurité. De plus, nous avons testé le système d'évitement d'obstacles. Ces expériences ont permis de constater une auto-adaptation des trajectoires ainsi qu'une auto-adaptation de la vitesse du véhicule.



---

QUATRIÈME PARTIE

---

## **Conclusion et Perspectives**

---



# CONCLUSION

---

Tout au long de cette thèse, nous avons abordé le problème de la navigation multi-véhicules et proposé des réponses à certaines questions. Cette contribution peut se décomposer en trois sous-parties :

- **Elaboration de modèles adaptés à la navigation multi-véhicules** : on a identifié deux principales tâches : la conduite en convoi et la navigation autonome. On propose, dans cette thèse, d'utiliser le paradigme des systèmes multi-agents pour développer ces deux comportements nécessaires à la navigation multi-véhicules.
- **Approche formelle : spécification et vérification des SMA réactifs** : dans le cas d'applications soumises à des contraintes sévères de sûreté de fonctionnement comme la navigation multi-véhicules. Il est indispensable d'élaborer des méthodes permettant de définir des modèles multi-agents réactifs, principalement de façon locale, tout en assurant l'émergence d'états globaux caractérisés par de bonnes propriétés.
- **Simulation et expérimentation** : compte tenu du domaine d'application cible de ce travail, nous avons implanté nos modèles dans un simulateur développé au sein de notre laboratoire, prenant en compte les caractéristiques physiques des véhicules. Le simulateur permet ainsi d'évaluer différents scénarios critiques avant de passer à une expérimentation sur des véhicules réels. Enfin, pour compléter notre étude, nous avons procédé à une étape d'expérimentation de nos modèles sur des véhicules de type CYCAB et GEM.

## 12.1 Bilan et apports

Nous avons construit, dans un premier temps, des modèles réactifs d'inspiration physiques pour la navigation multi-véhicules. Ensuite, nous avons appliqué une approche de spécification et de vérification adaptée aux systèmes multi-agents réactifs. Enfin, nous avons simulé et expérimenté ces modèles.

### 12.1.1 Proposition d'un modèle multi-agents réactifs pour la conduite en convoi

Nous avons, dans cette thèse, défini un modèle multi-agents appliqué à la conduite en convoi en utilisant des agents réactifs interagissant entre eux et avec leur environnement en fonction de lois physiques. L'accroche immatérielle a été réalisé avec un lien mécanique virtuel de type "masse-ressort" entre les véhicules. Le lien "masse-ressort" a été complété par un moment de torsion.

En fonction de la distance et de l'angle de courbure entre deux véhicules, une force  $\vec{F}$  résultante de la somme de la force du ressort, de l'amortisseur et de la force de torsion est appliquée au véhicule suiveur. L'agent véhicule élabore alors sa décision (accélération, direction) en appliquant le principe fondamental de la dynamique.

Chaque véhicule agent du train dispose ainsi d'un modèle d'interaction lui permettant de suivre le véhicule précédent. Par rapport aux travaux existants sur la conduite en convoi, notre approche peut être assimilée aux méthodes dites de "proche en proche" tout en résolvant ces principaux défauts. Le problème d'oscillation a été résolu à partir d'un mécanisme permettant une adaptation dynamique des paramètres du modèle (raideur du ressort, coefficient d'amortissement, ...) en fonction de la dynamique du véhicule. De plus, le moment de torsion qui complète le modèle "masse ressort", permet de remédier à l'anticipation en courbe.

### 12.1.2 Proposition d'un modèle multi-agents réactifs pour la navigation autonome

Contrairement à la conduite en convoi, nous nous intéressons ici au comportement d'un seul véhicule se déplaçant le long d'une trajectoire tout en évitant les obstacles. Ce comportement permet au véhicule de naviguer de manière complètement autonome le long d'un itinéraire prédéfini. Le véhicule doit être capable d'éviter des obstacles pouvant apparaître sur cette itinéraire et, éventuellement, adapter sa vitesse en fonction des contraintes générées par les obstacles.

Dans le modèle que nous proposons, le véhicule est considéré comme un système multi-agents. Les agents prennent collectivement la meilleure décision compte tenu de la perception des contraintes. L'environnement des agents est construit à partir des informations fournies par les capteurs du véhicule. L'organisation émergent de ce comportement est interprétée et transformée en une décision globale en utilisant des indicateurs globaux basés sur une analyse géométrique (position moyenne, vitesse moyenne des agents, ...) et sur la physique statistique.

### 12.1.3 Spécification et vérification de la conduite en convoi

De nombreuses applications de systèmes multi-agents réactifs doivent satisfaire des propriétés de sûreté. C'est le cas des convois multi-véhicules, qui constituent la principale problématique de cette thèse.

Pour répondre à cette exigence, nous avons utilisé une approche de vérification adaptée aux propriétés de sûreté, basée sur la spécification formelle et l'utilisation du model-checking. Nous avons suivi cette approche pour vérifier une propriété de sûreté relative aux convois multi-véhicules : l'absence de collision.

L'une des caractéristiques de cette approche est son caractère compositionnel qui nous a permis d'éviter l'explosion combinatoire lors de la vérification.

Ces expériences montrent aussi la pertinence de l'approche de vérification en tant qu'outil de dimensionnement : des scénarii de fonctionnement peuvent être définis en donnant des valeurs aux multiples paramètres du modèle pour déterminer les meilleures configurations qui satisfont la propriété de sûreté.

Si bien, nous n'avons pas accordé une attention considérable à l'aspect méthodologique, nous croyons que la démarche générale adoptée au cours de nos travaux peut donner lieu à quelques idées générales en la matière. En effet, nous avons identifié un certain nombre d'activités de modélisation, de spécification, d'analyse, ... ainsi que des relations de dépendance celles-ci. L'annexe E présente, de façon abstraite, une généralisation de la méthode suivie au cours de ces travaux.

### 12.1.4 Simulation et expérimentation

Les différents tests effectués, tant en simulation qu'avec des véhicules électriques du laboratoire SeT, nous ont permis de mettre en évidence que l'utilisation de SMAR pour la navigation multi-véhicules possède un certain nombre de propriétés intéressantes :

- **Adaptation aux véhicules** : le comportement d'agent a été mis en place sur différents véhicules : le *Robucab* et les *SetCar*. Ces deux implémentations illustrent l'adaptabilité du modèle proposé dans cette thèse.
- **Paramètres variables** : le modèle d'interaction est basé sur plusieurs paramètres. Le paramétrage variable, en fonction des caractéristiques dynamiques du système, permet de supprimer les oscillations de la distance inter-véhicule le long du train.
- **Erreur latérale et longitudinale** : les simulations et expérimentations ont permis de montrer que l'erreur latérale et longitudinale induite par la conduite en convoi est faible comparé aux autres approches dite de "proche en proche".
- **Faible coût** : le comportement intelligent du véhicule étant basé sur une perception locale et sans communication. Chaque véhicule est équipé d'un unique capteur pou-



vant être de type télémétrique.

## 12.2 Problèmes non résolus

Les méthodes proposées, dans cette thèse, pour la navigation multi-véhicules ont été simulé et expérimenté. Durant ces tests, des problèmes se sont manifestés :

- **Accroche immatérielle à haute vitesse** : la conduite en convoi n'a pas été testé à une vitesse supérieure à  $8 \text{ km/h}$  ( $2 \text{ m/s}$ ). Nous avons fait le choix de ne pas essayer avec une vitesse supérieure puisque, sur le véhicule électrique, nous avons un temps de latence de  $500 \text{ ms}$  entre l'envoi et l'application d'une consigne. La distance inter-véhicule souhaitée étant de  $180 \text{ cm}$ , nous ne pouvons assurer une réponse sûre avec une latence aussi importante.
- **Spécification et vérification de la conduite en convoi** : la vérification a été faite par rapport à un cas particulier de l'évolution : train de véhicules se déplaçant en ligne droite.
- **Spécification et vérification du modèle multi-agents réactifs pour la navigation autonome** : un modèle pour la navigation autonome a été défini dans cette thèse. Celui-ci a été simulé, expérimenté et présente des résultats encourageants, permettant à un véhicule d'éviter des obstacles pouvant se trouver sur sa trajectoire. Cependant, nous n'avons pas construit le modèle formel permettant d'effectuer des vérifications.
- **Navigation autonome et irrégularité du sol** : la perception des obstacles pour l'évitement est réalisée seulement sur un plan horizontal, ne permettant pas au véhicule de percevoir des trous, ....

## 12.3 Perspectives

Le travail effectué permet de soulever quelques questions qui pourraient faire l'objet de nouvelles recherches. Il est possible de décliner ces questions à partir de différents aspects abordés par nos travaux.

### 12.3.1 Du point de vue du modèle multi-agents réactifs pour la conduite en convoi

En formulant le modèle d'interaction, nous avons négligé l'effet de certains phénomènes (temps de réponse de l'asservissement, frottement, ...). Lors des expérimentations, il s'est avéré que, dans certains cas, ces effets provoquaient des erreurs non négligeables. Ceci nous conduit à affiner le modèle existant, pour prendre en compte ces phénomènes.

Il a y un aspect de la conduite en convoi que nous n'avons pas présenté : l'insertion et l'éjection de véhicule. Lors de l'insertion, un véhicule s'incorpore au convoi en mouvement.

Lors de l'éjection, un véhicule qui faisait parti du convoi en mouvement s'en extrait. Ces phases correspondent à des modes de fonctionnement autonome du véhicule. Nous pensons que l'insertion et l'éjection peuvent être abordé comme des cas particulier de la navigation autonome. Par conséquent nous comptons travailler à cette question en appliquant le modèle pour l'évitement d'obstacle développé au cours de cette thèse.

### **12.3.2 Du point de vue du modèle de SMAR proposé pour la navigation autonome**

Le modèle proposé pour l'évitement d'obstacle souffre actuellement du manque de la perception des irrégularités du sol. Pour remédier à cette problématique, on peut améliorer la perception de l'environnement en ajoutant un deuxième capteur télémétrique permettant la perception sur un plan vertical. A partir de ces informations, nous pouvons dupliquer le SMAR proposé et prendre une décision entre le SMAR percevant le sol et les obstacles sur le plan horizontal.

### **12.3.3 Du point de vue de la spécification et la vérification**

Le modèle formel de spécification utilisé pour la vérification inclut un composant représentant les caractéristiques physique du véhicule. Ce composant permet de prendre en compte de façon plus ou moins détaillé la réalité matérielle et par ce biais, d'augmenter notre confiance envers les résultats de la vérification.

