Méthodes et outils pour la conception de Systèmes de Transport Intelligents Coopératifs

Jérémy Sobieraj

▶ **To cite this version:**

Jérémy Sobieraj. Méthodes et outils pour la conception de Systèmes de Transport Intelligents Coopératifs. Système multi-agents [cs.MA]. Université Paris-Saclay; Université d'Evry-Val-d'Essonne, 2018. Français. NNT : 2018SACLE038 . tel-02042591

HAL Id: tel-02042591
https://hal.science/tel-02042591

Submitted on 20 Feb 2019

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

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

NNT : 2018SACLE038

Thèse de doctorat

Méthodes et outils pour la conception de Systèmes de Transport Intelligents Coopératifs

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université d'Evry-Val-d'Essonne, Laboratoire IBISC

École doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Evry, le mercredi 07 novembre 2018, par

M. JÉRÉMY SOBIERAJ

Composition du Jury :

M. Alexis Drogoul Directeur de Recherches Institut de Recherche pour le Développement (UMI 209 UMMISCO)	Rapporteur
Mme Marie-Pierre Gleizes Professeure Université Toulouse III Paul Sabatier (IRIT)	Rapporteure
M. Vincent Chevrier (président du jury) Professeur Université de Lorraine (LORIA)	Examinateur
M. Saïd Mammar Professeur Université d'Evry-Val-d'Essonne (IBISC)	Examinateur
M. Guillaume Hutzler Maître de Conférences HDR Université d'Evry-Val-d'Essonne (IBISC)	Directeur de thèse
Mme Hanna Klaudel Professeure Université d'Evry-Val-d'Essonne (IBISC)	Co-Directrice de thèse

Remerciements

J'attendais depuis longtemps ce moment où j'allais commencer à écrire ces mots. En effet, cela marque pour moi un point final sur l'écriture de mon manuscrit de thèse et signifie que je suis allé jusqu'au bout. Cet exercice immortalise les moments, les souvenirs et les personnes qui m'ont apporté et soutenu pendant cette longue période de trois ans mais quoi qu'il arrive, cela restera avant tout dans mes pensées à vie.

Tout d'abord, je tiens à remercier les professeurs Alexis DROGOUL et le professeur Marie-Pierre GLEIZES ainsi que les professeurs Vincent CHEVRIER (président du jury) et Saïd MAMMAR d'avoir accepté de faire partie des membres du jury respectivement en tant que rapporteurs et examinateurs. Je vous remercie tous pour l'intérêt que vous avez eu pour mes travaux de thèse. Merci également pour tous ces retours qui sont très précieux pour moi, d'une part à travers vos rapports, d'autre part à travers vos questions très pertinentes durant la soutenance.

J'arrive à ce moment où je dois parler de toi, Guillaume HUTZLER, mon directeur de thèse. Un grand merci de m'avoir fait confiance pendant cette grande aventure, en espérant que tu ne le regrettes pas. Merci pour ta simple présence, ta gentillesse, tes commentaires pertinents, tes corrections pointilleuses et ton scepticisme permanent qui m'a permis d'avancer pendant ces trois ans. Et un merci particulier sur cette phrase que tu m'as prononcée il y a huit ans déjà lors des journées portes ouvertes à l'université d'Evry : "Tu as le niveau en maths ? Alors fonce !", qui m'ont finalement convaincu d'aller dans cette université sans réellement savoir pourquoi.

C'est à ton tour, Hanna KLAUDEL, ma co-directrice de thèse. Un grand merci également d'avoir cru en moi tout le long. Merci pour ta bonne humeur, ta bienveillance, ton exigence sans faille et ton côté maternel qui m'a souvent rassuré. Merci également de m'avoir donné envie pendant toutes ces années à faire de la recherche à travers tes cours très intéressants et ta grande pédagogie.

Je tiens également à vous faire un dernier message groupé pour vous deux. Pendant ces trois ans, vous avez toujours été là pour me soutenir même dans les moments difficiles. Je sais ô combien j'ai été insupportable et je pense encore au nombre de fois où je vous ai fait peur lors de mes périodes de doutes. Je tiens à me faire pardonner pour ça. Vous avez été plus que des encadrants pour moi tout au

long de cette thèse et je continuerai à vous admirer pour la passion que vous avez pour ce métier et pour ce que vous êtes tout simplement.

Merci à l'actuel directeur du laboratoire le professeur Franck DELAPLACE d'avoir finalement été plus qu'un directeur, une personne bienveillante qui pousse tous les membres du laboratoire à aller toujours plus loin. Merci pour ta grande gentillesse, tes mises en scènes théâtrales pendant les repas du midi et ton humour particulier qui ne fait rire que moi sans doute.

Merci à l'actuel responsable de l'équipe COSMO (la meilleure bien évidemment) le professeur Franck POMMEREAU qui a toujours été présent dans certaines situations critiques et qui sait mettre en avant les qualités de chacun.

J'en arrive maintenant à toi Serenella CERRITO, professeure au laboratoire également. Je pense qu'on s'est toujours protégés mutuellement contre les moments difficiles et cela a beaucoup renforcé nos rapports. Merci pour cette merveilleuse joie que tu apportes au laboratoire ainsi que tes grandes qualités humaines. Merci tout simplement d'avoir été et d'être encore une amie, très précieuse, pour moi.

Tu m'as suivi depuis le début de mon cursus à l'université d'Evry et sans toi, je ne me serais peut-être pas intéressé à ce magnifique métier d'enseignant-chercheur. Merci à toi Tarek MELLITI, maître de conférences au laboratoire, de m'avoir tant apporté durant toutes ces années, de m'avoir fait confiance lors de mon stage de troisième année de licence. Je n'oublierai jamais ces moments de réflexion passés tous les deux devant le tableau blanc à vouloir révolutionner le monde (des réseaux d'automates booléens).

Pour finir avec l'équipe COSMO, je vais terminer avec le dernier venu. Tu es là que depuis un an mais tu m'as apporté beaucoup de choses qui m'ont vraiment aidé à terminer ma troisième année plus sereinement que prévu. Je parle évidemment de toi Sergiu IVANOV, maître de conférences au laboratoire IBISC. Un énorme merci pour ta gentillesse, ta grande bienveillance et pour cette collaboration dans laquelle j'ai pris beaucoup de plaisir. Je te souhaite le meilleur pour la suite. Tu apporteras beaucoup à ce laboratoire, j'en suis certain.

Pour continuer dans les collaborations, je tiens à remercier infiniment Lydie NOUVELIERE, professeure au laboratoire IBISC. Au delà du travail effectué ensemble, j'ai toujours vu en toi une encadrante cachée qui a toujours su me guider pour aller toujours loin et à avoir confiance en moi. Merci pour cette bienveillance.

Enfin, un merci à Sylvain SENE, actuellement professeur à l'université d'Aix-Marseille, de m'avoir fait également confiance pendant mon stage de troisième année, pour cette collaboration et de m'avoir beaucoup apporté à travers tes grandes qualités de chercheur.

Merci à Pascal PETIT, professeur agrégé à l'université d'Evry, d'avoir toujours été de bon conseil tant sur l'enseignement que sur des décisions à prendre en tant

que moniteur mais aussi pour mon avenir. Tu as toujours été pour moi un grand défenseur des étudiants et ils peuvent te remercier pour ça.

Un merci tout particulier à Jean-Christophe JANODET, professeur au sein de l'équipe AROBAS qui a toujours su être bienveillant notamment en tant qu'ancien directeur du département informatique. Merci également pour ton franc-parler qui m'a beaucoup aidé.

Pour finir avec les professeurs au laboratoire, merci à Jean-Marc DELOSME, également présent à l'équipe AROBAS, de m'avoir fait confiance pour me choisir en tant que chargé de TD pendant mon Master 2, et pour d'autres nombreux TD à essayer de révolutionner les systèmes d'évaluation tant bien que mal. Merci également d'avoir été un très bon référent de thèse avec tes remarques très pertinentes et plus généralement d'avoir été souvent là pendant cette période. Merci aussi à tous les autres professeurs et chercheurs du laboratoire que je n'ai pas cité.

Merci à toutes les secrétaires qui m'ont beaucoup dépanné de certaines démarches administratives parfois lourdes et qui ont brillé par leur efficacité. Merci notamment à vous Stéphanie DRUETTA, Carole LOYER et Patricia ROUSSEAU.

Je tiens à présent remercier particulièrement certaines personnes qui ont été des éléments clés pendant ces trois années. Tout d'abord, Laurent POLIGNY, anciennement ingénieur de recherche au laboratoire IBISC. J'ai beaucoup appris de ta vision de la vie et de tes connaissances sur tout. J'ai vraiment apprécié travailler avec toi cette dernière année de thèse. Merci pour ces grandes discussions philosophiques, ta gentillesse et ta confiance.

Merci beaucoup à Thierry MILLANT, régisseur du bâtiment mais beaucoup plus que cela à mes yeux. Tu as su me booster aux bons moments mais aussi m'aider avec tes prises de recul. Un merci sincère pour tout ce que tu m'as apporté mais aussi pour ce que tu as apporté à l'université et au laboratoire en général. Je sais que je pourrais toujours compter sur toi.

Merci à Murielle BOURGEOIS, secrétaire du laboratoire IBISC. Je n'oublierai jamais ces moments de rires passés à organiser les journées du laboratoire. Merci pour ta gentillesse, ta grande joie de vivre et pour avoir tout fait pour que ces années de thèse soient inoubliables.

Merci beaucoup à Wojciech TUREK, professeur à l'AGH University à Cracovie en Pologne. Lors de mon séjour de recherche, vous avez su me guider pour donner un vrai sens à mon sujet de thèse. J'ai beaucoup apprécié notre collaboration en espérant aller plus loin dans un avenir proche. Merci également au professeur Małgorzata Żabińska de m'avoir accueilli au sein ce de laboratoire. "Dziękuję bardzo !"

Merci beaucoup à Maxime DESBOIS, actuellement en deuxième année de Master en informatique mais aussi le premier stagiaire que j'ai encadré. Merci et encore

bravo pour le travail que nous avons effectué ensemble pendant cette période. Tu iras loin, je n'en doute pas.

Enfin, merci à toi Maxime GUERIAU, docteur en informatique et chercheur associé au Trinity College à Dublin en Irlande. Travaillant sur des thématiques proches, tu as toujours été d'une grande bienveillance et d'une gentillesse rare que ce soit à distance comme lors de notre rencontre en conférence. Je te souhaite le meilleur pour la suite.

J'en arrive à vous les doctorants (devenus docteurs depuis pour certains). Sans vous, la vie de laboratoire et ma thèse seraient complètement différentes et je ne pourrai jamais oublier ces moments passés avec vous. Merci à Elie EL RACHKIDI et Hoby RAKOTOARIVELO pour leur grande aide psychologique notamment en fin de thèse, Junheng CHENG et Peng WU pour cette merveilleuse ambiance que vous avez donnée au laboratoire par votre présence. Félicitations à tous les quatre pour votre diplôme de doctorat. Merci aux actuels doctorants, en particulier Yanting HOU et Yipei ZHANG pour leur gentillesse et sincérité rares, Yantong LI pour ces moments passés au laboratoire pendant la fermeture, Lauriane HUGUET que j'ai rencontré plusieurs fois en conférence avec qui j'ai partagé beaucoup de bons moments, Célia BIANE pour m'avoir supporté pendant trois ans dans le même bureau, Johan ARCILE pour notre collaboration scientifique et nos conversations philosophiques, Ludovic PLATON pour son calme et sa zénitude légendaire et enfin Clément BERTRAND, mon mentor doctorant pour tous ces jeux stupides créés pour se changer les idées mais aussi pour tes remarques pertinentes autour de mon sujet de thèse. Merci également et désolé à tous ceux que j'ai oublié ce citer. Pour certains, le chemin est encore long mais je crois en vous. Bon courage à vous tous !

Les stagiaires, bien que présents pendant une petite période, apportent toujours plus de vie au sein du laboratoire, notamment pendant les repas du midi. Un merci particulier à Gilles BELLIDO, Maria BARGUES RIBERA et Sara AGHAMIRI pour leur bonne humeur permanente au sein de mon bureau, Jean-Baptiste MUNIERES pour ces super bons moments passés en ta présence, William ATGER pour son humour particulier et nos énigmes tordues, Jérémie PARDO pour sa sincérité et sa bienveillance, Samy DAOUD pour nos grandes discussions notamment après 19 heures au laboratoire, Mathieu ARLES et Valentin BOCQUEL pour leur simple présence, et enfin, Colin ZACCOMER pour ta sincérité et pour toutes nos discussions qui m'ont vraiment aidé à tenir le coup pendant ces trois ans, un grand merci particulier pour toi. Une fois de plus, merci et désolé à tout ceux que je n'ai pas cité.

J'aimerais également remercier particulièrement les étudiants de l'association des étudiants d'informatique d'Evry ADIEVE avec qui j'ai passé des moments formidables. Merci à Alizée DOLAIS-PICOT, Simon CALDERAN, Charles-Adrien MICHEL, Alexandre PAGNAT, Geoffroy THIBORD, Magou CAMARA, Axelle ORDONNEAU, Jérémy LAFOSSE, Rémi DELAIRE et tous les autres. Merci d'avoir fait vivre

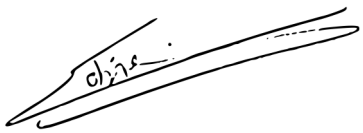
cette association et pourvu que ça dure ! Merci également à l'ensemble des étudiants qui m'ont eu en tant que chargé de TD. Ça a été un grand plaisir de jouer ce rôle pendant mes trois années de thèse.

Le laboratoire IBISC et tous ses membres sont devenus une seconde famille pour moi. Toutefois, je n'oublie pas mes amis qui, pendant ces trois années et même avant, ont toujours su me soutenir. Un merci sincère à Romain CHALUT et Jordi CHARPENTIER pour avoir toujours été là et pour ces super parties de Fortnite qui m'ont permis de me détendre et rigoler avec vous. Alexis MANGANIELLO pour sa grande gentillesse et pour avoir contribué à beaucoup de choses dans ma vie, Delphine LUGON pour cette randonnée inoubliable, Anthony MASSON et toute sa merveilleuse famille pour leur soutien sans faille tout le long, Coralie TANDON et Audrey WALTER pour leur grande sincérité et leur bonne humeur. Un merci particulier à Yoann VANHOESERLANDE pour sa grande sagesse et pour avoir toujours été là. Enfin, un dernier merci à toutes vos moitiés Florence BERTRAND, Kévin BESANCON, Dylan COQUELIN, Mylène DELAVIER, Mélanie DESPOIS, Jérémy GUIGGIA et Maïté JOUVIE qui représentent également beaucoup pour moi.

Cette partie est pour la personne qui partage ma vie depuis plus de deux ans maintenant. Tu es là depuis peu mais tu m'as beaucoup apporté pendant cette période. Tu es un énorme pilier pour moi et je sais que je pourrais toujours compter sur toi. C'est le début d'une grande aventure à deux. Je t'aime. Et toi aussi mon petit chat Leïna, qui n'a sans doute jamais compris ce que je faisais. Tu étais sans doute trop occupée avec tes cartons et tes croquettes. Mais merci beaucoup à toi quand même.

J'en arrive enfin à vous mes chers parents. Une énorme merci papa et maman d'avoir fait de moi ce que je suis aujourd'hui, de m'avoir toujours soutenu dans tout ce que je faisais (même si parfois, vous ne comprenez pas tout). Je ne peux être qu'extrêmement fier, heureux et chanceux d'avoir des parents comme vous, je vous aime. Un grand merci à mes grands-parents maternels mais aussi à ma grand-mère paternelle, partie trop tôt avant le début de ma thèse malheureusement, mais qui me regarde de là-haut et qui j'espère est fière.

Ces trois années, bien qu'éprouvantes, ont été riches en émotions, en rires et en rencontres. Cela est en grande partie grâce à vous. Merci du fond du cœur. . .



A mes parents,

A mes grands-parents,

A ma grand-mère partie trop tôt,

Sommaire

Introduction **1**

1 Positionnement **5**
1.1 Un véhicule autonome 6
1.2 Un véhicule communicant 11
1.3 Un véhicule virtuel 16
1.4 Synthèse 20

2 Un simulateur multi-agent d'analyse comportementale de véhicules **23**
2.1 Choix d'un outil de simulation 24
2.2 Un outil de simulation : GAMA 36
2.3 Contribution : extension de GAMA 41

3 Prise de décision et coopération entre véhicules communicants **55**
3.1 Représentation et évaluation de la dynamique des véhicules 57
3.2 Analyse du changement de voie $MOBIL$ 68
3.3 Un protocole de coopération entre véhicules 74
3.4 Prise en compte de véhicules non communicants sur le protocole . . 84
3.5 Synthèse 90

4 Méthodologie de conception de véhicules intelligents **93**
4.1 Analyse de nouvelles propriétés du modèle par vérification formelle . 94
4.2 Interfaçage du modèle avec des modèles plus réalistes 104
4.3 Synthèse : proposition d'une méthodologie 114

Conclusion **117**

Perspectives et discussion **119**

Références **122**

Liste des figures **I**

Liste des tableaux **V**

Introduction

Depuis l'antiquité, le transport de personnes et de marchandises est une nécessité pour le bon fonctionnement de la société avec des motivations évoluant au fil des années (économie, santé, emploi, éducation, défense, tourisme...). Aujourd'hui, il est représenté par des moyens de plus en plus sécurisés notamment grâce au développement de nouvelles technologies, la modernisation des infrastructures et la prévention. Parmi les modes de transport les plus utilisés notamment en Europe et en Amérique du Nord [32], on trouve la voiture. Bien que la voiture soit synonyme d'indépendance et appréciée par sa facilité d'utilisation, la route reste l'environnement cumulant le plus d'accidents, notamment en France (quatre roues, deux roues, piétons) [37]. De plus, une étude effectuée dans les années 2010 montre que plus de 90 % de ces accidents sont de cause humaine [96].

On pourrait donc imaginer que les décisions actuellement prises par le conducteur pour contrôler le déplacement du véhicule soient automatisées : c'est le cas des véhicules autonomes. Ces véhicules devraient respecter trois grandes exigences. Tout d'abord, la sécurité, où le véhicule ne devra pas engendrer un accident et respecter des lois imposées par la code de la route (distance de sécurité, limitation de vitesse...). Puis, l'efficacité, dont le but est d'avancer le plus rapidement possible sur la route pour atteindre un objectif final (rentrer à la maison, aller à son lieu de travail...). Enfin, le confort car, bien que l'on puisse avoir un véhicule sécurisé et efficace, il est préférable de ne pas effectuer des décélérations brusques afin que les passagers ne se sentent pas en danger.

