

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira de Béjaia



Faculté des Sciences Exactes  
Département d'Informatique

# MÉMOIRE DE MASTER RECHERCHE

En Informatique

Option : Intelligence Artificielle

## Thème

**Prévision et Évitement des collisions dans les systèmes de transports**

Réalisé par :

- BOUAZOUNI Mohamed Walid
- AZIB Yanis

Déposé le *Judi 05 Novembre* 2020 et évalué par le jury composé de :

Présidente	Dr ADEL Karima	Maître de conf. A	U. A/Mira Béjaia.
Examineur	Dr ATMANI Mouloud	Maître de conf. A	U. A/Mira Béjaia.
Encadreur	Dr BOULAHROUZ Djamila	Maître de conf. A	U. A/Mira Béjaia.

Béjaia, Novembre 2020.

## ✧ **Remerciements** ✧

Notre premier remerciement va droit à nos parents qui ont su nous donner courage, volonté et patience pour achever ce projet et tous les projets de la vie.

Nous tenons à remercier Mme BOULAHROUZ Djamilia, notre promotrice, pour son aide, ses conseils, ses encouragements et ses orientations qui nous ont guidés jusqu'à l'aboutissement de ce travail.

Nous remercions aussi les membres du jury d'avoir accepté de juger notre travail et de nous faire part de leurs remarques pertinentes.

Nous exprimons notre plus grande gratitude à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

# TABLE DES MATIÈRES

<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>iii</b>
<b>Liste des algorithmes</b>	<b>iv</b>
<b>Listes des tableaux</b>	<b>v</b>
<b>Notations et symboles</b>	<b>vi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Etat de l'art sur les véhicules intelligents et simulation</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Véhicules intelligents . . . . .	4
1.3 Comportement d'un véhicule intelligent . . . . .	5
1.3.1 Perception de l'environnement . . . . .	5
1.3.2 Prise de décision . . . . .	5
1.3.3 Exécution des actions . . . . .	6
1.4 Communication véhiculaire et STI . . . . .	6
1.5 Réseaux véhiculaires VANET . . . . .	7
1.5.1 La communication dans les réseaux VANET . . . . .	8
1.6 Modélisation et simulation du trafic routier . . . . .	9
1.6.1 Modélisation du trafic routier . . . . .	10
1.6.2 Simulation du trafic routier . . . . .	11
1.6.3 Présentation des simulateurs existants dans la littérature . . . . .	13
1.7 Conclusion . . . . .	13
<b>2 Positionnement des systèmes multi-agents dans les systèmes de transports</b>	<b>14</b>
2.1 Introduction . . . . .	14

2.2	Agents autonomes et systèmes multi-agents . . . . .	15
2.2.1	Définition du modèle agent . . . . .	15
2.2.2	Caractéristiques des agents . . . . .	16
2.2.3	De l'agent aux systèmes multi-agents . . . . .	16
2.2.4	Interactions dans un environnement multi-agents . . . . .	18
2.2.5	Plateformes de développement des systèmes multi-agents . . . . .	20
2.3	Application des SMA aux systèmes de transports . . . . .	20
2.4	Présentation des différents travaux réalisés existants dans la littérature . . . . .	21
2.4.1	Simulateur multi-agents de trafic coopératif . . . . .	21
2.4.2	Mécanisme d'adaptation de vitesse pour évitement de collision dans un réseau VANET . . . . .	23
2.4.3	Commande neuro-floue pour le suivi de trajectoire et l'évitement d'obstacles dans un véhicule autonome . . . . .	23
2.5	Conclusion . . . . .	25
<b>3</b>	<b>Approche de modélisation proposée et contribution</b>	<b>26</b>
3.1	Introduction . . . . .	26
3.2	Modélisation d'un véhicule . . . . .	26
3.3	Modèle agent et architecture . . . . .	28
3.3.1	Module de partage et communication d'informations . . . . .	28
3.3.2	Module de prévision et prédiction des collisions . . . . .	30
3.3.3	Journal des évènements . . . . .	33
3.3.4	Module de planification des actions à exécuter . . . . .	33
3.4	Évitement des collisions sur différents types de circulation routière . . . . .	35
3.4.1	Circulation à sens unique . . . . .	35
3.4.2	Circulation à double sens . . . . .	38
3.5	Conclusion . . . . .	39
<b>4</b>	<b>Simulation et résultats</b>	<b>40</b>
4.1	Introduction . . . . .	40
4.2	Déroulement de la simulation . . . . .	40
4.2.1	Présentation du simulateur VANET réalisé . . . . .	40
4.2.2	Étapes de simulation . . . . .	44
4.3	Discussion des résultats de simulation . . . . .	47
4.4	Conclusion . . . . .	49
	<b>Conclusion générale et perspectives</b>	<b>50</b>
	<b>Bibliographie</b>	<b>51</b>

## TABLE DES FIGURES

1.1	Comportement d'un véhicule intelligent . . . . .	5
1.2	Système de transport intelligent [8] . . . . .	7
1.3	Communication véhicule - véhicule . . . . .	8
1.4	Communication véhicule - infrastructure . . . . .	8
1.5	Communication hybride . . . . .	9
2.1	Agent en interaction avec son environnement . . . . .	15
2.2	Environnement multi-agents . . . . .	17
2.3	Exemple de modélisation d'un protocole d'interaction entre deux agents par le réseaux de Petri[39] . . . . .	19
3.1	Positionnement des roues . . . . .	26
3.2	Présentation des paramètres cinématique d'un véhicule . . . . .	27
3.3	Modules composant un agent . . . . .	28
3.4	Transsmission des messages informatifs . . . . .	29
4.1	Saisies des paramètres cinématiques . . . . .	41
4.2	Monde du simulateur et véhicules . . . . .	41
4.3	Environnement de simulation . . . . .	41
4.4	Composition de la plateforme JADE [38] . . . . .	43
4.5	AlphaMap . . . . .	44
4.6	HeightMap . . . . .	44
4.7	Interface représentant la communication entre deux véhicules par envois de messages . . . . .	45
4.8	Présentation du véhicule <i>B</i> en marron et véhicule <i>A</i> en bleu . . . . .	46
4.9	Cas d'arrêt . . . . .	49

## LISTE DES ALGORITHMES

1	Processus d'estimation des paramètres cinématiques d'un véhicule . . . . .	30
2	Processus de prédiction de collision . . . . .	32
3	Processus de planification des évènements . . . . .	36
4	Processus de dépassement à sens unique . . . . .	37
5	Processus de dépassement à double sens . . . . .	39

# LISTE DES TABLEAUX

4.1 Définition des paramètres cinématiques des véhicules . . . . .	46
--	----

## NOTATIONS ET SYMBOLES

<b>A</b>	ADSL	Asymmetric Digital Subscriber Line.
	ACL	Agent Communication Language.
<b>B</b>	BDI	Belief-Desire-Intention.
<b>C</b>	CORMAS	COmmon Resources Multi-Agent System.
<b>E</b>	ELU	Exponential Linear Unit.
<b>F</b>	FIPA	Foundation for Intelligent Physical Agent.
<b>G</b>	GPS	Global Positioning System.
<b>I</b>	IA	Intelligence artificielle.
	IAD	Intelligence Artificielle Distribuée.
	IDE	Integrated Development Environment.
<b>J</b>	JADE	Java Agent DEvelopment.
	jME	jMonkey Engine.
	JOGL	Java OpenGL.
<b>L</b>	LWJGL	Lightweight Java Game Library. <sup>2</sup>
<b>M</b>	MANET	Mobile Ad Hoc Network.
	MOVE	Mobility model generator for Vehicular Network.
<b>N</b>	NS2	Network Simulator. <sup>2</sup>
<b>O</b>	OMS	Organisation Mondiale de la Santé.
	OBU	On-Board Unit.
	OTcl	Object Tool command language.
<b>R</b>	RSU	Road Side Unit.
<b>S</b>	STI	Systèmes de Transport Intelligents.
	SUMO	Simulation for Urban Mobility.
	SMA	Système Multi-Agents.
<b>V</b>	VANET	Vehicular Ad-Hoc Network.
	V2V	Véhicule-à-Véhicule.
	V2I	Véhicules-à-Infrastructure.
	VEINES	VEhicles In NEtwork Simulator.



# Introduction générale

Avec l'apparition des réseaux sans fil d'un côté et l'avancement des technologies de la communication et de l'information de l'autre, il y'a eu l'émergence des **STI** « *Systèmes de Transport Intelligents* ». Il s'agit d'applications qui mettent en interactions plusieurs entités du réseaux routier tels que les véhicules, les infrastructures, les panneaux de signalisation dans le but de partager des informations pertinentes entre ces entités, et d'avoir un environnement complètement communiquant et coopératif et pratiquement observable.

Plusieurs technologies de communications peuvent être utilisées pour faire communiquer les entités, tels que les réseaux cellulaires et les réseaux satellitaires qui sont proposés pour assurer les communications à longue portée et permettre des services tels que la surveillance de l'état du véhicule, la localisation des véhicules, la diffusion des messages d'urgence et l'information sur le trafic, et aussi l'apparition d'une nouvelle technologie de communication, supportant les communications à moyenne et à courte portée qui peuvent être de **V2I** « *Véhicules-à-Infrastructure* » ou en mode ad hoc de **V2V** « *véhicule-à-véhicule* ». En effet, en conjuguant les données issues de la cartographie (**GPS** « *Global Positioning System* ») à celles détectées par les capteurs du véhicule (caméras, radars etc.) il devient possible de faire de cette interactivité une conduite coopérative et former un réseau dynamique qu'on appelle un réseaux **VANET** « *Vehicular Ad hoc Network* ».

Les problèmes routiers rencontrés par les automobilistes tiennent surtout à la communication inter-véhiculaire. En fait, la majorité des accidents peuvent être évités si l'un des conducteurs soit alerté avant la présence de la collision, mais le conducteur ne peut être attentif à tout et peut même passer à travers d'une alerte, de ce fait les véhicules devrait être autonomes et prendre la main au moment opportun pour éviter les accidents. Sécuriser nos routes est plus que primordiale, c'est pour cela que nous nous sommes intéressé à ce problème dans le but de trouver des solutions répondant à la problématique « *Évitement de collisions dans les systèmes de transport* ».

## Objectif du travail

Notre travail consiste en l'élaboration des solutions permettant l'évitement des collisions entre les entités mobiles dans les systèmes de transport sur différents types de circulation routière : sens unique, double sens.

Durant ce travail l'approche multi-agents est pleinement mise en avant que ce soit dans le traitement des informations, estimations, la prise de décision, organisation du trafic et protocole de communication. L'autonomie abordée dans ce travail est de niveau 3, c'est-à-dire que le conducteur a la totale liberté d'actions sur le véhicule, mais l'agent intervient au moment d'un éventuel danger mettant en péril le véhicule et provoquer une collision, tel que le non-respect de la distance de sécurité, la gestion des imprévus, etc.

La simulation dans les systèmes de transports passe par la définition d'un modèle de mobilité et trafic, de ce fait nous allons mettre l'accent sur la modélisation microscopique, et la représentation de la cinématique du véhicule par rapport à sa position, sa vitesse et sa direction.

Il faut noter que l'aspect de la simulation considérée dans ce travail est l'aspect comportemental, c'est à dire la simulation du comportement des véhicules sans prendre en compte la simulation réseau, donc nous considérons que les véhicules sont déjà connectés entre eux sur une partie de la route ou sur une distance que nous spécifions.

## Organisation du mémoire

Ce mémoire s'articule autour d'une introduction générale, de quatre chapitres, d'une conclusion générale et d'une bibliographie. Dans le premier chapitre, intitulé « *Etat de l'art* », nous mettons le point sur le concept des véhicules intelligents, sur l'apparition et les objectifs des STI, et enfin la modélisation et la simulation du trafic routier et les différents simulateurs proposés dans la littérature. Dans le deuxième chapitre, intitulé « *Positionnement des Systèmes Multi-Agents dans les Systèmes de Transports* », nous présentons le concept agent et système multi-agent et quelques plateformes de développement de ces derniers, par la suite nous allons aborder l'impact des SMA dans la résolution des problèmes liés au transport. Dans le troisième chapitre, intitulé « *Conception et modélisation* », nous nous penchons sur la spécification des agents du système, et le renforcement de nos idées par des algorithmes avec lesquels nous avons établi une solution à la problématique. Dans le quatrième chapitre, intitulé « *Simulation et résultats* », nous présentons la mise en application de la conception présentée dans le chapitre précédent en développant notre simulateur à temps réel ainsi que les différents outils de sa réalisation. Pour enfin conclure avec les différents résultats liés à l'évitement des collisions dans les systèmes de transport.

# Etat de l'art sur les véhicules intelligents et simulation

## 1.1 Introduction

Au cours de ces dernières années, une très grande importance a été donnée aux systèmes de transports, dans le but d'améliorer la sécurité routière, préserver l'environnement, réduire les déplacements, etc. Les efforts consentis pour résoudre ces problèmes ont ouvert de nouveaux domaines de recherche et d'application, à savoir les **STI** « *Systèmes de Transports Intelligents* », les réseaux **VANET** « *Vehicular Ad-Hoc Network* » et en fin la conduite automatisée des véhicules dans laquelle des nouvelles techniques sont exploités pour automatiser entièrement ou partiellement les tâches de la conduite.

Parmi les objectifs des STI, l'interconnexion des différentes entités du réseau routier pour s'échanger des informations pertinentes tels que la position, l'état de route, etc. et permettre ainsi à ces entités de mieux observer leurs environnements et être à l'écoute de l'état des routes.

Plusieurs solutions sont proposées dans la littérature pour résoudre les problèmes liés aux réseaux routiers dans le cadre des STI (Congestion, accidents, etc.), sauf que la mise en application de ces solutions pour les tests revient très chère et prend beaucoup de temps, pour cela nous employons d'autres méthodes tel que la simulation pour contourner ces problèmes.

Durant ce chapitre nous allons nous étaler sur le concept des véhicules intelligents, de l'apparition et des objectifs des STI et en fin sur la modélisation et la simulation du trafic routier et les différents simulateurs proposés dans la littérature.

## 1.2 Véhicules intelligents

Un véhicule intelligent est un véhicule possédant des fonctions autonomes, c'est-à-dire, le pouvoir de perception de l'environnement, et la prise de décision pour assister le conducteur en temps réel. Pour cela le véhicule doit posséder une panoplie d'équipements, des capteurs pour la perception, des actionneurs pour interagir avec l'environnement et un système de communication pour échanger des informations avec d'autres véhicules en vue d'une conduite collaborative, ou avec des infrastructures routières tels que les panneaux de signalisations, les informations sur le trafic, etc.

Les véhicules intelligents n'embarquent pas forcément le même nombre ou la même qualité d'équipements, ils peuvent être aussi bien des véhicules totalement autonomes, ou des véhicules semi autonomes. Ainsi nous pouvons distinguer plusieurs niveaux d'autonomies classés comme suit [7] :

- **Niveau 0. Pas d'autonomie ni d'assistance** : Niveau le plus simple où le conducteur est le seul responsable à bord. Toute la conduite est réalisée par le conducteur et elle n'est assistée en aucun cas par le véhicule.
- **Niveau 1. Assistance à la conduite** : Toujours pas de fonctions autonomes, en revanche quelques fonctions automatisées sont disponibles pour aider le conducteur. Nous pouvons citer le régulateur de vitesse, le freinage automatique d'urgence, etc.
- **Niveau 2. Automatisation partielle** : Toujours un peu plus d'assistance à la conduite, le conducteur peut superviser certaines tâches plus précises mais doit évidemment garder un oeil sur son environnement pour pouvoir reprendre rapidement le contrôle si un problème arrive. Nous pouvons citer le contrôle de maintien du véhicule en voie, assistance de parking, etc.
- **Niveau 3. Conduite autonome limitée** : À partir de ce niveau, les véhicules sont en capacité de lire et d'analyser leurs environnements de conduite et donc d'agir en conséquence. Ce niveau indique que le conducteur doit être capable de reprendre le contrôle du véhicule à tout moment, mais que le véhicule est parfaitement autonome dans certaines conditions de conduite.
- **Niveau 4. Autonomie élevée** : Aucune assistance de la part du conducteur n'est requise. Cependant l'autonomie totale se limite à des conditions prédéfinies : une zone géographique spécifique, par exemple une autoroute ou un parking avec lequel le véhicule est compatible, des conditions météorologiques favorables, etc.
- **Niveau 5. Autonomie élevée complète** : Ici, il n'y a plus d'intervention humaine ni de conditions de route, le véhicule est autonome à 100%. Il ne se repose en aucun cas sur l'humain et cumule alors toutes les tâches.