Le modèle que nous avons utilisé dans le cadre de ces travaux pourrait être enrichi pour prendre en compte un certains nombres de phénomènes (temps de latence, adhérence, ...).

### **12.3.4 Du point de vue applicatif**

Les simulations et expérimentations ont permis de valider les modèles proposés pour la navigation multi-véhicules définis dans cette thèse. Des aspects sont encore à améliorer en vue d'obtenir un comportement plus performant.

L'accroche immatérielle se base sur la perception de la distance inter-véhicules. Cette perception est réalisée par des capteurs télémétriques et des balises réfléchissantes. Elle peut être amélioré par une redondance obtenue grâce à l'ajout de capteurs vidéo, renforçant ainsi la sûreté en cas de défaillance des capteurs télémétriques.



---

CINQUIÈME PARTIE

---

**Annexes**

---



# INTERPRÉTATION DE L'ORGANISATION ÉMERGENTE D'UN SMAR BASÉE SUR LA PHYSIQUE STATISTIQUE

## A.1 Potentiels thermodynamique

L'évaluation globale d'un système par rapport à ses phénomènes locaux est une étude classique en physique statistique [McNaught and Wilkinson, 1997]. Cette évaluation est réalisée à l'aide de fonctions caractérisant l'état d'équilibre d'un système : les potentiels thermodynamiques [Balian, 2007]. Le minimum ou le maximum du potentiel suivant le type utilisé (Energie interne, Energie libre, ...) représente l'équilibre du système étudié. Ces potentiels sont utilisés [Reiss, 1997] pour le calcul de l'équilibre résultant d'une réaction chimique ou pour mesurer les propriétés de matériaux dans une réaction chimique. Ils sont appelés "potentiels" parce que dans un sens, ils décrivent la quantité d'énergie potentielle d'un système thermodynamique soumis à certaines conditions. Les potentiels thermodynamiques peuvent également être utilisés pour mesurer l'état global du système grâce à l'énergie locale mesurée sous certaines conditions de température, pression, .... comme le montre le tableau des potentiels les plus communs (cf. tableau A.1).

Potentiels thermodynamiques	Condition d'utilisation
Energie interne $U$	$S, V, N_i$ constants
Energie libre de Helmholtz $A$	$T, V, N_i$ constants
Enthalpie $H$	$S, P, N_i$ constants
Enthalpie libre de Gibbs $G$	$T, P, N_i$ constants

Figure A.1: Potentiels thermodynamiques les plus couramment utilisés

Avec  $T$  la température,  $S$  l'entropie,  $P$  la pression,  $V$  le volume et  $N_i$  le nombre d'éléments.

Le potentiel que nous avons considéré comme le plus intéressant pour l'étude des SMAR est l'*Energie libre de Helmholtz*  $A$  (cf. equation A.1) qui permet de mesurer le travail

fourni à partir d'une transformation réversible <sup>1</sup> dans un système fermé à une température constante :

$$A(T, V, N_i) = - \frac{\ln Z}{\beta}. \quad (\text{A.1})$$

Avec  $Z$  la fonction de partition et  $\beta$  la constante de la fonction de partition. Les conditions d'application de ce potentiel (cf. figure A.1), semble adaptées à l'étude des systèmes multi-agents réactifs. En effet, le volume du système peut est assimilé au volume de l'environnement du SMAR, le  $N_i$  aux nombres d'agents du système (cf. A.3). Quant à la température, nous considérons qu'il est possible d'en faire abstraction en première approximation en se limitant à dire que ça valeur est constante. Pour certains type de SMAR, il se pourrait qu'on trouve une correspondance entre des variables du système et une température.

L'étude de l'énergie libre de Helmholtz  $A$  permet de calculer des propriétés sur l'évolution du système. Par exemple, étant donné un système dans un état de déséquilibre initial, s'il évolue vers un état d'équilibre, la variation de l'énergie libre  $A$  est égale à la quantité maximale de travail. L'état d'équilibre se traduit ainsi par une variation nulle de l'énergie libre  $A$  (cf. figure A.2).

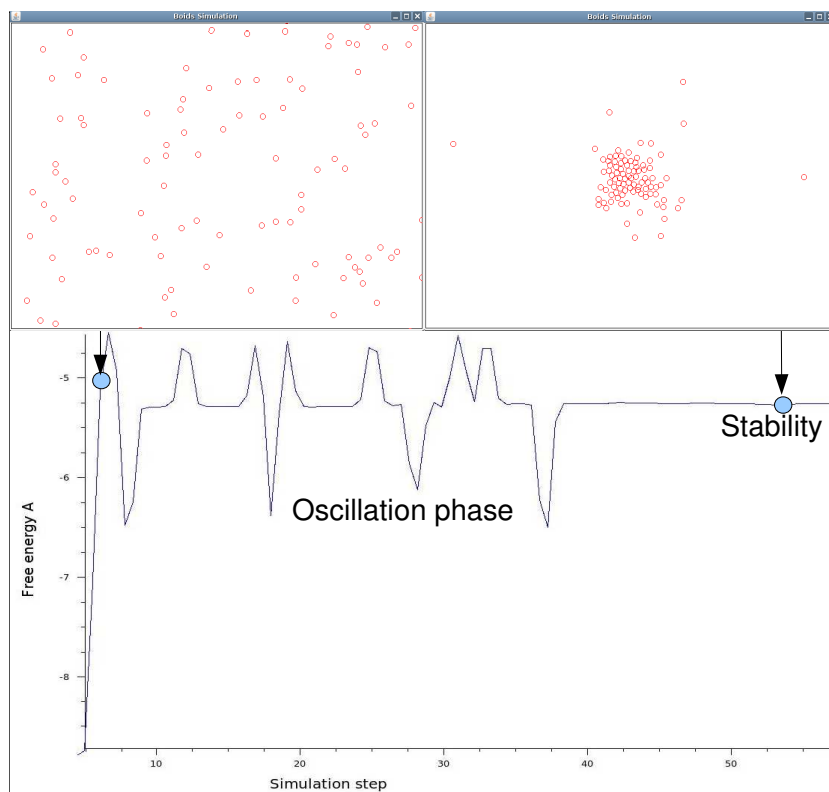


Figure A.2: Exemple de mesure de stabilité d'un système de boids [Contet et al., 2008]

<sup>1</sup>Une transformation réversible est un processus qui peut être "inversé" par le biais de changements infimes sans perte ou dissipation d'énergie dans le système

La figure A.2 présente la variation de la mesure de l'énergie libre  $A$  d'un système multi-agents réactifs simulant un vol d'oiseaux [Reynolds, 1987]. La stabilité de ce système est obtenue lorsque le groupe d'oiseaux est formé. Comme le montre la figure A.2 après une période d'oscillation lors de la construction du groupe d'oiseau, le système arrive à une stabilité.

Le calcul de l'énergie libre  $A$  est dépend du calcul de la fonction de partition  $Z$  et de son paramètre  $\beta$ . La fonction de partition mesure le degré de désordre d'un système. Cette fonction calcule la somme des micro-états possibles d'un élément en fonction de la valeur de leur énergie. Considérons un système qui possède un grand nombre d'états possibles (cf. [Balian, 2007]), dont la température  $T$ , le volume  $V$  et  $N_i$  le nombre d'élément sont constants. Indexons les états d'énergie (les micro-états) que le système peut occuper par  $j = 1, 2, 3, \dots$  et notons  $E_j$  l'énergie correspondant à chaque micro-état  $j$ . La fonction de partition s'écrit :

$$Z = \sum_j e^{-\beta E_j} \quad (\text{A.2})$$

où  $\beta$  est défini par  $\beta \equiv \frac{1}{k_B T}$  dans le cas d'une distribution de type Boltzmann-Gibbs. Cette constante normalise la fonction de partition en tenant compte des grandeurs que l'on souhaite mesurer.

La fonction de partition possède des propriétés intéressantes pour simplifier sa mesure. Supposons que nous avons  $N$  sous systèmes dans le système avec des énergies d'interaction négligeables. La fonction de partition du système est égale aux produits des fonctions de partition des sous système  $f_i$ .

$$Z = f_1 * f_2 * f_3 * \dots * f_N \quad (\text{A.3})$$

La figure A.3 montre l'utilisation de cette propriété pour calculer la fonction de partition  $Z$  en fonction de partition  $Z_1$  et  $Z_2$ .



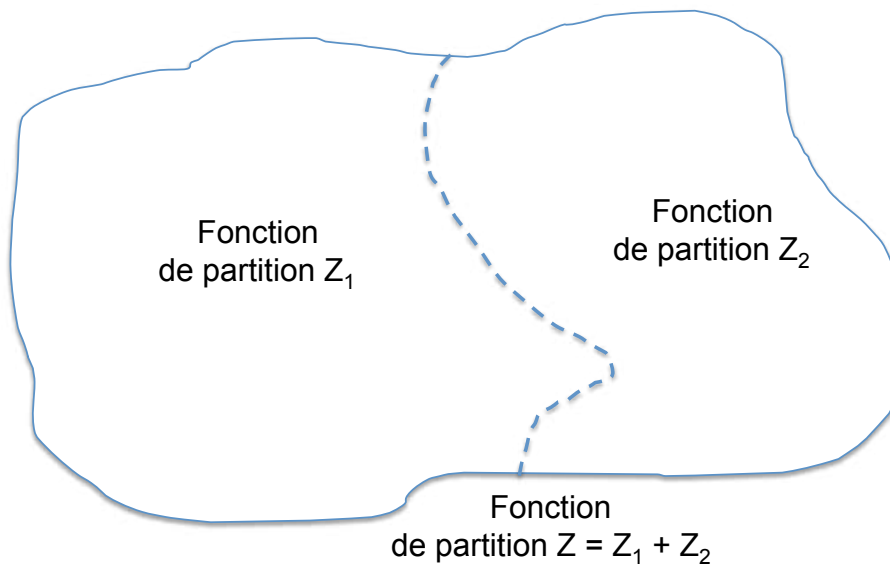


Figure A.3: Propriété intéressante de la fonction de partition

Le paragraphe suivant montre par le biais d'un exemple les différents calculs pour mesurer l'énergie libre  $A$  d'un système.

## A.2 Exemple : oscillateur harmonique

L'exemple présenté dans cette section (cf. [Balian, 2007], page 177) décrit l'évaluation de l'équilibre globale, à partir de l'énergie locale, d'un système mécanique composé d'une seule entité : l'oscillateur harmonique.

Un oscillateur harmonique (cf. figure A.4) est un corps relié à une base et en contact constant avec une source thermique de température  $T$ . La constante de raideur  $k$  intervient dans le calcul d'une force d'élasticité tandis que l'inertie dépend de la masse  $m$ .