On peut voir le véhicule autonome selon deux approches. La première, dans laquelle la voiture est totalement indépendante et où seuls les capteurs embarqués fournissent les informations nécessaires pour conduire (position des autres véhi-

cules, des panneaux de signalisation, obstacles...). La seconde, dans laquelle est ajoutée la possibilité de communiquer avec d'autres véhicules ou l'infrastructure de la route. Cette dernière option paraît plus complète que la première car elle permet une anticipation plus élevée grâce à une information plus globale de l'environnement mais aussi car elle permet la coopération entre véhicules autonomes, utile pour mieux analyser les intentions de chacun.

Ces véhicules intelligents vont progressivement remplacer le parc actuel des véhicules. Toutefois, ils vont être confrontés à deux problèmes. Tout d'abord, si l'on imagine des portions de routes contenant uniquement des véhicules autonomes, leur modèle de décision va être différent d'un constructeur à un autre et cela peut impliquer des incompatibilités. De plus, une situation intermédiaire va apparaître, pendant laquelle véhicules autonomes et véhicules non autonomes conduits par des êtres humains vont coexister. Cela va complexifier le modèle de décision du véhicule autonome car celui-ci doit également prendre en compte d'autres véhicules dont le comportement est imprévisible. Dans les deux cas, ce sont des environnements mixtes où chaque véhicule possède ses propres caractéristiques (temps de réaction, accélération...), et donc un modèle de décision propre à celles-ci. Aux exigences de sécurité, d'efficacité et de confort, on peut alors ajouter une quatrième exigence, la courtoisie, correspondant au fait que le véhicule ne doit pas avoir un impact trop négatif sur les autres véhicules et de manière générale, les autres éléments mobiles de son environnement.

Pour étudier ces comportements, une première solution est de les observer dans la vie réelle à travers des prototypes de véhicules autonomes. Bien qu'elle permette d'avoir des résultats réalistes en termes de dynamique du véhicule, le coût final peut être très élevé et il est difficile de tester tous les scénarios possibles, notamment les cas dangereux, pouvant mener à une collision. Pour remédier à ce problème, une autre solution est d'utiliser la simulation informatique permettant de modéliser le comportement des véhicules dans un environnement choisi pour ainsi étudier différentes sortes de scénarios dans un temps plus court. Cette simulation constituant une abstraction de la réalité, il faut donc s'assurer que le modèle est transposable sur un véhicule réel de manière aussi transparente que possible, c'est-à-dire sans

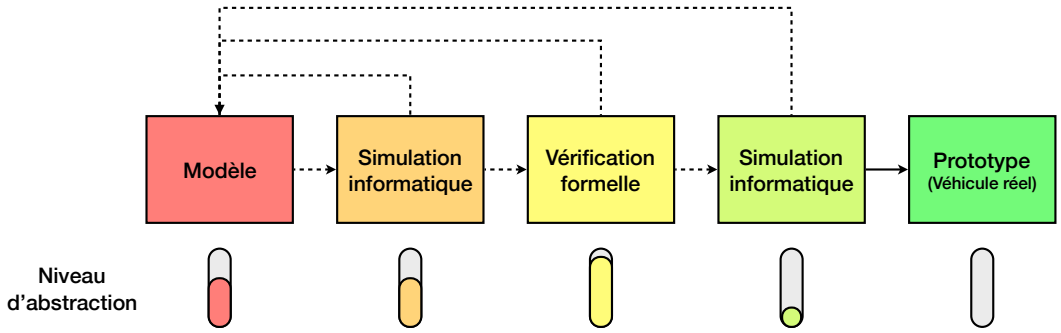


FIGURE 1: Méthodologie de conception d'un modèle de véhicule intelligent. En dessous, le niveau d'abstraction pour chaque étape.

perdre les caractéristiques intéressantes du modèle, telles qu'elles ont pu être observées en simulation. Par ailleurs, la simulation seule ne permet pas d'étudier, de manière exhaustive, les propriétés du modèle, pour garantir notamment sa sécurité dans toutes les situations possibles. Il faudra donc s'assurer, par des méthodes de vérification formelle (*model-checking*), que le modèle offre, en toutes circonstances, des garanties de sécurité souhaitées.

Ces deux besoins sont potentiellement contradictoires puisque les modèles de trajectoire en automatique, reproduisant le plus fidèlement la dynamique du véhicule, sont des modèles de très bas niveau, permettant très difficilement d'exprimer un modèle de décision. A l'autre extrême, les approches de vérification formelle nécessitent en général d'abstraire énormément le modèle de décision afin d'éviter une explosion de l'espace des états.

Le premier enjeu consiste donc à choisir le formalisme et le niveau d'abstraction adéquats pour pouvoir exprimer de manière riche des modèles de décision pour des véhicules autonomes, communicants et coopératifs. Le second enjeu est de s'assurer que ces modèles de décision sont compatibles, d'une part avec les approches de vérification formelle pour en garantir un certain nombre de bonnes propriétés, d'autre part avec les modèles de dynamiques pour les instancier sur des véhicules réels tout en conservant ces bonnes propriétés.

Actuellement, les différentes avancées autour des véhicules intelligents proposent des cas d'études aux environnements et niveaux d'abstraction multiples, rendant difficile toute comparaison ou compatibilité entre les modèles. L'objectif de cette thèse

est alors de définir un ensemble de méthodes et d'outils afin de proposer une méthodologie permettant de concevoir un modèle de véhicule à la fois fiable et réaliste. Pour cela, on imagine une architecture sous forme de couches (voir Figure 1). Dans un premier temps, on propose un modèle de véhicule, représentant à la fois son modèle de décision ainsi que les différentes entrées et sorties associées (perception, actions...). Ce modèle est testé et analysé via un outil de simulation défini au même niveau d'abstraction que le modèle. Ensuite, on valide les propriétés du modèle simulé par vérification formelle. A partir du modèle obtenu, on assure une compatibilité avec un outil de simulation au niveau d'abstraction plus bas (plus réaliste en termes de dynamique du véhicule). Enfin, l'étape de prototypage consiste à déterminer les éléments nécessaires à la mise en place du modèle dans un véhicule réel. A chaque étape de conception, il est possible de revenir du modèle initial permettant de l'affiner. Il repasse alors toutes les étapes de conception (simulation, vérification formelle puis simulation réaliste) afin de proposer un modèle pouvant être implanté dans un véhicule réel (prototype). Les différentes étapes de conception et liens vont être détaillés progressivement tout au long de ce manuscrit pour finalement obtenir une méthodologie pour la conception de Systèmes de Transport Intelligents Coopératifs.

Pour cela, nous allons dans un premier temps effectuer une analyse critique des avancées autour des véhicules autonomes, des véhicules communicants et de la simulation afin de préciser les critères conduisant au niveau d'abstraction (chapitre 1). A partir de ces critères, un outil de simulation va être choisi et amélioré pour y répondre totalement (chapitre 2). L'outil de simulation alors défini va être utilisé pour proposer un premier modèle de décision en prenant en compte des aspects de coopération et d'hétérogénéité de trafic (chapitre 3). Enfin, une méthodologie de conception de véhicules intelligents va être proposée, d'une part en établissant les éléments permettant de vérifier formellement un modèle simulé, d'autre part en proposant des solutions pour assurer une compatibilité entre le modèle conçu et un modèle de véhicule réaliste, proche d'un véhicule réel (chapitre 4).

1 | Positionnement

1.1 Un véhicule autonome 6

1.1.1 Niveau d'autonomie et capteurs 6

1.1.2 Avancées et défis 8

1.1.3 Analyse critique 10

1.2 Un véhicule communicant 11

1.2.1 Possibilités et niveaux de communication 11

1.2.2 Avancées et défis 13

1.2.3 Analyse critique 15

1.3 Un véhicule virtuel 16

1.3.1 Approches 16

1.3.2 Avancées et défis 17

1.3.3 Analyse critique 19

1.4 Synthèse 20

Dans cette partie, nous allons aborder les différentes avancées des véhicules intelligents sur des aspects d'autonomie, de communication et de simulation. Cela permet de faire un point sur leurs potentielles limitations afin de préciser les critères conduisant au choix du niveau d'abstraction pour le modèle ainsi que le type d'outil de simulation correspondant.

1.1 Un véhicule autonome

1.1.1 Niveau d'autonomie et capteurs

Considéré comme l'ancêtre de l'automobile [28], le fardier de Cugnot, créé en 1769 possède déjà les éléments que l'on connaît des véhicules actuels (un moteur, des roues, une direction, des freins. . .) avec une vitesse limitée à $4\ km.h^{-1}$. Depuis, le véhicule a beaucoup gagné en performance en atteignant des vitesses allant au-delà des limitations de vitesse autorisées sur les routes. Il a également gagné en sécurité en proposant différentes fonctionnalités le rendant plus autonome lui permettant de mieux éviter un danger proche.

Le terme "véhicule autonome" est un abus de langage car il regroupe un ensemble de véhicules dont la notion d'autonomie dépend du conducteur, du véhicule lui-même ou des deux. D'ailleurs, on appelle "Systèmes de Transport Intelligents" ou *ITS* les véhicules dont l'aspect décisionnel n'est pas totalement géré par le conducteur. Pour classifier cette notion d'autonomie, des normes ont été définies. En Amérique, la NHTSA (*National Highway Traffic Safety Administration*) propose un système à cinq niveaux [63]. En Europe, la classification de l'OICA (*Organisation Internationale des Constructeurs Automobiles*) définit six niveaux d'autonomie [64] (la principale différence étant la répartition du niveau 3 de la NHTSA en deux niveaux dans la norme européenne). La norme européenne définit donc l'autonomie d'un véhicule de la façon suivante :

— niveau 0 : le conducteur est totalement maître de son véhicule sur l'aspect longitudinal (sens de la route) comme sur l'aspect latéral (changement de voie). Il est toutefois possible d'avoir une aide du véhicule car celui-ci peut posséder des capteurs facilitant certaines manœuvres (exemple : avertissement sonore lors d'un franchissement de ligne continue) ;

— niveau 1 : le conducteur prend des décisions sur l'aspect longitudinal ou latéral tandis que le système s'occupe de l'aspect non géré par le conducteur (exemple : le régulateur de vitesse adaptatif ou *ACC*) ;

— niveau 2 : le conducteur devient un superviseur dans un cas d'utilisation prédéfini tandis que le système gère à la fois l'aspect longitudinal et latéral (exemple : l'aide au stationnement ou *park assist*) ;

— niveau 3 : bien que techniquement identique au niveau 2, la différence réside sur la non nécessité pour le conducteur d'être un superviseur pendant la manœuvre. Toutefois, le système doit être capable d'avertir le conducteur lorsqu'il ne peut plus gérer la conduite automatisée (exemple : la gestion automatisée des embouteillages sur des vitesses faibles) ;

— niveau 4 : le conducteur a cette fois un rôle de passager mais il peut toujours décider que le système prenne le contrôle ou non dans le cas concerné. Le système possède alors toutes les fonctionnalités du véhicule (exemple : gestion totale de la conduite dans un environnement restreint comme un parking pour garer automatiquement le véhicule ou aller chercher le passager) ;

— niveau 5 : de la même façon que les lignes de métro automatisées de Paris, le système est capable de prendre totalement le contrôle du véhicule dans n'importe quelle situation (exemple : les véhicules testés sur route actuellement comme les véhicules de Waymo [7]).

C'est à partir du niveau 3 que la législation peut tenir la machine pour responsable sur l'aspect décisionnel. Pour ces niveaux, on a donc besoin d'un ensemble de capteurs permettant de remplacer les différents sens de l'être humain nécessaires pour la conduite (la vue, l'ouïe et le toucher). Ces capteurs pourraient être capables d'observer l'environnement de façon plus précise, plus large et plus rapide qu'un être humain. On peut donc définir une liste de capteurs pouvant être nécessaires pour des véhicules autonomes de niveau 3 au minimum :

— le lidar : c'est de la télédétection par laser. Il permet de générer une carte en trois dimensions de l'environnement dans lequel se trouve le véhicule (routes, obstacles fixes ou mobiles. . .) ;

— caméra stéréoscopique : similaire aux yeux de l'être humain, elle permet de détecter les différents objets se trouvant dans son champ de vision (panneaux

de signalisation, obstacles, piétons...). La reconnaissance de ces objets se fait grâce à l'ordinateur embarqué ;

— la centrale à inertie : permet de déterminer l'accélération, la vitesse et la position du véhicule à un instant donné (généralement plus précis que l'antenne GPS) ;

— l'odomètre roue : permet de déterminer la distance parcourue par le véhicule ainsi que sa vitesse ;

— capteur à ultrasons : permet de détecter des objets sur de courtes distances (inférieures à trois mètres) ;

— antenne GPS : détermine la position exacte du véhicule via des coordonnées géographiques, obtenues grâce à un ensemble de satellites ;

— radars : ils permettent la détection des éléments environnants, leurs formes ainsi que leurs distances par rapport au véhicule.

L'ensemble de ces capteurs permettent *a priori* d'imaginer un véhicule capable de se passer du conducteur. Nous allons voir par la suite les différentes avancées des entreprises pour concevoir un tel véhicule.

1.1.2 Avancées et défis

Bien que certains états des États-Unis autorisent depuis peu la mise en circulation de véhicules autonomes sous certaines conditions [47], plusieurs tests de véhicules autonomes de niveau 4 ou 5 ont déjà été effectués dans un environnement réel dans certains contextes (concours, circuit...). Entre 1987 et 1995 est lancé le programme européen *PROMETHEUS* (*PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety*) ayant pour but principal d'améliorer l'environnement routier [95]. Dans le cadre de ce programme, l'ingénieur en aérospatiale *Ernst Dickmanns*, considéré comme le pionnier des véhicules sans conducteur, crée avec son équipe le véhicule autonome *VaMoRs*, qui est équipé de deux caméras, de plusieurs capteurs et d'un logiciel. Celui-ci a parcouru une vingtaine de kilomètres à une vitesse de $90\ km.h^{-1}$ sur une route sans présence de voitures.

En 1994, il réitère l'expérience avec *VaMP* constitué d'un équipement amélioré, testé sur une route composée de véhicules reproduisant un flux de trafic fluide. Il

avait la capacité de reconnaître les marquages sur la route, la présence des autres véhicules et pouvait changer de voie. Le véhicule pouvait atteindre une vitesse de $130\ km.h^{-1}$ [27].

En 2004, la *DARPA* (*Defense Advanced Research Projects Agency*) lance le *DARPA Grand Challenge* [1]. C'est une compétition entre voitures autonomes sans conducteur (autonomie de niveau 4) qui a lieu à l'ouest des Etats-Unis dans le désert des Mojaves sur un parcours de près de 250 kilomètres. Les voitures devaient réaliser le parcours en moins de 10 heures sans entrer en collision avec une autre voiture, même de manière involontaire. Le gagnant repartait avec un million de dollars. Cependant, aucune voiture n'a réussi à parcourir le trajet.

En 2005, la compétition reprend dans le désert sur une distance de 215 kilomètres avec un prix de deux millions de dollars pour le gagnant. Sur les 23 voitures qui ont participé, 5 ont parcouru le trajet entièrement dont 4 en moins de 10 heures. L'équipe de l'université Stanford *Stanley* remporte la compétition. Elle est dirigée par *Sébastien Thrun* qui n'est qu'autre qu'un des membres participant à la conception de la *Google Car*.

En 2007, la *DARPA* lance la *DARPA Urban Challenge* [1] dont la compétition s'organise à présent en milieu urbain en Californie. Le défi était d'effectuer un parcours de façon totalement autonome tout comme le *Grand DARPA Challenge*. Toutefois, l'idée était de prendre en compte les difficultés que l'on peut rencontrer en ville (intersections, cyclistes, piétons. . .) tout en respectant le code de la route. L'ensemble de ces compétitions a permis de mettre en avant l'avancée des voitures autonomes et la possibilité de les imaginer sur les routes actuelles. En effet, les voitures ont été capables de suivre une route, d'analyser différents obstacles et de réagir en conséquence.

A partir de 2010, on commence à évoquer la possibilité de tester des véhicules de niveau 4 et 5 sur les routes actuelles. Le véhicule ayant le plus médiatisé cette pratique est la *Google Car*. C'est un projet lancé vers 2010 par *Google* et dirigé par *Sebastian Thrun* (co-inventeur de *Google Street View* et gagnant du *Grand DARPA Challenge 2005*) [85]. Les prototypes proposés avant mai 2014 étaient des modèles semi-automatisés (niveau 4). *Google* affirme avoir parcouru plus d'un million de kilomètres avec l'ensemble des *Google Car* testées sur les routes. Le nouveau modèle,

dont la vitesse est limitée à $40\ km.h^{-1}$ est considéré comme un modèle de niveau 5 dans lequel se trouve un superviseur prêt à réagir en cas de danger non détecté par le véhicule. En 2016, le projet *Google Car* devient le projet *Waymo* et atteint plus de 8 millions de km de tests sur différents lieux des États-Unis (niveau 5) [7].

Depuis 2014, les tests de véhicules de niveau 4 et 5 continuent mais on propose déjà aux conducteurs des véhicules de niveau 3 permettant une autonomie dans certaines situations. Le plus médiatisé actuellement reste le modèle de Tesla qui propose à la vente des véhicules de niveau 3 composés d'un système appelé *Auto-Pilot* permettant une autonomie partielle sur les autoroutes. En 2017, Audi lance le modèle *A8* capable de gérer des embouteillages sur une autoroute jusqu'à une vitesse de $60\ km.h^{-1}$ (non mis en vente pour le moment). D'autres entreprises (Uber, Ford, Renault, Toyota, BMW...) testent également des véhicules de niveau 3 dans le but de les commercialiser d'ici peu.

Depuis plusieurs années, le défi de beaucoup d'entreprises du marché est de proposer une commercialisation de leurs véhicules autonomes vers 2020. De par les tests actuellement effectués et les limites de ces véhicules, les véhicules autonomes concernés seraient de niveau 3.