## 1.3 Comportement d'un véhicule intelligent

Le comportement d'un véhicule intelligent est généralement basé sur trois étapes permettant de passer de l'observation et de l'analyse de l'environnement du véhicule à la commande envoyée aux actionneurs qui assurent le contrôle du véhicule. Ces trois étapes, répétées cycliquement (Voir Figure 1.1), sont généralement les suivantes :

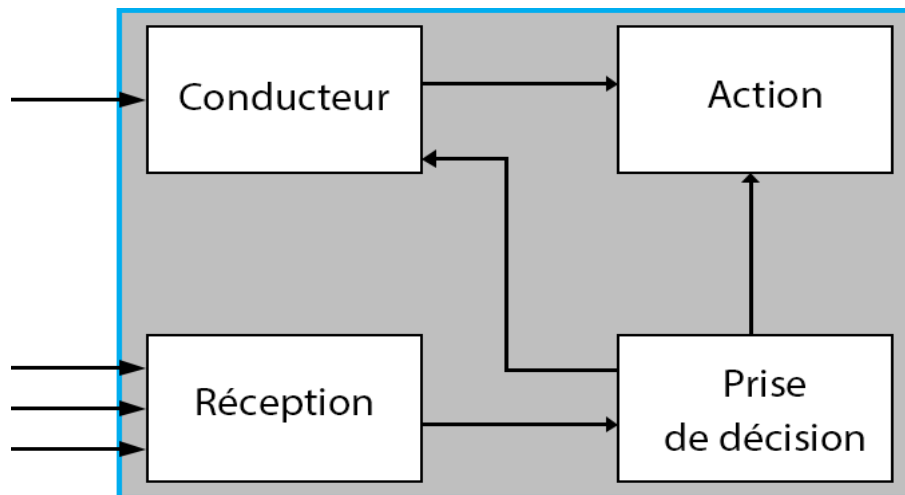


FIGURE 1.1 – Comportement d'un véhicule intelligent

### 1.3.1 Perception de l'environnement

La perception consiste à récupérer les informations relatives à l'environnement du véhicule en utilisant des capteurs appropriés. Comme exemple, nous pouvons citer :

- **Camera** : La caméra est un capteur de prise de vue destiné à capter des images successives de l'environnement et permettre de les restituer avec une impression de mouvement [3].
- **Radar** : Système qui utilise des ondes électromagnétiques pour détecter la présence d'un objet et de déterminer sa position et sa vitesse.
- **GPS** : Le capteur **GPS** « *The Global Position System* » permet la localisation des objets grâce au système satellite. Il offre des capacités de navigation et de notification en temps réel [4].

### 1.3.2 Prise de décision

Grace aux informations fournis par les capteurs, l'unité de traitement évalue ces informations et prend des décisions appropriées pour préserver la sécurité des usagers de la route et le confort des passagers.

Notons qu'en vue d'aider à la prise de décision, deux tâches sont indispensables : la localisation du véhicule et la perception de son environnement [35].

- **Localisation du véhicule**

Généralement la position du véhicule est mesurée par GPS. La localisation du véhicule permet la planification des chemins à emprunter en temps réel ou à des fins mettant le confort des passagers en priorité.

- **Perception de l'environnement du véhicule**

Nous distinguons deux types de perception d'environnement :

- **Perception de l'environnement local** : Chaque véhicule a besoin de percevoir les informations nécessaires concernant ses alentours grâce à ses capteurs ou grâce aux informations reçues par d'autres entités routières. La perception de l'environnement par le véhicule permet d'anticiper ses actions pour éviter toute sorte d'imprevu [35].
- **Perception de l'état du véhicule** : La prise de décision passe par la connaissance de l'état dynamique du véhicule tels que la vitesse, l'accélération, la direction, etc.

### 1.3.3 Exécution des actions

C'est l'application des décisions prises par l'unité de traitement au niveau des actionneurs physiques du véhicule.

## 1.4 Communication véhiculaire et STI

De nos jours, nos véhicules sont équipés de plus en plus de nouvelles fonctionnalités, dans le but d'améliorer le confort des usagers de la route et leurs sécurités. En vu du nombre de véhicule en circulation et des accidents engendrés, la sécurité routière est devenue un thème d'actualité, où ingénieurs, chercheurs, constructeurs proposent leurs solutions pour essayer de minimiser les cas d'accidents routiers, en employant des moyens technologiques avancés, parmi les plus importants, l'émergence des réseaux sans fils et leurs adoptions dans les infrastructures routières.

Par ailleurs la technologie de communication réseau ne cesse d'évoluer, des réseaux filaires aux réseaux sans fil, bien de technologies d'accès réseau et partage ont émergées (**ADSL** « *Asymmetric Digital Subscriber Line* », Wifi, etc.). Elles sont devenues indispensables pour notre vie quotidienne, pratiquement tout le monde s'en sert pour partager de l'information et être à l'écoute de l'actualité.

L'adoption des réseaux sans fil dans les infrastructures routières ont fait apparaître ce qu'on appelle les STI. En dotant les véhicules mais également les infrastructures routières et urbaines de moyens de communication, nos routes deviennent plus sûres.

L'information est le coeur des STI, les composantes du réseau routier doivent être en perpétuelle communication (comme présenté dans la Figure 1.2). Les données provenant des STI peuvent fournir de l'information en temps réel sur l'état actuel de l'environnement ou pour la planification des déplacements.



FIGURE 1.2 – Système de transport intelligent [8]

Parmi les services pertinents des STI, mentionnons les suivants [21] :

- Système de détection d'incidents et systèmes d'alerte de collisions.
- Contrôle du respect des limites de vitesse.
- Surveillance des conditions météorologiques et de l'état des routes et diffusion des informations relatives à ces conditions.

La possibilité de communication inter-véhiculaires ou infrastructure-véhicule fait partie des concepts importants des STI, cette communication engendre la formation des réseaux véhiculaires VANET.

## 1.5 Réseaux véhiculaires VANET

Les réseaux VANET sont en effet une classe émergente des réseaux sans fil, ils peuvent être considérés comme une particularité des réseaux **MANET** « *Mobile Ad Hoc Network* » [29] où les noeuds mobiles sont des véhicules intelligents. Comme tout autre réseau Ad hoc, les véhicules peuvent communiquer entre eux ou avec des stations de base placées tout au long des routes (Demander des informations, accéder à internet, etc.). Parmi ses objectifs :

- **Une conduite collaborative sécurisée** : Transmission des messages d'urgence (freinage, collision, danger quelconque, etc.). [21]
- **Une conduite plus conviviale et confortable pour le conducteur et ses passagers** : Partage de contenu, publicité, tourisme, internet. [21]
- **Information sur l'environnement en temps réel** : Etat des routes, planification du chemin à emprunter, embouteillages.
- **Une circulation plus organisée** : Le partage d'informations et synchronisation entre les feux de signalisation offrant une conduite plus fluide et dégager.

### 1.5.1 La communication dans les réseaux VANET

Les réseaux VANET sont basés sur la communication et l'échange d'information, nous pouvons distinguer 3 modes de communication dans les VANETs :

- **Communications véhicule à véhicule (V2V)** : Dans ce mode, chaque véhicule est équipé pour communiquer directement avec d'autres véhicules de sorte qu'aucune infrastructure et installation n'est nécessaire, ce qui rend la communication moins coûteuse et plus flexible (Voir Figure 1.3).



FIGURE 1.3 – Communication véhicule - véhicule

- **Communications véhicule à infrastructure (V2I)** : Celle-ci est une architecture centralisée basée sur des stations de base dans leurs communication. Les véhicules garantissent des communications avec l'infrastructure en utilisant des points d'infrastructure (Voir Figure 1.4). Ces points d'accès sont également connus sous le nom **RSU** « *Road Side Units* », situés dans certaines sections critiques de la route, tels que les feux de circulation, intersections, ou les stops, afin d'améliorer l'expérience de conduite et la rendre plus sûre. Cette architecture peut être utilisée dans les scénarios comme l'accès à Internet, état de la circulation, contrôle de vitesse, etc. [51][26]

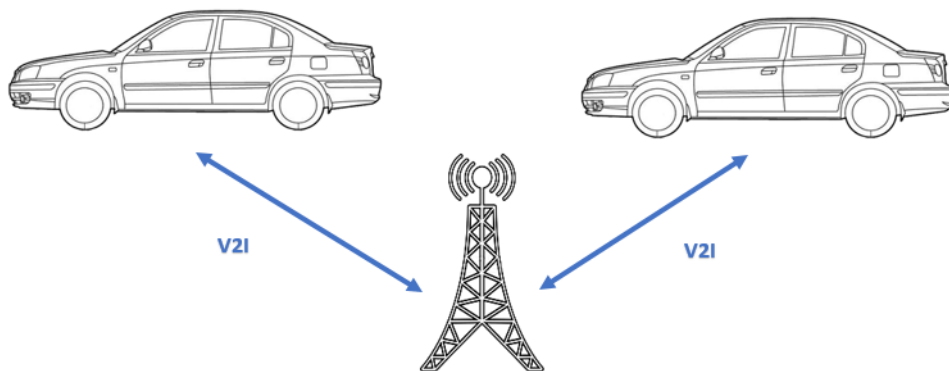


FIGURE 1.4 – Communication véhicule - infrastructure

- **Communications hybrides (V2V-V2I)** : Ce mode de communication est produit lors de la combinaison du mode de communication de véhicule à véhicule avec le mode de communication de véhicule avec utilisation d'infrastructures (Voir Figure 1.5). L'utilisation des deux méthodes de communication simultanée est le point fort des VANETs. Ce mode hybride permet de diffuser efficacement les informations des applications sur courtes et longues portées en utilisant la topologie dynamique des VANETs.



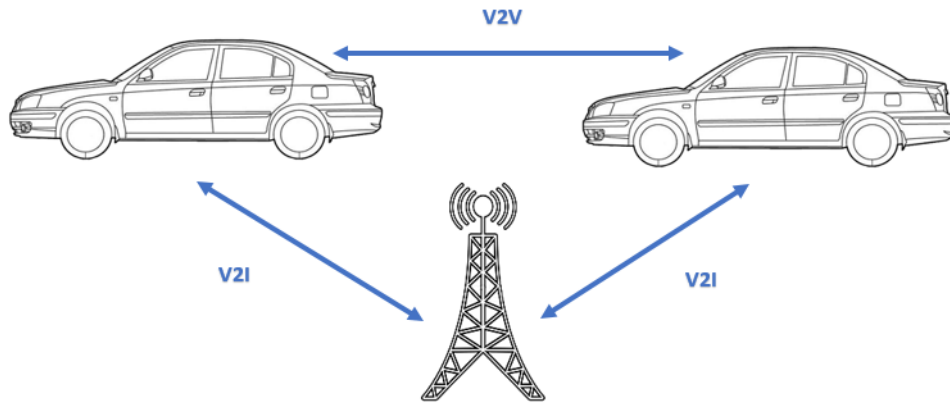


FIGURE 1.5 – Communication hybride

Généralement les types d'informations communiquées dans un réseau VANET s'orientent sur la diffusion des messages tels que [32] :

- **Messages de contrôle** : Un message de contrôle est généré à intervalle régulier. Conventionnellement chaque véhicule émet un message de contrôle toutes les 100 ms. Ce message appelé aussi « *beacon* » contient la position, la vitesse, la direction et l'itinéraire du véhicule émetteur. Grâce aux messages de contrôle chaque véhicule se crée une vue locale de son voisinage.
- **Messages d'alerte** : Un message d'alerte est généré lorsqu'un événement est détecté. Cela peut être la détection d'un accident, d'un obstacle ou la réception d'un autre message d'alerte. Le message d'alerte doit être émis à intervalle régulier afin d'assurer la pérennité de l'alerte.
- **Autres messages** : Il s'agit de tous les autres messages qui ne sont pas des messages d'alerte ou de contrôle. Ces messages ne sont généralement pas répétés à intervalle régulier. En effet, cela peut être par exemple un message de transaction financière ou l'envoi de courrier électronique. Tous les messages reçus seront stockés dans un cache des messages récemment reçus . Chaque message se verra associer une durée de vie dans le cache.

## 1.6 Modélisation et simulation du trafic routier

Toute nouvelle solution passe par un processus d'évaluation et de validation avant son éventuel déploiement. Le moyen idéal de réaliser cette tâche est de pouvoir effectuer des tests dans des environnements réels. Cependant, de par la nature distribuée, l'environnement et la topologie complexe des réseaux véhiculaires, il est extrêmement délicat d'évaluer la solution dans un environnement pareil. La modélisation et la simulation est le moyen le plus utilisé pour contourner ce problème.

## 1.6.1 Modélisation du trafic routier

La modélisation du trafic routier est un domaine qui traite la simplification de la circulation automobile [25]. Le but est de pouvoir modéliser le trafic routier dans diverses situations et qui se rapproche de la réalité.

Des études ont mené à l'élaboration des modèles permettant la simplification et la compréhension du phénomène réel. Étant donné leurs multitudes, il semble utile de les classer par la façon avec laquelle ils reproduisent l'écoulement des véhicules et le niveau du détail avec lequel ils représentent les systèmes du trafic. Une telle classification a été proposée par *Hoogendoorn et Bovy* qui ont trouvé quatre types de modèles [48] :

- **Modèles Microscopiques** : Décrivent le comportement des entités des systèmes (les véhicules et les conducteurs) aussi bien que leurs interactions à un niveau élevé de détail (individuellement).
- **Modèles SousMicroscopiques** : Représentent en détail le comportement des commandes de véhicule (par exemple changement de la vitesse) en correspondance avec les conditions d'environnement (les modèles SousMicroscopiques sont assimilés aux modèles Microscopiques).
- **Modèles Macroscopiques** : Décrivent le trafic à un niveau élevé d'agrégation comme l'écoulement d'un fluide sans distinguer ses composantes, le trafic est représenté d'une façon globale en utilisant des caractéristiques comme le débit, la densité et la vitesse. Le niveau général de détail étant réduit.
- **Modèles Mésoscopiques** : Indiquent le comportement des individus en termes probabilistes. A cet effet, le trafic est représenté par (de petits) groupes d'entités du trafic, dont les activités et les interactions sont décrites à un niveau moyen de détail.

### 1.6.1.1 Modèles Microscopiques

Dans ces modèles, les véhicules sont considérés uniquement de manière individuelle, le véhicule est vu comme une entité indépendante de l'environnement [36].

Classiquement, les modèles microscopiques ont pour principale vocation de pouvoir décrire les comportements individuels des usagers pour deux situations de conduite [25] :

- Le comportement en poursuite, correspondant au comportement d'un conducteur en réponse aux actions du véhicule qui le précède. Nous distinguons deux types de conduite :
  - **Cas de conduite libre** : Le véhicule roule avec la vitesse désirée si le véhicule qui le précède est suffisamment loin.
  - **Cas de conduite en poursuite** : Si un véhicule est précédé par un autre qui est suffisamment proche, le véhicule doit adapter son comportement par rapport au véhicule qui le précède.

- Le comportement de changement de voie au sens large. Cela comprend les manoeuvres de dépassement en section courante tout comme les manoeuvres d'insertion.

Comme exemples citons les modèles microscopiques, les plus souvent rencontrés dans la littérature : Le Modèle **Follow the leader** [44][22], le Modèle à **Distance de sécurité** [37].

### 1.6.1.2 Modèles Macroscopiques

Les modèles macroscopiques décrivent le trafic à un niveau élevé d'agrégation[20]. L'objectif de ces modèles est de pouvoir caractériser le comportement global du trafic, à une échelle d'étude relativement importante. Ils sont donc tout particulièrement utilisés dans le cadre de la modélisation des grands réseaux [25]. Leurs applications courantes couvrent la simulation du trafic en vue de la planification et de la conception des infrastructures, mais aussi la gestion dynamique du trafic et l'évaluation de ces mesures de gestion à posteriori.