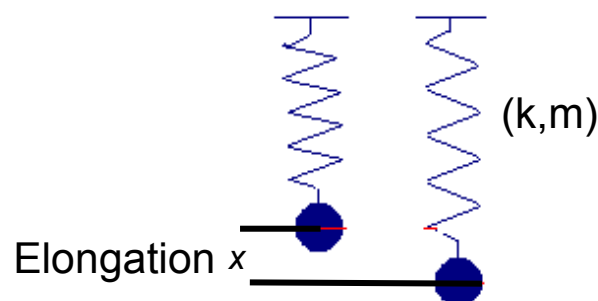


Figure A.4: Oscillateur harmonique

On se propose ici de calculer le potentiel thermodynamique  $A$ , qui décrit l'état d'équi-

libre du système. Seules les grandes étapes du calcul sont présentées pour illustrer l'utilisation de la physique statistique et de la thermodynamique. La première étape consiste à calculer l'énergie du système :

$$E_{\text{systeme}} = E_{\text{cinetique}} + E_{\text{potentiel}} = \frac{1}{2}mv^2 + \frac{1}{2}kx^2 \quad (\text{A.4})$$

avec  $v$  la vitesse du corps et  $x$  son déplacement. A partir de cette énergie, on exprime la fonction de partition  $Z$  :

$$Z = e^{-\beta E_{\text{systeme}}} \quad (\text{A.5})$$

avec  $\beta = \frac{1}{k_B T}$ ,  $k_B$  la constant de Boltzmann-Gibbs.

A partir de ceci, l'évolution conduisant à l'équilibre du système peut être étudié [Balian, 2007].

Toutefois, l'application de la physique statistique et de la thermodynamique aux multi-agents doit respecter certaines conditions qui sont présentées dans le paragraphe suivant.

### A.3 Utilisation de la physique statistique dans le cadre des systèmes multi-agents

Peut-on appliquer la physique statistique pour mesurer des propriétés globales d'un système multi-agents réactifs ? Nous allons énumérer les limites et les conditions d'une telle application :

– **Conditions d'utilisation des potentiels thermodynamiques :**

Les potentiels thermodynamiques sont des fonctions caractérisant l'état du système sous certaines conditions. Celles imposées par l'énergie libre de Helmholtz  $A$  sont un volume  $V$ , un nombre d'élément  $N_i$  et une température  $T$  constants. Les conditions volume et nombre d'éléments constant sont réalistes pour un SMAR. En effet, le volume du système est assimilé au volume de l'environnement où évoluent les agents lequel est naturellement limité. De la même manière, le nombre d'éléments est assimilé au nombre d'agents du système, pouvant être considéré constant pour un grand nombre d'applications. La dernière condition d'utilisation concerne la température  $T$  du système. Cette hypothèse peut être supposé vraie puisque les SMAR ne sont pas influencés par la température du système. Pour conforter notre supposition, nous pouvons remarquer que cette supposition a déjà été faite dans *The Mathematical Theory of Communication* [Shannon, 1949] pour justifier l'utilisation de l'entropie statistique [Balian, 2007].

– **Constante  $\beta$  de la fonction de partition :**

Ce paramètre a été introduit pour normaliser la fonction de partition par rapport à sa distribution. Pour la distribution de Boltzmann-Gibbs, on définit  $\beta = \frac{1}{k_B T}$  avec  $k_B$  une constante de proportionnalité permettant d'égaliser la valeur de l'entropie de la mécanique statistique à l'entropie classique de *Clausius*<sup>2</sup>. Le choix de la constante  $\beta$  dépend du système d'unités dans lequel l'énergie est mesurée. Par exemple, on suppose  $\beta = 1$  en théorie de l'information [Shannon, 1949] ou de la communication (cf. [Balian, 2007] page 103). Nous avons adopté le même choix, comme pour un système d'information ou de la communication la constante  $\beta$  égale à 1.

– **Nombre d'éléments :**

Il est généralement supposé que la physique statistique requiert un grand nombre d'éléments (corps, molécules, ...) comme condition d'application. L'étude statistique est un outil mathématique qui permet de prévoir l'évolution du système, à partir d'informations acquises antérieurement. Pour utiliser cet outil, nous nous basons sur le grand nombre d'états possibles du système macroscopique. Ce grand nombre d'états possibles peut par exemple correspondre, dans le cas d'un SMAR situé, aux nombreuses positions possibles d'un agent dans l'environnement. On peut ainsi appliquer la physique statistique à un système composé d'un seul et unique agent (cf. [Balian, 2007], page 177) ou d'un nombre arbitraire d'agents.

---

<sup>2</sup>énoncé de la deuxième loi de la thermodynamique (1850)

## EXPÉRIMENTATION DES CYBERCARS

---

Le déplacement automatique a fait l'objet de plusieurs expérimentations depuis 1990.

### B.1 Aéroport de Schiphol 1997

C'est à la fin 1997 que l'aéroport de Schiphol à Amsterdam a mis en service le "Parking Hopper", les premiers véhicules routiers automatisés pour le transport de personnes. Quatre navettes fonctionnent ainsi 24 heures sur 24 pour aller de sa voiture au terminal de bus. Ces véhicules (les ParkShuttles de la société Frog Navigation Technologies) fonctionnent à la demande comme un " ascenseur horizontal ". En 1999, un deuxième système a été mis en place par Frog pour assurer la liaison entre une gare de la banlieue de Rotterdam et un centre d'affaires. Ce deuxième système comporte 7 navettes automatisées. On peut avoir une description détaillée du système de navette de l'aéroport d'Amsterdam<sup>1</sup>.



Figure B.1: Véhicules automatisés de l'aéroport de Schiphol

Le guidage est assuré via des plots ou des balises dissimulées dans les pistes de circulation. On peut reconnaître facilement ces balises, représentées par un petit cercle blanc ou gris d'une dizaine de centimètres, visible sur la chaussée.

Si aucun véhicule n'est présent en station, il suffit d'appuyer sur le bouton d'appel et un véhicule viendra. Une pression sur le bouton vert pour ouvrir les portes permettant d'embarquer dans un des véhicules de 6 places assises à 4 roues. Ainsi, dans toutes les stations se trouvent un bouton d'appel. Les véhicules ne peuvent passer d'une ligne à l'autre qu'à la station principale et au niveau des garages (Non accessibles au public). Si on emprunte

---

<sup>1</sup><http://www.foudurail.net/ville/hopper.html>

un véhicule sur la ligne B, il faut rester sur cette même ligne. On ne peut changer de ligne qu'à la station principale.<sup>2</sup>

## B.2 CyberCars/CyberMove 2001

Fort de ces recherches et des premières applications dans le public, l'INRIA a proposé à la Commission Européenne le double programme de recherche CyberCars/CyberMove concernant les véhicules automatisés urbains. Les véhicules étudiés dans ces projets sont désormais appelés des **Cybercars**, selon une terminologie utilisée par le sénateur *Trégouët*, président de la Commission d'Innovation au Sénat. L'objectif principal du projet de CyberCars/CyberMove est d'accélérer le développement et la diffusion de ce système de transport en améliorant les exécutions et en abaissant le coût.



Figure B.2: Cycabs de la présentation du projet CyberCars/CyberMove

La ville d'*Antibes-Juan les Pins* et les partenaires des projets européens CyberCars/CyberMove ont organisé du 1 au 13 juin 2004 une démonstration de nouveaux véhicules urbains entièrement automatisés. Cette démonstration était une première européenne puisqu'elle a permis de tester en grandeur réelle l'intégration de véhicules entièrement automatisés dans un réseau urbain dédié, en cohabitation avec des modes doux (piétons, 2 roues). Le site du *Port Vauban*, au cœur même de la ville historique d'*Antibes-Juan les Pins*, a été sélectionné en tant que site test parmi une dizaine d'autres sites Européen. Cette expérimentation est l'aboutissement des résultats des travaux de recherche menés dans le cadre du 5<sup>ième</sup> Programme Cadre de Recherche Européen, qui pendant 3 ans a mobilisé 15 partenaires académiques, industriels et bureaux de conseil, coordonnées par l'INRIA. Le projet CyberMove a été financé par l'unité " Ville de demain et héritage culturel " des programmes de recherche européens Énergie, Environnement et Développement durable<sup>3</sup>.

<sup>2</sup><http://www.altivis.fr/-Un-modele-europeen-des-TIC-dans-.html>

<sup>3</sup><http://www.cybermove.org/>

Le résultat technique est probant : une navette a circulé du 3 au 13 juin 2004 le long du port de plaisance, transportant au total près de 3 000 personnes. Du point de vue urbanistique, il a été démontré que ce type de mobilité douce gagne, sur la voiture, au moins 2000  $m^2$  par kilomètre de voirie. Autrement dit, il est envisageable de restaurer le domaine public au profit des piétons et de l'animation sociale, tout en assurant la mobilité urbaine.

Enfin, l'enthousiasme général des premiers usagers semble indiquer une acceptation sociale aisée. La ville d'Antibes a d'ailleurs décidé d'étudier l'aménagement de la ligne telle qu'elle a été initialement projetée, sur la totalité du parcours (une boucle de 3 *Kms*). Cette ligne devrait permettre ainsi aux visiteurs venant en voiture de laisser leur véhicule à l'entrée de la ville dans un parking gratuit et de se rendre au port ou à la vieille ville sans engorger ni polluer le site.

### B.3 Le projet BODEGA 2003

Le projet **Bodega**<sup>4</sup> a pour but la navigation autonome et sûre en environnement urbain. Ce projet implique plusieurs partenaires issus du CNRS, de l'Université et de l'INRIA. Le but de ce projet est de développer des techniques pour la navigation automatique de véhicules en milieu urbain non instrumenté. Il est réalisé par une collaboration entre l'Ensil de Limoges, l'Utc à Compiègne, le Lasma à Clermont-Ferrand, le projet Icare de l'Inria Sophia-Antipolis, et les projets Texmex et Lagadic de l'Irisa.

Par rapport au projet MOBIVIP, le projet BODEGA est focalisé sur la " planification et l'exécution robuste de missions de déplacement dans des environnements urbains non instrumentés ". Le véhicule issu du projet BODEGA devra être capable de réaliser les tâches suivantes :

- Apprentissage automatique de la carte de l'environnement par navigation en mode manuel des véhicules
- Production d'une cartographie de type SIG (Système d'Information Géographique) comportant des amers (points de repères) visuels identifiés lors de l'apprentissage.