1.1.3 Analyse critique

A l'évidence, on remarque assez rapidement le véhicule automatisé connaît actuellement de grandes avancées. En effet, il est possible de concevoir un véhicule capable d'effectuer un déplacement longitudinal et latéral sans intervention humaine sur tout type de route (autoroutes, carrefours...) tout en respectant les règles imposées par le code de la route (lignes blanches, panneaux de signalisation, feux tricolores...). Il est également possible d'adapter sa vitesse face à un véhicule situé devant le véhicule autonome.

Toutefois, une première limitation apparaît au niveau des conditions météorologiques extrêmes (neige, pluie, brouillard) où le véhicule autonome (tout comme le conducteur) connaît une réduction de sa visibilité et perd également en adhérence, impliquant un renforcement des contraintes de sécurité dans ces situations. Actuellement, les essais sur route des véhicules autonomes se font dans des conditions où tout élément naturel pouvant dégrader la perception et l'adhérence du véhicule

ne sont pas présents [45].

Une deuxième limitation concerne l'interprétation que le véhicule fait des images capturées par les capteurs. Un accident survenu entre un piéton et un véhicule autonome d'Uber pose cette question de reconnaissance [58]. En effet, le logiciel a interprété le piéton survenant à droite du véhicule comme étant un obstacle fixe, et donc, considéré comme non dangereux pour le véhicule. Cela pose le problème du niveau de tolérance que l'on accepte lors de la reconnaissance. Ainsi, un bas niveau de tolérance produira un véhicule qui va s'arrêter fréquemment en raison d'un trop grand nombre d'obstacles reconnus, et inversement, un haut niveau de tolérance va avoir un impact sur la sécurité.

Enfin, une dernière limitation pose la question de la présence de véhicules conduits par des êtres humains dans l'environnement routier actuel. En effet, bien que globalement, les conducteurs aient un comportement rationnel, il n'est pas possible de déterminer la motivation et l'intention de chacun à un instant donné. De plus, contrairement à un conducteur humain, le véhicule peut avoir des difficultés à effectuer des gestes de courtoisie (en plus du clignotant) pour faciliter une manœuvre (comme dans une voie d'insertion par exemple). Ce problème peut également se poser si l'on considère un environnement constitué de véhicules autonomes ayant des modèles décisionnels différents du véhicule dans lequel on se trouve.

1.2 Un véhicule communicant

1.2.1 Possibilités et niveaux de communication

Pour mettre en place cette notion de courtoisie, en plus d'un véhicule capable de percevoir son environnement de la façon la plus précise possible, il est possible d'ajouter une fonction grâce à laquelle ce véhicule aurait la possibilité de communiquer avec son environnement. C'est ce que l'on appelle les *C-ITS* ou Systèmes de Transport Intelligents Coopératifs [26]. Il est possible de matérialiser ces *C-ITS* grâce à différentes normes conçues pour créer des réseaux véhiculaires appelés *VANETs* ou Réseaux Véhiculaires Ad-Hoc [97]. La technologie de communication spécialisée pour les Systèmes de Transport Intelligents est appelée *DSRC* (*Dedicated Short Range Communications*) [25] basée sur la norme *802.11p* [50]. Elle

reprend les technologies du *Wi-Fi* auxquelles on ajoute des fonctionnalités permettant des communications à haute vitesse tout en réduisant les latences. En Europe, la norme est appelée *ITS-G5* et est définie par l'*ETSI* (*European Telecommunications Standards Institute*) [31]. Cette norme permet deux types de communications sur les routes (voir Figure 1.1) [23] :

— Les communications *V2V* (*Vehicle to Vehicle*) : à partir de dispositifs de communication intégrés aux véhicules appelés *OBU* (ou Unités Embarqués), un véhicule peut envoyer une information à un autre véhicule (*Unicast*) ou à un groupe de véhicules (*Broadcast*) ;

— Les communications *V2I* (*Vehicle to Infrastructure*) ou *I2V* (*Infrastructure to Vehicle*) : le principe est le même que les communications *V2V* mais dans ce cas, les véhicules ont la possibilité de communiquer avec des unités routières situées sur la route appelées *RSU* (ou Unités de Bord de Route) et inversement.

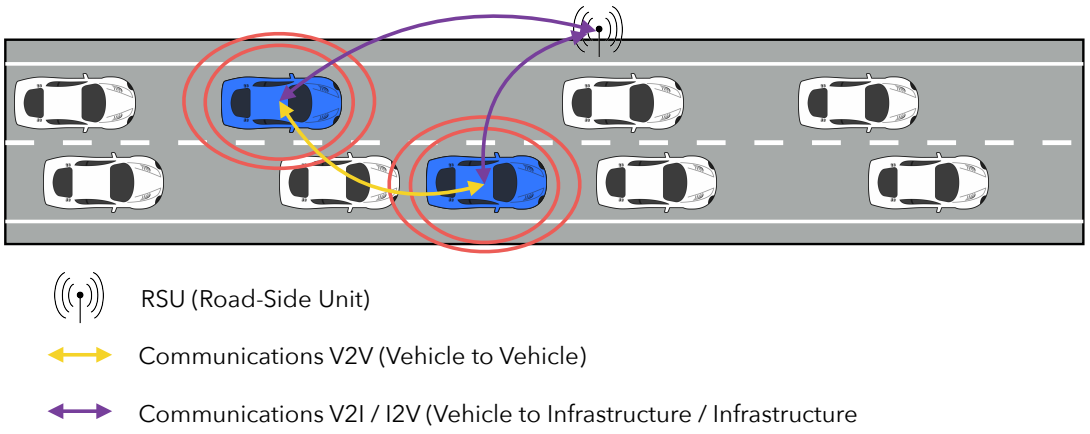


FIGURE 1.1: Types de communications entre véhicules communicants. En jaune, les communications entre véhicules (V2V). En violet, les communications avec l'infrastructure (V2I ou I2V).

Les *RSU* peuvent détecter un danger et en informer les véhicules situés dans leur zone de perception afin d'anticiper ce danger (prévention d'un embouteillage, d'un accident. . .). Il est également possible d'imaginer un ensemble de *RSU* sur les routes afin de propager une information entre les différentes unités. Concernant les communications *V2V*, on peut proposer une classification selon plusieurs niveaux de la même manière que les niveaux d'autonomie des véhicules autonomes présentés précédemment. Plus le niveau est élevé, plus la communication a un impact sur le réseau véhiculaire concerné par ces communications :

— *niveau C0* : Aucune communication informatisée. Les véhicules ne peuvent communiquer directement avec les autres véhicules. Toutefois, les conducteurs ont la possibilité de communiquer via des signes de la main ou des signes lumineux (qui auront un impact essentiellement sur des véhicules conduits par des êtres humains) ;

— *niveau C1* : L'envoi d'informations personnelles (position, vitesse, accélération, informations reçues d'autres véhicules. . .). Cela permet d'avoir une connaissance plus précise de la route et ainsi, réagir différemment lors d'une situation précise ;

— *niveau C2* : L'envoi d'une intention à un ou plusieurs véhicules ("Je vais accélérer", "Je vais aller ici bientôt"). Cela permet aux autres véhicules d'être au courant de la potentielle action future du véhicule. La différence avec le niveau C1 est qu'elle concerne un futur proche et la décision peut être non réalisée ou irréalisable ;

— *niveau C3* : L'envoi d'une intention avec attente d'une réponse de type "Oui" ou "Non". Cette fois-ci il est possible d'avoir un feedback suite à cette intention permettant éventuellement au véhicule de changer d'avis ;

— *niveau C4* : La mise en place d'un dialogue entre plusieurs véhicules (négociation). Cela reprend les aspects des niveaux C2 et C3 avec un but différent : chaque véhicule doit, par l'intermédiaire d'un compromis, prendre une décision qui lui convient ("Mon action me paraît juste pour moi, l'est-elle pour toi aussi ?").

Cette classification permet notamment un positionnement plus clair sur l'utilisation de la communication entre véhicules mais aussi un positionnement des différentes avancées sur les réseaux véhiculaires.

1.2.2 Avancées et défis

Tout comme pour les véhicules autonomes, on note également différentes innovations autour des véhicules communicants. Elles sont intéressantes car elles proposent des solutions sur des véhicules aux niveaux d'autonomie différents.

Parmi ces avancées, on peut citer le *platooning*. Un *platoon* est un ensemble de voitures situées les unes derrière les autres à la manière d'un train composé de

wagons. Comme un train, chaque *platoon* est dirigé par la voiture de devant appelée *leader*. Les autres voitures, les *followers*, répondent aux ordres donnés par le *leader* et effectuent les mêmes actions que lui.

Plusieurs projets associés au *platooning* ont vu le jour. Le plus connu d'entre eux est le projet *SARTRE* (*Safe Road Trains for the Environment*) lancé en 2009 [73]. Il propose un principe reposant à la fois sur des véhicules semi-automatisés (niveaux 3 et 4) et sur des véhicules actuels (niveaux 0 à 2). Au sein du *platoon*, le *leader* est représenté par un camion dirigé par un conducteur professionnel. Les *followers* sont des véhicules semi-automatisées. Le principe est simple : initialement, aucune voiture n'est reliée au camion. Lorsqu'une voiture semi-automatisée s'approche de lui, le camion lui envoie une demande afin de rejoindre le *platoon*. C'est le passager du véhicule qui décide d'accepter ou non de le rejoindre (niveau C3 de communication). En cas d'acceptation, le véhicule passe en conduite automatisée et se place derrière le camion. Pour reprendre la main, le conducteur de la voiture fait une demande au camion de quitter le *platoon*. Toutes les voitures suivant le camion effectuent la même trajectoire que celui-ci. Le nombre de *followers* conseillé est de 15.

Des tests ont été effectués en janvier 2011 en Suède sur les pistes d'essais de *Volvo* avec un camion suivi d'une voiture. Le conducteur de la voiture avait la possibilité de faire autre chose pendant qu'il se trouvait dans le *platoon* (lire un journal, boire du café...). En janvier 2012, le projet *SARTRE* a été testé sur les routes de Barcelone en Espagne. Le test a été concluant avec un camion suivi de trois voitures roulant à une vitesse de $90\ km.h^{-1}$, avec des distances de sécurité inférieures à 6 mètres.

L'institut *VeDeCom* utilise des *RSU* sur ses pistes d'essais pour mieux anticiper un danger proche (niveau C1) [92]. Un *RSU* est posé sur la route et situé près d'un véhicule à l'arrêt. Celui-ci va envoyer des messages à intervalles réguliers à un groupe de véhicules situés plus loin (niveau 3). Lorsque le groupe de véhicules reçoit le message, celui-ci se met à ralentir afin de ne pas entrer en collision avec le véhicule à l'arrêt.

D'autres entreprises se lancent sur l'utilisation de la communication afin d'accroître la connaissance de l'environnement (niveau C1), à travers plusieurs projets comme le projet *SCOOP* qui a déployé près de 3000 véhicules sur 2000 kilomètres

de routes à partir de 2016 [8]. Ces projets misent en majorité sur un accroissement des communications entre véhicules, routes et piétons afin d'une part de compléter les informations déjà obtenues grâce aux capteurs d'autres véhicules autonomes, d'autre part afin de proposer des véhicules connectés à des conducteurs pour mieux être informés d'un danger proche.

1.2.3 Analyse critique

Au vu des différentes avancées sur le véhicule communicant, on remarque que l'utilisation des communications se restreint généralement à une amélioration des connaissances physiques de l'environnement. Cela permet d'obtenir des informations sur des véhicules plus éloignés permettant une anticipation des décisions, ce que ne permet pas l'utilisation seule de capteurs d'observation qui ont une représentation plus locale de l'environnement. La seule exception réside sur la notion de *platooning* qui propose à la fois des décisions basées sur l'intention et fait le lien avec l'environnement routier actuel.

Les communications V2I / I2V peuvent être intéressantes pour propager une information locale à une portion de route. Cela permet aux véhicules d'anticiper un danger futur par exemple. De plus, à l'aide de plusieurs bornes, il est possible de propager l'information sur une portion plus large de la route. Cela a le pouvoir d'informer à l'avance un véhicule, améliorant son anticipation. Si l'on imagine que ces bornes diffusent leur information par l'intermédiaire d'un panneau d'affichage, elles seraient également utiles pour les véhicules actuels. Toutefois, l'ajout de ces bornes impose une amélioration de la structure des routes et donc un coût supplémentaire. Il peut toutefois être intéressant d'observer l'apport de ce type de communication combiné aux communications entre véhicules.

L'utilisation même des communications pose toutefois plusieurs limitations. Une première concerne la perte d'informations. Les récentes expérimentations ont montré que la perte d'informations était négligeable mais il peut être important de la prendre en compte afin d'imaginer un système plus fiable [24]. Une seconde limitation pose la question de l'intrusion au sein du véhicule. Il faut concevoir un système totalement sûr pour éviter toute possibilité de contrôler un véhicule à distance ou de diffuser des fausses informations. Une dernière limitation concerne les poten-

tiels problèmes d'accord entre l'information donnée au niveau des capteurs et celle reçue des communications. Quelle information est considérée comme étant la plus fiable ?

Peu d'avancées ont été effectuées sur les niveaux de communication supérieurs à C1, basés sur l'intention du véhicule. Contrairement aux autres niveaux, elle implique de connaître les décisions futures d'un véhicule et donc de s'assurer que les véhicules environnants ne vont pas l'impacter négativement. Si c'est le cas, il est possible de faire négocier les véhicules entre eux, afin qu'ils effectuent des décisions compatibles avec les décisions des véhicules environnants.

1.3 Un véhicule virtuel

1.3.1 Approches

Toutes les innovations présentées jusqu'à maintenant concernent des véhicules réels testés sur des routes ou pistes d'essais. Cela permet une expérimentation réaliste car cela respecte la physique de l'environnement. Cependant, cela impose d'utiliser des prototypes coûteux et la mise en place de dispositifs également coûteux dans le cadre de la conception d'une route intelligente par exemple. La simulation informatique peut être une solution au problème de coût en imaginant le véhicule comme une entité virtuelle située dans un environnement virtuel. Différentes approches ont été développées autour de la simulation des véhicules autonomes parmi lesquelles on pourrait distinguer trois grandes familles : l'approche orientée sur le réalisme, sur le trafic et sur les agents.

L'approche orientée sur le réalisme consiste à décrire la dynamique du véhicule par un ensemble d'équations sous une forme d'espace non linéaire. Cet ensemble d'équations est basé sur l'application du principe fondamental de la dynamique pour prendre en compte les principaux efforts affectant la dynamique du véhicule. Un modèle couplé de dynamique longitudinale et latérale est généralement utilisé pour reproduire la conduite humaine en association avec le comportement du véhicule. Ce modèle couplé est paramétré par un ensemble de valeurs issues du prototype expérimental du véhicule (masse, inertie...).

Une deuxième approche s'intéresse en priorité au trafic, notamment son évolu-

tion au cours du temps à travers trois points de vue [90] :

— macroscopique : on étudie les véhicules dans leur globalité dans une grande zone géographique ;

— microscopique : on étudie les véhicules de façon individuelle sur une zone géographique plus restreinte ;

— mésoscopique : dans cette combinaison des zones microscopiques et macroscopiques, on étudie un flux de véhicules dans une zone plus restreinte que les modèles macroscopiques tout en ayant la possibilité d'étudier des groupes de véhicules de façon individuelle.

Le niveau d'abstraction est alors plus élevé que dans l'approche orientée sur le réalisme. De plus, selon les points de vue, il est possible de définir un temps discret, permettant alors par exemple de garantir certaines propriétés du modèle par vérification formelle (point abordé dans la partie 4.1).

Enfin, la troisième approche fondée sur la construction d'un système constitué de plusieurs agents. Un agent est une entité ayant une certaine autonomie, évoluant dans un environnement qu'il peut percevoir impliquant une évolution de son comportement et de ses buts au cours du temps [34]. Chaque agent a également la possibilité d'interagir avec d'autres agents ou avec son environnement pour échanger une information ou négocier une future décision à prendre [15]. Les Systèmes Multi-Agents peuvent ainsi très bien représenter un ensemble de véhicules (agents) se trouvant sur une route (environnement) et ayant la possibilité de communiquer entre eux (interactions) [29].

Ces trois approches apportent différentes informations et permettent d'étudier différentes propriétés à différents niveaux d'abstractions. On pourrait donc imaginer un outil de simulation capable de prendre en compte ces trois approches à la fois afin d'observer des propriétés différentes. Nous allons donc étudier les différentes avancées autour des outils de simulation afin de mieux déterminer les possibilités de chacun.

1.3.2 Avancées et défis

Nous avons évoqué les différents véhicules autonomes proposés par différentes entreprises. Chacun de ces véhicules dispose d'une représentation virtuelle de l'en-

vironnement représentant l'ensemble des éléments observables en temps réel. Si l'on prend l'exemple de la représentation virtuelle de l'environnement sur les véhicules autonomes de Waymo (Figure 1.2), on obtient une représentation en deux dimensions de l'environnement routier, auquel s'ajoute une surcouche d'éléments en trois dimensions tels que les véhicules environnants (en violet), les piétons (en jaune), les cyclistes (en rouge), les panneaux des signalisation ainsi que le chemin que le véhicule autonome doit effectuer. Bien que cela ne soit pas de la simulation, on peut supposer que la conception de ces véhicules sont basés sur des simulations proposant ce niveau de réalisme. D'autres outils permettent cette représentation précise de l'environnement. C'est le cas des simulateurs de conduite qui permettent de reproduire au plus proche la réalité de l'environnement routier réel. Parmi ces outils, on peut citer SCANeR de l'entreprise OKTAL [4] ou encore OpenDS [60], une alternative gratuite et open-source.