Trois modèles sont les plus reconnus aujourd'hui dans la littérature pour représenter la dynamique du trafic au niveau macroscopique : le modèle d'équilibre **LWR** [41] [43], le modèle de non équilibre **ARZ** [15] [27] et le modèle multi-classes d'origine-destination **MOD** [33][28].

### 1.6.1.3 Modèles Mésoscopiques

Les modèles mésoscopiques utilisent un niveau de détail plus fin que les modèles macroscopiques [25]. L'idée sous-jacente à cette approche est qu'effectuer des calculs sur des quantités agrégées à un niveau macroscopique ne fournit pas une description suffisamment détaillée des évolutions du trafic, mais suivre et gérer les véhicules de manière individuelle est trop coûteux en temps de calcul.

Pour pallier à ce problème, ces modèles proposent un compromis en composant approche macroscopique et approche microscopique. Ils représentent le trafic par des « *paquets* » de véhicules regroupés en fonction de certaines caractéristiques [45].

Comme exemples citons les modèles mésoscopiques, les plus souvent rencontrés dans la littérature : Le modèle **CONTRAM**, le modèle **DYNAMIT**, le modèle **FASTLANE**.

## 1.6.2 Simulation du trafic routier

La simulation est largement reconnue comme une des techniques les plus efficaces pour l'analyse et la conception des systèmes mise à la disposition des concepteurs et des chercheurs. Elle peut être appliquée dans divers domaines, tels que l'analyse des systèmes de services, les systèmes de production, les systèmes naturels, les systèmes informatiques, etc. Cette technique connaît actuellement un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, que par le besoin de simuler par des réalisations de plus en plus complexes.

Le but est de reproduire la réalité physique de l'environnement véhiculaire dans diverses situations et

ainsi réduire au maximum l'écart entre les résultats de la simulation et la réalité après application de la solution proposée.

Deux aspects peuvent être distingués dans la simulation des véhicules intelligents :

- L'aspect protocolaire qui consiste à modéliser les protocoles et mécanismes utilisés par les services et applications réseau, permettant de reproduire les mécanismes utilisés dans les différentes couches qui interviennent dans les communications réseau tels que [16] :
  - L'établissement des connexions au niveau transport.
  - L'adressage au niveau réseau ou encore la politique d'accès au canal.
  - Ordonnancement au niveau liaison de données.
- L'aspect comportement qui est axé sur la modélisation du comportement des entités du réseau routier. Les déplacements des véhicules sont liés aux structures des routes (intersections, panneaux de signalisation, etc.) et aux stations de base routières (infrastructures) que ce soit dans les autoroutes ou au sein d'une zone urbaine.

#### 1.6.2.1 Etapes de simulation

Le processus de simulation du trafic routier exige quatre étapes qui doivent être exécutées. Dans ce qui suit, nous allons exposer ces différentes étapes.

- **Génération de la carte**

Dans la première étape, nous avons besoin d'une zone de simulation appropriée au scénario désiré pour visualiser le mouvement des entités mobiles [21].

Cette zone de simulation peut être une carte que nous pouvons créer ou tout simplement l'importer à partir des bases données en ligne tel que **Google Maps** ou l'extraire du site **OpenStreetMap** [2].

- **Simulation de modèle de mobilité et trafic**

Un modèle de mobilité reflète le comportement et les déplacements des entités mobiles dans un réseau routier, où le but est de représenter au mieux les conditions de cette mobilité dans un contexte particulier du monde réel [21].

- **Configuration du moyen de communication**

Dans les STI, les entités du trafic routier sont capables de communiquer, il est important de prendre en considération dans la simulation cet aspect.

La communication passe par l'application d'un protocole de communication permettant la gestion de l'interaction entre les différentes entités du système.

- **Simulation et analyse**

La simulation permet de tester et visualiser à moindre coût les résultats sous forme de graphes faciles à analyser et à interpréter.

### 1.6.3 Présentation des simulateurs existants dans la littérature

Il existe plusieurs simulateurs dans la littérature. Dans ce qui suit, nous allons présenter les plus significatifs.

- **Le simulateur de trafic SUMO** : « *Simulation for Urban Mobility* » est un simulateur microscopique du trafic routier. Il prend en charge le trafic au sol multimodal et intermodal. SUMO modélise les véhicules individuels et leurs interactions à l'aide de modèles de comportement de suivi de voiture, de changement de voie et d'intersection. Il utilise également des modèles de piétons pour simuler le mouvement des personnes et leurs interactions avec les véhicules [6].
- **Le simulateur MOVE** : « *Mobility model generator for Vehicular Network* » est un outil qui permet la génération rapide des modèles de mobilité réels pour la simulation des VANETs et se basant sur SUMO. La sortie du simulateur MOVE produit un modèle réel sous forme d'un fichier trace qui peut être directement utilisé dans plusieurs simulateurs de réseaux. MOVE comprend également un outil de visualisation qui permet aux utilisateurs de visualiser les traces de mobilité générées [24].
- **NS2** : « *Network Simulator* » est un simulateur à événements discrets orienté objet, écrit en **C++** avec une interface qui utilise le langage **OTcl** « *Object Tool command langage* ». A travers ces deux langages il est possible de modéliser tout type de réseau et de décrire les conditions de simulation : La topologie réseau, le type du trafic qui circule, les protocoles utilisés, les communications qui ont eu lieu, etc.  
Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Pour reprendre la terminologie objet, il sert à définir les classes. Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc .
- **VEINES** : « *VEhicles In NEtwork Simulator* » est un simulateur open-source de communication inter-Véhiculaire, composé d'un simulateur de réseau basé sur les événements et un modèle de micro-traffic routier.

## 1.7 Conclusion

Durant ce chapitre, nous avons tout d'abord présenté le concept des véhicules intelligents, de la façon dont ils perçoivent leurs environnements et traitent les informations émises par leurs capteurs, puis nous avons parlé des STI et de l'importance qu'ils donnent à la sécurité routière et aussi la possibilité de communication entre les véhicules et les infrastructures routières grâce aux réseaux VANET. En fin nous avons abordé la modélisation du trafic routier ainsi que la simulation qui représente un moyen efficace pour tous les chercheurs ou ingénieurs souhaitant mettre en application leurs idées à moindre coût.

# Positionnement des systèmes multi-agents dans les systèmes de transports

## 2.1 Introduction

Dans le chapitre précédent, nous avons découvert les STI, l'importance des réseaux véhiculaires et les interactions possibles conduisant aux partages d'informations entre les entités routières pour des faits divers.

Cependant les informations captées par ces entités doivent passer par un processus de prise de décision pour juger l'utilité de ces informations et extraire les données permettant de synchroniser leurs comportements pour ensuite appliquer les bonnes actions. Environnement distribué, voitures autonomes, prise de décision, etc. toutes ces notions nous orientent directement vers l'utilisation des **SMA** « *Système Multi-Agents* ».

Concept provenant de l'intelligence artificielle distribuée, les SMA mettent en interactions plusieurs entités autonomes appelées agents dans un environnement qu'ils partagent. Les SMA sont efficaces pour la modélisation et la simulation des systèmes nécessitant l'interaction de plusieurs entités et la résolution des problèmes complexes impliquant la coopération des entités du même environnement.

Ainsi dans ce chapitre nous allons définir le concept agent et les systèmes multi-agents, ensuite nous allons présenter quelques plateformes de développement agent, puis nous parlerons de l'importance des SMA dans la résolution des problèmes liés aux systèmes de transports et en fin nous présenterons quelques travaux trouvés dans la littérature appliquant cette approche pour la résolution des problèmes liés aux systèmes de transports.

## 2.2 Agents autonomes et systèmes multi-agents

Les SMA forment un paradigme prometteur pour la conception d'applications complexe. Une de leurs plus grande force vient dans la distribution de l'expertise sur un ensemble d'agents, capable de travailler dans un environnement commun et résoudre les éventuelles conflits [46].

Un environnement distribué permet d'offrir plusieurs avantages, nous citons la modularité, la rapidité de résolution des problèmes de par le parallélisme et la fiabilité des informations/résultats.

### 2.2.1 Définition du modèle agent

Actuellement il existe plusieurs définitions de ce qu'est un agent, nous pouvons citer celle donnée par *Rusell* " *Un agent est tout entité qui peut être considérée comme percevant son environnement grâce à des capteurs et qui agit sur cet environnement via des effecteurs* " [49] (voir Figure 2.1). Cette définition a ensuite été successivement complétée en introduisant les concepts de rationalité, d'autonomie et d'autres propriétés toutes aussi importantes.

Globalement un agent est une entité physique ou virtuelle possédant ses propres ressources, dotée des capacités de perception et d'interaction avec son environnement et qui peut éventuellement partager le même environnement avec d'autres agents et communiquer avec eux ou leur offrir des services. Un agent est animé par un ensemble d'objectifs dont son comportement tend à les satisfaire en prenant en compte les ressources et compétences dont il dispose, en fonction de sa perception, de ses représentations et des communications qu'il reçoit.

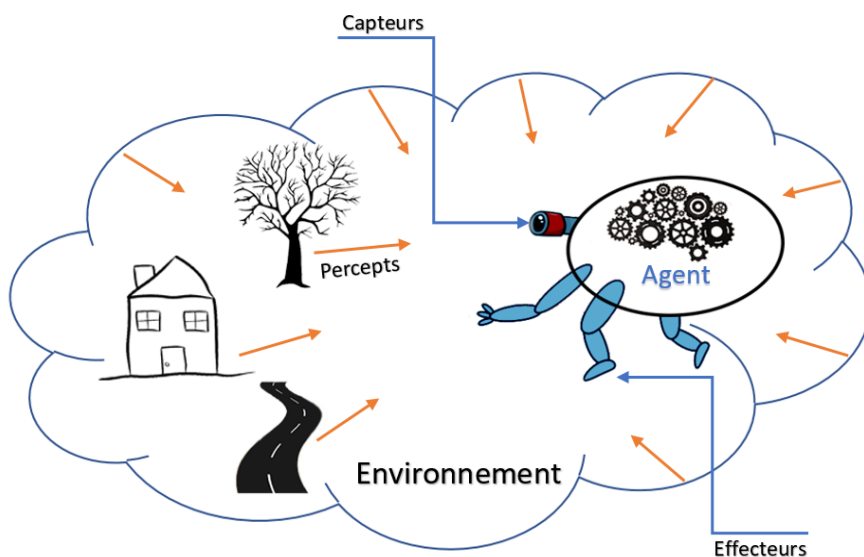


FIGURE 2.1 – Agent en interaction avec son environnement

## 2.2.2 Caractéristiques des agents

Les agents possèdent plusieurs caractéristiques qui varient en fonction de la complexité des tâches qu'ils vont exécuter et qui les distinguent d'autres entités, nous citons :

- **Sociabilité** : Les agents sont destinés à fonctionner dans un environnement distribué, avec lequel ils interagissent avec d'autres agents.
- **Réactivité** : Un agent est capable de percevoir son environnement et de répondre en temps réel aux changements.
- **Proactivité** : Un agent doit être capable de prendre des initiatives, se mettre dans des situations inconnues (nouvel état) pour évaluer et tirer les enseignements de cette nouvelle situation. C'est un comportement opportuniste orienté but.
- **Autonomie** : Un agent doit évoluer sans l'intervention d'un tiers qu'il soit un autre agent ou un humain, il doit contrôler ses actions et son état interne.
- **Rationalité** : La rationalité se réfère au succès escompté compte tenu de ce qui a été perçu. Si perceptions insuffisantes alors résultats non satisfaisants [17].

## 2.2.3 De l'agent aux systèmes multi-agents

Comme présenté dans la Figure 2.2, un Système Multi-Agents est un système composé des éléments suivants [31] :

- **Un environnement  $E$**  : Un espace disposant d'une métrique et d'un support de communication (Voir section *Environnement des tâches*).
- **Un ensemble d'objets  $O$**  : Ce sont des objets qui sont situés dans l'environnement  $E$ . Il est donc possible à tout moment, d'associer à  $O$  une coordonnée dans  $E$ .
- **Un ensemble d'agents  $A$**  : les entités actives et autonomes du système.
- **Un ensemble d'interactions  $I$**  : Elles permettent aux agents  $A$  d'interagir avec d'autres agents  $A$  ou de modifier l'environnement  $E$ .
- **Un ensemble d'opérateurs  $O_p$**  : Permettant aux agents de percevoir, produire, consommer, transformer et manipuler des objets  $O$ .



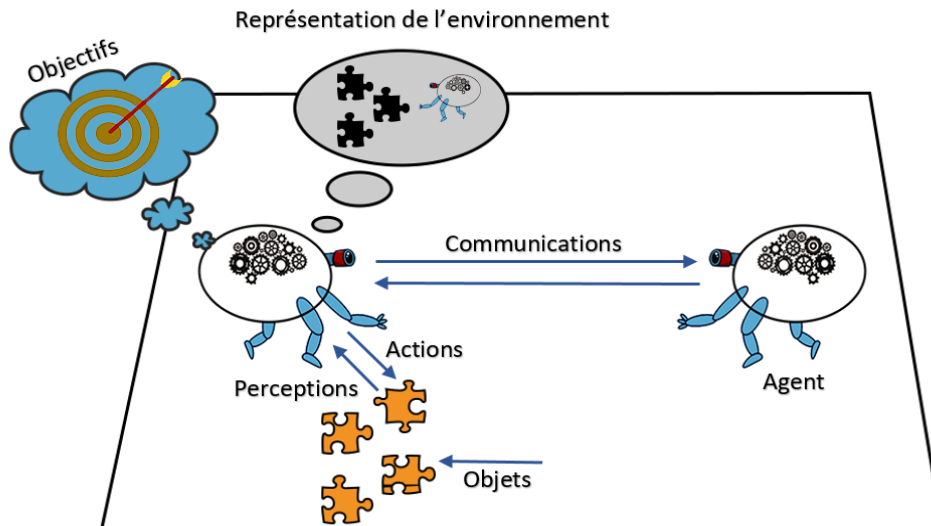


FIGURE 2.2 – Environnement multi-agents

De fait de la distribution, une approche multi-agents permet de mettre en évidence des propriétés telles que la collaboration, la coopération, et la négociation entre les agents, afin de réaliser des objectifs, qu'ils soient individuels ou collectifs.

Notons que selon le type d'applications, l'environnement peut avoir différentes propriétés à savoir [49] :

- **Environnement entièrement observable ou partiellement observable** : Les agents ont accès à tout ou une partie de l'environnement.
- **Environnement mono-agent ou multi-agents** : Agents opérants seule ou en plusieurs dans un environnement.
- **Environnement déterministe ou stochastique** : Capacité de l'agent à déterminé ou pas l'état suivant de l'environnement à partir de l'état courant.
- **Environnement épisodique ou séquentiel** : Dans un environnement épisodique, l'expérience de l'agent est divisée en épisodes. A chaque épisode, l'agent reçoit des informations et exécute une action, l'action exécutée par l'agent n'a d'effet que sur l'épisode courant, contrairement à un environnement séquentiel où l'expérience de l'agent est continue, et chaque action exécutée par l'agent influence l'état future de son environnement.
- **Environnement statique ou dynamique** : Si l'environnement de l'agent n'évolue pas pendant sa prise de décision alors c'est un environnement statique sinon c'est un environnement dynamique.
- **Environnement discret ou continu** : Un environnement est dit discret s'il y a un nombre limité de perceptions et d'actions distinctes possibles, à l'inverse d'un environnement continu où ses perceptions et actions appartiennent à un intervalle de valeurs continues.

## 2.2.4 Interactions dans un environnement multi-agents

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent ou tout simplement de l'informer sur l'état de son environnement. Elles permettent aux agents de participer à la satisfaction d'un but global s'il s'agit d'une coopération ou de résoudre des conflits afin d'atteindre des buts individuels dans le cas de négociations entre agents antagonistes. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent.

Les agents ne peuvent pas agir directement sur l'état et le comportement d'un autre agent, pour cela les interactions sont mises en oeuvre par un transfert d'informations entre agents, soit par perception, soit par communication. Pour définir les interactions entre agent deux approches sont envisageables : Les interactions directes par envoi de messages, et les interactions indirectes par émission / perception des signaux via l'environnement commun.