### B.4 MobiVip 2005

Le projet **MobiVip** (Véhicules Individuels Publics pour la Mobilité en centre ville) est un projet de recherche et développement du PREDIT 3<sup>5</sup>. Il réunit 5 laboratoires de recherche et 7 organismes privés, en vue d'expérimenter, de démontrer et d'évaluer l'impact des Véhicules Intelligents sur les services de mobilité en centre ville.

---

<sup>4</sup><http://jazz.ensil.unilim.fr/bodega/>

<sup>5</sup><http://www.predit.prd.fr/predit3/homePage.fo>



Figure B.3: Projet MobiVip

Les recherches menées par le projet MobiVip visent l'expérimentation de " briques technologiques " pour le déploiement des services de mobilité en milieu urbain. Ces services sont basés sur un système de transport Véhicules Individuels Publics et sur un système d'information qui s'intègre à la politique de gestion des déplacements dans un centre ville.

Pour atteindre cet objectif, plusieurs domaines de recherche convergent au sein de MobiVip : la conduite assistée et automatisée, les télécommunications, la modélisation des transports, et l'évaluation de service. Le projet a présenté des démonstrations sur des sites expérimentaux : Nancy est l'un d'eux, comme avant elle, la ville d'Antibes<sup>6</sup>.

## B.5 CRISTAL 2008

Le Cristal est un projet de transport urbain labellisé par le pôle de compétitivité "Véhicule du Futur" d'Alsace Franche-Comté. Ce projet consiste en la réalisation d'un nouveau mode de transport en commun, caractérisé par :

- Flotte de véhicules disponible en tout temps et lieu dans un périmètre défini.
- Répond à une demande ponctuelle instantanée (sans réservation), dispersée dans le temps et dans l'espace.
- Système de transport complémentaire aux autres systèmes existants.

Fonctionnalités en mode navette (conducteur professionnel) :

- Répond à une demande de déplacement planifiée (événement, type de service spécifique) ponctuellement concentrée dans le temps et l'espace.
- Offre la possibilité d'adapter la capacité par attelage en fonction de la demande réelle.

L'LOHR Industrie est le porteur industriel du projet CRISTAL et les partenaires industriels/universitaires sont : Transitec, GEA, VULog, SeT, LORIA, LASMEA, IMARA.

<sup>6</sup><http://www-sop.inria.fr/mobivip/>

Le processus de pré-industrialisation a été lancé à la mi-juin 2007 et le premier prototype est attendu d'ici le milieu 2012. Deux études territoriales sont en cours à Strasbourg et Montbéliard. Elles sont destinées à développer ce transport innovant dans des conditions d'intégration qui auront été préalablement optimisées.

Les projets qui ont été présentés précédemment démontrent bien la volonté de créer des infrastructures permettant une autonomie complète pour le transport urbain. Le projet CRISTAL repose sur l'accroche immatérielle entre véhicules pour faciliter l'accrochage et le désaccrochage des véhicules du train.





# MODÈLE D'INTERACTION

## C.1 Paramètres du modèle d'interaction

Les agents perçoivent des forces, au sens physique du terme, des autres agents :

- **Force d'attraction du ressort** : cette force s'applique entre un couple d'agent, le précédent et le suivant. Elle permet de réaliser une attraction du véhicule suivant sur la trajectoire du précédent.
- **Force d'amortissement** : cette force est comparable à la force réalisé par un piston. Elle permet de freiner le véhicule suiveur afin d'éviter les collisions.

### C.1.1 Force théorique du modèle d'interaction :

#### C.1.1.1 Force d'attraction du ressort :

La force d'attraction peut-être apparentée à la force d'un ressort idéal. Selon la loi de *Hooke*, un ressort étiré, de par son élasticité, exerce une force de rappel proportionnelle à l'allongement du ressort.

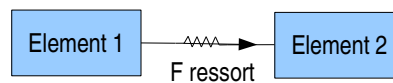


Figure C.1: Repère utilisé

Expression de la force :

$$\vec{F}_{ressort} = -k(l - l_0)\vec{e}_{12}. \quad (C.1)$$

$k$  étant la constante, appelé raideur du ressort

$\vec{e}_{12}$  est le vecteur unitaire de l'axe  $Element_1 Element_2$  orienté de  $Element_1$  vers  $Element_2$ .

$l$  : longueur du ressort.

$l_0$  : longueur à vide du ressort.

### C.1.1.2 Force d'amortissement du piston :

La force d'amortissement est apparentée à la force produite par un piston,

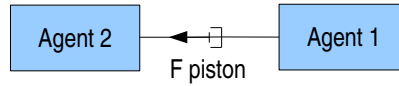


Figure C.2: Forces subies par deux éléments reliés par un piston

$$\vec{F}_{Piston} = -\lambda * \Delta \vec{V}. \quad (C.2)$$

$\lambda$  étant la constante d'amortissement.

$\Delta \vec{V}$  : différence de vitesse entre le suiveur et le précédent.

### C.1.2 Paramètre du modèle d'interaction :

Le modèle d'interaction est basé sur 6 paramètres : masse  $m$  du véhicule suiveur, raideur  $k$ , amortissement  $h$ , longueur à vide  $l_v$ , raideur de torsion  $k_t$  et amortissement de torsion  $h_t$  :

- La masse  $m$  est un paramètre permettant de prendre en compte l'inertie du véhicule. En effet, plus la masse est importante, plus le véhicule mettra du temps à réagir.
- La raideur  $k$  représente la capacité du véhicule à répondre face à une variation de la distance inter-véhicule.
- Le paramètre  $h$  permet d'amortir le système "masse-ressort" afin de ne pas avoir d'oscillation.
- La longueur à vide  $l_v$  correspond à la distance pour laquelle le système "masse-ressort" ne réagit plus. Par conséquent cette distance correspond à la distance de suivi inter-véhicule souhaitée.
- La raideur de torsion  $k_t$  et l'amortisseur de torsion  $h_t$  sont les deux paramètres du modèle "masse-ressort" sur l'axe vertical.

# CAPTEURS

---

## D.1 Capteurs télémétriques

Les capteurs télémétriques<sup>1</sup> fonctionnent sur le principe de la mesure par temps de vol : une impulsion lumineuse très courte est émise par une diode laser infrarouge (en classe 1, non dangereuse pour l'œil), simultanément une horloge est démarrée. La partie de la lumière diffusée par un obstacle quelconque et se trouvant dans l'axe d'émission revient vers le capteur et est détectée par un récepteur photo-électronique. Le signal de réception arrête l'horloge. Par ailleurs le faisceau lumineux émis est dévié par un miroir tournant, permettant ainsi de couvrir une surface.

A partir de la mesure de temps séparant la réception de l'émission, il est possible de calculer la distance du capteur à la cible. Connaissant par ailleurs la position angulaire de chaque "tir", on peut en déduire la position exacte de l'objet dans un plan (coordonnées polaires).

Les capteurs télémétriques possèdent de nombreux avantages par rapport aux systèmes de perception vidéo :

1. Large champ de mesure important jusqu'à 180°
2. Très précis quelle que soit la distance
3. Indépendance des tailles, forme et couleur de l'objet
4. Mesure contours en temps réel
5. Pas de marquage spécial
6. Peut être installé dans n'importe quelle position
7. Système actif ne nécessitant pas d'éclairage spécifique
8. Utilisable dans tous les environnements

---

<sup>1</sup>[www.sick.fr](http://www.sick.fr)

### D.1.1 LMS 200

Le LMS 200 (cf. figure D.1) est un système de mesure à laser sans contact qui scrutent leur environnement en deux dimensions (radar laser). Ces capteurs sont principalement conçus pour l'usage en intérieur et n'utilisent aucun réflecteur ni aucun repère. Grâce à leur haute résolution, il est possible de confier à ces systèmes des tâches qui jusqu'à maintenant, soit étaient impossibles, soit nécessitaient un matériel considérable.



Figure D.1: Télémètre laser LMS 200

Caractéristiques techniques :

1. Portée : 80 M
2. Résolution : +/-10 mm
3. Précision : +/-30 mm
4. Résolution angulaire : 0.25 °/0.5 °/1°
5. Temps réponse : 52 / 26 / 13 ms
6. Ouverture entre : 100° et 180°
7. Adapté au brouillard

Dans notre cas, nous avons paramétré le LMS200 avec un pas de un degré et une ouverture de 180°.

## D.1.2 IBEO

Le scanner ibeo LUX<sup>2</sup> se distingue en particulier par sa robustesse dans tous les environnements climatiques (cf. image D.2). Il balaye et mesure en même temps dans 4 plans parallèles. De plus, il peut évaluer, par mesure et par plan, jusqu'à 3 réflexions successives, et ceci dans une plage de distance comprise entre 0,3 et 200 mètres. Il fonctionne dans les conditions environnementales les plus défavorables, comme, par exemple, sous la pluie ou la neige.

Sa mesure reste fiable quelles que soient les conditions climatiques et de luminosité. Le boîtier du capteur a été conçu de forme compacte pour pouvoir s'intégrer directement dans un véhicule. Le scanner à laser ibeo LUX fournit une évaluation complète de l'environnement sous forme d'objets avec leurs propriétés pertinentes - position, vitesse, type (véhicule, piéton et cycliste).



Figure D.2: Télémètre laser IBEO

Caractéristiques techniques :

1. Portée : 0.3 m à 200 m
2. Résolution angulaire : 0.1° à 1°
3. Fréquence : 25 HZ
4. Ouverture entre : 100°

Dans notre application, nous avons paramétré le capteur IBEO avec un pas angulaire de 0.1° et une ouverture de 100°.

---

<sup>2</sup><http://www.ibeo-as.com/english/default.asp>

## D.2 Capteur de type GPS

La Cinématique temps réel (Real Time Kinematic, en anglais ou RTK)<sup>3</sup> est une technique de positionnement par satellite basée sur l'utilisation de mesures de phase des ondes porteuses des signaux émis par le système GPS, GLONASS ou Galileo. Une station de référence fournit des corrections en temps réel permettant d'atteindre une précision de l'ordre du centimètre.