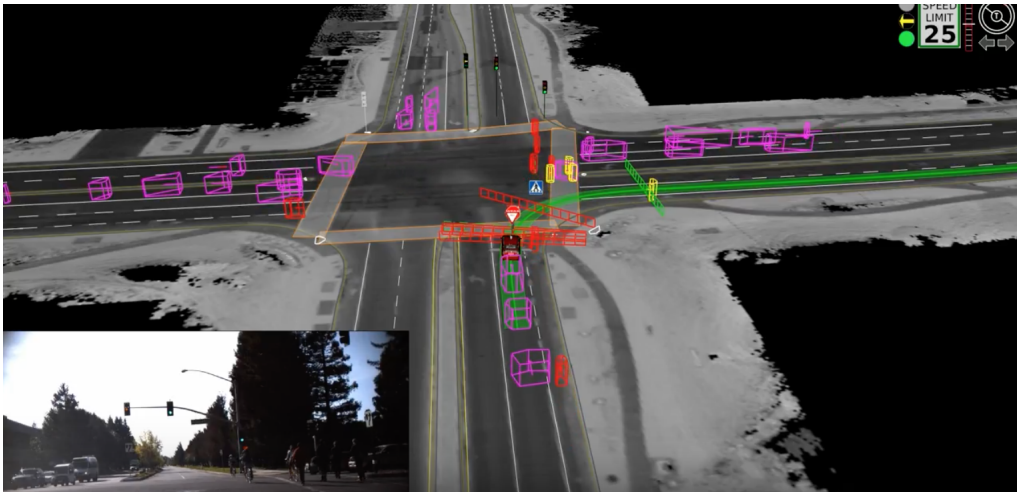


FIGURE 1.2: Perception du lidar des véhicules autonomes de Waymo (© Google).

Concernant l'approche trafic, on retrouve deux grandes catégories d'outils qui diffèrent notamment sur leurs aspects de gratuité et d'ouverture. La première catégorie regroupe les outils propriétaires permettant la modélisation, la simulation et l'analyse du trafic routier de façon complète et intuitive. Ils permettent généralement une représentation en deux ou trois dimensions de l'environnement et proposent des véhicules autonomes et/ou communicants en respectant la norme des ITS. Parmi ces outils, on peut citer Aimsun [22], Paramics [21] ou encore Vissim [33]. Dans la deuxième catégorie, on retrouve des outils gratuits et open-source. Bien que plus limités que les outils cités précédemment, il est possible de concevoir des

extensions afin d'obtenir un outil plus complet. Ils proposent généralement des environnements en deux dimensions dans lesquels on peut simuler et analyser des flux de trafic dans divers environnements. Parmi les outils les plus connus, on retrouve SUMO [55], MatSIM [12] ou encore MovSIM [88].

L'approche agent a un statut particulier dans ce contexte car les outils ne font pas toujours référence à l'environnement routier et aux véhicules. Ces outils proposent généralement une approche orientée agents de par la structure de l'outil permettant de modéliser les agents et leurs différents aspects (environnement, action, perception...) et/ou via un langage de programmation basé sur les agents. Cela permet de mieux séparer les différents aspects propres aux agents (environnement, action, perception...). Différents outils existent pour représenter ces systèmes, et plusieurs comparatifs existent pour choisir les outils selon le système souhaité [6] [14]. Parmi ces outils, on peut citer l'outil propriétaire AnyLogic [18] qui permet de concevoir des Systèmes Multi-Agents via une interface intuitive ou encore Cormas [19] qui est une plate-forme de simulation open-source et gratuite. D'autres outils utilisent un langage de programmation orienté agents accompagné d'un outil de simulation comme NetLogo [94] ou encore GAMA [83] qui ajoute la possibilité de communiquer entre agents.

1.3.3 Analyse critique

Bien que les outils des approches orientés sur le réalisme ont l'avantage de représenter un environnement le plus fidèle à la réalité, il n'est pas toujours évident de travailler sur l'aspect décisionnel du véhicule en raison du grand nombre de paramètres à modifier. L'approche trafic, grâce à son niveau d'abstraction généralement plus élevé, va permettre d'étudier plus facilement des propriétés de flux de trafic mais peut se limiter lorsque l'on souhaite ajouter des nouveaux aspects tels que la communication par exemple. L'approche agent ne répond généralement pas à des aspects trafic mais propose une liberté plus grande sur la conception du système et sur le modèle décisionnel des véhicules.

Ainsi, la combinaison des approches réaliste, trafic et agent pourrait permettre de constituer un simulateur "idéal". Toutefois, il est très rare de trouver un outil de simulation qui mette en œuvre les trois approches à la fois. Par exemple, si l'on

considère un simulateur orienté agents, il est souvent difficile de représenter l'aspect trafic. De même qu'un simulateur conçu pour bien représenter le trafic pourra ne pas présenter directement un niveau de réalisme attendu initialement. Pour des raisons similaires, un simulateur reproduisant bien le réalisme des véhicules va souvent manquer de l'aspect agent. Selon les approches, les simulateurs vont permettre de modéliser et d'étudier des propriétés différentes à des niveaux de réalisme et d'échelle différents. Ainsi, l'outil de simulation couvrira un sous-ensemble plus ou moins grand de ces trois approches.

1.4 Synthèse

Toutes les limitations des véhicules autonomes présentées montrent que l'on est encore loin d'avoir un modèle décisionnel parfait et qu'il est difficile d'imaginer à l'heure actuelle un véhicule autonome de niveau 5 capable de s'adapter sur une route fréquentée également par des véhicules non autonomes. Cependant, imaginer un véhicule capable de rouler sur les différentes routes existantes dans la vie réelle tout en respectant la signalisation n'est plus utopique aujourd'hui. Bien que les véhicules souffrent encore de limitations techniques notamment sur la gestion des conditions météorologiques ou dans la reconnaissance d'images, on peut facilement se positionner sur un véhicule dont l'aspect longitudinal et latéral est totalement assuré par le véhicule lui-même (véhicules de niveau 3 à 5). Toutefois, d'autres questions se posent lorsque l'on souhaite ajouter des véhicules ayant des modèles décisionnels différents. Sans interaction, il difficile de déterminer ni leur motivation ni leur intention. On peut alors imaginer un véhicule capable de coopérer avec d'autres véhicules autonomes afin d'une part, arriver à un accord commun, d'autre part, anticiper un danger proche.

La vision actuelle du véhicule communicant, consistant à effectuer des échanges d'information entre véhicules et via l'infrastructure paraît compatible dans un environnement constitué de véhicules à la fois autonomes et non autonomes. De plus, elle permettrait de palier au manque d'information des capteurs sur une zone plus étendue, empêchant de détecter des dangers futurs. Notre objectif étant d'établir des stratégies de coopération entre véhicules autonomes et communicants dans un environnement hétérogène, on se dirigera vers des niveaux de communication C2

minimum, tout en observant les différentes intentions futures des véhicules afin de prendre en compte la notion de courtoisie entre les véhicules.

La simulation s'avère être un élément indispensable à la conception des véhicules intelligents. Cependant, le choix de l'outil adapté peut se révéler particulièrement difficile en raison des différentes approches existantes. La combinaison des approches agent et trafic permet d'obtenir un système où l'on se concentre sur l'autonomie et les interactions d'un agent tout en conservant une expertise trafic impliquant une représentation fidèle de l'environnement routier. L'approche orientée sur le réalisme permet de représenter de façon précise la dynamique et les actions d'un véhicule via des trajectoires précises. Elle est d'ailleurs la plus étudiée pour effectuer le passage entre la simulation et un prototype de véhicule réel. Toutefois, il est parfois difficile de concevoir un modèle de décision à partir d'un grand nombre de paramètres au niveau du véhicule, nous incitant donc à les réduire. De plus, cette réduction permet de se combiner plus facilement à des outils de vérification formelle.

Finalement, le simulateur idéal est un outil gratuit et open-source, capable de modéliser une route et des agents de façon simple et intuitive tout en ayant la possibilité de créer son propre modèle décisionnel pour le véhicule et les véhicules environnants. Dans le chapitre 2, nous proposons un comparatif entre différents outils de simulation donnant plus de détails sur leurs distinctions et les différentes solutions apportées à l'outil choisi afin de répondre totalement aux approches souhaitées.

2 | Un simulateur multi-agent d'analyse comportementale de véhicules

2.1 Choix d'un outil de simulation 24

2.1.1 Motivations 24

2.1.2 Critères d'évaluation et analyse comparative 25

2.1.3 Tableau comparatif et choix 34

2.2 Un outil de simulation : GAMA 36

2.2.1 Présentation 36

2.2.2 Fonctionnement de l'extension trafic 39

2.3 Contribution : extension de GAMA 41

2.3.1 Représentation de l'environnement 41

2.3.2 Prise de décision 47

2.3.3 Optimisations et performances 50

2.3.4 Contenu de l'extension et synthèse 53

2.1 Choix d'un outil de simulation

2.1.1 Motivations

Comme nous l'avons dit précédemment dans la partie 1.4, l'objectif est de trouver un outil alliant l'approche multi-agent pour ses aspects comportementaux et coopératifs ainsi que l'approche trafic pour ses aspects de modélisation et d'analyse de l'environnement véhiculaire. Aucun outil existant ne permet de concevoir exactement le système souhaité. Il a fallu donc déterminer un ensemble d'outils susceptible de concevoir un tel système.

Tous les simulateurs présélectionnés répondent à des critères d'accessibilité et de fiabilité. En effet, il était nécessaire d'obtenir un outil facilement accessible sur le web et mis à jour récemment et/ou régulièrement. Toutefois, des critères supplémentaires étaient requis pour chaque catégorie. Pour les simulateurs de type trafic, le choix s'est fait selon leur popularité dans le domaine du trafic. Concernant les simulateurs multi-agents, nous avons choisi ceux proposant des premiers outils de modélisation et d'étude de trafic afin de se rapprocher au maximum d'un outil alliant les deux approches. Enfin, un simulateur de conduite a été choisi car il paraissait susceptible de répondre aux deux approches trafic et agent bien que ce soit un simulateur proposant un niveau d'abstraction plus faible que celui choisi. De plus, nous avons fait le choix de présenter quelques outils payants, généralement plus complets, afin d'établir des comparaisons intéressantes. Chaque simulateur représente donc une approche différente pour modéliser et simuler des Systèmes de Transport Intelligents. Pour répondre à ce besoin, nous avons retenu sept outils, répartis entre trois catégories :

1. Les simulateurs de trafic
   - Aimsun [22] : l'*Advanced Interactive Microscopic Simulator for Urban and non-urban Networks* (Aimsun) est un logiciel de modélisation et de simulation de transports créé en 1993 et développé en C++ par TSS-Transport Simulation Systems ;
   - *MovSim* [90] : le *Multi-model Open-source Vehicular-traffic SIMulator* (MovSim) a été initié en 2011 par Arne Kesting à l'Université Technique de Dresde en Allemagne ;

— SUMO [55] : le logiciel de *Simulation of Urban MObility* (SUMO) a été développé en 2001 en collaboration entre le Centre d'Informatique Appliqué de Cologne (ZAIK) et l'Institut des Systèmes de Transport au Centre Allemand pour l'Aéronautique (DLR).

2. Les simulateurs multi-agents
   — AnyLogic [18] : AnyLogic est un logiciel de simulation multi-méthode (systèmes à événements discrets, dynamiques et à base d'agents) développé en Java et conçu en 2000 par The AnyLogic Company ;
   — GAMA [83] : GAMA (*Gis & Agent-based Modelling Architecture*) est un simulateur de systèmes multi-agents conçu en 2010 par l'unité de recherche UMMISCO (*Unité mixte internationale de Modélisation Mathématique et Informatique des Systèmes COmplexes)* à l'initiative d'Alexis Drogoul ;
   — MATSIM [12] : le *Multi-Agent Transport SIMulation*, permettant la représentation des systèmes de transport à grande échelle, a été créé vers 2005 par deux groupes de recherche dirigés par Kai Nagel et Kay W. Axhausen ainsi que la société Senozon, spécialisée dans les transports et la planification des systèmes de transport.
3. Les simulateurs de conduite
   — OpenDS [60] : OpenDS (*Open-source Driving Simulator*) est un simulateur de conduite en trois dimensions créé en 2012 par le groupe automobile du Centre allemand de recherche pour l'intelligence artificielle (DFKI GmbH).

2.1.2 Critères d'évaluation et analyse comparative

Un ensemble de critères d'évaluation est nécessaire afin d'orienter la recherche vers un simulateur respectant les approches agent et trafic tout en ayant la possibilité de se diriger vers une approche orientée sur le réalisme. Nous les avons regroupés en six catégories afin de déterminer l'ensemble des aspects qui peuvent être intéressants pour effectuer ce choix. Ces critères sont détaillés tout au long d'une analyse comparative des différents outils de simulation permettant de déterminer leurs possibilités mais aussi leurs limites.

Nous avons installé puis testé chaque simulateur sur une machine récente de type bureautique. Afin de bien explorer les fonctionnalités proposées, nous avons simulé puis analysé un premier modèle de route. Enfin, pour les simulateurs open-source, nous avons exploré le code source pour observer les possibilités de personnalisation et d'extension.

2.1.2.1 Caractéristiques générales

Il s'agit ici d'analyser les caractéristiques principales du logiciel, c'est-à-dire de savoir si le logiciel est gratuit, libre et multiplateforme. Il est également important de voir si l'installation se fait instantanément ou demande un ensemble d'étapes préliminaires.

Prix. Seuls GAMA, MATSIM, MovSim et SUMO proposent des outils totalement gratuits. OpenDS dispose d'une version gratuite ainsi que de versions plus complètes mais payantes. Aimsun propose plusieurs versions payantes mais il permet la simulation microscopique dans une version gratuite limitée (sans sauvegarde). Enfin, AnyLogic est payant mais possède une version gratuite pour une utilisation hors recherche (personnelle et éducation).

Open-source. Exceptés Aimsun et AnyLogic qui sont des logiciels propriétaires, tous les outils proposés sont open-source (et disponibles sur GitHub).

Multiplateforme. Tous les simulateurs fonctionnent sur les systèmes d'exploitation principaux (Windows, Linux, Mac OS).

Installation. L'installation de Aimsun, AnyLogic et GAMA se fait à partir d'un exécutable. MATSIM et OpenDS s'exécutent directement à partir d'un fichier au format .jar (Java). L'installation de SUMO et de MovSim, demandant quelques étapes préliminaires, se fait assez simplement et des efforts ont été faits au niveau de la documentation.

2.1.2.2 Prise en main

Lorsque le simulateur est installé, il est important d'avoir une interface et une documentation claires pour une bonne prise en main. De plus, en cas de problème,

il est nécessaire de savoir s'il est possible de contacter les concepteurs et/ou la communauté du simulateur concerné.

Documentation. Les simulateurs disposant de la documentation la plus riche sont Aimsun et AnyLogic (manuel PDF, tutoriels textuels et vidéos, vidéos de démonstration, études de cas, articles de recherche. . .). MATSIM propose un manuel de documentation très complet et des tutoriels textuels pour les utilisateurs ainsi qu'un guide pour les développeurs. La documentation de GAMA est complète avec un manuel, une base de données de chaque fonctionnalité du langage GAML ainsi qu'un grand nombre d'exemples de tous types. Le site officiel de SUMO possède un wiki complet ainsi qu'un tutoriel détaillé avec un premier exemple à modéliser. MovSim fournit une documentation basée sur des exemples afin de démarrer assez facilement sur une première simulation. De plus, l'ensemble des modèles implémentés se trouve dans le livre très complet *Traffic Flow Dynamics* [90], recensant l'ensemble des modèles existants dans la littérature. La combinaison de ces deux documentations ainsi qu'un contact possible avec les concepteurs par mail rend la prise en main du logiciel plutôt accessible. On regrettera cependant l'absence d'une documentation de départ ou d'un wiki spécialisé pour le simulateur. Concernant OpenDS, la documentation dépend de la version choisie. Ainsi, la version gratuite propose des tutoriels vidéo tandis que les versions payantes donnent accès à une documentation sous forme d'un wiki.

Communauté. Pour Aimsun et AnyLogic, une équipe est disponible pour répondre aux différentes questions (délai de 48 heures pour AnyLogic). Ils est possible et simple de contacter les principaux concepteurs des autres simulateurs. En outre, Aimsun, MATSIM, Sumo et OpenDS possèdent des groupes de discussion où la communauté est active.

Simplicité. De par la présence d'une interface "user-friendly", Aimsun et AnyLogic se présentent comme les plus simples d'utilisation sans nécessité de regarder à tout prix la documentation. GAMA est simple d'utilisation notamment grâce à la présence de GAML qui est un langage orienté agent proche de Java intuitif et simple d'utilisation (seul un peu de documentation autour du langage est nécessaire). MATSIM, MovSim, OpenDS et SUMO proposent une modélisation à partir de fichiers en

entrée et une simulation à partir d'un environnement graphique (pour SUMO, possibilité également de passer par un terminal de commandes). Bien qu'assez simple d'utilisation, une documentation est nécessaire afin d'écrire les fichiers d'entrée correctement. Toutefois, OpenDS est moyennement simple d'utilisation en raison d'une documentation un peu faible sur la conception de ses fichiers en entrée.

2.1.2.3 Simulation et analyse

Pour lancer une simulation et effectuer son analyse, il peut être intéressant d'avoir un environnement de simulation permettant d'effectuer certains réglages rapidement et efficacement. Il faut également savoir comment nous pouvons observer et analyser notre simulation (aspect graphique, courbes, exportation dans un fichier...). Enfin, il est nécessaire d'analyser les performances du simulateur qui peuvent être limitées sur une machine moins performante que les machines actuelles.

Environnement de simulation.

Comme nous l'avons dit précédemment, Aimsun et AnyLogic disposent d'un environnementf de simulation "user-friendly" constitué de nombreuses fonctionnalités classées par catégories. MovSim, SUMO, GAMA présentent une interface simple permettant de lancer, accélérer, ralentir, mettre en pause ou arrêter la simulation. En plus de ces éléments, AnyLogic et GAMA ont l'avantage de créer une interface adaptée à chaque utilisation (visuel seul, visuel et courbes...). Les interfaces de MATSIM et d'OpenDS permettent uniquement de lancer la simulation en choisissant les fichiers d'entrée de son choix.

Sortie de simulation.

Sur Aimsun tout comme sur AnyLogic, il est possible d'observer l'avancement de chaque moyen de transport en temps réel sur une route en deux ou trois dimensions, le tout dans un visuel très clair et personnalisable. Bien que l'aspect visuel ne soit pas aussi développé que pour les deux outils précédents, GAMA, MovSim et SUMO possèdent également cette fonctionnalité et peuvent également envoyer les résultats de la simulation dans un fichier de sortie. Sur OpenDS, il est possible d'observer la simulation en trois dimensions à partir d'un véhicule que l'on peut contrôler avec les touches du clavier. Sur l'outil d'origine, MATSIM peut uniquement

enregistrer les résultats dans un fichier de sortie. Toutefois, une extension, Via [2], permet d'avoir un aspect graphique personnalisable en deux dimensions. Celle-ci est payante pour un usage recherche (gratuit et limité pour un usage personnel).

Outils d'analyse.