### 2.2.4.1 Formes d'interactions entre les agents

Les interactions entre agents peuvent être de forme basique très simple, se résumant à un simple échange intentionnel et explicite d'informations, comme elles peuvent être de forme plus évoluée selon que les entités aient des connaissances plus étendues, des capacités de planifier ou encore de faire évoluer leurs raisonnements par apprentissage en observant les actions des autres. Les interactions entre agents peuvent varier selon le contexte : coexistence, compétition ou coopération [18], nous distinguons :

- **Coopération** : La coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents, le processus de décomposition doit tenir compte des ressources disponibles et des compétences des agents [31].
- **Collaboration** : La collaboration suppose que les agents doivent coordonner leurs actions pour pouvoir disposer de l'ensemble de leurs compétences pour éventuellement réaliser des objectifs communs.
- **Coordination** : La coordination se traduit par un comportement individuel visant à servir ses propres intérêts tout en essayant de satisfaire le but global du système [30]

Le processus fait en sorte que les agents agissent ensemble tout en bénéficiant du travail des uns et des autres au travers d'interactions positives (une certaine tâche favorisant ou améliorant une autre tâche), au lieu de se nuire au travers d'interactions négatives (une tâche empêchant ou bloquant une autre) [47].

- **Compétition** : Nous distinguons deux types de compétitions :
  - **Compétition individuelle** : Les agents luttent pour atteindre leurs objectifs individuels indépendamment des autres agents.
  - **Compétition collective** : Lorsque les agents n'ont pas les compétences suffisantes, ils doivent se regrouper au sein de coalitions ou d'associations pour parvenir à atteindre leurs objectifs.
- **Négociation** : La négociation intervient lorsque des agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents.

Les interactions regroupent les formes de communication qui nécessitent la mise en place d'un protocole de communication, qui peut prendre la forme d'un langage [39] où les agents s'envoient des messages structurés pour communiquer. C'est l'approche mise en avant par la **FIPA** « *Foundation for Intelligent Physical Agent* », qui a spécifié un standard de communication agents, **ACL** pour « *Agent Communication Language* ». En effet, FIPA-ACL est basé sur la théorie des actes de langage. Il définit des actes de communication associés à diverses finalités [18], tels que le passage d'informations, la réquisition d'informations, la négociation, la distribution des tâches et en fin la manipulation des erreurs.

#### 2.2.4.2 Modélisation des interactions

Les interactions entre agents sont souvent représentées avec des approches logiques, ou en utilisant des outils graphiques tels que les automates à états finis ou les réseaux de Pétri (Voir figure 2.3)

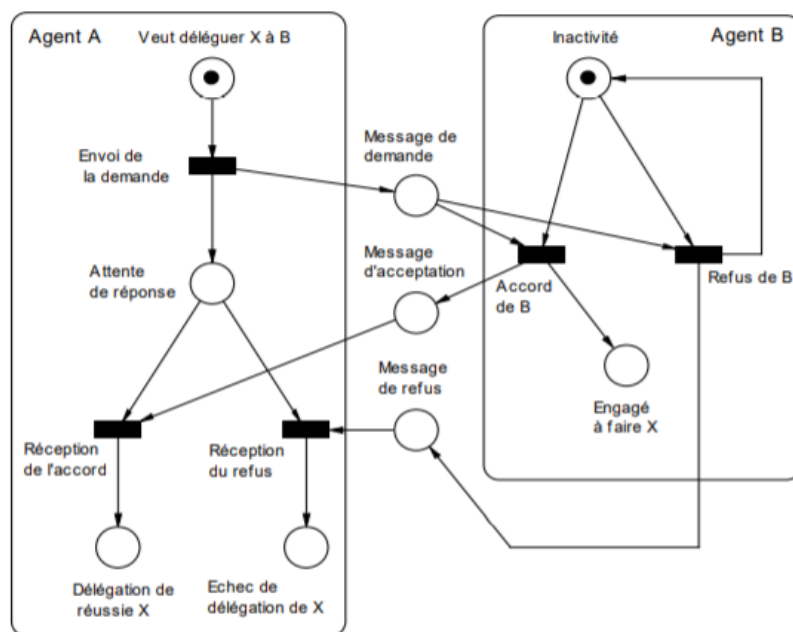


FIGURE 2.3 – Exemple de modélisation d'un protocole d'interaction entre deux agents par le réseaux de Pétri[39]

Modéliser les interactions permet d'une part à un observateur de comprendre la façon dont elles se déroulent et d'autre part de concevoir le comportement d'un agent qui respecte, par exemple, un protocole de communication. La modélisation permet ainsi de visualiser le système, de le simuler, d'analyser son comportement et de vérifier ses propriétés.

## 2.2.5 Plateformes de développement des systèmes multi-agents

Plusieurs plateformes de développement ont vu le jour pour faciliter le développement des Systèmes multi-agents et leurs déploiements, nous citons quelques-unes [9] :

- **JADE** : « *Java Agent DEvelopment* » est un framework de développement de systèmes multi-agents, open-source et basé sur le langage Java. Il offre en particulier un support avancé de la norme FIPA-ACL, ainsi que des outils de validation syntaxique des messages entre agents basés sur les ontologies.
- **JADEx** : C'est une plate-forme agent développée en Java par l'*Université de Hambourg* qui se veut modulaire, compatible avec de nombreux standards et capable de développer des agents suivant le modèle **BDI** « *Belief-Desire-Intention* ».
- **CORMAS** : « *COmmon Resources Multi-Agent System* » est un framework de développement de systèmes multi-agents, open-source et basé sur le langage de programmation orientée objet Small-Talk. Spatialisé, il est centré sur des problématiques de recherche en sciences du développement et de négociation entre acteurs.
- **MAGIQUE** : C'est une plate-forme pour agents physiquement distribués écrite en Java et fournissant un modèle de communication originale d'appel à la cantonade. Dans MAGIQUE, les compétences sont dissociées des agents. L'architecture des agents et les différentes compétences sont développées séparément. Les compétences sont ensuite greffées comme plugin dans les agents au gré du concepteur.

## 2.3 Application des SMA aux systèmes de transports

La transposition des notions orientées agent au domaine des transports est en accord avec les caractéristiques des deux domaines, on y trouve des notions d'autonomies, de distribution et d'environnement partiellement observable, il est donc logique que l'on retrouve des concepts tirés du paradigme agents dans le domaine des transports [39].

Nous considérons que les entités telles que des moyens de transport et les infrastructures peuvent être modalisés sous forme d'agents intelligents capable d'interagir dans un environnement SMA, dans le but de résoudre divers problèmes liés à la gestion des réseaux routiers. Citons comme exemple :

- **Gestion des intersections** : La gestion des intersections est un problème récurrent dans le milieu routier, la solution la plus utilisée est l'installation des feux de signalisation dans les endroits critiques. Le cycle de feux de ces entités est généralement périodique, mais en dotant ces entités et les moyens de transport de la technologie agent et les faire coopérer, les feux de signalisation pourront adapter le cycle de feux et de le réguler par rapport à la densité du trafic dans un secteur donné.
- **Réduction des accidents** : L'environnement routier est un environnement partiellement observable, le conducteur a peu d'informations sur les intentions des autres conducteurs et les informations d'autres véhicules, ce qui engendre beaucoup de situations de conflits qui se termine généralement par des accidents.  
  
La collaboration et la coopération des agents permet d'anticiper les tâches des autres entités routières, rendre l'environnement de l'agent mieux observable et synchroniser leurs comportements pour éviter les accidents routiers.
- **Gestion de la congestion** : Problème le plus étudié dans le système de transport, de par le nombre d'entités mobiles en circulation. L'application des SMA permet d'envisager à tout moment des modèles d'organisation permettant de fluidifier la circulation et d'anticiper des échappatoires pour réduire la densité routière dans un secteur en particulier.
- **Autres problèmes** : D'autres problèmes de gestion liés au système de transport sont d'actualité tels que : la régulation du trafic routier, gestion des parkings, etc. les SMA peuvent offrir diverse solutions à la résolution de ces problèmes.

## 2.4 Présentation des différents travaux réalisés existants dans la littérature

Dans cette section, nous présenterons trois travaux significatifs ayant trait à notre problème et qui ont enrichi la littérature par des techniques de résolution des différents problèmes liés à la gestion des STI.

### 2.4.1 Simulateur multi-agents de trafic coopératif

La réalisation de ce travail [40] s'est fait dans un environnement où les véhicules sont connectés et partiellement automatisés, capable d'échanger des informations entre eux et sont dotés aussi de capteurs embarqués qui leurs fournissent des connaissances par lesquelles ils construisent une représentation de l'environnement dans lequel ils évoluent.

L'article apporte une présentation d'un modèle générique qui sera appliqué et testé en simulation dans le but d'évaluer la stratégie coopérative vu du côté performance et sécurité avec une fiabilité de l'information.

- **Solution proposée** : L'idée consiste à tirer profit des informations fiables pour prévenir les mouvements des véhicules et régulariser le flux du trafic. Le procédé se fait en interaction entre trois couches qui sont :
  - **Couche physique** : Consiste à la localisation du véhicule et au calcul des inter-distances entre les véhicules.
  - **Couche communication** : Ce sont les types de messages envoyés entre les véhicules, deux types de messages sont proposés : Les messages de mesures et les messages de confiance.
  - **Couche confiance et fiabilité de l'information** : L'auteur propose d'introduire le concept de confiance comme un mécanisme capable d'affronter les problèmes de fiabilité des capteurs en adoptant un modèle qui utilise une structure de données correspondant à un réseau de confiance nommé *TrustNet*. Ce dernier détermine des représentations numériques de la confiance calculée à partir des expériences acquises par les agents, et dans ce modèle la confiance entre agents correspond à l'opinion de la majorité.
- **Implémentation** : Le modèle présenté dans l'article est implémenté dans *MovSim* en lui ajoutant le concept agents et intégrer l'architecture de modélisation du trafic coopératif, dont l'objectif est de garantir une perception locale et partielle de leurs environnements, de modéliser leurs interactions complexes et d'assurer un processus autonome aux véhicules.
- **Résultats** : Pour évaluer les performances, le modèle doit répondre à une amélioration significative du flux du trafic en termes d'homogénéité et de sécurité.

Deux expérimentations de simulation ont été faites, la première étudie l'influence de la stratégie coopérative sur l'homogénéité du trafic, quant à la deuxième, elle a rapport avec la validation de l'utilisation de la confiance pour évaluer la fiabilité des capteurs.

- **Première expérience** : Les résultats présentés dans la première expérience de l'article montrent que même un faible pourcentage de véhicules connectés permet de prévenir la propagation des ondes de congestion, en général, les résultats montrent l'influence de la stratégie coopérative sur l'homogénéité de flux du flux du trafic, et même avec une faible portion de véhicules connectés.
- **Deuxième expérience** : Les résultats fournis par la deuxième expérimentation montrent l'effet positif du modèle de confiance sur le comportement physique des véhicules au détriment des informations erronées envoyées par les capteurs.

## 2.4.2 Mécanisme d'adaptation de vitesse pour évitement de collision dans un réseau VANET

L'article [19] présente un système de régularisation de vitesse de manière à éviter des collisions dans un réseau routier, les véhicules dans l'environnement sont connectés via un réseau VANET et dotés chacun d'un agent permettant de prendre en compte les informations fournies par le capteur du véhicule et celles reçues de la part des autres agents via une communication réseau.

- **La solution proposée** : L'apport de ce travail, se repose sur le « *Mécanisme d'adaptation de vitesse* » qui va mener à une conduite avec une vitesse régularisée en conséquence afin de garder la distance de sécurité pour éviter des collisions potentielles.

Périodiquement, des véhicules s'envoient des informations tels que leurs position, vitesse et distance qui les séparent. Si un véhicule précède un autre, grâce aux informations reçues du véhicule qui le devance, l'agent calcule le temps et la distance potentielle de collision qui est définie comme suit :

$$\begin{cases} X_A(t=0) + V_A(t=0) \times t + \frac{1}{2} a_{A(\min)} \times t^2 = X_B(t) \\ V_A(t) = V_B(t) \end{cases}$$

et ajuste sa vitesse de manière à éviter la collision.

- **Implémentation et résultats** : Le mécanisme est testé avec différentes variations de distances relatives au temps et à la vitesse, il a été montré qu'il n'y aurait pas de collisions pour une décélération  $a = -12\text{m/s}^2$  ou moins, ainsi la décélération recommandée est  $a_{\min} < -9\text{m/s}^2$ , il a été constaté aussi durant la phase de simulation, que si un véhicule roule à une vitesse de 30m/s et son prédécesseur à une vitesse nulle, la collision aura lieu à 2.5s, et pour les autres cas de simulation ou la décélération est fixée à  $-6\text{m/s}^2$ ,  $-9\text{m/s}^2$  et  $-12\text{m/s}^2$ , les collisions n'auront jamais lieu.

Comme résultat général, la valeur de décélération optimale recommandée est  $a_{\min} < -4\text{m/s}^2$ .

## 2.4.3 Commande neuro-floue pour le suivi de trajectoire et l'évitement d'obstacles dans un véhicule autonome

Le travail qu'apporte le document [42] consiste en l'usage des commandes neuro-floue dont l'objectif est d'implémenter un système embarqué capable d'assister le conducteur dans l'évitement d'obstacles et de collisions, bien-sûr, ce système va traiter des informations sur l'environnement captées grâce à des capteurs radars, ultrasonores ainsi que des caméras stéréovisions.

- **Solution proposée** : Par intermédiaire entre le véhicule et l'environnement, les capteurs perçoivent l'environnement, et transmettent les informations acquises au module de calcul (réseau neuro-flou coopératif) et qui donnera en sortie l'angle de déviation et la pression de freinage des roues.

Un Réseau de neuro-flou est dit coopératif vu son architecture qui vise l'usage des réseaux de neurones et les systèmes flous en parallèle, ce dernier est un mécanisme d'apprentissage des réseaux de neurones artificielles et détermine les fonctions d'appartenance du système d'inférence floue.

- **Implémentation** : Comme fonction d'activation, l'auteur a opté pour la fonction **ELU** « *Exponential Linear Unit* » avec un algorithme d'optimisation *ADAM* et un taux d'apprentissage de 0.0005 avec 16 itérations.

Concernant l'angle de braquage du volant comme donnée de sortie, est due à la considération de trois données d'entrée, à savoir, les images de caméras placées à droite, celles de gauche et celles du centre, et pour évaluer la prédiction, la mesure d'erreur moyenne quadratique a été utilisée, pondérée par le carré de l'erreur

- **Résultats** : Grâce au contrôle neuronal (réseau de neurones à convolution), les images prises par les caméras seront traitées, dans le but de maintenir le véhicule sur la route sans dérapage et de suivre un chemin prédéfini.

Concernant l'évolution de l'erreur lors de l'apprentissage, elle est en diminution de manière significative sur les premières phases de l'apprentissage, et à la fin, la courbe de l'évolution devient linéaire indiquant que le réseau a atteint l'erreur minimale. Les résultats donnés pour l'erreur sur la suite des commandes exécutées en fonction du nombre d'itérations, est décroissante et tend vers zéro. Avec un résultat pareil, l'identification des obstacles est garantie. La précision au cours de l'apprentissage croît au fil des années, elle a atteint 97% en 12 ans, cela démontre la rapidité du traitement de l'information. Certes, même avec les réseaux entraînés, il faut lui faire subir des tests afin de vérifier son efficacité. Une remarque aussi pertinente dans la phase d'apprentissage concernant l'évolution de l'erreur sur les données du test, elle est décroissante de manière exponentielle, ce qui veut dire que le système sera performant si on lui fait subir plusieurs tests.

En se basant sur les données captées par les capteurs, le véhicule agira automatiquement pour éviter le risque de collisions grâce au système de freinage et de direction, ceci est principalement dû au contrôleur flou.

Les résultats de la simulation du contrôleur flou démontrent l'efficacité de la commande à base de la logique floue pour les différentes fonctions du système à savoir, l'évitement d'obstacle et de collision et le maintien du véhicule en route.