Figure D.3: GPS Pro Flex 500

Le GPS du laboratoire SeT (cf. image D.3) est un PROFLEX 500<sup>4</sup>, dont voici les caractéristiques techniques :

1. Multi-constellation : GPS + Glonass + SBAS, et bientôt Galileo
2. 75 canaux, sortie données brutes et position à 10Hz (bientôt 20Hz)
3. Connectivité élargie : 1 port RS232/RS422, 2 ports RS232, 1 port USB, 1 port Ethernet, sortie 1PPS, compatible Bluetooth, marqueur d'évènements

<sup>3</sup>[http://fr.wikipedia.org/wiki/Real\\_Time\\_Kinematic](http://fr.wikipedia.org/wiki/Real_Time_Kinematic)

<sup>4</sup><http://www.promagellangps.com/fr/news/releases/index.asp>

# APPROCHE DE MODÉLISATION, SPÉCIFICATION ET VÉRIFICATION POUR DES SMAR D'INSPIRATION PHYSIQUE

---

## E.1 Perspective

Les applications de systèmes multi-agents réactifs (SMAR) se caractérisent, entre autres, par deux aspects. Premièrement, le rôle primordial que jouent les interactions, aussi bien agent-agent qu'agent-environnement. Deuxièmement, le fait que beaucoup d'applications (dont celles qui incluent des dispositifs matériels en général mobiles, situés dans un environnement physique) doivent satisfaire un certain nombre de propriétés de sûreté.

Nous présentons ici une approche de conception de SMAR qui intègre ces deux aspects. La prise en compte des propriétés de sûreté passe par la construction d'un modèle formel du comportement de chaque classe d'agent, où les interactions sont modélisées à partir de mécanismes inspirés de la physique. Les lois physiques des dits mécanismes fournissent des fonctions permettant le calcul des actions en fonction des perceptions. On obtient également des conditions pour la satisfaction des propriétés de sûreté grâce à l'analyse du modèle physique des interactions (cinématique, énergétique, ...).

Nous allons présenter cette approche sous la forme d'un ensemble d'activités : construction d'un modèle, analyse et vérification. Certaines de ces activités sont en relation de dépendance, au sens où une activité fournit des informations nécessaires à l'autre.

Dans ce qui suit, les statecharts jouent un rôle qui est dû aux capacités de ce langage pour l'expression d'aspects qui interviennent dans la description du comportement d'une classe d'agents. En particulier, la hiérarchie des états d'un statechart suivant deux types de décomposition en sous-états, sont adaptés à la description de comportements qui se décomposent en sous-comportements.



### E.1.1 Vue générale

L'approche proposée est composée des activités suivantes :

- **Construction d'un modèle abstrait de comportement** : pour chaque classe d'agent, construction d'un modèle de comportement, sous la forme d'un statechart qui spécifie comment les différentes activités de l'agent s'articulent, du point de vue comportemental.
- **Spécification physique des mécanismes d'interaction** : définition des liens fonctionnels entre percepts et actions, déterminés à partir d'un mécanisme inspiré de la physique.
- **Analyse du modèle physique des interactions** : l'objectif est, d'une part, de prouver des propriétés indépendantes de la mise en oeuvre telles que la stabilité, et d'autre part, de formuler des conditions qui se rattachent aux propriétés sûreté, objet de la vérification.
- **Construction du modèle de spécification** de la classe d'agent, sous la forme d'une classe *OZS*, sur la base des modèles abstraits de comportement, en intégrant les fonctions de perception et de calcul des actions à partir des percepts.
- **Vérification des propriétés de sûreté** : traduction du modèle *OZS* vers SAL et vérification par model-checking.
- **Implémentation des comportements** d'agent, à partir des modèles de spécification vérifiés.

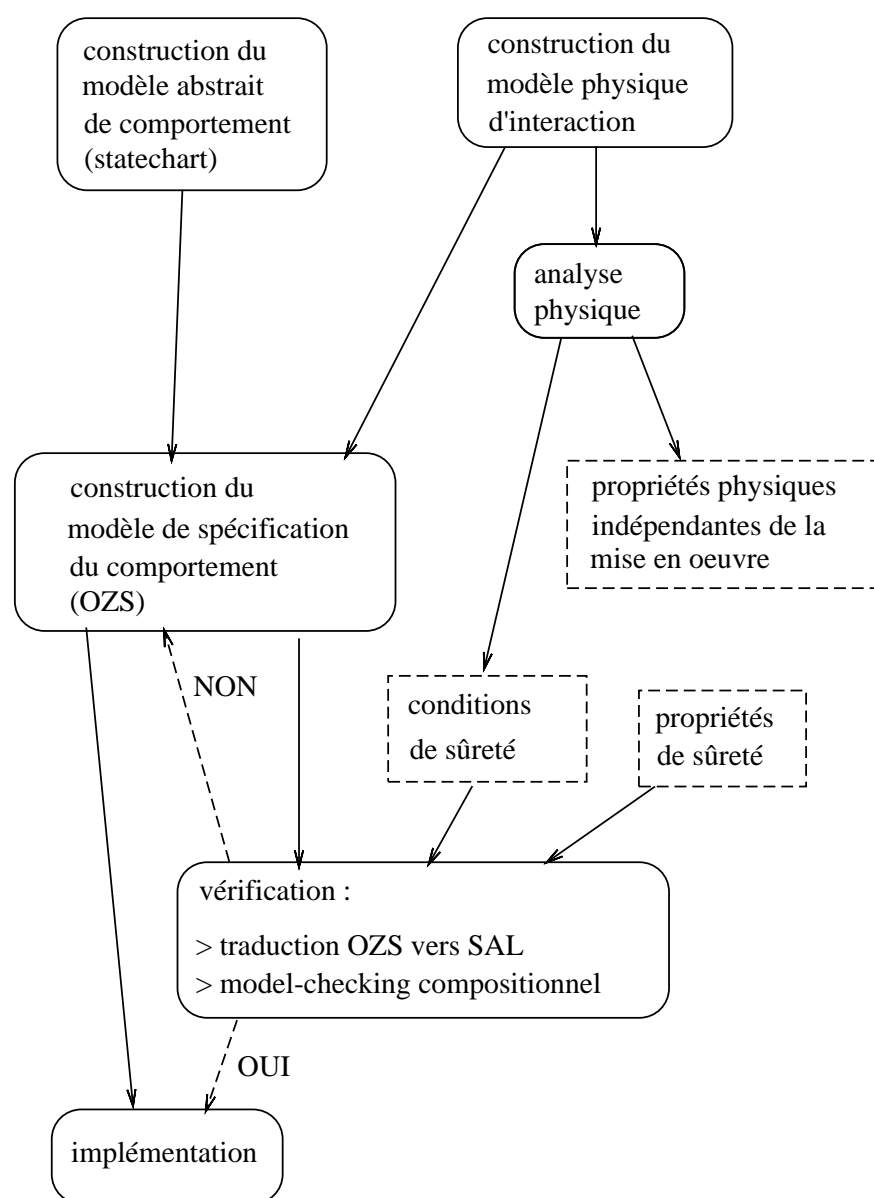


Figure E.1: Dépendance entre activités

La figure E.1 présente la relation de dépendance entre activités. Cette relation exprime le fait qu'une activité  $A_2$  puisse dépendre de données ou informations qui proviennent de  $A_1$  ( $A_1 \rightarrow A_2$ ). Considérons avec un peu plus de détail chacune des activités mentionnées.

### E.1.2 Construction d'un modèle abstrait de comportement

Il s'agit de définir, pour chaque classe d'agent, la structure du statechart qui modélisera le comportement des entités de la classe. Le comportement d'une classe d'agent présente souvent cette particularité : un agent intègre un ensemble de comportements, associés aux différents composants de son architecture. Ces sous-comportements s'intègrent suivant des

considérations :

Chaque composant de l'architecture de l'agent doit contribuer avec un sous-comportement, modélisable par un statechart. Chacun de ces sous-comportements s'intégrant dans le comportement de la classe d'agent, suivant une combinaison de deux mécanismes, qui correspondent aux décompositions AND et OR des états d'un statechart. La figure E.2 illustre les deux types de composition. l'expression  $\langle cond \rangle$  formule la condition de transition entre sous-comportements. Le comportement de chaque composant de l'architecture d'une classe d'agent est spécifié à partir de ses fonctions, en se basant en général sur des interactions. En gros, chaque composant intervient dans l'élaboration de percepts à partir de ses interactions avec l'environnement, dans la production de réaction adressées à l'environnement, dans la détermination de l'état interne de l'agent, ou une combinaison des trois activités.

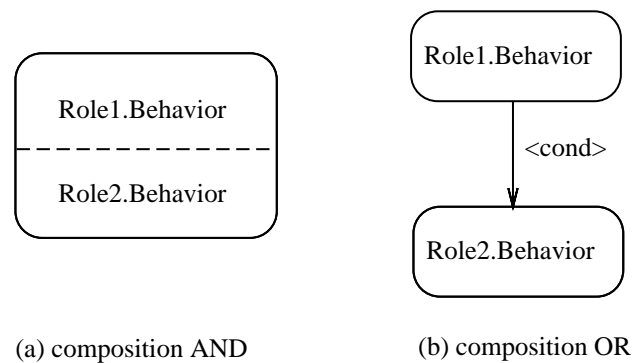


Figure E.2: Composition des rôles

### E.1.3 Construction du modèle physique d'interaction

Cette activité consiste en la définition d'un modèle qui lie les valeurs produites par les fonctions de perception aux valeurs des variables qui déterminent les réactions de l'agent. L'adoption d'un modèle d'inspiration physique permet de faire en sorte que ce lien soit fonctionnel : il est défini à partir des lois qui régissent le mécanisme physique adopté. Le modèle physique d'interaction n'est pas destiné à constituer une boucle de régulation du comportement de l'agent, mais plutôt à fournir une référence pour une boucle d'asservissement. Dans ce cas, le schéma ou architecture fonctionnelle correspond à une double boucle de régulation, comme l'illustre la figure E.3.

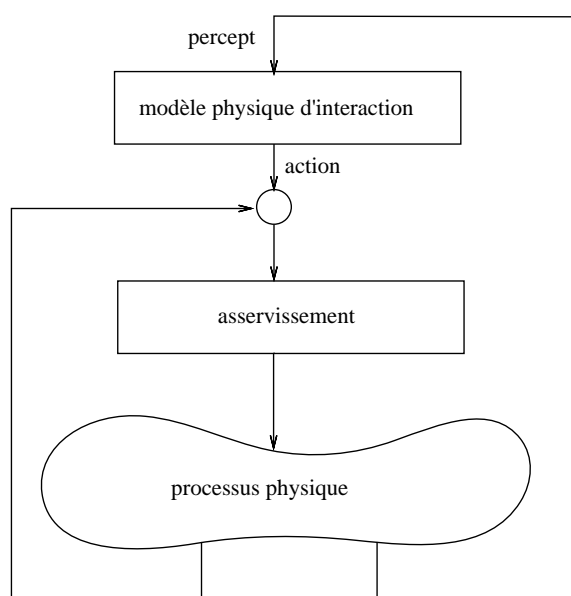


Figure E.3: Boucle perception-action

La boucle d'asservissement porte sur la grandeur physique qui matérialise l'action calculée par l'intermédiaire du modèle physique d'interaction. On peut cependant envisager des cas où la boucle interne d'asservissement n'est pas présente et où le modèle physique d'interaction sert en même temps de régulateur. Dans ces cas, l'analyse du modèle physique d'interaction doit en principe aborder la question des propriétés de base d'un régulateur, telle que la stabilité.