Aimsun est constitué d'un grand nombre d'outils d'analyse prédéfinis. Il peut afficher plusieurs aspects de la route (densité, vitesse, pollution, consommation d'essence...) directement sur le visuel en deux dimensions ou encore via des courbes. AnyLogic, GAMA, OpenDS, MovSim peuvent étudier les résultats par l'intermédiaire de courbes ou directement sur la simulation (trajectoire du véhicule, vitesse...). GAMA peut effectuer ceci grâce à une extension permettant une étude de trafic routier [82]. MATSIM permet d'analyser la simulation via des courbes ou directement sur la simulation à partir de Via. SUMO manque d'un ensemble d'outils d'analyse permettant l'étude de ces résultats (nécessité d'effectuer l'analyse nous-mêmes à partir du fichier contenant les données).

Performances.

Il n'est pas toujours facile de déterminer les performances d'un simulateur car elles ne dépendent pas uniquement du nombre de véhicules mais aussi du niveau de complexité de l'environnement et des véhicules. Pour ces raisons, nous donnons un ordre de grandeur en nombre de kilomètres de voies saturées de véhicules. Ces véhicules ont un modèle de décision s'adaptant au prochain véhicule situé devant eux et changeant de voie en observant les véhicules environnants. Avec ces suppositions, Aimsun et AnyLogic permettent de simuler et d'observer facilement au moins 100 kilomètres de voies, au moins 20 kilomètres de voies pour GAMA, MATSIM, MovSim, SUMO et au moins 5 kilomètres de voies pour OpenDS, plus limité notamment de par son environnement en trois dimensions. C'est une limite qui permet d'observer des éléments de façon raisonnable, c'est-à-dire avec un temps simulé supérieur au temps réel de simulation.

2.1.2.4 Simulation de trafic

Dans le cas de l'échelle microscopique, il s'agit de voir si l'implémentation des lois de poursuite et des changements de voie a été faite dans le simulateur, les

possibilités en termes de modélisation des routes ainsi que la diversité des moyens de transport que l'on peut modéliser.

Lois de poursuite existantes.

MovSim est l'outil proposant le plus large choix de lois de poursuite et de modèles de changement de voie (Gipps [38], IDM [89], MOBIL [54]...). SUMO utilise les modèles de Gipps mais des extensions permettent un modèle dérivé de Krauss [3]. Aimsun utilise les modèles de Gipps également. GAMA utilise un modèle dérivé d'IDM. AnyLogic, MATSIM et OpenDS n'utilisent pas de modèles de lois de poursuite.

Modélisation de la route à partir de données géographiques.

A l'exception de MovSim, tous les logiciels peuvent modéliser une route à partir de données géographiques. Les solutions les plus simples et les plus complètes sont proposées sur Aimsun et AnyLogic qui permet d'importer les routes de son choix (à partir de données OpenStreetMap (OSM), GIS, CAD...). Il est par ailleurs possible de retirer les routes qui ne nous intéressent pas afin d'avoir un modèle sur-mesure. GAMA, MATSIM et SUMO permettent d'importer des fichiers au format shapefile et de choisir les types de routes souhaités ou non (outil assez intuitif sur MATSIM et SUMO, moyennement intuitif sur GAMA). OpenDS peut récupérer des routes et bâtiments construits à l'aide du logiciel CityEngine (version payante).

Modélisation manuelle de routes.

A l'exception de GAMA, tous les logiciels peuvent modéliser une route manuellement. Sur Aimsun et AnyLogic, la modélisation est extrêmement simple car leur interface est basée sur le *drag-n-drop*, ce qui permet donc de construire tout type de route (autoroute, voie d'insertion, rond-point...) à la souris. MATSIM, MovSim et SUMO passent par un ensemble de fichiers au format XML. Ainsi, on peut imaginer la route de son choix en l'assimilant à un graphe indiquant la direction des véhicules ainsi que le nombre de voies (MovSim respecte le standard OpenDrive [46]). Sur OpenDS, la modélisation manuelle peut se faire également via CityEngine.

Modélisation de routes particulières.

Seuls Aimsun, AnyLogic et SUMO permettent de modéliser des routes particu-

lières telles que les voies de bus, les rails et les passages pour piétons (également parkings pour Aimsun).

Diversité des moyens de transports.

Aimsun, AnyLogic et SUMO proposent la plus grande variété de moyens de transport : tous types de quatre roues, deux roues, trains et piétons. GAMA et MATSIM permet de modéliser tous types de quatre roues (sauf les bus pour GAMA) et les piétons. Quant à MovSim et OpenDS, ils peuvent créer uniquement des véhicules à quatre roues (excepté les bus).

Création de nouveaux moyens de transport.

Avec AnyLogic, bien qu'il possède déjà une grande variété de transports, il est très simple de créer le moyen de transport de son choix en imposant des règles précises (voies autorisées, vitesse maximale, accélération). GAMA possède également cette fonctionnalité en utilisant le langage GAML (simple de définir un agent plus petit qu'une voiture pour un piéton par exemple). Il est assez simple de le faire sur MATSIM, MovSim, OpenDS et SUMO directement dans le code source. Quant à Aimsun, il n'est pas possible de créer de nouveaux moyens de transport.

2.1.2.5 Simulation de systèmes multi-agents

L'aspect agent est considéré comme important car il peut être nécessaire d'avoir une diversité d'agents aux comportements différents. De plus, les interactions entre eux (communication, négociation) doivent être également possible.

Diversification du comportement des agents.

Sur AnyLogic, GAMA et SUMO, il est simple de donner un comportement précis à chaque agent et de définir des probabilités (chemin que doit effectuer le véhicule, vitesse maximale, probabilité de changer de voie, d'avoir un accident, de respecter les priorités...). Sur Aimsun, il est possible de modifier facilement les caractéristiques d'une portion (chemin que doit effectuer le véhicule, vitesse maximale, détection d'événements...). Sur MovSim, il est possible de choisir parmi plusieurs lois de poursuite et de modifier les paramètres de chacune facilement. Sur MATSIM, cela se fait à partir du fichier d'entrée où l'on peut définir le chemin que doit effectuer le véhicule mais pas de prises de décisions différentes (comportement prédéfini).

Sur OpenDS, il est assez simple de définir la trajectoire de chaque véhicule en lui donnant un ensemble de points à suivre.

Ajout de nouveaux comportements.

Sur AnyLogic, il est possible de modéliser le comportement du véhicule selon notre choix : par une loi de poursuite que nous définissons nous-même ou par l'intermédiaire d'un arbre de décision. Sur GAMA, au-delà de la loi de poursuite, il est possible d'affecter simplement à chaque véhicule son propre comportement. En effet, chaque agent entre dans une catégorie d'espèces (*species*) que l'on peut définir grâce au langage GAML. Sur MovSim, il est simple de créer une nouvelle classe Java définissant un nouveau comportement à suivre pour un ou plusieurs véhicules. Via un kit de développement, il est assez simple de définir ses propres lois de poursuite sur Aimsun. Sur OpenDS, on peut effectuer des modifications de vitesses et d'accélérations selon les situations à partir d'une classe Java existante. De plus, une mise à jour récente permet une fonctionnalité d'auto-pilotage afin d'avoir des véhicules autonomes. Sur SUMO, il n'est pas aussi simple de proposer son propre comportement pour chaque véhicule ou de simuler des situations imprévues (accidents, perte de communication, véhicule d'urgence. . .). Enfin, sur MATSIM, il est plus difficile de créer une diversité de comportements pour les agents. En effet, l'outil paraît plus adapté pour une étude de flux de trafic ce qui fait que la flexibilité de l'outil se situe plus, par exemple, sur le choix de la densité de véhicules sur une portion de route ou encore le chemin qu'ils réalisent selon l'heure de la journée par exemple.

Communication.

Exceptés MATSIM et OpenDS, tous les simulateurs ont la capacité de faire communiquer les différents types d'agents entre eux. Aimsun possède une extension permettant la conception de véhicules connectés (ITS). Sur SUMO, l'extension Veins [81], permet la communication entre les véhicules. Sur AnyLogic, il est possible d'envoyer des messages aux autres agents. Sur GAMA, une extension a été créée respectant les standards de la FIPA avec la possibilité d'utiliser le langage de communication ACL (*Agent Communication Language*) [35] ainsi que la négociation entre agents grâce à l'implémentation du modèle de négociation CNP (*Contract Net*

Protocol) [78]. Enfin, une contribution récente [44] reprend la base de MovSim tout en ajoutant un aspect multi-agent et communicant à celui-ci (non disponible pour le moment).

2.1.2.6 Aller plus loin

Il est évident que nous n'allons pas trouver un outil de simulation répondant exactement à l'ensemble des contraintes imposées par l'approche que nous souhaitons développer. Il est donc nécessaire de voir si un tel outil peut offrir une liberté de personnalisation, de création, et s'il est possible d'implémenter de nouveaux éléments facilement et efficacement. Ces nouveaux éléments, sous forme de modules, peuvent être intéressants pour la communauté du simulateur choisi et enrichir celui-ci.

Liberté de personnalisation et de création.

AnyLogic et GAMA sont les outils proposant la plus grande liberté de création. En effet, au-delà de la diversité des routes, moyens de transport et comportements, il est possible de créer toutes sortes d'éléments (agent virtuel, obstacle, accident, événement imprévu) et de leur donner l'aspect visuel de notre choix. MovSim, bien que limité en termes de moyens de transport, offre une bonne liberté en termes de construction de routes et de véhicules aux comportements différents. Aimsun propose une grande liberté de personnalisation et de création que ce soit au niveau de la modélisation des routes ou des moyens de transport ainsi qu'une assez bonne flexibilité sur les comportements de véhicules. OpenDS permet assez facilement de créer un contrôleur pour un véhicule. Toutefois, de par son aspect plus réaliste, il est plus difficile de lui attribuer des actions élémentaires (changement de voie, tourner à droite. . .). SUMO et MATSIM sont moins personnalisables que les autres simulateurs. Toutefois, SUMO offre une bonne liberté de création de routes et de moyens de transport différents.

Création d'extensions (plug-ins).

Sur Aimsun, la construction de fonctionnalités sous forme de blocs rend le logiciel de simulation modulaire et il est possible d'obtenir une application en Java du modèle créé. GAMA, MATSIM, MovSim, OpenDS et SUMO permettent la création d'extensions à partir du code source. Sur Aimsun il est possible, grâce aux kits

de développement, de créer ses propres extensions sur certains éléments (lois de poursuite, changements de voie...). On peut également automatiser un ensemble de tâches via des scripts en Python.

Clarté du code source.

GAMA, MovSim et SUMO proposent un code clair et bien commenté, notamment dans les parties du code susceptibles d'être modifiées. MATSIM et OpenDS possèdent un code source bien commenté mais manquent parfois de clarté sur les noms de certaines classes. Aimsun et AnyLogic sont des logiciels propriétaires, le code source n'est donc pas disponible.

2.1.3 Tableau comparatif et choix

A partir des critères d'évaluation cités et détaillés précédemment, il convient de proposer un tableau comparatif (Tableau 2.1) résumant les possibilités de chacun selon les critères détaillés précédemment. Pour plus de lisibilité, nous avons créé un code couleur. Ainsi, pour chaque case du tableau, les couleurs iront du rouge pour une caractéristique négative au vert pour une caractéristique positive, représentant un sous-critère d'un outil de simulation.

Grâce à ce code couleur, on remarque assez rapidement que deux simulateurs se démarquent : Aimsun et AnyLogic. Ce sont deux outils très complets permettant la modélisation et la personnalisation d'un modèle très rapidement. Malheureusement, ce sont deux outils payants et propriétaires. L'ensemble des simulateurs restants ne proposent pas des fonctionnalités aussi riches que Aimsun et AnyLogic. Notre choix doit donc être un compromis entre les fonctionnalités proposées, la facilité ainsi que la liberté de création. SUMO et MATSIM sont deux bons outils de modélisation de trafic mais se limitent très rapidement lorsque l'on souhaite gérer des situations "inhabituelles". De plus, MATSIM se présente plus comme un outil d'étude de flux de trafic. OpenDS propose une alternative intéressante. En effet, bien que ce soit un simulateur de conduite, il permet la modélisation de véhicules plus réalistes à travers un environnement en trois dimensions. Son code est facilement modifiable et il devient facile de donner un comportement particulier à un véhicule. Il peut donc être intéressant de se diriger vers cet outil si l'on souhaite

avoir des modèles plus réalistes à l'avenir.

Les deux outils restants, MovSim et GAMA, ne sont pas aussi riches et simples d'utilisation que les deux simulateurs payants. On peut donc imaginer deux façons de voir les choses : partir d'un simulateur orienté trafic auquel on ajoute un aspect agent (MovSim) ou partir d'un simulateur agent auquel on ajoute des aspects trafic. GAMA possède une extension proposant un aspect trafic [82]. De même, bien que non disponible en ligne, une extension a été créée sur MovSim proposant des aspects de communication et multi-agent [44]. Toutefois, GAMA permet la communication entre agents, propose une grande liberté de création, de personnalisation et permet de définir des comportements différents pour chaque agent de façon efficace. C'est pour ces raisons que nous nous sommes orientés vers GAMA comme outil pour la modélisation et la simulation des véhicules autonomes communicants.

		Aimsun	AnyLogic	Gama	MatSIM	MovSim	OpenDS	SUMO
Date de création		1993	2000	2010	2005	2010	2012	2001
Langage de programmation		Langage propriétaire / C++	Langage propriétaire / Java	Java / GAML	Java	Java	Java	C++
Caractéristiques générales	Prix	Version gratuite limitée / Version payante	Version gratuite (hors recherche) / Version payante	Gratuit	Gratuit	Gratuit	Version gratuite / Version payante plus complète	Gratuit
	Open-source	Non	Non	Oui	Oui	Oui	Oui	Oui
	Multiplateforme	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	Installation	Simple	Simple	Simple	Simple	Assez simple	Simple	Assez simple
Prise en main	SImplicité	Très simple	Très simple	Simple	Assez simple	Assez simple	Moyennement simple	Assez simple
	Documentation	Très complete	Très complete	Complète	Complète	Assez complète	Assez complète	Complète
	Communauté	Très active	Très active	Active	Active	Active (concepteurs)	Active	Active
Simulation et analyse	Environnement de simulation	Très complet	Très complet	Complet	Assez complet	Complet	Assez complet	Complet
	Sortie de simluation	Très complet	Très complet	Complet	Complet (outil payant)	Complet	Complet	Complet
	Outils d'analyse	Très complets	Complets	Complets	Complets (outil payant)	Complets	Complets	Limités
	Performance	Très puissant	Très puissant	Puissant	Puissant	Puissant	Assez puissant	Puissant
Simulation de trafic	Lois de poursuite existantes	Un	Non	Un	Non	Un grand nombre	Non	Plusieurs
	Modélisation de la route à partir de données géographiques	Très simple et très complète	Très simple et très complète	Moyennement simple et assez complète	Assez simple et assez complète	Non	Assez simple et complète (logiciel tiers)	Assez simple et assez complète
	Modélisation manuelle de la route	Très simple et très complète	Très simple et très complète	Non	Assez simple et assez complète	Assez simple et assez complète	Assez simple et complète (logiciel tiers)	Assez simple et très complète
	Modélisation de routes particulières	Très complete	Complète	Non	Non	Non	Non	Complète
	Diversité des moyens de transports	Quatre roues et deux roues, trains, piétons	Quatre roues et deux roues, trains, piétons	Véhicules, poids lourds, piétons	Quatre roues, piétons	Véhicules, poids lourds	Véhicules, poids lourds	Quatre roues et deux roues, trains, piétons
	Création de nouveaux moyens de transport	Impossible	Très simple	Simple	Assez simple	Assez simple	Assez simple	Assez simple
Simulation multi-agent	Diversification du comportement des agents	Simple et complète	Simple et complète	Simple et complète	Simple et assez complète	Assez simple et complète	Assez simple et assez complète	Simple et complète
	Ajout de nouveaux comportements	Assez simple	Très simple	Simple	Moyennement simple	Simple	Assez simple	Moyennement simple
	Communication	Oui	Oui	Oui	Non	Oui (non disponible en ligne)	Non	Oui
Aller plus loin	Liberté de personnalisation et de création	Bonne	Très bonne	Très bonne	Moyennement bonne	Assez bonne	Bonne	Moyennement bonne
	Création d'extensions (plug-ins)	Assez simple mais limité	Simple	Assez simple	Assez simple	Assez simple	Assez simple	Assez simple
	Clarté du code source	Code source non disponible	Code source non disponible	Clair et bien commenté	Assez clair et bien commenté	Clair et bien commenté	Assez clair et bien commenté	Clair et bien commenté

-				+

TABLEAU 2.1: Tableau comparatif des simulateurs présentés [79].

2.2 Un outil de simulation : GAMA

2.2.1 Présentation

GAMA est un simulateur multi-agent gratuit et open-source permettant la modélisation et la simulation d'un ensemble d'agents. Un agent est une entité qui possède des caractéristiques qui lui sont propres et peut également interagir avec les autres agents de son environnement. Plusieurs agents aux mêmes comportements peuvent être modélisés permettant l'étude de leurs interactions entre-eux. Chaque

agent est mis à jour grâce à un pas de temps propre à l'outil de simulation. Cet agent peut être statique ou dynamique, communicant ou non, coopératif ou non. Ainsi, il est possible d'imaginer un système constitué :

— d'un agent statique représentant la route (pouvant éventuellement communiquer avec son environnement) ;

— d'agents dynamiques représentant les véhicules connectés et non connectés (communicants et coopératifs avec certains types d'agents uniquement) ;

— d'autres éléments pouvant s'ajouter à l'environnement : piétons, autres moyens de transport, bornes sur les routes...