## 2.5 Conclusion

Dans ce chapitre nous avons présenté en premier lieu les agents, leurs caractéristiques qui accentuent leurs importances, puis nous avons abordé les SMA et les interactions pouvant conduire à la réalisation d'un système coopératif et intelligent pour la modélisation et la résolution des problèmes complexes tel que le problème qui fait l'objet de notre étude " *Evitement des collisions dans les systèmes de transports* ", en suite nous avons parlé des SMA dans les moyens de transport et les problèmes pouvant être résolus en appliquant cette approche, enfin nous avons terminé par la présentation des travaux réalisés par des auteurs sur les problématiques liées aux systèmes de transports.

# Approche de modélisation proposée et contribution

## 3.1 Introduction

L'élaboration de nos solutions passe par la conception et modélisation des systèmes interagissant, assurant la coordination et la synchronisation de ces derniers dans le but d'obtenir les résultats souhaités.

Durant ce chapitre nous allons nous pencher sur la spécification des agents du système et le travail qui a été proposé dessus en concrétisant (implémentant) nos idées par des algorithmes. Enfin nous allons conclure en abordant les solutions proposées pour répondre à la problématique sur différents types de circulation.

## 3.2 Modélisation d'un véhicule

La modélisation du mouvement d'un véhicule revient à spécifier la cinématique des paramètres tels que la position, la vitesse et la direction du véhicule. Le mouvement est conféré aux véhicules par deux actions, à savoir sa vitesse et l'orientation des roues directrices. Comme présenté dans la Figure 3.1, une voiture est généralement composée de 4 roues.

- 2 roues sur l'essieu avant, qui sont directrices dans notre cas.
- 2 roues sur l'essieu arrière.

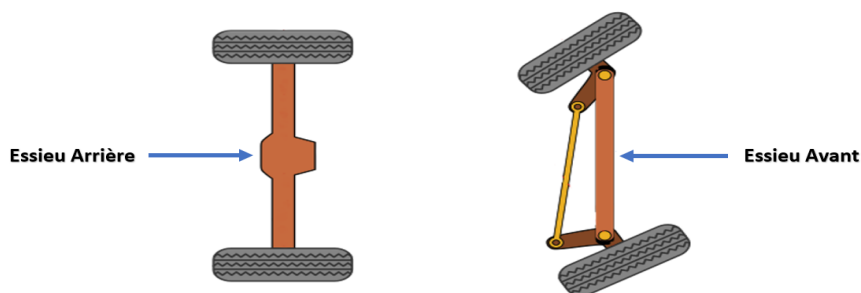


FIGURE 3.1 – Positionnement des roues

De ce fait les roues de l'essieu avant n'ont pas forcément la même direction que les roues de l'essieu arrière [50]. La vitesse du véhicule dépend fortement des actions d'accélération/décélération (Freinage) appliquées au véhicule.

Nous supposant que le véhicule est représenté dans un repère en 2<sup>ème</sup> dimensions  $R(O, \vec{X}, \vec{Y})$  et son état est défini sous un espace  $M((x, y), (\vec{v}_x, \vec{v}_y), (\varphi, \alpha))$  (Voir Figure 3.2) où [21] :

- $x, y$  : Position latérale et longitudinale.
- $v_x, v_y$  : Vitesse latérale et longitudinale (que nous considrons égale à 0).
- $\varphi, \alpha$  : L'angle de braquage du véhicule et des roues.

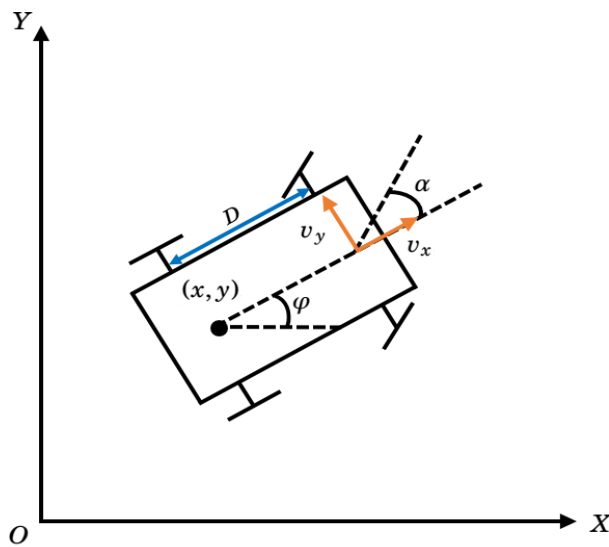


FIGURE 3.2 – Présentation des paramètres cinématique d'un véhicule

Par un simple raisonnement géométrique en s'appuyant sur la Figure 3.2 et les formules de trajectoire [1][21], nous obtenons les équations cinématiques suivantes :

$$\left\{ \begin{array}{l} x_{t+1} = x_t + v_t \times \cos(\varphi_t) \times t + \frac{1}{2} \times a \times t^2 \\ y_{t+1} = y_t + v_t \times \sin(\varphi_t) \times t + \frac{1}{2} \times a \times t^2 \\ \varphi_{t+1} = \varphi_t + v_t \times \frac{\tan(\alpha)}{D} \times t \\ v_{t+1} = v_t + a \times t \\ \alpha_{t+1} = \alpha_t + \eta \times t \end{array} \right. \quad (3.1)$$

- $\eta$  : Représente la vitesse d'orientation imposée aux roues directrices.
- $a$  : Accelération du véhicule.
- $D$  : Distance entre les deux essieux du véhicule.

### 3.3 Modèle agent et architecture

Les véhicules évoluent dans un environnement distribué basé sur la technologie agent. Un agent doit être capable de prendre les décisions appropriées pour éviter tout type de collisions, pour cela nous avons doté les agents de différents modules permettant la réception et la transmission des messages personnalisés, l'analyse des informations et prévision, planification des évènements et en fin un journal des évènements (Voir Figure 3.3).

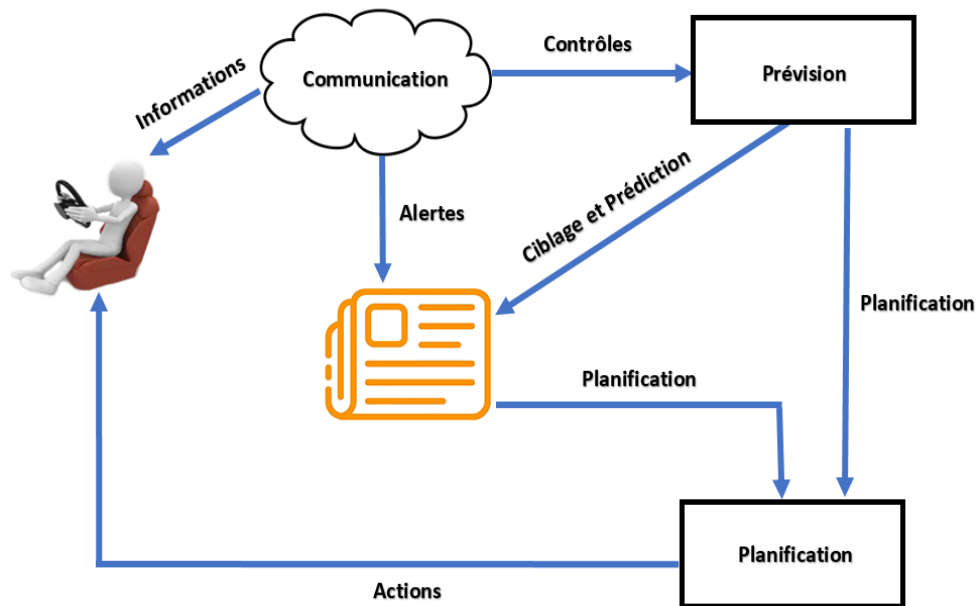


FIGURE 3.3 – Modules composant un agent

#### 3.3.1 Module de partage et communication d'informations

La communication entre les différents véhicules se fait par l'application d'un protocole de communication proposé par FIPA et basé sur les messages ACL (Agent Communication Language). Nous distinguons 4 types de messages :

- **Messages de contrôle** : Type de messages qu'un véhicule envoie périodiquement (10 messages/seconde) aux autres véhicules connectés sur son réseau, ce type de message a pour but d'informer les autres véhicules de ses paramètres, à savoir : sa position, sa vitesse, sa vitesse max, sa direction, son accélération, la voie de la route empruntée.
- **Messages d'alerte** : Type de messages qu'un véhicule envoie aux autres véhicules en cas de détection d'un évènement pouvant mettre en danger le réseau routier, nous pouvons citer : Les embouteillages, un accident de la route, Les travaux dans un tronçon de la route, etc.

Les paramètres envoyés dans ce type de message sont : Evènement, la position, la vitesse, la direction et la voie. Ces paramètres sont automatiquement enregistrés dans le journal des évènements.

• **Messages de dépassement** : Types de messages qu'un véhicule envoie aux autres véhicules en cas de dépassement, ce type de message a pour but :

- Informer le véhicule qui le devance de l'intention de dépassement pour qu'il n'exécute pas une action pouvant mener en danger la procédure de dépassement.
- Informer le véhicule qui le précède de la procédure de dépassement.
- Le véhicule qui le devance peut intervenir avant la procédure de dépassement au cas où y a un danger et conseiller le véhicule voulant dépasser soit de reporter ou d'annuler la procédure pour éviter un danger.
- La mise à jour d'informations.

Les paramètres envoyés dans ce type de messages sont : la direction de dépassement, la position et la vitesse.

• **Messages informatifs** : Type de message qu'un véhicule envoie au conducteur sur le tableau de bord, dans le but d'informer ce dernier des actions éventuelles qu'il va exécuter et la cause, pour ne pas surprendre le conducteur, et lui donner l'opportunité de changer son comportement pour éviter des collisions.

Ce type de message est envoyé périodiquement à chaque  $\frac{1}{5}$  de la distance qui sépare la voiture à la position d'exécution de l'action (Voir Figure 3.4).

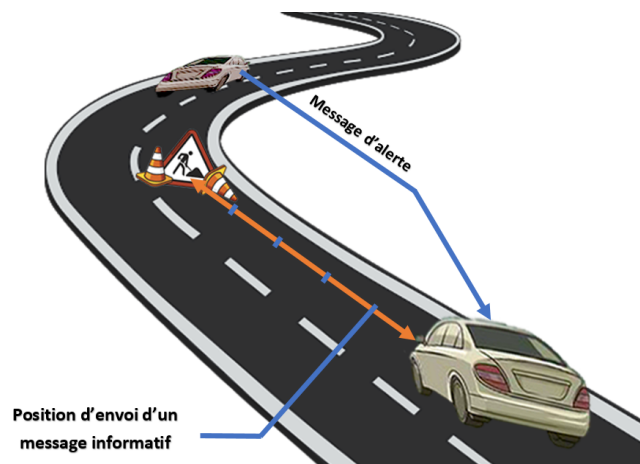


FIGURE 3.4 – Transsmission des messages informatifs

Les paramètres envoyés dans ce type de message sont : la cause, la distance de séparation et le type d'action à exécuter.

### 3.3.2 Module de prévision et prédiction des collisions

Module permettant l'analyse et le traitement des informations. Ce module a pour but de faire des estimations et des prédictions des collisions à partir des données reçues. La prise de décision et le calcul des paramètres d'actions sont planifiés à partir des résultats émis par le traitement des différents processus de ce module. Pour cela nous distinguons 2 processus tout aussi importants : Processus d'estimation et le Processus de prédiction des collisions.

#### 3.3.2.1 Processus d'estimation des paramètres cinématiques

Le processus d'estimation permet d'estimer les valeurs futures des paramètres cinématiques d'un véhicule (à un certain temps  $t$  ou une position) à partir des valeurs précédentes acquises par les échanges de messages entre ces entités mobiles. Les paramètres pris en compte dans cette estimation sont : le temps, la position, la direction, la vitesse, la vitesse maximale et en fin l'accélération. Nous présentons le processus d'estimation par cet algorithme :

---

**Algorithme 1** : Processus d'estimation des paramètres cinématiques d'un véhicule

---

**Données** :  $t$  : temps, position, vitesse, vitesse max, accélération

**Résultat** : position, direction, vitesse, accélération

```
1 Chercher la vitesse du véhicule à  $t$  ;
2 si la vitesse à  $t >$  à vitesse max alors
3   | Chercher  $t'$  pour que la vitesse == vitesse max;
4   | Mise à jour des paramètres à  $t'$ ;
5   | retourner Estimation_paramètres( $t'$ , positions, direction, vitesse max, accélération = 0)
6 sinon
7   | Mise à jour des paramètres cinématiques à  $t$ ;
8 fin
```

---

1. L'algorithme est initié par tout les agents des véhicules du réseau routier, pour estimer les paramètres cinématiques souhaités.
2. L'algorithme reçoit en entrée les paramètres tels que la position du véhicule, sa direction, sa vitesse instantanée, sa vitesse maximale, son accélération et en fin le temps d'estimation des paramètres.
3. En premier lieu, l'algorithme va prédire la vitesse du véhicule à  $t$ , en utilisant les différentes formules cinématique 3.1.
4. La condition présentée dans la ligne 2 permet de tester si le véhicule a dépassé sa vitesse maximale à  $t$ , or un véhicule ne peut dépasser sa vitesse maximal (estimation dans le pire cas).
  - Si elle s'avère vrai, alors l'algorithme cherchera à prédire le temps auquel la vitesse du véhicule sera maximale, et en fin mettre les paramètres à jour pour le temps calculé et refaire le processus d'estimation à partir du temps calculé.
  - Sinon l'algorithme met à jour les paramètres cinématiques pour  $t$  et retourne les résultats.

### 3.3.2.2 Processus de prédiction des collisions

Le processus de prédiction des collisions est le processus permettant la prédiction des éventuelles collisions avec les véhicules dans le système de transports. Ce processus est déclenché à chaque réception d'un nouveau message de la part de ces entités.

Le processus de prédiction passe par 2 étapes importantes exécutées séquentiellement, à savoir : La détection des véhicules cibles et la prédiction des collisions.

- **Détection** : Généralement les collisions se passent quand un véhicule qui précède un autre percute le véhicule qui le devance ou quand deux véhicules de sens opposé se croisent sur la même voie de la route. Nous sommes partis de ce principe pour élaborer notre algorithme de détection qui stipule que :

- Si un véhicule est devancé par d'autres véhicules en étant sur la même voie de la route, le véhicule le plus proche représente un véhicule cible.
- Si un véhicule est dans une portion de la route où d'autres véhicules roulent dans le sens opposé, le véhicule le plus proche représente un véhicule cible. Dans ce cas la collision est directement envisageable, le véhicule cible est enregistré dans le journal des événements.

- **Prédiction** :

Une fois les véhicules ciblés, l'algorithme de prédiction de collision va être exécuté pour détecter, si une collision va se produire entre ces deux entités au cours du temps ou pas.

Soit un véhicule  $A$  précédant un véhicule  $B$ , la prédiction de la collision revient à résoudre cette équation :

$$\begin{cases} x_A + v_A \times \cos(\varphi_A) \times t + \frac{1}{2} \times a_A \times t^2 = x_B + v_B \times \cos(\varphi_B) \times t + \frac{1}{2} \times a_B \times t^2 \\ \frac{1}{2} \times (a_A - a_B) \times t^2 + (v_A \times \cos(\varphi_A) - v_B \times \cos(\varphi_B)) \times t + (x_A - x_B) = 0 \\ \frac{1}{2} \times (a_A - a_B) \times t^2 + (v_A - v_B) \times t + (x_A - x_B) = 0 \quad \text{avec } \varphi = 90^\circ \\ A^2 + B + C = 0 \end{cases} \quad (3.2)$$

Notre algorithme est présenté sous cette forme :

---

**Algorithme 2** : Processus de prédiction de collision

---

**Données** : positions, direction, vitesses, vitesses maximal, accélération

**Résultat** : Vrai ou Faux

```
1 tant que Position estimée du véhicule A < Position estimée véhicule B faire
2   si Accélération des deux véhicules égale à 0 alors
3     retourner Vitesse véhicule A > Vitesse véhicule B;
4   sinon
5     si Tout les paramètres du véhicules A < Paramètres du véhicule B alors
6       retourner Faux;
7     sinon
8       Chercher  $t$  avec l'application de la formule 3.2;
9       Appel à Estimation_paramètres(positions, direction, vitesses, vitesses maximal,
10        accélération,  $t$ ) ; // Estimation des paramètres cinématiques des deux véhicules à  $t$ 
11     fin
12   fin
13 retourner Vrai;
```

---

L'algorithme de prédiction de collision, est un algorithme itératif, qui estime les paramètres des véhicules étudiés à chaque itération. Ces itérations permettent d'étudier les changements d'accélération des véhicules suivant leurs vitesses.