#### E.1.4 Analyse du modèle physique d'interaction

Comme l'anticipe la section E.1.3, l'un des buts de cette activité est l'étude des propriétés communément associées à la mise en oeuvre de boucles de régulation. Un deuxième objectif est la validation de l'applicabilité des lois physiques du mécanisme adopté en tant que modèle d'interaction, lorsque l'on tient compte des propriétés propres aux dispositifs et objets matériels qui composent le système. En d'autres termes, cette activité assure la pertinence du passage du virtuel au réel, en analysant la réponse du système matériel lorsqu'on fait intervenir le mécanisme d'interaction. Un troisième objectif de cette activité consiste en la formulation de conditions qui vont jouer le rôle d'hypothèses dans le processus de vérification des propriétés de sûreté. Ces conditions induisent une prise en compte de certains aspects de la réalité matérielle lors de la vérification.

#### E.1.5 Construction du modèle de spécification de la classe d'agent

Cette activité consiste principalement à raffiner le modèle abstrait de comportement d'une classe d'agent  $A_i$ , pour obtenir une classe *OZS*  $C_i$  qui sera le modèle de spécifica-

tion fonctionnelle et comportementale de  $A_i$ . Le schéma d'état de la classe  $C_i$  contiendra les déclarations des variables d'état. Les schémas d'opération de la classe  $C_i$  contiendront les définitions des fonctions  $F_p$  et  $F_a$  qui produisent respectivement les percepts et les actions d'un agent de la classe. Ces fonctions résultent de l'élaboration du modèle physique d'interaction (section E.1.3). Le schéma de comportement de la classe  $C_i$  résulte du raffinement du statechart qui constitue le modèle abstrait de comportement de la classe (section E.1.2). Ce raffinement consiste essentiellement à détailler les statecharts définissant les sous-comportements mentionnés au paragraphe E.1.2. Les opérations définies dans les schémas d'opération de la classe  $OZS$ , qui sont liées aux actions de perception ou de réaction de l'agent, seront invoquées lors des transitions entre états qui modélisent les sous-comportements.

### E.1.6 Verification

L'activité de vérification doit aboutir à établir avec la force d'une preuve, que des propriétés de sûreté sont satisfaites par un système, en fonction du modèle de spécification. Cette activité fait appel à un environnement ou une boîte à outils de vérification qui fournit, en général, un formalisme  $\mathcal{F}$  pour la construction des modèles à vérifier, un langage  $\mathcal{L}$  pour l'expression des propriétés à vérifier et des outils de preuve et/ou de model-checking. L'activité se déroule en procédant aux étapes suivantes :

- Construction d'un modèle de système multi-agents, par instantiation des différentes classes d'agent. Si nécessaire, un modèle fonctionnel et comportemental de l'environnement sera également incorporé. La construction du modèle d'environnement peut résulter d'activités analogues à celles qui ont conduit aux modèles de classes d'agents.
- Construction de l'objet sémantique associé, qui est une entité syntaxique du formalisme  $\mathcal{F}$ . Dans notre cas, il s'agit d'un système de transition conforme au langage de modélisation de la boîte à outils SAL.
- Expression des propriétés de sûreté à l'aide du langage  $\mathcal{L}$ . Dans le cas de SAL, il s'agit d'un fragment de la logique temporelle.
- Vérification proprement dite à l'aide, dans notre cas, des model-checkers de SAL.

Au chapitre 8, nous proposons une méthode de vérification basée sur une règle de vérification compositionnelle adaptée à la vérification de systèmes tels que les SMAR, composés d'entités autonomes en interaction. La méthode de vérification proposée, par son caractère compositionnel, permet de mieux maîtriser le risque d'explosion combinatoire de l'espace d'états globaux du système, qui est l'un des enjeux de la vérification.

### **E.1.7 Implémentation**

L'implémentation est la transcription dans un langage de programmation tel que le *java* ou le *C++* du système multi-agents spécifié précédemment. Une attention particulière doit être portée sur la définition des agents et de leurs communications. Des outils SAL, en cours d'élaboration, permettent de transformer la spécification SAL directement en *java* ou en *C++*.



---

SIXIÈME PARTIE

---

# **Bibliographie**

---





---

# Bibliographie

---

- Martín Abadi and Leslie Lamport. Composing specifications. *ACM Trans. Program. Lang. Syst.*, 15(1) : 73–132, 1993. ISSN 0164-0925. doi : <http://doi.acm.org/10.1145/151646.151649>.
- Jean-Raymond Abrial. *The B-Book : Assigning Programs to Meanings*. Cambridge University Press, 1996.
- J. L. Austin. *Quand dire c'est faire*. Le Seuil, 1970.
- T. Balch. Hierarchic social entropy : An information theoretic measure of robot group diversity. *Autonomous Robots*, 2000.
- R. Balian. *From Microphysics to Macrophysics*. Springer, 2007.
- E Bonabeau, M Dorigo, and G Theraulaz. *Swarm Intelligence : From Natural to Artificial Systems*. Oxford University Press, 1999.
- J. Borenstein, Y. Koren, and Senior Member. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7 :278–288, 1991.
- Nikolaos Bourbakis and Michael Findler. Smart cars as autonomous intelligent agents. *Proceedings of the International Conference on Tools with Artificial Intelligence*, pages 25 – 32, 2001. ISSN 1063-6730. URL <http://dx.doi.org/10.1109/ICTAI.2001.974445>. Smart cars ;Autonomous intelligent agents ;Traffic patterns ;.
- Edward Chang. Compositional verification of reactive and real-time systems. *Stanford University Stanford, CA, USA*, 1993.
- Samuel Colin, Arnaud Lanoix, Olga Kouchnarenko, and Jeanine Souquières. Towards validating a platoon of cristal vehicles using csp11b. In *AMAST 2008 : Proceedings of the 12th international conference on Algebraic Methodology and Software Technology*, pages 139–144, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-79979-5. doi : [http://dx.doi.org/10.1007/978-3-540-79980-1\\_11](http://dx.doi.org/10.1007/978-3-540-79980-1_11).
- Samuel Colin, Arnaud Lanoix, Olga Kouchnarenko, and Jeanine Souquières. Using csp11b components : Application to a platoon of vehicles. *Formal Methods for Industrial Critical Systems*, pages 103–118, 2009. doi : [http://dx.doi.org/10.1007/978-3-642-03240-0\\_11](http://dx.doi.org/10.1007/978-3-642-03240-0_11).
- Jean Michel Contet, Franck Gechter, Pablo Gruer, and Abder Koukam. Multiagent system model for vehicle platooning with merge and split capabilities. *Third International Conference on Autonomous Robots and Agents ICARA*, pages 41–46, 2006.
- Jean Michel Contet, Franck Gechter, Pablo Gruer, and Abder Koukam. Evaluation of global system state thanks to local phenomenona. *biennial European Conference on Artificial Intelligence (ECAI), Grèce, Patras, 21 au 27 juin 2008. Impact Factor : 0.69 (top 38.49%)*, 2008.
- Jean Michel Contet, Franck Gechter, Pablo Gruer, and Abder Koukam. Bending virtual spring-damper : a solution to improve local platoon control. *International Conference on Computational Science (ICCS)*, 2009.

- Massimo Cossentino, Nicolas Gaud, Vincent Hilaire, Stéphane Galland, and Abderrafîaa Koukam. ASPECS : an agent-oriented software process for engineering complex systems. *Autonomous Agents and Multi-Agent Systems*, 2009.
- Pascal Daviet and Michel Parent. Longitudinal and lateral servoing of vehicles in a platoon. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 41 – 46, 1996. URL <http://dx.doi.org/10.1109/IVS.1996.566349>. Automatic driving ;Platooning techniques ;.
- Leonardo De Moura, Harald Rueß, and Maria Sorea. Bounded model checking and induction : From refutation to verification. *Computer Aided Verification*, 2725, 2003.
- Y. Demazeau. From interactions to collective behaviour in agent-based systems. *European Conference on Cognitive Science*, 1995.
- Alexis Drogoul. *De la simulation multi-agents a la résolution collective de problèmes*. PhD thesis, University Paris 6, 1993.
- J Dugoff, P Fanches, and L Segal. An analysis of tire traction properties and their influence on vehicle dynamic performance. *In SAE*, 1970.
- Roger Duke and Gordon Rose. *Formal Object Oriented Specification Using Object-Z*. Cornerstones of Computing, Palgrave Macmillan, 2000.
- Jacques Ferber. *Les Systèmes Multi-Agents*. InterEditions, 1995.
- Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations : An organizational view of multi-agent systems. In Paolo Giorgini, Jörg P. Müller, and James Odell, editors, *AOSE*, volume 2935 of *Lecture Notes in Computer Science*, pages 214–230. Springer, 2003. ISBN 3-540-20826-7. URL <http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2935&page=214>.
- U. Franke, F. Boettiger, Z. Zomotor, and D. Seeberger. Truck platooning in mixed traffic. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1 – 6, 1995. URL <http://dx.doi.org/10.1109/IVS.1995.528248>. Truck platooning ;Mixed traffic ;Autonomously guided trucks ;.
- Hans Fritz. Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic : Experimental results from the chauffeur project. *IEEE Conference on Control Applications - Proceedings*, 2 :1348 – 1352, 1999. URL <http://dx.doi.org/10.1109/CCA.1999.801168>. Heavy duty trucks ;.
- William L Garrard and Reggie J Caudill. Dynamic behavior of strings of automated transit vehicles. *SAE Transactions*, pages pp. 1365–1378, 1977.
- Nicolas Gaud, Franck Gechter, Stéphane Galland, and Abderrafîaa Koukam. Holonic multiagent multilevel simulation : Application to real-time pedestrians simulation in urban environment. In *Twentieth International Joint Conference on Artificial Intelligence, IJCAI'07*, pages 1275–1280, Hyderabad, India, 2007.
- F. Gechter, V. Chevrier, and F. Charpillat. A reactive multi-agent system for localization and tracking in mobile. In *In 16th IEEE International Conference on Tools with Artificial Intelligence - ICTAI'2004.*, 2004.
- Franck Gechter. *Modèle multi-agent réactif pour la fusion de percepts : application à la localisation et au suivi en robotique mobile*. PhD thesis, Université Henri Pointcaré, Nancy 1, 2003.
- S.K. Gehrig and F.J. Stein. Elastic bands to enhance vehicle following. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 597 – 602, 2001. URL <http://dx.doi.org/10.1109/ITSC.2001.948727>.