2.2.1.1 Modélisation

Pour mettre en place ce système, GAMA utilise un langage orienté agents appelé GAML (basé sur le langage Java). C'est un langage de programmation simple et intuitif pouvant être utilisé par tout chercheur, même non informaticien. Ce langage permet de créer un système constitué d'agents très rapidement et simplement. Comme nous pouvons le voir en figure 2.1, un fichier GAML est constitué des éléments suivants :

— *global* : il va contenir toutes les variables globales du modèle, c'est-à-dire les éléments qui seront accessibles dans tout le modèle. Ces variables peuvent permettre à l'utilisateur de définir l'initialisation du modèle mais aussi d'établir quelques statistiques pour son analyse ;

— *species* : il va contenir les caractéristiques d'un agent en particulier. Tout comme dans le paradigme objet, l'agent va contenir différentes caractéristiques qui lui sont propres. On peut également à chaque pas de temps mettre à jour un ensemble de variables grâce à l'attribut *reflex*. Chaque agent peut-être constitué d'un ensemble de *skills* permettant d'ajouter des caractéristiques supplémentaires à l'agent. Ainsi, *moving* permet de faire avancer un agent et de le diriger vers un point cible. On peut enfin donner un aspect visuel à l'agent grâce à l'attribut *aspect* ;

— *experiment* : il va permettre de lancer une simulation du modèle. Cette simulation peut se faire de façon indirecte en enregistrant les données de chaque

agent dans un fichier ou de façon directe en ajoutant une interface graphique du modèle grâce à l'attribut *display*.

New Model.gaml

```
1 model NewModel
2
3 global {
4     /** Insert the global definitions, variables and actions here */
5     int nb_cars <- 5;
6
7     init {
8         create cars number: nb_cars {
9             init_pos <- rnd(0,50);
10            location <- {init_pos, 30.0};
11            length_car <- rnd(2.5,4.0);
12            speed_th <- rnd(1.0,3.0);
13        }
14    }
15 }
16
17 species cars skills: [moving] {
18    int id;
19    float speed_th;
20    int init_pos;
21    float length_car;
22    point target <- {90.0,30.0};
23
24    reflex update_location {
25        speed <- speed_th;
26        do goto target: target;
27    }
28
29    aspect base {
30        draw rectangle (length_car,1) color: #black;
31        draw circle(1) at: target color: #red;
32    }
33 }
34
35 experiment NewModel type: gui {
36    /** Insert here the definition of the input and output of the model */
37    output {
38        display carSimulation type: java2D background: #lightgreen
  refresh: every(1 #cycle) {
39            species cars aspect: base refresh: every(1 #cycle);
40        }
41    }
42 }
43
```

Page 1

FIGURE 2.1: Squelette d'un fichier GAML sur GAMA avec un début de simulation de véhicules.

2.2.1.2 Analyse

Avec GAMA, il est simple d'observer des propriétés selon un ensemble de critères et de les valider par simulation. Le simulateur permet l'analyse du modèle ainsi créé via (voir 2.2) :

— une interface graphique : on peut directement observer les phénomènes visuellement en deux ou trois dimensions ;

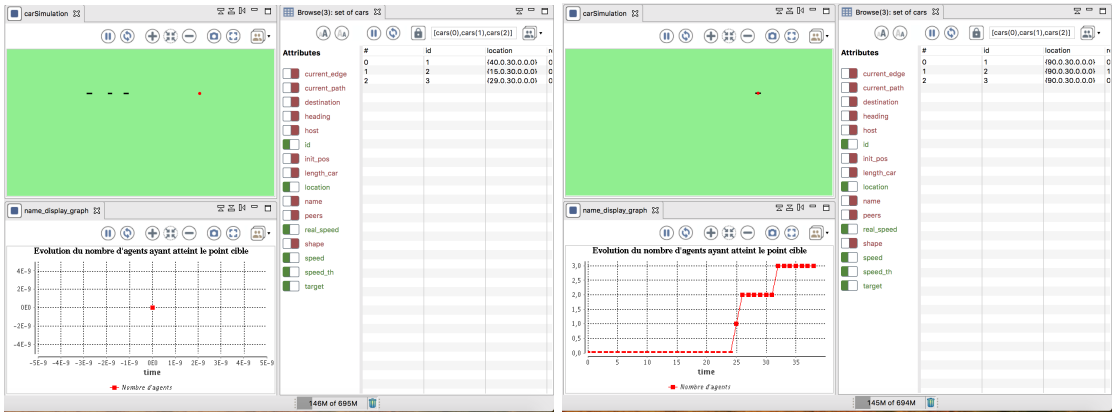


FIGURE 2.2: Analyse de trois véhicules se dirigeant vers le point cible au début de la simulation (à gauche) et à la fin (à droite).

— des courbes : on peut observer la variation de certaines caractéristiques grâce à des courbes. On peut ainsi voir l'évolution de ces caractéristiques en temps réel ;

— un tableau d'agents : à chaque lancement d'une simulation, il est possible d'accéder à un tableau reprenant l'ensemble des agents et leurs caractéristiques dans un tableau et ainsi voir leur évolution au cours du temps.

GAMA propose un ensemble complet de fonctionnalités pour la modélisation, la simulation et l'analyse d'un système multi-agent. De plus, différentes extensions sont disponibles, parmi lesquelles une implémentation du langage FIPA-ACL [35] permettant la communication entre agents, ou l'implémentation d'une extension de trafic routier [82].

2.2.2 Fonctionnement de l'extension trafic

Une extension de trafic routier est proposée sur GAMA répondant à un manque d'outils pour les non-informaticiens pour modéliser et analyser un flux de trafic comme le font MATSIM ou encore SUMO. De plus, bien que GAMA puisse charger des routes sous forme de fichiers géométriques via des données GIS, il ne permet pas de considérer le comportement du véhicule sur une échelle plus basse. En effet, il n'est pas possible initialement d'étudier le changement de voie ni la prise en compte de feux tricolores par exemple. Une extension de trafic a donc été développée par les concepteurs de la plate-forme apportant deux améliorations : une pour la modélisation des routes et l'autre pour la modélisation des véhicules.

2.2.2.1 Modélisation de la route

Deux *skills* ont été ajoutés sur GAMA : *road Skill* et *roadNode Skill*. Ils permettent la définition de routes à la manière d'un graphe orienté où chaque noeud peut indiquer un point de départ, d'arrivée ou de transition. On peut donc partir d'une route existante à partir de données géographiques disponibles sur le web (OpenStreetMap, Google Earth. . .). De plus, il est possible d'obtenir des routes à plusieurs voies et définir les différentes parties qui nous intéressent sur les données initiales (n'afficher que les autoroutes, prendre en compte uniquement les routes secondaires. . .).

2.2.2.2 Modélisation des véhicules

Pour la modélisation des véhicules, un *skill* a été créé : *advanced driving skill*. Il permet de modéliser des véhicules avec plusieurs caractéristiques utiles :

— la possibilité de planifier son itinéraire : à la manière d'un système de navigation, le véhicule sait vers quelle route il se dirige à chaque instant ;

— la définition du comportement longitudinal du véhicule : à partir d'un modèle de décision inspiré de la loi de poursuite *Intelligent Driver Model* (IDM) [89], il est possible d'avoir un véhicule ajustant sa vitesse par rapport au véhicule qu'il suit ;

— la variabilité des comportements : chaque véhicule peut changer de voie à tout moment en définissant la probabilité de changer de voie mais aussi la probabilité de respecter le code de la route (feux tricolores, stops. . .).

2.2.2.3 Limitations

Grâce à cette extension, il est possible, à partir d'une route existante, de définir un scénario avec des véhicules ayant des comportements différents pour étudier le flux de trafic. Toutefois, tout comme MATSIM et SUMO, il n'est pas possible de définir directement son propre comportement car on est limité par le modèle de décision existant (*IDM*). Cependant, l'outil GAMA le permet grâce à son aspect agent. De plus, il n'est pas possible de créer des routes manuellement, c'est-à-dire, imaginer une route sur un dessin et la reproduire directement. Enfin, l'extension ne prend pas en compte les routes spécialisées (passage piéton, ligne de chemin de fer, arrêt de bus. . .) ce qui permettrait de se rapprocher encore plus de l'environnement existant

dans la vie réelle. C'est pour ces raisons que nous avons décidé de développer une extension pour GAMA.

2.3 Contribution : extension de GAMA

L'objectif de cette extension est de répondre aux besoins suivants :

— La construction manuelle de l'environnement routier (comme le propose l'outil SUMO) ;

— L'ajout d'éléments de communications spécifiques aux ITS (comme le propose l'extension de *MovSim* [44]) ;

— La conception simple d'un système de prise de décision pour les véhicules (à la manière de l'outil AnyLogic) ;

Ainsi, il sera possible de modéliser un large éventail d'environnements et de simuler un grand nombre de situations quel que soit le niveau d'autonomie et de communication de chaque véhicule.

2.3.1 Représentation de l'environnement

2.3.1.1 Fonctionnement

L'objectif ici est de donner la possibilité à l'utilisateur de reproduire un scénario précis d'une route avec des caractéristiques que l'on donnera en entrée. Cette conception de la route doit être un exercice simple et intuitif afin de ne pas trop impacter son but principal : la simulation de scénarios.

Parmi les outils open-source présentés précédemment pour modéliser manuellement des routes, la plupart assimilent la route à un graphe décrit sous forme d'un fichier XML. Toutefois, la représentation via des fichiers XML peut être rapidement laborieuse notamment pour des utilisateurs non informaticiens. L'idée est donc de proposer à l'utilisateur un langage simplifié et intuitif qu'il utilisera à partir d'un fichier texte. Ce langage sera traduit sous forme d'un fichier XML. Enfin, GAMA traduira ce fichier XML afin de construire une route conformément aux attentes de l'utilisateur.

Pour la traduction du fichier texte au fichier XML, nous souhaitons utiliser un compilateur. Celui-ci va dans un premier temps procéder à une analyse lexicale du texte en entrée afin de le convertir en un ensemble de *tokens* (unités lexicales). A partir de ces *tokens*, l'analyseur syntaxique (*parser*) va créer une représentation sous forme d'arbre illustrant la structure du programme. Enfin, le compilateur va générer du code bas niveau afin d'effectuer la traduction.

Parmi les outils existants, Lex et Yacc [57] sont les références en terme d'analyse lexicale et syntaxique. On peut également citer ANTLR [36] permettant de générer du code pour de nombreux langages (Java, C, Python,...). JavaCC [41] possède les mêmes caractéristiques que ce dernier mais n'est exécutable que sous Java. Toutefois, de par sa popularité, il possède une documentation très riche et une communauté très active. De plus, il est très efficace dans la description des erreurs lexicales et syntaxiques. C'est pour ces raisons que nous nous sommes orientés vers ce dernier.

Enfin, la traduction d'un fichier XML sous GAMA se fait uniquement ligne par ligne. Ainsi, il est nécessaire de créer une fonction sur GAMA permettant une meilleure reconnaissance de ces fichiers. Cette structuration permet de récupérer très facilement l'ensemble des éléments donnés en entrée. Ces données sont temporairement stockées dans des tableaux accessibles initialement afin de construire l'environnement avant le début de la simulation.

2.3.1.2 Cahier des charges

Notre objectif était de pouvoir construire un ensemble de routes représentatives d'un large panel de cas. Nous nous sommes concentrés sur des situations d'autoroutes dont les éléments constitutifs peuvent être découpés en plusieurs segments :

— route normale (*Normal*) : une route droite qui possède une ou plusieurs voies ;

— élargissement de voie (*Enlarge*) : une route qui s'élargit par la gauche ou la droite ;

— rétrécissement de voie (*Reduce*) : une route qui se rétrécit par la gauche ou la droite ;

— voie d'insertion ou d'accélération (*OnRamp*) : une route permettant de rejoindre la route principale par la gauche ou la droite ;

— sortie d'autoroute (*Exit*) : une route permettant de quitter la route principale (généralement par la droite dans les autoroutes françaises).

De plus, plusieurs informations seront nécessaires pour la construction de ces routes :

— un identifiant (unique) ;

— le type de route souhaité ;

— le nombre de voies (notamment pour les routes de type *Normal* et *Exit*) ;

— la longueur de la route ;

— la largeur de chaque voie de la route ;

— la vitesse maximale autorisée sur la route ;

— le sens par lequel on place les routes de type *Enlarge*, *Reduce* et *OnRamp* ;

— la taille de la portion de route où l'on peut s'insérer (pour le type *OnRamp*) ou sortir (pour le type *Exit*) ;

Enfin, pour le positionnement manuel de chaque véhicule, deux solutions seront proposées : soit placer un seul véhicule, soit placer un train de véhicules permettant d'ajouter rapidement un ensemble de véhicules ayant les mêmes caractéristiques.

Pour le positionnement manuel d'un véhicule, les informations suivantes sont nécessaires :

— les identifiants indiquant la route et la voie où se trouve le véhicule ;

— le comportement désiré pour ce véhicule ;

— sa position longitudinale initiale (le véhicule est situé par défaut au centre de la voie, c'est-dire entre les lignes séparant chaque voie ou route) ;

— la vitesse initiale du véhicule.

Pour l'ajout d'un train de véhicules, on doit ajouter les éléments suivants :

— la position du véhicule situé en queue du train de véhicules ;

— la position maximale du véhicule situé en tête du train de véhicules ;

— le nombre de véhicules situés dans le train (limité par le paramètre précédent) ;

— l'inter-distance entre les véhicules.

Ces différentes informations seront totalement paramétrables. Toutefois, il est possible que le changement de certaines de ces informations ne soit pas nécessaire. Dans ce cas là, une valeur par défaut sera attribuée pour celles-ci.

2.3.1.3 Langage de modélisation et de positionnement des routes et des véhicules

Afin d'instancier les routes et les véhicules de façon intuitive et simple, nous avons décidé de mettre en place un langage permettant à l'utilisateur de définir facilement les différentes entités qu'il souhaite créer. Ce langage doit contenir les différentes informations détaillées dans le cahier de charges défini précédemment. Le langage proposé a donc la forme définie ci-dessous en 2.1 :

$$X_{id} < ty > \{p_i, \ldots, p_n\} \ (v_i, \ldots, v_n); \tag{2.1}$$

Dans l'exemple défini ci-dessus, on construit un élément de type X (lettre R pour une route, C pour un véhicule, MC pour un train de véhicules) ayant pour identifiant id. Cet élément est de type ty constitué de plusieurs paramètres dont n définis manuellement. A chaque paramètre p_i est associée une valeur v_i. L'avantage de ce langage est qu'il permet d'instancier toutes sortes d'agents que l'on souhaite créer (au-delà de ceux déjà définis). Ainsi, dans le cas de routes, le type ty représente son type et dans le cas de véhicules, son comportement. L'autre avantage de ce langage est la liberté qu'on laisse à l'utilisateur de choisir les paramètres qu'il souhaite donner et l'ordre dans lequel il les définit. Les autres paramètres ont une valeur définie par défaut dans l'outil. On peut donc construire une route en écrivant uniquement son nom, son identifiant et son type.

Comme on peut le voir en Figure 2.3, la définition des routes par le langage est traduite en un fichier XML contenant toutes les informations nécessaires à la modélisation de celle-ci sur GAMA. Enfin, nous avons ajouté une gestion des erreurs en vérifiant par exemple que les paramètres ont des valeurs conformes à leurs types. Si un type de route ou un constructeur est inconnu, l'utilisateur est également notifié.

$$R_1 < N > \{nb\}(2);$$
$$R_2 < E > \{nb,\ ws,\ beg\}(3,\ 30,\ 1);$$
$$R_3 < I > \{w,\ right\}(60,\ 1);$$

```
<road type="<NORMAL>" id="1" nbLane="2" width="100" height="3.5" attachedFirst="0"
speed="36.11" width_spec="20" side="L" attachedRight="0" attachedLeft="0" />

<road type="<ENLARGE>" id="2" nbLane="3" width="100" height="3.5" attachedFirst="1"
speed="36.11" width_spec="30" side="L" attachedRight="0" attachedLeft="0" />

<road type="<INSERTION>" id="3" nbLane="1" width="60" height="3.5" attachedFirst="0"
speed="36.11" width_spec="20" side="L" attachedRight="1" attachedLeft="0" />
```

FIGURE 2.3: En haut, le langage définissant trois routes. En bas, la traduction du langage en un fichier XML contenant toutes les informations relatives au cahier des charges de l'utilisateur.

Pour positionner les routes les unes par rapport aux autres, nous avons défini quatre types d'opérateurs :

— l'opérateur '.' de la forme $\{R_i.R_j\}$: signifie que l'on aura la route R_i suivie de la route R_j ;

— l'opérateur '/' de la forme $\{R_i/R_j\}$: signifie que la route R_i sera située au-dessus de la route R_j ;

— les opérateurs '(' et ')' de la forme $\{(R_i)\}$: permettent de regrouper un ensemble de routes afin d'y appliquer les opérateurs "." et "/" ;

— l'opérateur ';' de la forme $\{R_i;\}$: signale la fin d'un groupe de routes. Ainsi, l'ensemble de routes construites après cet opérateur sera positionné en dessous du groupe précédent sur GAMA.

Comme on peut le voir en Figure 2.4, on a un groupe de routes qui se construit de la gauche vers la droite et du haut vers le bas. Chaque route possède un identifiant unique ce qui permet de positionner facilement les véhicules. De plus, cela donne la possibilité de créer un itinéraire de route pour chaque véhicule pour ainsi donner un objectif précis à chacun.

2.3.1.4 Modélisation de capteurs sur la route

Comme nous l'avons dit précédemment, il est possible avec ce langage de représenter tous types d'agents en plus des véhicules et des routes. On peut donc

$$
\begin{array}{l}
R_1 < N > \{nb\}(2); \\
R_2 < E > \{nb,\ ws,\ beg\}(3,\ 30,\ 1); \\
R_3 < I > \{w,\ right\}(60,\ 1); \\
(R_1/R_3).R_2;
\end{array}
$$

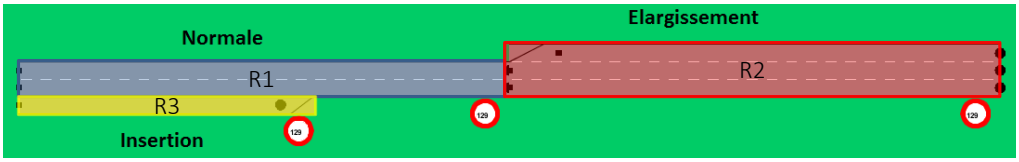


FIGURE 2.4: Positionnement des routes à partir du cahier des charges spécifié par l'utilisateur grâce au langage.

imaginer une route connectée, c'est-à-dire qui analyse son environnement et qui diffuse un ensemble d'informations nécessaires pour les véhicules. La diffusion de ces informations peut se faire via des unités de bord de route (*On-Board Unit*) qui vont être placées à des endroits précis de la route.

Ces unités de bord de route peuvent facilement se positionner grâce au langage cité précédemment. Pour cela, les informations suivantes sont nécessaires :

— l'identifiant de la route où se trouve l'*OBU* ;

— sa position par rapport à cette route ;

— sa zone de perception (en mètres) ;

— la fréquence d'envoi des informations.