1. L'algorithme est exécuté par l'agent du véhicule  $A$ , pour prédire l'éventualité d'une collision ou pas avec le véhicule  $B$ .
2. L'algorithme reçoit en entrée les paramètres tels que la position du véhicule, sa vitesse instantanée, sa vitesse maximale et fin son accélération.
3. l'algorithme retournera une valeur de vérité, montrons si une collision va se produit à travers le temps ou pas.
4. En premier lieu l'algorithme testera si à chaque itération (après estimation), le véhicule  $A$  arrivera a dépassé le véhicule  $B$  à un certain temps.
5. L'estimation des paramètres se fera à partir du processus d'estimation.
6. L'algorithme va arrêter son exécution selon 3 cas :
  - Dans le meilleur des cas, les paramètres du véhicule  $A$  sont inférieur aux paramètres du véhicule  $B$ , c'est à dire leurs accélérations, vitesses et en fin leurs vitesses Max.
  - Si à un temps calculé la position de  $A$  est supérieur à la position de  $B$ .
  - Suite aux changements d'accélération des deux véhicules (véhicule atteignant la vitesse max), à partir de cet état, la collision est envisageable que si la vitesse du véhicule  $A$  est supérieur à la vitesse du véhicule  $B$ .



7. Si la collision existe, le véhicule  $B$  sera enregistré dans le journal des évènements avec ses paramètres.

### 3.3.3 Journal des évènements

Le journal des évènements est un historique de tous les évènements qui vont se produire à un certain temps : collision avec un véhicule, embouteillage ou accidents dans les prochaines mètres, etc. Le journal des évènements est divisé en deux parties : la partie enregistrant les paramètres envoyés dans les messages d'alertes et la partie enregistrant les véhicules avec lesquels des collisions peuvent se produire.

Cette séparation va permettre à l'agent de distinguer les différentes tâches à planifier et avoir une vue précise des états futurs du système. Pour chaque partie, l'ordonnancement se fait par le calcul de l'évènement le plus proche à réaliser.

La suppression d'un évènement passe par deux approches différentes :

- Une fois la planification calculée.
- Réception de plusieurs messages de contrôle de la part de plusieurs agents à partir d'une position supérieure à celle de l'évènement sans que le même évènement ne soit notifié.

### 3.3.4 Module de planification des actions à exécuter

Le module planification sert à planifier les actions à exécuter par rapport aux évènements issues du journal des évènements. Les algorithmes de ce module permettent d'estimer la vitesse, la position et les actions à entreprendre. Suivant le type d'évènements, nous distinguons deux cas de figures :

- **Évènements de collisions avec les véhicules** : Le code de la route [23] stipule que la distance de sécurité entre deux voitures doit être égale à 2 fois la vitesse du véhicule en  $m/s$ , par exemple si un véhicule roule à  $20m/s$  alors la distance de sécurité doit être égale à  $20 * 2 = 40m$ .

Le but de cette approche est de trouver un algorithme capable de prédire les paramètres permettant d'adapter la vitesse d'un véhicule de tels sorte qu'il préserve une distance de sécurité tel qu'elle est énoncée dans le code de la route. Pour ce faire nous devons avoir connaissance des formules tels que la distance de freinages, le temps de réaction, équation de collision 3.2 :

- **Temps de réaction** : Le temps de réaction, c'est le temps qui sépare le temps de perception et le temps d'exécution, pendant le temps de réaction, la voiture continue de rouler à la même vitesse.

La durée du temps de réaction est en moyenne d'une seconde pour un conducteur en bonne condition physique. Mais elle varie en fonction de l'état et de la concentration de ce dernier (une demie seconde à 3 secondes) [13].

– **La distance de freinage** : La distance de freinage est calculée grâce à cette formule[10] :

$$d_{\text{freinage}} = \frac{v_f^2 - v_i^2}{2 \times g \times f} \quad (3.3)$$

- \*  $v_i, v_f$  : vitesse initiale et vitesse finale exprimées en  $m/s$ .
- \*  $g$  : La décélération du véhicule (négative).
- \*  $f$  : Coefficient d'adhérence de la chaussée où  $f \in [0, \dots, 1]$ 
  - $f \geq 0.6$  : Route sec et en bonne état.
  - $f \leq 0.5$  : Route mouillée et peut être en mauvaise état.

– **Formule d'estimation de la position de freinage** : Soit un véhicule  $A$  précédant un véhicule  $B$ , si la vitesse du véhicule  $A$  est supérieure à la vitesse du véhicule  $B$ , alors le véhicule  $A$  doit adapter sa vitesse de telle sorte qu'il respecte la distance de sécurité tel qu'elle est définie.

L'objectif de cette formule, est de trouver la position idéale de freinage de telle sorte à respecter la distance de sécurité et éviter une collision avec le véhicule  $B$ . En combinant les différentes formules étudiées, nous obtenons :

$$\begin{cases} x_{At} + 2 \times v_{Bt} + d_{\text{freinage}} + x_{A\text{temps réaction}} = x_{Bt} \\ \frac{1}{2} \times (a_A - a_B) \times t^2 + (v_A - v_B) \times t + (x_A - x_B) + 2 \times v_{Bt} + d + x_{A\text{temps réaction}} = 0 \end{cases} \quad (3.4)$$

$$v_{Bt} = v_B + a_B \times t. \quad (3.4a)$$

$$\begin{aligned} d_{\text{freinage}} &= \frac{v_A^2 t - v_B^2 t}{2 \times g} \quad \text{avec } f = 1 \\ &= \frac{(v_A^2 + v_A \times a_A \times t + a_A^2 \times t^2) - (v_B^2 + v_B \times a_B \times t + a_B^2 \times t^2)}{2 \times g} \\ &= \frac{a_A^2 - a_B^2}{2 \times g} \times t^2 + \frac{a_A \times v_A - a_B \times v_B}{2 \times g} \times t + \frac{v_A^2 - v_B^2}{2 \times g} \end{aligned} \quad (3.4b)$$

$$\begin{aligned} x_{\text{temps réaction}} &= v_A t + \frac{1}{2} \times a_A \\ &= a_A \times t + v_A + \frac{1}{2} \times a_A \end{aligned} \quad (3.4c)$$

- **Évènements reçus par les messages d'alertes** : L'environnement routier est un environnement à propriété dynamique, ses états changent d'une seconde à l'autre, plusieurs évènements peuvent être interceptés dans l'environnement routier, nous pouvons citer comme exemples : les embouteillages, les accidents, les travaux dans un tronçon d'une route, etc.

Une fois réception d'un message d'alerte, la planification de cet évènement permet d'estimer la position de freinage permettant de ralentir à l'arrivée de l'évènement et les actions nécessaires, tel que le changement de voie par exemple.

### **3.4 Évitement des collisions sur différents types de circulation routière**

Les situations de collisions sont récurrentes, et elles se produisent sur différents types de route. Durant ce travail nous nous sommes penchés sur les collisions se produisant dans les sens uniques et les doubles sens en employant les algorithmes vus dans les sections précédentes.

Il faut noter que le conducteur est toujours au courant de la situation et des actions que l'agent va entreprendre grâce aux messages informatifs affichés dans le tableau de bord et peut intervenir à tout moment en employant le bon comportement.

#### **3.4.1 Circulation à sens unique**

La circulation en sens unique est un type de circulation routière qui impose aux conducteurs de rouler sur une route particulière dans un sens, sans qu'il y'ait un sens opposé [11]. Ces routes sont généralement des routes à 2 voies ou 3 (Autoroutes).

Plusieurs situations de collisions peuvent se produire dans les sens uniques, dû au non-respect des distances de sécurité, une mauvaise manoeuvre de dépassement, la non prévision des risques, etc.

##### **3.4.1.1 Adaptation de vitesse**

Lors de la réception de chaque message de contrôle, l'agent lance le processus de prévision pour cibler les véhicules pouvant mener à une collision. A partir des prédictions faites par ce même agent, il estime les paramètres tels que la position de freinage et la vitesse de réduction calculées en utilisant les formules 3.4 et ainsi réussir à préserver la distance de sécurité entre deux véhicules. Tous les paramètres calculés seront planifiés.

### 3.4.1.2 Évènements

Comme présenté dans l'algorithme 3, Lors de la réception d'un message d'alerte, l'agent calcule les paramètres telles que la position de freinage et la vitesse d'adaptation à partir du processus de prédiction des collisions, puis selon l'évènement il anticipe les actions à entreprendre pour éviter tout type d'accident dès son arrivée sur lieu de l'évènement. Tous les paramètres calculés et les actions anticipées seront planifiés.

---

**Algorithme 3** : Processus de planification des évènements

---

**Données** : Agents, Position du véhicule, Journal d'évènements

```
1 Position_Freinage = Procédure Estimation_Position_Freinage(Paramètres dans évènement) ;
2 Planification(Position_Freinage);
3 tant que Position_Véhicule <= Planification.Position faire
4   | Message informatifs aux conducteurs(Planification.Position - Position_Véhicule) ;
5   | si Planification.Position == Véhicule.Position alors
6     |   | si Travaux alors
7       |   |   | Action_Freinage();
8         |   |   | si Voie Travaux == Voie véhicule A alors
9           |   |   |   | Changement de Voie();
10          |   |   |   | fin
11          |   |   | sinon
12          |   |   |   | Autres conditions;
13          |   |   |   | fin
14          |   |   | fin
15 fin
16 Supprimer de Planification(Planification.position);
```

---

1. L'algorithme est initié par tout les agents des véhicules roulant vers la direction de l'évènement.
2. L'algorithme reçoit en paramètres l'agent qui va exécuter les différentes actions, la position du véhicule et en fin le journal d'évènements.
3. L'algorithme procède tout d'abord au calcul de la position de freinage permettant à l'agent d'actionner le freinage une fois arriver à la position calculée.
4. Une fois la position de freinage calculé, il planifie les actions à entreprendre à partir des données qui lui ont été reçus par les véhicules émetteurs du message.
5. Arrivé à l'évènement, l'agent actionne le freinage et applique d'autre actions qu'ils avait planifiées suivant différentes situation (Changement de voie pour les travaux, etc.).

### 3.4.1.3 Dépassement

Le dépassement est le fait pour un véhicule, de changer de voie pour doubler un autre véhicule roulant à une vitesse inférieure. Selon les pays les dépassements peuvent se faire par la gauche (dans notre cas) ou par la droite (Angleterre, etc.).

Pour un bon dépassement il faut assurer [23] :

- Le véhicule menant la procédure de dépassement a assez de réserve en accélération pour dépasser.
- Vérifier qu'aucun véhicule de derrière n'a lancé une procédure de dépassement.
- Le véhicule dépassant n'accélère pas ou n'exécute pas d'action menant en péril la procédure.

Comme présenté dans l'algorithme 4. Le processus de dépassement passe par la perception approfondie de l'environnement véhiculaire. La procédure est initiée par le conducteur en allumant les clignotants du côté gauche.

---

#### Algorithme 4 : Processus de dépassement à sens unique

---

**Données :** Agents, Environnement

```
1 Action_Procédure_Dépassement() ; // Initiée par le conducteur
2 Si aucun véhicule de derrière ou devant n'a initié la procédure de dépassement;
3 Transmission_messages(Véhicule_devant, Véhicule_derrière, Message_Dépassement);
4 Confirmation ou pas de la procédure par le véhicule de devant ;
5 si Voie libre alors
6 | Permettre la procédure de dépassement;
7 | Mise_à_jour_Positions() ;
8 sinon
9 | si Procédure Prédiction_Collision(Agent) alors
10 | | Bloquer la procédure()
11 | sinon
12 | | Permettre la procédure de dépassement;
13 | | Mise_à_jour_Positions() ;
14 | fin
15 fin
```

---

1. En premier lieu l'agent du véhicule entamant la procédure de dépassement vérifie si aucun véhicule de derrière n'a entamé la même procédure, en consultant les messages reçus ou la position des agents se trouvant derrière le véhicule.
2. Si la voie est libre, l'agent informe automatiquement les véhicules autour de la procédure de dépassement.

Une fois que le message de dépassement soit transmis au véhicule du devant, il vérifie s'il n'y a aucun véhicule devant qui gêne :

- Si oui l'agent (entamant la procédure de dépassement) lance le processus de prédiction de collision pour vérifier si le dépassement va mener à une situation de collision ou pas. Son action dépendra du résultat du processus.
- Sinon il valide la procédure.

### **3.4.2 Circulation à double sens**

La circulation en double sens est un type de circulation routière qui impose aux conducteurs de rouler sur une route avec des voies de sens opposé. C'est un type de circulation routière où les accidents sont les plus fréquents à cause des dépassements mal gérés.

Durant ce type de circulation nous nous intéresserons qu'au cas de dépassement, les autres cas étudiés dans les sens unique sont similaires et sont parfaitement applicables sur ce type de circulation.

#### **3.4.2.1 Dépassement**

Dépasser un véhicule sur une route à double sens nécessite de se placer momentanément sur la voie opposée, en contre sens de la circulation. Pour réaliser ce dépassement, les mêmes étapes que le dépassement dans une route à sens unique sont appliquées, cette fois ci en vérifiant la distance et la vitesse des véhicules venant en sens inverse, en plus de ceux venant de derrière [14].

Comme présenté dans l'algorithme 5, la procédure de dépassement dans ce type circulation routière est une extension de la procédure de dépassement en sens unique. La décision passe par la négociation entre 3 entités, à savoir : Le véhicule initiant la procédure de dépassement (le processus), le véhicule dépassant et en fin le véhicule venant du sens inverse. Si aucun accord n'est trouvé entre Le véhicule initiant la procédure de dépassement et le véhicule venant du sens inverse, le processus de dépassement est annulé.

Notre algorithme est présenté comme suit :

---

**Algorithme 5** : Processus de dépassement à double sens

---

**Données** : Agents,voitures, Environnement

```
1 Confirmation ou pas de la procédure par le véhicule de devant;
2 si Voie libre alors
3   | Permettre la procédure de dépassement;
4   | Mise_à_jour_Positions() ;
5 sinon
6   | Transmission_messages(Véhicule_Sens_opposé, Message_Dépassement);
7   | Lancement de la procédure de prédiction de collision par les deux agents;
8   | si Deux prédiction menant à aucune collision alors
9     | Permettre la procédure de dépassement;
10    | Mise_à_jour_Positions() ;
11    | sinon
12    | Bloquer la procédure();
13    | fin
14 fin
```

---

### 3.5 Conclusion

A travers ce chapitre, nous avons proposé des solutions permettant de répondre à la problématique principale du travail à savoir "*Évitement des collisions dans le système de transports*", en procédant par la conception et la modélisation des différentes entités du système. Nous avons défini les différents modules ajoutés ainsi que différents algorithmes lui permettant de faire des traitements. En fin nous avons terminé en abordant les solutions proposées en s'appuyant sur la modélisation des différentes interactions et les algorithmes élaborés.

# Simulation et résultats

## 4.1 Introduction

Durant ce chapitre nous mettrons en application la conception proposée dans le chapitre 3 en développant notre simulateur 3D en temps réels en employant différents outils que nous allons présenter, puis nous allons présenter les différents résultats liés à l'évitement des collisions dans les systèmes de transports dans les sens uniques.

## 4.2 Déroulement de la simulation

Notre simulation passe par l'application des différents algorithmes proposés dans le chapitre 3 dans un simulateur en suivant des étapes définies permettant le bon déroulement de la simulation.

Dans cette section nous présenterons tout d'abord le simulateur que nous avons développé et qui nous a permis de faire nos tests. Nous justifierons pourquoi nous avons créé notre propre simulateur alors qu'il existe des simulateurs déjà développés dans la littérature, puis nous présenterons les différentes étapes de simulations que nous avons suivies pour parvenir à avoir des résultats que nous présenterons dans la section suivante.