- Ottmar Gehring and Hans Fritz. Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 117 – 122, 1997. URL <http://dx.doi.org/10.1109/ITSC.1997.660461>. Truck platooning;Vehicle to vehicle communication;Longitudinal control concept;Two layered control concept;String stability ;.
- J. Christian Gerdes and J. Karl Hedrick. "loop-at-a-time" design of dynamic surface controllers for nonlinear systems. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 124(1) : 104 – 110, 2002. ISSN 0022-0434. URL <http://dx.doi.org/10.1115/1.1433804>. Surface controllers;Control engineering;Sliding controllers;Sequential design procedure ;.
- S Goss, J.L Beckers, R andDeneubourg, S Aron, and J.M pasteels. trail laying and trail following can solve foraging problems for ant colonies. *Behavioural Mechanisms of food Selection*, 20, 1990.
- Pablo Gruer, Vincent Hilaire, Abder Koukam, and P. Rovarini. Heterogeneous formal specification based on object-z and statecharts : semantics and verification. *Journal of Systems and Software*, 70(1-2) :95–105, 2004.
- Orna Grumberg and David E. Long. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems*, 16, 1991.
- H. Habrias and B. Griech. Formal specification of dynamic constraints with the b method. *Proceedings of the International Conference on Formal Engineering Methods, ICFEM*, pages 304 – 314, 1997. URL <http://dx.doi.org/10.1109/ICFEM.1997.630437>. Formal specifications;Dynamic constraints ;.
- S. Halle and B. Chaib-draa. A collaborative driving system based on multiagent modelling and simulations. *Transp. Res. C, Emerg. Technol. (UK)*, 13(4) :320 – 45, 2005. ISSN 0968-090X. URL <http://dx.doi.org/10.1016/j.trc.2005.07.004>.
- J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6) :21 – 32, 1994. ISSN 0272-1708. URL <http://dx.doi.org/10.1109/37.334412>.
- Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Developing organised multiagent systems using the MOISE. *IJAOSE*, 1(3/4) :370–395, 2007. URL <http://dx.doi.org/10.1504/IJAOSE.2007.016266>.
- P. Ioannou and Z. Xu. Throttle and brake control systems for automatic vehicle following. *IVHS Journal*, 1 (4) :345 –, 1994. ISSN 1065-5123.
- N.R. Jennings, T.J. Norman, and P. Faratin. Adept : An agent-based approach to business process management. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 27(4) :32 – 39, 1998. ISSN 0163-5808.
- Seong Min Jeon and Jae Weon Choi. Design of a hybrid controller for the platoon maneuvers. *SICE 2001. Proceedings of the 40th SICE Annual Conference. International Session Papers (IEEE Cat. No.01TH8603)*, pages 32 – 7, 2001. URL <http://dx.doi.org/10.1109/SICE.2001.977801>. automated highway systems ;AHS ;safety ;highway infrastructure ;traffic automation ;large scale multiagent complex dynamical system ;hierarchical hybrid controller ;safety distance policy ;platoon maneuvers ;merge ;split ;.
- S.M. Jeon, J.W. Choi, and H. Hashimoto. Hybrid modeling and control for a platoon maneuvers. *IEEE International Symposium on Industrial Electronics*, 3 :1542 – 1547, 2001. URL <http://dx.doi.org/10.1109/ISIE.2001.931935>. Hybrid system ;Platoon maneuvers ;Vehicle dynamics ;.
- Bengt Jonsson. Compositional specification and verification of distributed systems. *ACM Trans. Program. Lang. Syst.*, 16(2) :259–303, 1994. ISSN 0164-0925. doi : <http://doi.acm.org/10.1145/174662.174665>.

- K.R.S. Kodagoda, W.S. Wijesoma, and E.K. Teoh. Fuzzy speed and steering control of an agv. *IEEE Transactions on Control Systems Technology*, 10(1) :112 – 120, 2002. ISSN 1063-6536. URL <http://dx.doi.org/10.1109/87.974344>. Proportional integral derivative (PID) controllers ;
- Abderrafiaa Koukam. Sujet : De la modélisation des systèmes discrets à la spécification formelle des systèmes multi-agents : méthodes, formalismes et outils. *Habilité à Diriger des Recherches (HDR), Université de Bourgogne, Rapporteurs : J. Kouloumdjian, J.P. Müller, A. Napoli, Examineurs : L. Chen, P. Deschizeaux, P. Gourriat, J.M. Pallo, K. Yetongnon.*, 28 Janvier 2000.
- Arnaud Lanoix. Event-b specification of a situated multi-agent system : Study of a platoon of vehicles. In *TASE '08 : Proceedings of the 2008 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, pages 297–304, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3249-3. doi : <http://dx.doi.org/10.1109/TASE.2008.39>.
- Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- C. Laugier, I. Paromtchik, and M. Parent. Developing autonomous maneuvering capabilities for future cars. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 68 – 73, 1999. URL <http://dx.doi.org/10.1109/ITSC.1999.821029>. Autonomous maneuvering ;Automated urban vehicles ;
- G.D. Lee and S.W. Kim. A longitudinal control system for a platoon of vehicles using a fuzzy-sliding mode algorithm. *Mechatronics*, 12(1) :97 – 118, 2002. ISSN 0957-4158. URL [http://dx.doi.org/10.1016/S0957-4158\(00\)00063-5](http://dx.doi.org/10.1016/S0957-4158(00)00063-5). Control algorithms ;
- Hyeongcheol Lee and Masayoshi Tomizuka. Adaptive vehicle traction force control for intelligent vehicle highway systems (ivhss). *IEEE Transactions on Industrial Electronics*, 50(1) :37 – 47, 2003. ISSN 0278-0046. URL <http://dx.doi.org/10.1109/TIE.2002.807677>.
- Horst Leipholz. *An introduction to the stability of Dynamic systems and rigid bodies*. Translation, 1987.
- C. Liang and H. Peng. Optimal adaptive cruise control with guaranteed string stability, 1998. URL [citeseer.ist.psu.edu/liang99optimal.html](http://citeseer.ist.psu.edu/liang99optimal.html).
- Xiao-Yun Lu, H.-S. Tan, S. Shladover, and J.K. Hedrick. Nonlinear longitudinal controller implementation and comparison for automated cars. *Trans. ASME, J. Dyn. Syst. Meas. Control (USA)*, 123(2) :161 – 7, 2001. ISSN 0022-0434. URL <http://dx.doi.org/10.1115/1.1368115>. nonlinear longitudinal controller ;automated cars ;platooning ;dynamic sliding mode controller ;dynamic back-stepping multiple surface controller ;adaptive sliding mode controller ;
- P. Lucidarme, O. Simonin, and A. Liegeois. Implementation and evaluation of a satisfaction/altruism based architecture for multi-robot systems. *Proceedings - IEEE International Conference on Robotics and Automation*, 1 :1007 – 1012, 2002. ISSN 1050-4729. URL <http://dx.doi.org/10.1109/ROBOT.2002.1013487>. Multirobot systems ;Minirobots ;Multiple cooperating robotic vehicles ;Mobile manipulators ;
- A. M. Lyapunov. *The general problem of the stability of the motion*. Taylor & Francis, 1992.
- Samer A. Masoud and Ahmad A. Masoud. Constrained motion control using vector potential fields, 2000.
- D. H. McMahon, J. K. Hedrick, and S. E. Shladover. Vehicle modelling and control for automated highway systems. *Proceedings of the American Control Conference*, pages 297 – 303, 1990. ISSN 0743-1619. Automated Vehicle Platoons ;
- A. D. McNaught and A. Wilkinson. *Compendium of Chemical Terminology, The Gold Book, Second Edition*. Blackwell Science, 1997.

- David Meignan, Olivier Simonin, and Abderrafiaa Koukam. Simulation and evaluation of urban bus-networks using a multiagent approach. *Simulation Modelling Practice and Theory*, 15(6) :659–671, July 2007. doi : 10.1016/j.simpat.2007.02.005. URL <http://dx.doi.org/10.1016/j.simpat.2007.02.005>.
- Fabien Michel. The irm4s model : the influence/reaction principle for multiagent based simulation. In *AAMAS '07 : Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5. doi : <http://doi.acm.org/10.1145/1329125.1329289>.
- Javier Minguez, Associate Member, and Luis Montano. Nearness diagram (nd) navigation : Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20 :2004, 2004.
- John J. Moskwa and J. Karl Hedrick. Nonlinear algorithms for automotive engine control. *IEEE Control Systems Magazine*, 10(3) :88 – 93, 1990. ISSN 0730-6598. URL <http://dx.doi.org/10.1109/37.55129>.
- S Moujahed, O Simonin, A Koukam, and K ghedira. A reactive agent based approach to facility location : Application to transport. *Workshop on Agents in Traffic and Transportation in the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006.
- S. Moujahed, N. Gaud, and D. Meignan. A self-organizing and holonic model for optimization in multi-level location problems. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 2, pages 1053–1058, June 2007. doi : 10.1109/INDIN.2007.4384921.
- Sana Moujahed. *Approche Multi-Agents Réactive pour l'Optimisation de Systèmes Spatialement Distribués et Dynamiques : Application aux Problèmes de Positionnement*. PhD thesis, Université de Franche-Comté, Université de Technologie de Belfort-Montbéliard, 2007.
- M.R.Jean. Emergence et sma. *5eme Journees Francophones sur l Intelligence Artificielle Distribuee et les Systemes Multi-Agents*, Quinqueton, Thomas, Trousse eds, Hermes, p. 323-342, 1997., 1997.
- Jean-Pierre Muller. Emergence of collective behavior and problem solving. In *Proceedings of Engineering Societies in the Agents World, ESAW 03, LNAI 3071.*, pages 1–21. Springer Verlag, 2004.
- J. Bom P. Martinet, B. Thuilot. Autonomous navigation and platooning using a sensory memory. *International IEEE Conference on Intelligent Robots and Systems, IROS'06, Beijing, China, October 2006*, 2006.
- H. Van Dyke Parunak. "go to the ant" : Engineering principles from natural multi-agent systems, 1999.
- H.V.D. Parunak. Go to the ant. In *Engineering Principles from Natural Agent Systems. Annals of Operations Research*, 1997.
- H.V.D. Parunak and S. Brueckner. Entropy and self-organization in multi-agent systems. *International Conference on Autonomous Agents*, 2001.
- Howard Reiss. *Methods of Thermodynamics*. Courier Dover Publications, 1997.
- Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. *Computer Graphics (ACM)*, 21(4) :25 – 34, 1987. ISSN 0097-8930. BEHAVIORAL MODELING ;OBJECT MODELING ;SCENE UNDERSTANDING ;.
- Marcel Riess. Qualification d'un télémètre à balayage laser pour la robotique mobile : Intégration et expérimentations. Master's thesis, Grenoble, 2000.
- Stuart J. Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Prentice Hall, (second edition 2003), 1st edition, January 1995. ISBN 0137903952.