Ainsi, on pourrait imaginer la création d'un *OBU* à partir du langage de la façon suivante en 2.2 :

$$O_1\{idr,\ w,\ per,\ fr\}(2,\ 50,\ 200,\ 0.5); \tag{2.2}$$

où l'on obtient une unité de bord de route ayant l'identifiant 1, posé à w mètres de la route ayant l'identifiant idr, avec un rang de perception de per mètres et une fréquence d'envoi de messages de fr secondes.

Il est donc tout à fait possible d'imaginer un grand choix d'éléments connectés à la route qui peuvent s'initialiser très facilement grâce au langage créé. Le fonctionnement de ces éléments connectés et leur aspect visuel seront ensuite développés sur GAMA. On arrive à une modélisation manuelle des routes et des véhicules proche de l'outil SUMO bien qu'elle ne soit pas aussi complète et on dispose d'une grande

liberté concernant la mise en place d'éléments connectés pour les Systèmes de Transport Intelligents à la manière de *MovSim*.

2.3.2 Prise de décision

2.3.2.1 Raisonnement

Dans la grande majorité des simulateurs de trafic routier, il est possible de dérouler un ensemble de scénarios à partir de comportements prédéfinis au niveau des véhicules. Toutefois, il est généralement difficile de modifier le modèle de décision propre au véhicule, permettant uniquement une analyse de flux de trafic. L'objectif est d'avoir la possibilité de mettre en place le modèle décisionnel d'un véhicule. L'outil de simulation s'en rapprochant est AnyLogic qui permet la modélisation multi-méthode, c'est-à-dire la possibilité de combiner trois grandes approches dans un même modèle : la modélisation multi-agent, de systèmes dynamiques et par événements discrets. L'intérêt est de pouvoir représenter un modèle à des niveaux d'abstraction différents.

Afin de se rapprocher de ce type de modélisation, l'objectif est de concevoir un agent représentant un véhicule contenant des caractéristiques principales (position, vitesse, accélération...) et également un ensemble de caractéristiques permettant d'avoir une représentation plus précise de son environnement (détermination des véhicules environnants, obstacles...). On peut concevoir alors un nouveau modèle hérité du modèle générique qui contiendra le modèle comportemental du véhicule. La prise de décision fonctionne donc selon un va-et-vient entre le modèle générique et le modèle spécialisé de la façon suivante (voir figure 2.5) : tout d'abord, le modèle de véhicule générique reçoit l'ensemble des données du véhicule (étape 1) à un instant donné (semblable à une boîte noire). Celui-ci calcule la nouvelle accélération et modifie ses propres caractéristiques selon le modèle décisionnel (étape 2). Cette accélération est envoyée au modèle générique (étape 3) qui a pour rôle de mettre à jour les caractéristiques principales de l'agent (étape 4).

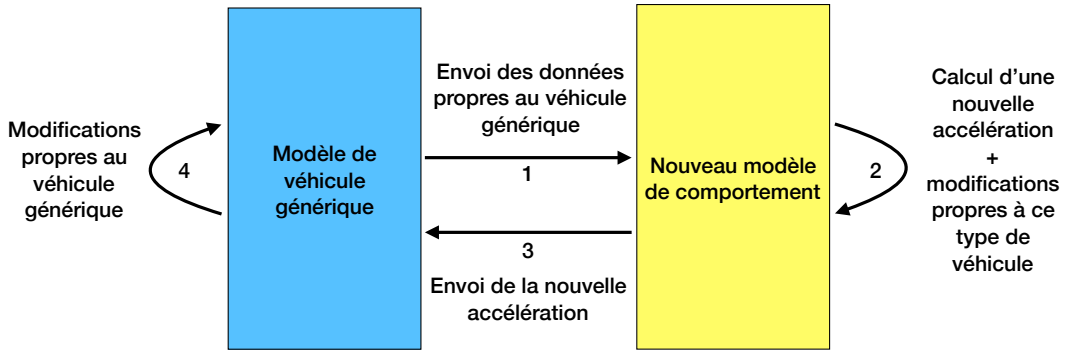


FIGURE 2.5: Fonctionnement global de la prise de décision.

2.3.2.2 Représentation sur l'outil et mise à jour

Afin de construire le modèle de décision de son choix, il est important de déterminer les caractéristiques propres au véhicule. Comme on peut le voir dans le diagramme de classes en Figure 2.6, nous distinguons trois types de classes de véhicules de base :

— *ObjectCar* : il contient toutes les caractéristiques de base du véhicule ainsi que les fonctions utiles pour celui-ci. De plus, il possède le "*skill*" *moving* permettant notamment de se diriger vers un point cible (chaque fin de voie contient un point cible) ;

— *Car* : hérite des caractéristiques d'*ObjectCar* et contient l'ensemble des *reflexes* du véhicule. Il est responsable de la mise à jour des caractéristiques du véhicule à un instant donné et permet ainsi de créer la boîte noire ;

— *VirtualCar* : hérite également des caractéristiques d'*ObjectCar* et permet de définir un véhicule virtuel associé à un élément de type *Car* afin d'observer différents aspects, comme une prédiction future d'un véhicule par exemple. Il est plus léger que *Car* car il n'est pas obligé de se mettre à jour à chaque pas de temps et il n'est pas considéré comme un véhicule présent sur la route.

Afin de créer son modèle de décision, il suffit de créer une nouvelle classe héritée de *Car* (*NewDecisionModel* sur la Figure 2.6). On peut y définir de nouvelles caractéristiques, de nouvelles fonctions et de nouveaux *reflexes*. Nous avons créé un modèle reprenant la loi de poursuite *IDM* et le modèle de changement de voie *MOBIL* (*IdmMobilModel* sur la Figure 2.6). Il est important de préciser que les *re-*

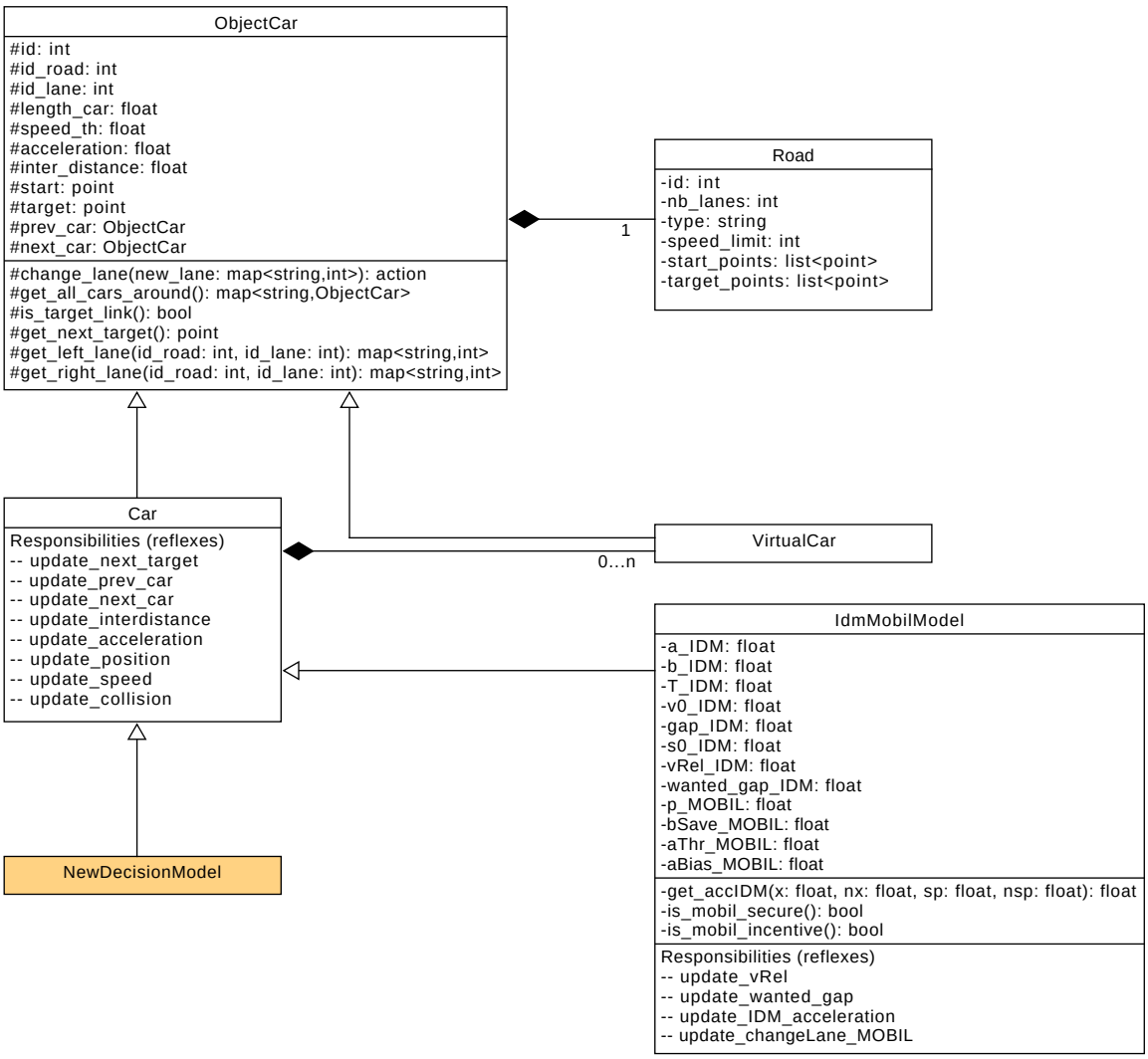


FIGURE 2.6: Diagramme de classes de l'extension.

flexes de l'élément fils se mettent à jour avant l'élément mère. Ainsi, lorsque notre modèle de décision a effectué ses calculs, le modèle générique prend le relais pour mettre à jour la valeur de la nouvelle accélération, la position du véhicule ainsi que sa vitesse.

Il est donc simple de concevoir son propre modèle de décision et de le représenter de la façon de son choix (équation différentielle, loi de poursuite, arbre de décision, théorie des jeux. . .). De plus, il est tout à fait possible et très facile d'ajouter des éléments sur *ObjectCar* et *Car* afin d'enrichir le fonctionnement du véhicule donnant ainsi plus de liberté sur le niveau d'abstraction que l'on souhaite définir. Cela n'altère pas son fonctionnement initial car ce sont des éléments qui vont améliorer la perception du véhicule (ajout de nouveaux capteurs).

2.3.3 Optimisations et performances

L'ajout d'un modèle de décision pour chaque véhicule va complexifier le système impliquant un temps de simulation plus long. L'objectif ici est de trouver des moyens d'optimiser le système afin de réduire ce temps de simulation offrant ainsi plus de liberté sur la conception du système. A partir d'indicateurs, on analysera la performance de plusieurs modèles en variant la complexité du système.

2.3.3.1 Moyens d'optimisation de cette extension

Plusieurs éléments ont été implémentés dans cette extension et des optimisations ont été nécessaires afin d'accélérer le temps de simulation. Parmi elles :

— la mise à jour de l'environnement (routes et véhicules) ;

— la mise en place des véhicules virtuels permettant de mettre en évidence une décision future d'un véhicule ;

— la détermination des véhicules qui précédent et qui suivent ;

Grâce à une modélisation initiale des routes, celle-ci n'a pas (ou très peu) d'impact sur le temps de simulation. En effet, cela peut se représenter comme une boîte noire disponible tout le long de la simulation par tous les véhicules. L'optimisation va donc reposer sur la fréquence à laquelle les véhicules vont avoir à accéder à cette boîte noire. De plus, la séparation du modèle de véhicule en deux (véhicule générique et nouveau modèle) permet d'avoir une représentation modulaire du système et cela permet d'éviter d'alourdir le modèle de base.

Concernant les véhicules virtuels, ils ne se mettent jamais à jour automatiquement. Ils permettent de représenter une prédiction temporaire d'un véhicule facilement. Il est possible de le mettre à jour si cette prédiction suit un comportement particulier de la même façon qu'un vrai véhicule. Lorsque le véhicule associé quitte la route, les véhicules virtuels sont détruits.

L'une des possibilités du véhicule est de connaître les véhicules situés autour de lui. Par exemple, pour déterminer le véhicule qui suit, une solution triviale de ce problème est d'observer parmi tous les agents situés sur la voie le premier se trouvant à

une position longitudinale supérieure au véhicule étudié. Comme cette solution s'applique pour tous les agents du système, on aura une complexité en temps de $O(n^2)$ à chaque pas de temps de simulation. Afin de remédier à ce problème, on crée un tableau associatif (*map*) dont la clé est représentée par l'identifiant de chaque route. Sa valeur sera également représentée par un tableau associatif dont la clé représente l'identifiant de chaque voie et la valeur la liste ordonnée des véhicules s'y trouvant. Ainsi, à chaque création de véhicule, celui-ci se place en première position de la liste. L'avantage de cette représentation est que cela permet d'accéder au véhicule qui suit pour chaque véhicule avec une complexité en temps de $O(n)$. De plus, on met à jour ces éléments uniquement dans les cas suivants :

— lors de l'initialisation du véhicule ;

— lorsqu'un véhicule quitte la route ;

— lors d'un changement de voie.

2.3.3.2 Performances de l'outil avec cette extension

L'objectif est de déterminer quel est l'impact de cette extension sur le simulateur GAMA. Pour cela, nous allons tester trois modèles de décision de véhicules différents, du plus simple au plus complexe :

— un véhicule n'observant pas son environnement et conservant une vitesse constante tout au long de la simulation ;

— un véhicule ne changeant pas de voie, dont l'aspect longitudinal (représentant le sens de la route) est contrôlé par la loi de poursuite IDM, calculant donc son accélération par rapport au véhicule situé devant lui sur la voie ;

— un véhicule également contrôlé par IDM sur l'aspect longitudinal mais aussi par MOBIL pour l'aspect latéral (changement de voie).

Nous allons calculer le temps que met chacune des simulations à atteindre plusieurs temps de simulation (1, 5, 10, 30 et 60 minutes). Autrement dit, nous allons voir s'il est possible de simuler ces modèles plus rapidement que si on les observaient sur une route réelle. Le pas de temps de simulation est égal à 100 ms. L'environnement étudié est une route constituée de deux voies dont on va tester deux tailles différentes : 400 m et 2000 m. Pour varier le flux contenu sur la route, nous définissons un débit de véhicules générés en début de route : un véhicule tous les

2 secondes ($1800\ veh.h^{-1}$) et un véhicule tous les 5 secondes ($720\ veh.h^{-1}$). Ces simulations ont été testées sur un ordinateur portable de type bureautique constitué d'un processeur à 2.7 GHz à deux cœurs et de 8 Go de RAM.

Sur les Figures 2.7 et 2.8 se trouvent les temps de simulation obtenus selon les modèles de décision détaillés précédemment, respectivement sur une route de 400 mètres et de 2000 mètres. Pour chaque modèle de décision, on remarque les éléments suivants :

— le modèle de décision du véhicule simple n'impacte pas réellement le temps de simulation car celui-ci est généralement seulement 2 à 3 fois plus long qu'une route de 400 mètres sans véhicules (3 à 8 fois plus long pour une route de 2000 mètres). Simuler l'équivalent d'une heure de circulation de route ne pose aucun problème (inférieur à 1 minute) ;

— la loi de poursuite *IDM* impacte un peu plus le temps de simulation. En effet, chaque agent a une perception plus grande que son environnement qu'il met continuellement à jour à chaque pas de temps. On n'observe pas de problèmes particuliers pour simuler 1 heure de circulation de route. Par rapport au véhicule simple, l'outil met 3 à 5 plus de temps sur une route de 400 mètres (3 à 6 fois plus sur une route de 2000 mètres) ;

— dans le cas de la combinaison *IDM* et *MOBIL*, on observe plus d'éléments en raison de l'ajout du changement de voie. Ainsi, on commence à obtenir des temps plus long à partir de 30 minutes de circulation réelle (bien que toujours inférieurs à une simulation en temps réel). Par rapport au modèle *IDM*, l'outil met 2 à 3 fois plus de temps sur une route de 400 mètres (3 à 4 fois plus de temps sur une route de 2000 mètres).

Nombre de pas de temps observés	Equivalent en temps réel (en minutes)	Aucun véhicule	Véhicule simple		IDM		IDM + MOBIL	
			Tous les 5 secondes	Tous les 2 secondes	Tous les 5 secondes	Tous les 2 secondes	Tous les 5 secondes	Tous les 2 secondes
600	**1**	0min 0s	0min 0s	0min 0s	0min 1s	0min 1s	0min 3s	0min 3s
3000	**5**	0min 0s	0min 1s	0min 1s	0min 3s	0min 6s	0min 9s	0min 15s
6000	**10**	0min 1s	0min 2s	0min 3s	0min 6s	0min 12s	0min 15s	0min 31s
18000	**30**	0min 3s	0min 5s	0min 8s	0min 19s	0min 39s	0min 37s	1min 36s
36000	**60**	0min 5s	0min 10s	0min 16s	0min 36s	1min 17s	1min 8s	3min 10s

FIGURE 2.7: Temps de simulation pour les trois différents modèles de décision sur une route de **400 mètres**.

Nombre de pas de temps observés	Equivalent en temps réel (en minutes)	Aucun véhicule	Véhicule simple		IDM		IDM + MOBIL	
			Tous les 5 secondes	Tous les 2 secondes	Tous les 5 secondes	Tous les 2 secondes	Tous les 5 secondes	Tous les 2 secondes
600	**1**	0min 0s	0min 0s	0min 0s	0min 1s	0min 5s	0min 2s	0min 6s
3000	**5**	0min 0s	0min 2s	0min 3s	0min 7s	0min 26s	0min 22s	1min 18s
6000	**10**	0min 1s	0min 3s	0min 6s	0min 15s	0min 47s	0min 45s	2min 50s
18000	**30**	0min 3s	0min 11s	0min 20s	0min 48s	2min 4s	2min 20s	8min 56s
36000	**60**	0min 5s	0min 21s	0min 40s	1min 40s	3min 52s	4min 42s	18min 4s

FIGURE 2.8: Temps de simulation pour les trois différents modèles de décision sur une route de **2000 mètres**.