### 4.2.1 Présentation du simulateur VANET réalisé

Le développement d'un simulateur est une tâche rude, plusieurs critères doivent être pris en considération de telle sorte à simuler parfaitement la réalité, à savoir l'environnement, les variables prises en compte dans notre simulation, etc. mais plusieurs raisons nous ont poussées à développer notre propre simulateur, parmi les plus importantes :

- Avoir une liberté de développement.
- Ne plus se soucier des différentes licences d'utilisations des autres simulateurs.
- Enrichir le simulateur par des fonctionnalités que nous jugeons nécessaires par rapport à nos besoins et les supprimer si nous jugeons que ça a un impact sur les résultats (Fonctionnalité lourde et gourmande en ressources).



- L'importance d'utilisation du langage de programmation JAVA combiné avec la plateforme JADE. Dans la littérature il n'existe pratiquement pas des simulateurs programmés en JAVA combinant à la fois des technologies multi-agents.
- Enrichir la littérature en proposant notre propre simulateur.
- C'est gratuit !

Nous définissons notre simulateur, comme un programme permettant de simuler le comportement des véhicules intelligents dans un monde en temps réels suivant le modèle microscopique, toute en utilisant la force des SMA dans la prise de décision selon différentes situations que nous pourrions retrouver dans un environnement routier.

Le paramétrage des différents paramètres cinématiques se fait entièrement par ligne de code en JAVA (Voir Figure 4.1), permettant ainsi l'extension du simulateur par un développeur ou chercheur selon ses besoins et les paramètres qu'il souhaite prendre en considération sans se soucier de l'existence d'une interface et sa modification ce qui peut se révéler lourd et coûteux en terme de temps de développement.

```
public Cinematique(float positionZ, float positionX, float vitesse, float vitesseMax, float directionRoues,
                  float directionVehicule, float acceleration, float freinage) {

    this.positionX = positionX;
    this.positionZ = positionZ;
    this.vitesse = vitesse;
    this.vitesseMax = vitesseMax;
    this.acceleration = acceleration;
    this.freinage = freinage;
    this.directionRoues = directionRoues;
    this.directionVehicule = directionVehicule;
}
```

FIGURE 4.1 – Saisies des paramètres cinématiques

Une fois le programme exécuté, l'interface du simulateur s'affiche permettant l'affichage du monde et le contrôle du véhicule concerné à partir des touches du clavier (Voir Figures 4.2 et 4.3) .

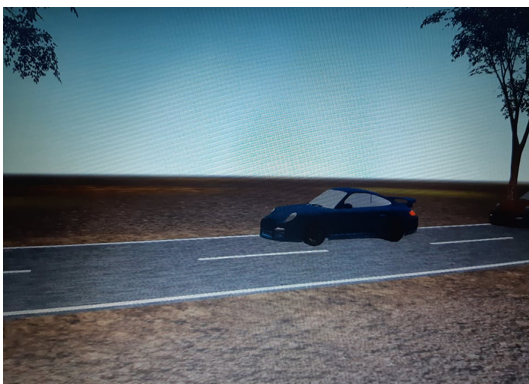


FIGURE 4.2 – Monde du simulateur et véhicules



FIGURE 4.3 – Environnement de simulation

#### 4.2.1.1 Outils de développement et environnement utilisés

Pour développer notre simulateur nous avons utilisées plusieurs outils existants qui nous ont facilité son développement, nous pouvons citer :

- **Langage de programmation JAVA** : Le langage de programmation utilisé est le **JAVA** qui est à la fois un langage de programmation orienté objet. C'est l'un des langages les plus récents et les plus souvent utilisés. Notre choix s'est porté sur l'usage du JAVA pour l'implémentation de notre solution, sa portabilité, sa robustesse et sa simplicité, ceci représente un critère de taille quand il s'agit d'un développement d'une application 3D et à base des SMA tel l'objet de notre proposition.
- **Environnement de développement Eclipse** : **Eclipse** est l'environnement de développement utilisé dans notre travail, cet environnement est un projet, décliné et organisé en un ensemble de sous-projets de développement logiciel. Lancé par la fondation Eclipse, il vise le développement d'un environnement de production de logiciels libre qui se veut extensible, universel et polyvalent [34].

Parmi les grandes réussites de l'open-source, Eclipse figure dans ce cercle, il vise notamment le support de tout langage de programmation, est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs de logiciels et sociétés de services.

En principe nous avons choisi Eclipse car c'est un **IDE** « *Integrated Development Environment* » open-source simple et facile à utiliser et aussi pour sa popularité au sein de la communauté des développeurs. La figure suivante montre son interface.

- **Plateforme JADE** : Les systèmes multi-agents peuvent être réalisés avec n'importe quel langage de programmation, mais plus particulièrement avec les langages orientés objet vu que le concept de l'agent est très proche de celui de l'objet. Les deux concepts partagent plusieurs propriétés telles que l'encapsulation, l'héritage et l'échange de messages, en revanche les agents sont dotés d'autres caractéristiques que l'objet ne possède pas à savoir l'autonomie, le raisonnement et la sociabilité. Un agent peut posséder différents comportements, il est concrètement représenté par un thread de contrôle JAVA.

Il existe plusieurs plateformes dédiées aux SMA. Dans ce travail, nous avons opté pour la plateforme **JADE** « *Java Agent DEvelopment Framework* ». Cette dernière est une plateforme multi-agents créée par le laboratoire *TILAB* [38]. Elle permet le développement de systèmes multi-agents et est implémentée entièrement en JAVA, le choix de la plateforme JADE est dû d'une part à sa gestion de l'interopérabilité et d'autre part à sa souplesse et sa flexibilité. Cette plateforme multi-agents peut être distribuée sur plusieurs machines et pas nécessairement sur le même système d'exploitation, les configurations peuvent être modifiées au démarrage des agents en les déplaçant d'une machine à une autre, ce qui permet une très grande portabilité des agents.

JADE, possède un middleware qui facilite le développement des SMA [38], contient (voir Figure 4.4) :

- **Un environnement d'exécution** : l'environnement où les agents peuvent vivre et évoluer. Cet environnement doit être activé pour pouvoir lancer des agents.
- **Une librairie de classes** : que les développeurs utilisent pour implémenter en code leurs agents.
- **Une suite d'outils graphiques** : qui facilitent la gestion et la supervision de la plateforme des agents.

Chaque instance de JADE est appelée conteneur et celui-ci peut contenir plusieurs agents. Ainsi, un ensemble de conteneur constitue une plateforme et chaque plateforme doit contenir un conteneur spécial appelé main-container et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement. Le main container se distingue des autres conteneurs car il contient toujours deux agents spéciaux appelés AMS et DF qui se lancent automatiquement avec le main-container.

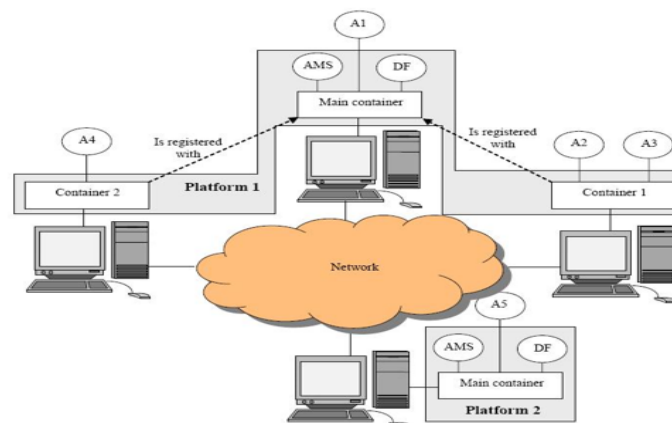


FIGURE 4.4 – Composition de la plateforme JADE [38]

- **JMonkeyEngine** : **jME** « *jMonkey Engine* » est un moteur de jeu libre spécialement conçu pour le développement 3D moderne comme il utilise extensivement la technologie de shader. jME est écrit en Java et utilise **LWJGL** « *Lightweight Java Game Library* » pour son rendu par défaut (un autre système de rendu basé sur **JOGL** « *Java OpenGL* » est disponible) [5].
- **Photoshop** : Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 par Adobe, il est principalement utilisé pour le traitement des photographies numériques, mais sert également à la création d'images [12].

## 4.2.2 Étapes de simulation

Nous définissons dans cette section les différentes étapes de simulation.

### 4.2.2.1 Création de la map

La création de la map revient à créer les différents endroits nous permettant de simuler nos solutions suivant différentes situations, les maps peuvent être extraites à partir des images satellites fournies par des services tels que *GoogleMaps*, ou créer à partir des logiciels de création tel que *Photoshop*.

Nous avons procédé à la création de la map en utilisant un logiciel de création à savoir Photoshop, en premier lieu nous avons créé une AlphaMap permettant de définir les contours du terrain et le placement des textures (Chaque couleur représente une texture que nous pouvons appliquer). Comme exemple sur la Figure 4.5 le vert représente la route, le rouge les champs (Herbe, terrain, etc.) et le bleu pour tout autre ajout sur l'environnement tels que les trottoirs par exemple

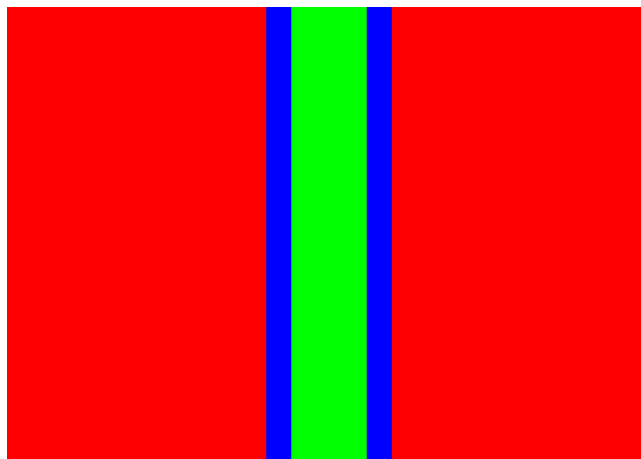


FIGURE 4.5 – AlphaMap

Nous avons ajouté à cela une HeightMap qui représente une cartographie des sols surélevé permettant la création des montagnes par exemple (Voir Figure 4.6).

- **Blanc** : Sol élevé pouvons représenter une montagne par exemple.
- **Noir** : Sol plat.

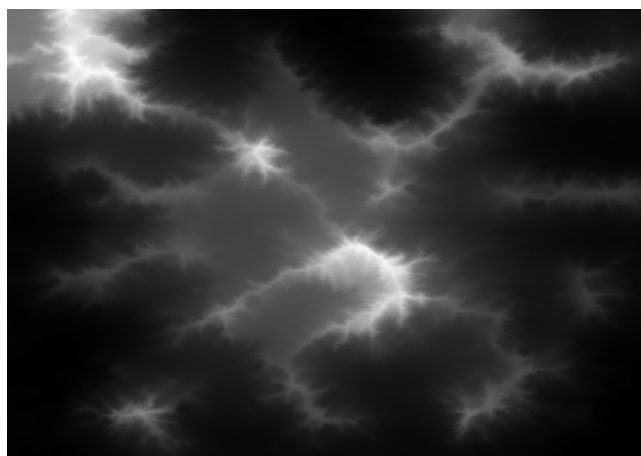


FIGURE 4.6 – HeightMap

#### 4.2.2.2 Simulation de modèle de mobilité et trafic

Un modèle de mobilité reflète le comportement et les déplacements des entités mobiles dans un réseau routier, où le but est de représenter au mieux les conditions de cette mobilité dans un contexte particulier du monde réel [21].

Le comportement des véhicules dans notre simulation, tend vers un comportement qui respecte le code de la route, avec des exceptions menées par les conducteurs, c'est-à-dire :

- Dépassement dangereux.
- Non-respect de la distance de sécurité.
- Etc.

#### 4.2.2.3 Configuration du moyen de communication

La communication passe par l'application d'un protocole de communication permettant la gestion de l'interaction entre les différentes entités du système.

L'utilisation des SMA permet à l'application l'usage d'un protocole de communication efficace à savoir les ACLMessages permettant la gestion des interactions des différents agents (Voir Figure 4.7).

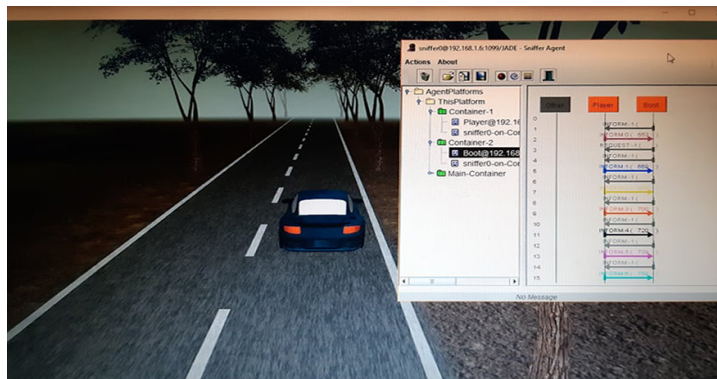


FIGURE 4.7 – Interface représentant la communication entre deux véhicules par envois de messages

#### 4.2.2.4 Simulation

Exécution du simulateur suivant différents paramètres et type de circulation. Nous avons choisi de simuler le comportement permettant l'adaptation de la vitesse du véhicule selon un type de circulation en sens unique selon différents paramètres cinématiques, en considérant les 4 hypothèses suivantes :

- Pas d'effets aérodynamiques (frottement = 0).
- Pas de transfert de charge.
- Aucune latence dans la communication inter-véhiculaires.
- Le temps ne change pas !

Nous considérons deux véhicules  $A$  et  $B$  présentés dans la Figure 4.8, le véhicule  $B$  devance le véhicule  $A$ . Le véhicule  $A$  doit adapter sa vitesse durant toute la simulation de telle sorte à respecter une distance de sécurité relativement suffisante qu'il lui permettra d'éviter une collision avec le véhicule  $B$  suivant leurs différents paramètres cinématiques (voir Tableau 4.1) :



FIGURE 4.8 – Présentation du véhicule  $B$  en marron et véhicule  $A$  en bleu

Véhicules	Véhicule $A$	Véhicule $B$
Paramètres		
Accélération	$a_a = 3m/s^2$	$a_b = 2m/s^2$
Vitesse	$v_a$	$v_b$
Vitesse maximale	$v_{max_a}$	$v_{max_b} = 20m/s$
Vitesse initiale	$v_{initial_a}$	$v_{initial_b}$
Force de freinage (décélération)	$f_a = 10m/s^2$	$f_b = 10m/s^2$
Distance de séparation	$d$	

TABLE 4.1 – Définition des paramètres cinématiques des véhicules

#### Remarque

Les paramètres auxquels nous avons fixés les valeurs dans le tableau 4.1, resterons inchangés durant toute la simulation.

## 4.3 Discussion des résultats de simulation

Dans cette section, nous discuterons des différents résultats obtenus durant la simulation en variant à chaque partie les valeurs des paramètres cinématiques des deux véhicules non fixés dans le tableau 4.1.

### 1. Pour $v_{initial_a} = 0m/s$ , $v_{max_a} = 30m/s$ , $v_{initial_b} = 10m/s$ et $d = 200m$ :

Le véhicule *A* démarre avec une vitesse initiale de  $0m/s$  pour une distance séparant le véhicule *B* de  $200m$  qui lui est déjà à une vitesse de  $10m/s$  au moment où le véhicule *A* démarre.

Nous constatons dans ce cas de figure, que le véhicule *A* arrive à adapter parfaitement sa vitesse de telle sorte qu'il passe d'une vitesse de  $30m/s$  au moment où il décide de freiner à  $20m/s$  qui est la vitesse du véhicule *B* à cet instant, tout en laissant une distance de sécurité de  $48m$ , qui peut s'expliquer :

- La distance de sécurité, qui représente deux fois la vitesse du véhicule *B*.
- Le temps de réponse relativement court pour un agent intelligent.

### 2. Pour $v_{initial_a} = 0m/s$ , $v_{max_a} = 30m/s$ , $v_{initial_b} = 0m/s$ et $d = 75m$ :

Les deux véhicules démarrent en même temps avec une vitesse initiale de  $0m/s$  pour une distance les séparant de  $75m$ .

Nous constatons dans ce cas de figure, que le véhicule *A* arrive à adapter parfaitement sa vitesse de telle sorte qu'il passe d'une vitesse de  $24m/s$  au moment où il décide de freiner à  $16.3m/s$  qui est la vitesse du véhicule *B* à cet instant.