- John A. Sauter, Robert Matthews, H. Van Dyke Parunak, and Sven Brueckner. Evolving adaptive pheromone path planning mechanisms. In *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 434–440, New York, NY, USA, 2002. ACM. ISBN 1-58113-480-0. doi : <http://doi.acm.org/10.1145/544741.544841>.
- A. Scheuer. *Planification de chemins à courbure continue pour robot mobile non-holonyme*. PhD thesis, Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble, 1998.
- Alexis Scheuer, Olivier Simonin, and François Charpillet. Safe Longitudinal Platoons of Vehicles without Communication. Research Report RR-6741, INRIA, 2008. URL <http://hal.inria.fr/inria-00342719/en/>.
- Sridhar Seshagiri and Hassan K. Khalil. Longitudinal adaptive control of a platoon of vehicles. *Proceedings of the American Control Conference*, 5 :3681 – 3685, 1999. ISSN 0743-1619. URL <http://dx.doi.org/10.1109/ACC.1999.782453>. Automated vehicles ; Vehicle dynamics ;
- C.E. Shannon. *The mathematical theory of communication*. University of Illinois Press, 1949.
- Shahab Sheikholeslam and A. Desoer, Charles. System level study of the longitudinal control of a platoon of vehicles. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 114(2) :286 – 292, 1992. ISSN 0022-0434. Longitudinal Control Law ; Platoon of Vehicles ;
- Shahab Sheikholeslam and Charles A. Desoer. Longitudinal control of a platoon of vehicles. *Proceedings of the American Control Conference*, pages 291 – 296, 1990. ISSN 0743-1619. Longitudinal Control ; Vehicle Platoons ;
- Shahab Sheikholeslam and Charles A. Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information : A system level study. *IEEE Transactions on Vehicular Technology*, 42(4) :546 – 554, 1993. ISSN 0018-9545. URL <http://dx.doi.org/10.1109/25.260756>.
- Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1) :51–92, March 1993. ISSN 00043702. doi : 10.1016/0004-3702(93)90034-9. URL [http://dx.doi.org/10.1016/0004-3702\(93\)90034-9](http://dx.doi.org/10.1016/0004-3702(93)90034-9).
- Reid Simmons. The curvature-velocity method for local obstacle avoidance. In *In Proc. of the IEEE International Conference on Robotics and Automation*, pages 3375–3382, 1996.
- Olivier Simonin and Franck Gechter. An environment-based methodology to design reactive multi-agent systems for problem solving. In *(Lecture Note in Artificial Intelligence) LNAI 3830/2006 : Environment for Multiagent Systems II (revised and selected papers of E4MAS 2005) D. Weyns, H. Van Dyke Parunak, and F. Michel (Eds)*, pages 32–49. Springer-Verlag Berlin Heidelberg, 2006.
- Graeme Smith. *The Object-Z specification language*. Kluwer Academic Publishers, Norwell, MA, USA, 2000. ISBN 0-7923-8684-1.
- Graeme Smith and Luke Wildman. *Model Checking Z Specifications Using SAL*. Springer Berlin, Heidelberg, 2005.
- R. G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. *Morgan Kaufmann*, pages 357–366, 1988.
- J. M. Spivey. *Understanding Z : a specification language and its formal semantics*. Cambridge University Press, New York, NY, USA, 1988. ISBN 0-521-33429-2.
- D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *Automatic Control, IEEE Transactions on*, 41(3) :349–357, Mar 1996. ISSN 0018-9286. doi : 10.1109/9.486636.

- Han-Shue Tan, Juergen Guldner, Chieh Chen, Satyajit Patwardhan, and Benedicte Bougler. Lane changing with look-down reference systems on automated highways. *Control Engineering Practice*, 8(9) : 1033 – 1043, 2000. ISSN 0967-0661. URL [http://dx.doi.org/10.1016/S0967-0661\(00\)00026-5](http://dx.doi.org/10.1016/S0967-0661(00)00026-5). Lane changing ;Look down reference systems ;Automated vehicles ;.
- H. Van Dyke Parunak, Sven Brueckner, and John Sauter. Digital pheromone mechanisms for coordination of unmanned vehicles. In *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 449–450, New York, NY, USA, 2002. ACM. ISBN 1-58113-480-0. doi : <http://doi.acm.org/10.1145/544741.544843>.
- S.C. Warnick and A.A. Rodriguez. A systematic antiwindup strategy and the longitudinal control of a platoon of vehicles with control saturations. *IEEE Trans. Veh. Technol. (USA)*, 49(3) :1006 – 16, 2000. ISSN 0018-9545. URL <http://dx.doi.org/10.1109/25.845117>. antiwindup strategy ;longitudinal control ;vehicle platooning ;control saturations ;nonidentical vehicles ;nominal controller ;nonlinear controller ;saturating nonlinearities ;stability preservation ;error governor ;IVHS problem ;proportional control ;nonlinear feedback designs ;15-car platoon simulation ;nonlinear vehicle models ;.
- D. Weyns, V. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems, state of the art and research challenges. In *Post-proceedings of the first International Workshop on Environments for Multiagent Systems, LNAI vol 3374*. Springer Verlag, 2005.
- Kirsten Winter and Graeme Smith. Compositional verification for object-z. *Lecture notes in computer science ISSN 0302-9743*, pages 280–299, 2003.
- Michael Wooldridge and Nicholas R. Jennings. Intelligent agents : Theory and practice. *Knowledge Engineering Review*, 10 :115–152, 1995.
- Soo-Yeong Yi and Kil-To Chong. Impedance control for a vehicle platoon system. *Mechatronics (UK)*, 15 (5) :627 – 38, 2005. ISSN 0957-4158. URL <http://dx.doi.org/10.1016/j.mechatronics.2004.12.002>.
- Karim Zeghal. *Vers une theorie de la coordination d'actions, application a la navigation aerienn*e. PhD thesis, Université de Paris 06, Paris, FRANCE, 1994.
- Y. Zhang, Elias B. Kosmatopoulos, Petros A. Ioannou, and C.C. Chien. Autonomous intelligent cruise control using front and back information for tight vehicle following maneuvers. *IEEE Transactions on Vehicular Technology*, 48(1) :319 – 328, 1999. ISSN 0018-9545. URL <http://dx.doi.org/10.1109/25.740110>. Autonomous intelligent cruise control (AICC) ;Tight vehicle following maneuvers ;.





Abstracts



## Modèle multi-agents réactifs pour la navigation multi-véhicules : Spécification formelle et Vérification

Cette thèse propose des modèles multi-agents réactifs fondés sur un cadre formel pour la vérification de propriétés et les valider par la simulation et l'expérimentation en considérant la navigation multi-véhicules comme domaine d'application. La navigation multi-véhicules soulève plusieurs problématiques : perception de l'environnement, communication inter-véhicule, évitement d'obstacle, ... Dans ce contexte, nous avons abordé plus particulièrement les aspects suivants :

- Concernant la navigation multi-véhicules nous avons abordé deux problèmes spécifiques : la conduite en convoi ou platooning et la navigation autonome. En ce qui concerne la navigation autonome, nous avons mis l'accent sur l'évitement d'obstacles.
- Concernant l'approche proposée : nous adoptons les systèmes multi-agents réactifs, dont les interactions sont inspirées de la physique. Pour la conduite en convoi, nous proposons un modèle d'interaction basé sur la physique classique. En ce qui concerne l'évitement d'obstacles, nous adoptant un modèle inspiré de la physique statistique.
- Nous plaçons les systèmes multi-agents réactifs dans un cadre formel pour la vérification des propriétés, compte-tenu des contraintes de sécurité imposées par la classe d'applications cible. Pour faire face à la complexité des modèles, nous proposons une règle et une méthode de vérification compositionnelle. Cela nous a permis de vérifier la satisfaction d'une propriété de sûreté essentielle : la non-collision entre véhicules lors de la conduite en convoi.
- Nous abordons également la question de la validation du système multi-agents par la simulation et l'expérimentation : nous avons contribué au développement d'un simulateur de la conduite en convoi et la navigation autonome. Le simulateur prend en compte les caractéristiques physiques des véhicules et est couplé à un outil de visualisation 3D. Nous avons aussi expérimenté nos modèles sur des véhicules tels que le Cycab et le GEM Car.

**Mots-clés :** Systèmes multi-agents réactifs, navigation multi-véhicules, spécification formelle, vérification.

---

## Reactive multi-agent models for the multi-vehicle navigation problem : formal specification and verification

The work presented in this thesis is concerned with providing reactive multi-agent models for the multi-vehicle navigation problem. The approach bases on formal specification and allows verification of the specification models, validation by simulation and implementation. Multi-vehicle navigation addresses several issues : environment perception, inter-vehicle communications, obstacle avoidance,... In this thesis, we focus on the following aspects :

- Concerning multi-vehicle navigation, we address two specific problems : platooning and autonomous navigation. As for autonomous navigation, we focused on obstacle avoidance.
- We propose a reactive multi agent system approach, with physics-inspired interaction models. Concerning platooning, the interaction model bases on classical physics. Concerning obstacle avoidance, the physical model bases on statistical physics.
- Multi vehicle navigation requires compliance with safety properties. We put the emphasis on compositional verification aspects, as a method to verify the validity of those safety properties. The verification method is iterative and bases on a compositional verification rule that applies to distributed reactive systems.
- We also address the issue of the validation of multi-agent system using simulation and experimentation : we have contributed to the development of a simulator of driving in convoy and autonomous navigation. The simulator takes into account vehicle's physical characteristics and is coupled to a 3D visualization tool. We also tested our models on real vehicles such as Cycab and GEM Car.

**Keywords :** Reactive multi-agent systems, physics inspired interaction models, multi-vehicle navigation problem, formal specification, verification.