Au moment où le véhicule *A* décide d'arrêter son freinage sa vitesse est de  $17.7m/s$ , et la vitesse du véhicule *B* est de  $18m/s$  tout en laissant une distance de sécurité de  $38.3m$ , qui peut s'expliquer :

- La distance de sécurité, qui représente deux fois la vitesse du véhicule *B*.
- Le temps de réponse relativement court pour un agent intelligent.
- Une distance de freinage plus courte, au vu d'une petite différence de vitesse entre les deux voitures.

Nous constatons par ailleurs, que le résultat est plus proche d'une distance de sécurité tels que le code de la route l'a définie. Ce qui peut s'expliquer par la faible vitesse avec lesquelles les deux voitures roulaient.

**3. Pour  $v_{\text{initial}_a} = 0m/s$ ,  $v_{\text{max}_a} = 30m/s$ ,  $v_{\text{initial}_b} = 0m/s$  et  $d = 10m$  :**

Les deux véhicules démarrent en même temps avec une vitesse initiale de  $0m/s$  pour une distance les séparant de  $10m$ .

De par une distance de séparation plus petite, nous constatons dans ce cas de figure que plusieurs évènements se passent de tels sorte que le véhicule  $A$  maintient toujours la distance de sécurité.

- a. A  $t = 2s$  le véhicule  $A$  se trouve à une vitesse de  $6m/s$  et le véhicule  $B$  à une de vitesse de  $4m/s$  pour une distance les séparant de  $6m$ .
- b. L'agent du véhicule  $A$  bloque l'accélération du véhicule  $A$  de telle sorte que le véhicule  $B$  prenne une distance suffisante pour respecter la distance de sécurité.
- c. L'accélération n'est débloquée qu'à la demande du conducteur en relâchant la pédale d'accélération ou par une petite action de freinage de telle sorte à assurer que le conducteur ne puisse pas accélérer directement sans laisser une distance suffisante.

L'agent arrive parfaitement à maîtriser la situation, même si à un moment l'action de freiner pour attendre que le véhicule  $B$  prenne une distance suffisante peut représenter un handicap, mais ici nous privilégions avant tout la sécurité des passagers !

**4. Pour  $v_{\text{initial}_a} = 25m/s$ ,  $v_{\text{max}_a} = 30m/s$ ,  $v_{\text{initial}_b} = 20m/s$  et  $d = 40m$  :**

A l'instant initial la vitesse des véhicules  $A$  et  $B$  sont respectivement à  $25m/s$  et  $20m/s$  pour une distance les séparant de  $30m$ .

De par une distance de séparation plus petite, et des vitesses élevées nous constatons dans ce cas de figure que le véhicule  $A$  n'arrive pas à adapter parfaitement sa vitesse de tels sorte qu'il préserve une distance de sécurité qui devait être de  $40m$  or nous avons obtenu après freinage du véhicule  $A$  une distance de  $11.5m$ , plusieurs causes peuvent mener à ce résultat :

- a. Une force de freinage petite.
- b. La réception des informations, reçues en retard, dû à l'entrée tardive du véhicule  $B$  ou  $A$  dans leurs champs respectifs réseau.
- c. Une distance de séparation relativement petite.
- d. Des vitesses trop grandes.



5. Pour  $v_{\text{initial}_a} = 25m/s$ ,  $v_{\text{max}_a} = 30m/s$ ,  $v_b = 0$  et  $d = 100m$  :

A l'instant initial la vitesse du véhicule *A* est de  $25m/s$ , le véhicule *B* est complètement à l'arrêt, pour une distance les séparant de  $30m$ .

De par l'échange de messages d'alerte, le véhicule *A* anticipe l'arrêt du véhicule *B*, et calcule parfaitement la position du freinage pour éviter une collision, la distance à l'arrêt est de  $5m$  (Voir figure 4.9)



FIGURE 4.9 – Cas d'arrêt

Nous constatons de par les résultats obtenus, que l'agent arrive à adapter sa vitesse parfaitement dans la majorité des cas où une distance de séparation est suffisamment grande lui permettant d'anticiper et de faire le calcul adéquat, mais dès que la distance séparant deux véhicules est relativement courte l'agent peine à des grandes vitesses, qui peut s'expliquer par différentes raisons liées à la configuration du véhicule tels que la force de freinage, ou liée aux matériels réseaux tels que la portée réseaux des infrastructures et la latence réseaux.

## 4.4 Conclusion

Durant ce chapitre nous avons présenté le simulateur que nous avons développé, ainsi que les étapes de simulations, puis nous avons terminé par une mise au point des différents résultats obtenus par l'application de nos solutions proposées dans le chapitre 3, dans le cas d'adaptation de la vitesse d'un véhicule lui permettant de respecter la distance de sécurité pour éviter une collision dans les systèmes de transports.

## Conclusion générale et perspectives

L'intérêt grandissant, au sujet des systèmes d'évitement de collisions comme étant une composante essentielle dans la conception des véhicules autonomes de futures générations, a motivé la communauté scientifique et industrielle à développer des véhicules équipés de ces systèmes.

Dans ce projet de fin d'études, nous avons montré que l'utilisation des SMA demeure une technique judicieuse pour le partage de données et transmission d'informations. Par ailleurs, l'interaction et la communication entre les agents permettent une meilleure organisation ainsi qu'une meilleure adaptabilité. Rappelons que notre contribution s'est résumée d'une part à la proposition d'une architecture de communication SMA, et d'autre part à la conception et l'implémentation d'un simulateur de véhicules autonomes de degrés restreint essentiellement pour l'évitement de collisions dans les systèmes de transports.

Plusieurs simulations ont été effectuées en variant divers paramètres. En ressortant avec des résultats satisfaisants dans chaque cas simulés, le système prend toujours le contrôle de la vitesse du véhicule dans le cas où une collision potentielle s'impose.

Comme perspectives, il serait intéressant d'implémenter et de coopérer entre les SMA et de leurs adopter des algorithmes liés aux réseaux de neurones, de façon à ce que le système pourrait même aller à nous avertir d'une future collision loin dans le temps grâce aux expériences vécues, ajouter a cela l'amélioration de nos algorithmes de prédiction de tels sorte a prendre en compte différents cas imprévus que nous n'avons pas réussi à considérer durant notre analyse et conception et en fin améliorer le simulateur, lui ajoutant des technologies de communications VANET lui permettant de simuler au mieux les risques réseaux liés a la communication telles que la latence et la perte de connexion.

# Bibliographie

- [1] <http://res-nlp.univ-lemans.fr/NLP-C-M01-G01/co/Contenu.html>, Consulté le 1 Septembre 2020.
- [2] <https://www.openstreetmap.org>, Consulté le 15 Juillet 2020.
- [3] <https://fr.wikipedia.org/wiki/Cam>, Consulté le 18 Août 2020.
- [4] [https://fr.wikipedia.org/wiki/GPS \(assistant de navigation\)](https://fr.wikipedia.org/wiki/GPS_(assistant_de_navigation)), Consulté le 18 Août 2020.
- [5] <https://fr.wikipedia.org/wiki/JMonkey-Engine>, Consulté le 18 Septembre 2020.
- [6] <https://sumo.dlr.de/docs/FAQ.html>, Consulté le 19 Août 2020.
- [7] <https://mbamci.com/6-niveaux-voiture-autonome>, Consulté le 21 Décembre 2019.
- [8] <https://www.egis-group.com>, Consulté le 21 Mars 2020.
- [9] <https://fr.wikipedia.org/wiki/Systeme-multi-agents>, Consulté le 25 Août 2020.
- [10] <https://fr.wikipedia.org/wiki/Distance-d'arret>, Consulté le 25 Novembre 2019.
- [11] <https://fr.wikipedia.org/wiki/Circulation-en-sensunique>, Consulté le 25 Novembre 2019.
- [12] <https://fr.wikipedia.org/wiki/Adobe-Photoshop>, Consulté le 25 Septembre 2020.
- [13] <https://www.zerotracas.mma/news/4536/adapter-sa-vitesse-le-temps-de-reaction.html>, Consulté le 3 Septembre 2020.
- [14] <https://www.lepermislibre.fr/permis-conduire/conseils-conduite/croisement-depassement-code-route>, Consulté le 3 Septembre 2020.
- [15] A.Aw and M.Rasclé. "resurrection of second order models of traffic flow". *SIAM Journal on Applied Mathematics*, pages 916–938, 2000.
- [16] A.Kahina. "Modélisation Et Etude De Performances Dans Les Réseaux VANET". PhD thesis in Computer Science, Université de Technologie de Belfort Montbéliard, 2012.
- [17] B.Djamila. Technologie des agents. Technical report, Université de Béjaïa.

- [18] B.Djamila. " *La Négociation Dans les Systèmes Multi-agents : Modèles, Protocoles et Formalismes* ". PhD thesis in Computer Science, Université de Constantine 2, 2016.
- [19] Benkirane, Said, Jadir, and Ahmed. " adapted speed mechanism for collision avoidance in vehicular ad hoc networks environment ". *International Journal of Advanced Computer Science and Applications*, 9, 01 2018.
- [20] C.Dimon. " *Contributions à la modélisation et la commande des réseaux de trafic routier* ". PhD thesis in Computer Science, Ecole Centrale de Lille, 2012.
- [21] D.Bektache. " *Application et Modélisation d'un protocole de communication pour la sécurité routière* ". PhD thesis in Computer Science, Badji Mokhtar Annaba, 2014.
- [22] D.C.Gazis, R.Herman, and R.Rothery. " non-linear follow -theleader models of traffic flow ". *Opns*, 1961.
- [23] Institut Français d'information juridique. " *Code de la route* ". 2020.
- [24] F.Karnadi, Z.Mo, and K.Lan. " rapid generation of realistic mobility models for vanet ". In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 2506–2511, 2017.
- [25] G.Costeseque. " modélisation du trafic routier : passage du microscopique au macroscopique ". Master's thesis, ENTPE, 2011.
- [26] H.Moustafa and G.Bourdon. " vehicular networks deployment view : Applications, deployment architectures and security mean ". *Ubiquitous Computing and Communication Journal*, 2008.
- [27] H.M.Zhang. " a non-equilibrium traffic model devoid of gas-like behavior ". *Transportation Research*, pages 275–190, 2002.
- [28] H.M.Zhang and W.L.Jin. " kinematic wave traffic flow model for mixed traffic ". *Transportation Research*, pages 197–204, 2002.
- [29] I.Salhi, S.M.Senouci, and Y.Gourhant. " collecte d'informations dans un réseau de véhicule ". Master's thesis, Badji Mokhtar Annaba, 2008.
- [30] Jennings and R. Nick. " commitments and conventions : The foundation of coordination in multi-agent systems ". *The Knowledge Engineering Review*, 8(3) :223–250, 1993.
- [31] J.Ferber. " *Les Systèmes Multi-Agents. Vers une intelligence collective. Interéditions* ". 1995.
- [32] J.Petit. " *Cout de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaire* ". PhD thesis in Computer Science, Toulouse 3 Paul Sabatier, 2011.
- [33] J.P.Lebacque. " the godunov scheme and what it means for first order traffic flow models ". pages 647–677, 1996.

- [34] K.Darryl. " *Reviews and analysis in programming environments and developer tools* ", 2005.
- [35] K.Moghraoui. " *Gestion de l'anonymat des communications dans les réseaux véhiculaires ad hoc sans fil (VANETs)* ". PhD thesis in Computer Science, Québec À Trois-Rivières, Juillet 2015.
- [36] K.Wissam. " *Contribution à la modélisation graphique de système de systèmes* ". PhD thesis in Computer Science, Université lille 1, 2012.
- [37] L.A.Pipes. " an operational analysis of traffic dynamics ". In *Journal of Applied Physics*, volume 24, pages 274–281, 1953.
- [38] L.B.Fabio, C.Giovanni, and G.Dominic. " *Developing Multi-agent System with JADE* ". 2007.
- [39] M.Guériau. " *Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic* ". PhD thesis in Computer Science, Université de Lyon, 2016.
- [40] M.Guériau, R.Billot, F.Armetta, S.Hassas, and N.E.El-Faouzi. " un simulateur multiagent de trafic coopératif ". 2015.
- [41] M.H.Lighthill and G.B.Whitham. " on kinematic waves 2 : a theory of traffic flow on long crowded roads ". volume 229, pages 317–345. *Proceedings of the Royal Society*.
- [42] O.Florence, E.Dieudonné, P.Steve, I.Nelson, Y.Lionel, and A.Zongo. " commande neuro-floue pour le suivi de trajectoire et l'évitement d'obstacles dans un véhicule autonome ". *Afrique Science*.
- [43] P.I.Richards. " shockwaves on the highway ". *Operations Research*, pages 42–51, 1956.
- [44] R.Hermana, E.W.Montroll, R.B.Potts, and R.Rothery. " traffic dynamics : Analysis of stability in car-following ". *Opns*, 1959.
- [45] R.Jayakrishnan, H.S.Mahmassani, and T.Y.Hu. " an evaluation tool for advanced traffic information and management systems in urban networks ". *Transportation Research*, pages 129–147, 1994.
- [46] S.Labidi and W.Lejouad. " de l'intelligence artificielle distribuée aux systèmes multi-agents ". Technical report, INRIA, 1993.
- [47] S.Paquet. " *Distributed Decision-Making and Task Coordination in Dynamic, Uncertain and Real-Time Multiagent Environments* ". PhD thesis in Computer Science, Université de LAVAL QUEBEC, 2006.
- [48] S.P.Hoogendoorn and P.H.L.Bovy. " state of the art of vehicular traffic flow modelling ". volume 215, pages 283–303, Août 2001.
- [49] S.Russell, , P.Norvig, and F.Popineau. " *Intelligence artificielle : Avec plus de 500 exercices* ". Pearson Education. Pearson, 2010.

- [50] T.D.Gillespie, SAE International (Society), and Society of Automotive Engineers. " *Fundamentals of Vehicle Dynamics* ". Premiere Series Bks. Society of Automotive Engineers, 1992.
- [51] Y.Qian, K.Lu, and N.Moayeri. " ieee global telecommunications conference publication ". In IEEE, editor, *A Secure Vanet Mac Protocol For DSRC Applications*, 2008.

# RÉSUMÉ

Durant ces dernières années le nombre de véhicules possédés par foyer ne cesse de croitre, et ça implique inévitablement un trafic routier augmenté. Ainsi un taux d'accident de plus en plus élevé causé principalement par les humains à savoir prise d'alcool, fatigue, somnolence et le non-respect de la distance de sécurité.

Avec l'apparition des réseaux sans fil et l'avancement des technologies de communications et de l'information cela a induit une interaction entre plusieurs entités du réseau routier à savoir, les véhicules et les infrastructures, et le partage des informations les plus pertinentes relatives au trafic.

Notre contribution a montré que l'usage des SMA reste une méthode judicieuse pour le partage de données et la transmission d'informations, par ailleurs, les agents permettent une meilleure organisation ainsi qu'une meilleure adaptabilité. Dans un sens restreint, notre apport s'est résumé d'une part à la proposition d'une architecture de communication SMA, et d'autre part, à la conception d'un simulateur de véhicules autonomes de degré restreint, conçu essentiellement pour l'évitement de collisions dans les systèmes de transport routier.

**Mots clés** : Distance de sécurité ; Système multi-agent ; Évitement de collisions ; Réseau routier ; Véhicules intelligents

# ABSTRACT

In recent years the number of vehicles owned per household continues to grow, and that inevitably implies increased road traffic. This leads to high accident rate caused mainly by humans at know alcohol intake, fatigue, drowsiness and non-compliance with the distance of security.

With the emergence of wireless networks and the advancement of communications and information this has led to an interaction between several entities of the road network namely, vehicles and infrastructure, and sharing the most relevant traffic information.

Our contribution has shown that the use of SMA remains a sound method for the sharing of data and the transmission of information, in addition, agents allow for better organization as well as better adaptability. In a limited sense, our contribution has been summed up on the one hand to the proposal of an SMA communication architecture, and on the other hand, to the design of a limited degree autonomous vehicle simulator, designed mainly for collision avoidance in transport systems road.

**Key words** : Safety distance ; Multi-Agent System ; Collision Avoidance ; Road Network ; Smart Vehicles ;