Ainsi, il est possible de concevoir un modèle de décision basé sur des modèles existants sans trop impacter les performances du simulateur. Il est évident que l'ajout de communications ou la simulation de plusieurs routes en parallèle augmente encore le temps de simulation mais il reste raisonnable. D'ailleurs, on peut facilement trouver des solutions d'optimisation en simulant une route infinie à partir d'une route de petite taille, similaire à une route sous forme d'anneau (possible en utilisant le paramètre *torus* sur le simulateur GAMA).

2.3.4 Contenu de l'extension et synthèse

Actuellement, l'outil est disponible en ligne sous forme de prototype contenant un ensemble de modèles GAMA. Le projet GAMA, comprenant près de 5000 lignes de code, contient les modèles suivants (avec leurs noms dans le projet entre parenthèses) :

— des modèles permettant de récupérer les informations du fichier XML (*Initialization*) afin de concevoir les véhicules (*CarConstruction*) et les routes (*RoadConstruction*) ;

— l'ensemble des variables (*GlobalVariables*) et fonctions (*GlobalReflexes*) accessibles pour tout agent du simulateur. Les fonctions contiennent notamment un algorithme de théorie des jeux ainsi qu'une partie du protocole de communication détaillé dans le chapitre 3 ;

— un modèle permettant la génération automatique et aléatoire de véhicules (*RandomCarGeneration*) ;

— un modèle regroupant l'ensemble des variables que l'on souhaite tester (*StatsVariables*) dont plusieurs exemples sont disponibles ;

— un modèle représentant une route générique (*Road*) ;

— les modèles de véhicules génériques permettant de concevoir un objet ayant les caractéristiques d'un véhicule (*ObjectCar*) et proposant un modèle de décision générique pour le véhicule (*Car*) ;

— des modèles de décision héritant tous du modèle de décision générique. Parmi les modèles proposés : la loi de poursuite IDM (*IDM*), le modèle de changement de voie (*MOBIL*), un modèle de véhicule simulant des embouteillages (*TrafficJam*), un modèle de théorie des jeux associé à IDM (*IDMGameTheory*) et d'autres modèles dérivés de ceux-ci.

A ces modèles s'ajoute l'outil de compilation afin d'initialiser de façon intuitive les routes à partir d'un fichier texte. Le projet est disponible à l'adresse suivante :

`https://ibisc.univ-evry.fr/~jsobieraj/projectVehicleGAMA`

Cette extension permet de rajouter à GAMA une expertise trafic plus complète pour concevoir des véhicules intelligents. Grâce à la conception plus intuitive du modèle de décision, il est possible de proposer son propre modèle à partir d'un modèle initial répondant aux caractéristiques principales du véhicule. Ainsi, le lien entre le modèle du véhicule et l'outil de simulation est simplifié. Dans le chapitre 3, nous allons définir le niveau d'abstraction du modèle du véhicule notamment à partir de modèles existants. Ce chapitre a un double objectif : d'une part, proposer la mise en place d'une méthode pour la conception du modèle de décision d'un véhicule, d'autre part, montrer l'intérêt d'un protocole de coopération dans des environnements routiers comme la voie d'insertion, pouvant être dangereux du point de vue de la sécurité et anxiogène pour les êtres humains.

3 | Prise de décision et coopération entre véhicules communicants

3.1 Représentation et évaluation de la dynamique des véhicules 57

3.1.1 Aspect longitudinal 57

3.1.2 Aspect latéral 58

3.1.3 Choix de modélisation et description des modèles 59

3.1.4 Indicateurs de performance 62

3.2 Analyse du changement de voie $MOBIL$ 68

3.2.1 Limitations du modèle 68

3.2.2 Améliorations du modèle 72

3.3 Un protocole de coopération entre véhicules 74

3.3.1 Objectifs et modèle de décision 74

3.3.2 Formalisation du protocole 75

3.3.3 Résultats et comparaisons avec $MOBIL$ et $MOBIL_{safe}$ 82

3.4 Prise en compte de véhicules non communicants sur le protocole . . . 84

3.4.1 Motivations 84

3.4.2 Étude de cas 1 : Ajout de véhicules autonomes non communicants dans l'environnement 86

3.4.3 Étude de cas 2 : Ajout de véhicules non autonomes et non communicants dans l'environnement 88

3.5 Synthèse 90

Avant de mettre en place des protocoles de coopération entre véhicules, il est important de définir les différents scénarios où la communication peut être un apport. En effet, un véhicule équipé d'un ensemble de capteurs observant l'environnement peut suffire pour certaines situations où la simple analyse de la dynamique des véhicules environnants permet de répondre à l'ensemble des problèmes que l'on peut rencontrer. Parmi ces situations, on peut citer la gestion de la vitesse d'un véhicule par rapport au véhicule situé devant lui, comme le fait un véhicule constitué d'un système *ACC* (*Adaptive Cruise Control*) [70] ou encore le positionnement d'un véhicule au centre de sa voie (*lane keep assist*) [49].

Pour d'autres situations, il est nécessaire d'anticiper pour éviter un danger dans un temps futur. Par exemple, on peut imaginer la prévention d'un embouteillage grâce à l'envoi d'une alerte aux véhicules situés derrière nous. Cela permet de créer un train de véhicules ayant des vitesses très proches évitant le phénomène d'accordéons souvent présent lors d'embouteillages. De plus, on peut supposer que les véhicules envoient leurs intentions futures. Par exemple, dans le cas d'un carrefour, l'envoi des différentes intentions des véhicules peut réduire les chances que l'ensemble des décisions de chaque véhicule mène à une collision.

Toutefois, le simple envoi d'une alerte ne permet pas toujours à un ensemble de véhicules d'arriver à un accord commun. Pour cela, on peut faire communiquer un groupe de véhicules afin de faciliter une manœuvre à faire ou d'optimiser le trafic. Dans certains cas, les buts des véhicules peuvent être différents, voire contradictoires, rendant la coopération plus difficile. On le voit notamment dans les situations de voies d'insertion où l'objectif d'un véhicule dans la voie d'insertion est de rejoindre la voie principale tandis que les véhicules situés sur la voie principale souhaitent optimiser leurs vitesses maximales. Actuellement sur les routes, les conducteurs adoptent parfois des stratégies de négociation afin de convenir d'un accord commun facilitant l'insertion d'un véhicule. L'objectif ici est de reproduire ce phénomène de façon automatisée afin de faciliter l'insertion sans impacter négativement les véhicules environnants (freinage brusque, non respect des distances de sécurité. . .). De plus, cette étude de cas permet de mieux définir le modèle du véhicule afin de le décomposer en plusieurs parties indépendantes.

3.1 Représentation et évaluation de la dynamique des véhicules

Ici nous allons préciser comment nous allons représenter et évaluer la dynamique des véhicules sur l'aspect longitudinal (dans l'axe de la route) et l'aspect latéral (dans la direction perpendiculaire à l'axe de la route).

3.1.1 Aspect longitudinal

Dans le cas de modèles microscopiques où l'on s'intéresse à la dynamique individuelle du véhicule, la dimension longitudinale de la route se caractérise par un modèle calculant la position future d'un véhicule et sa vitesse par rapport à son accélération, calculée par rapport au véhicule qui le précède (s'il y en a un). Parmi ces modèles, on distingue les modèles discrets et continus (en espace et en temps).

Les modèles discrets peuvent être représentés par les automates cellulaires représentant la route sous forme de cases. Chaque véhicule va répondre à un ensemble de règles déterminant dans quelle case de la route il va se trouver au prochain pas de temps. Parmi les modèles d'automates cellulaires les plus connus, on peut citer celui de Nagel et Schreckenberg [61]. Il permet de représenter un flux de trafic mais il souffre de quelques limitations lorsqu'on souhaite déterminer des propriétés plus précises comme l'inter-distance entre deux véhicules de par la représentation discrète de la route.

Toutefois, les modèles continus peuvent représenter de façon plus précise la dynamique des véhicules ainsi que des propriétés plus précises prenant en compte une distance ou un temps. Ces modèles sont représentés par des lois de poursuite (*car-following models*), qui calculent l'accélération du véhicule à partir du véhicule situé devant lui (déterminant ainsi sa vitesse et sa position). Les premiers modèles (Reuschel [72] et Pipes [68]) appliquent une règle élémentaire des lois de poursuites : la distance minimale entre deux véhicules (appelée aussi "distance de sécurité") doit être proportionnelle à l'inter-distance entre les deux véhicules. Un des modèles les plus connus est celui de Gipps [38] qui propose un modèle sans collisions tout en ayant la possibilité de faire varier certains paramètres propres aux véhicules (taille du véhicule, décélération souhaitée, temps de réaction...). Sa po-

pularité vient probablement de sa simplicité d'implémentation, son efficacité d'utilisation et sa capacité à s'adapter à un large panel de situations avec des valeurs de décélérations réalistes par rapport aux capacités de décélération d'un véhicule réel. D'autres modèles existent également dans la littérature comme les modèles linéaires de Helly [48] et les modèles de points d'action (AP) [13] se rapprochant d'un arbre de décision dans lequel on choisit une décision selon les conditions actuelles de l'environnement. Concernant des modèles plus récents, Treiber et Kesting ont proposé l'Intelligent Driver Model (IDM) au début des années 2000 permettant, tout comme le modèle de Gipps, de faire varier plusieurs paramètres.

3.1.2 Aspect latéral

Les modèles de changement de voie permettent la prise en compte de la dimension latérale de la route. Dans leur grande majorité, ces modèles sont compatibles avec les lois de poursuite. Toutefois, bien qu'il soit possible de les représenter graphiquement par un changement de voie progressif, la durée de changement de voie n'est pas prise en compte dans ces modèles. Ainsi, le changement de voie se fait par l'intermédiaire d'un "saut" : à l'instant t, on détecte une situation où l'on peut changer de voie et à l'instant $t+1$, le changement de voie est effectué et le véhicule situé à l'arrière sur la nouvelle voie observe la présence de ce véhicule.

En simulation microscopique, différents modèles de changement de voie ont été réalisés que l'on peut classer dans quatre grandes catégories [69] :

— *modèles basés sur des règles (Gipps, CORSIM...)* [39] [65] : les véhicules décident de changer de voie selon un arbre de décision constitué d'un ensemble de conditions fixes. Par conséquent, ils ont l'avantage d'être simples d'implémentation. Toutefois, il est parfois difficile de calibrer les paramètres du modèle et il n'est pas possible de représenter des comportements de véhicules différents ;

— *modèles à choix discrets (Ahmed, Toledo...)* [5] [86] : ce sont des modèles basés sur des probabilités d'effectuer un changement de voie obligatoire (MLC) ou des changements de voie impliquant un avantage de vitesse (DLC). Ils permettent donc de donner des résultats sous forme de probabilités contrairement aux modèles basés sur les règles indiquant une réponse binaire (oui

ou non). Cependant, de type de modèle impose de calculer les fonctions de probabilités pour déterminer l'intérêt des différents choix possibles.

— *modèles d'intelligence artificielle (Artificial neural, Fuzzy-Logic...)* [87] [77] : ils fonctionnent selon un ensemble de données d'entrées que l'on va analyser afin de déterminer le statut des véhicules environnants. Ils permettent ainsi de prendre en compte l'imprécision de la perception humaine. Toutefois les fonctions sont complexes et requièrent une grande quantité initiale de données ;

— *modèles basés sur la récompense ($MOBIL$, LMRS...)* [54] [75] : l'objectif de ces modèles est de décider de changer de voie ou non dans le but de maximiser le bénéfice pour les véhicules, ils se rapprochent des modèles basés sur les règles. Ce sont des modèles faciles à implémenter mais aussi très flexibles (possibilité d'ajouter des récompenses supplémentaires facilement). De plus, ils proposent un ensemble de paramètres d'entrée permettant une variabilité des comportements des véhicules. Toutefois, ces modèles ont des difficultés pour gérer les situations de congestion et de changement de voie obligatoire (notamment pour $MOBIL$).

3.1.3 Choix de modélisation et description des modèles

Finalement, pour l'aspect longitudinal, notre choix se tournera vers IDM, motivé par les raisons suivantes :

— il peut décrire un véhicule autonome utilisant un système de régulateur de vitesse adaptatif (ACC) mais il peut également reproduire le comportement d'un véhicule conduit par l'homme sur l'aspect longitudinal ;

— cette variété de représentations peut être obtenue grâce à la possibilité de changer les valeurs d'un ensemble de paramètres initiaux ;

— c'est un modèle auquel on peut ajouter différentes couches de décision sans impacter le modèle de base et, dans une plus grande mesure, la sécurité du véhicule.

Quant à l'aspect latéral, nous souhaitons un modèle de changement de voie dont l'implémentation ne pose pas de difficultés particulières et permettant de représenter un large choix de comportements de véhicules comme le permet IDM sur la dimension longitudinale. Ainsi, nous nous sommes orientés vers un modèle basé

sur la récompense, plus précisément $MOBIL$. A partir du modèle IDM, $MOBIL$ va établir un compromis entre obtenir un gain de vitesse et ne pas faire perdre en vitesse les véhicules situés derrière lui. Il permet d'observer à un instant donné si un changement de voie est considéré comme étant sécurisé (*safety*) et motivé (*incentive*).

3.1.3.1 Description d'IDM

Considérons deux véhicules comme décrit dans la Figure 3.1 : le véhicule suiveur i (dont le comportement est déterminé par IDM) et le véhicule leader $i-1$ (dont le comportement ne correspond pas nécessairement à IDM). A chaque pas de temps, le véhicule i met à jour son accélération, qui varie selon deux critères principaux : d'une part, le véhicule i tend à atteindre une vitesse maximale v_0 autorisée sur la portion de route (vitesse de croisière), d'autre part, il doit également respecter une distance de sécurité minimale s^* avec le véhicule le précédant (qui varie en fonction de la vitesse relative entre les deux véhicules).

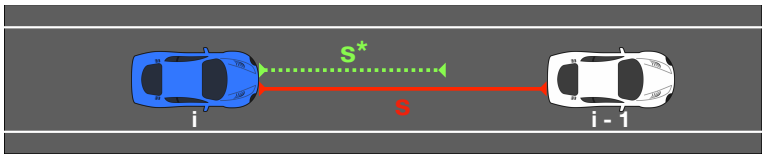


FIGURE 3.1: Principe d'IDM : à chaque pas de temps, le véhicule i adapte sa vitesse par rapport au véhicule $i-1$

L'équation IDM calculant l'accélération du véhicule est la suivante :

$$\frac{dv_i}{dt} = a \left[1 - \left(\frac{v_i}{v_{lim}} \right)^{\delta} - \left(\frac{s^*(v_i, \Delta v_i)}{s} \right)^2 \right] \tag{3.1}$$

où l'inter-distance désirée $s^*(v_i, \Delta v_i)$ est :

$$s^*(v_i, \Delta v_i) = s_0 + \max \left[0, \left(v_i \cdot T + \frac{v_i \cdot \Delta v_i}{2\sqrt{a \cdot b}} \right) \right] \tag{3.2}$$

avec a l'accélération maximale, v_i la vitesse du véhicule i, v_{lim} la vitesse maximale autorisée du véhicule i, δ le coefficient d'accélération (coefficient d'"agressivité"), $\Delta v_i = v_i - v_{i-1}$ la vitesse relative, s_0 l'inter-distance minimale avec le véhicule qui suit, T le temps de réaction estimé et b la décélération souhaitée.

Intuitivement, IDM va appliquer une accélération égale à a si aucun véhicule n'est situé devant lui sur la même voie. Deux termes ont un impact sur la valeur d'accélération. Le terme $\frac{v_i}{v_{lim}}$ réduit la valeur de l'accélération au fur et à mesure que le véhicule se rapproche de sa vitesse limite (v_{lim}). Quant au terme, $\frac{s^*(v_i,\Delta v_i)}{s}$, il s'assure qu'une certaine distance de sécurité (s^*) est respectée par rapport à l'inter-distance avec le véhicule situé devant lui à un instant donné. Ainsi, un véhicule se respectant pas sa distance de sécurité a tendance à décélérer. A l'inverse, un véhicule accélère s'il n'a pas encore atteint sa vitesse maximale ou respecte les distances de sécurité.

3.1.3.2 Description de $MOBIL$

Comme on peut le voir sur la Figure 3.2, on représente le véhicule actuel comme étant *M*, *L* pour le véhicule situé devant *M* et *F* le véhicule situé derrière *M*. On pose également *M'* la potentielle position du véhicule actuel *M* sur la voie cible ainsi que les positions des véhicules situés devant (*L'*) et derrière (*F'*) lui sur la voie cible. Enfin, on calcule une accélération a_i correspondant à l'accélération d'un véhicule i calculée par IDM avant un potentiel changement de voie de M et une accélération $\hat{a}_i$ d'un véhicule i calculée par IDM après un potentiel changement de voie du véhicule M avec $i \in \{F, F, M, M'\}$.

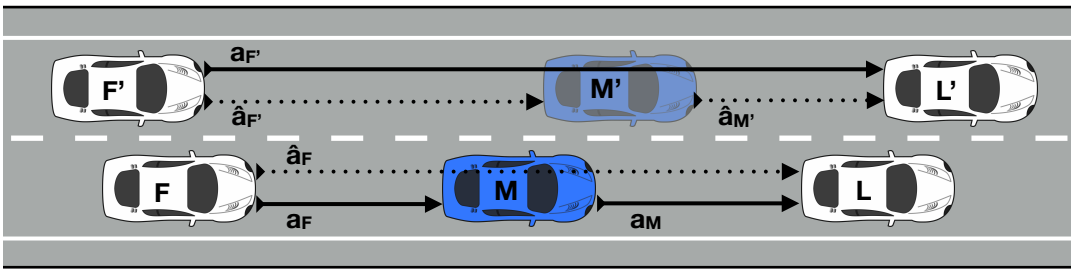


FIGURE 3.2: Positionnement des variables pour $MOBIL$.

Afin de déterminer si le critère de sécurité est respecté, on observe si la future accélération du véhicule calculée par IDM ne passe pas en dessous d'une valeur

limite de décélération appelée b_{safe}. Ainsi, on obtient le critère de sécurité (*safety*) représenté par l'équation 3.3.

$$MOBIL_{safe} = (\hat{a}_{M'} > -b_{safe}) \tag{3.3}$$

Le critère de motivation (*incentive*) observe si les gains d'accélération pour les véhicules *M* et *F* sont intéressants et s'ils ne vont pas trop impacter négativement le véhicule *F'*. Ainsi, on obtient le critère de motivation représenté par l'équation 3.4.

$$MOBIL_{incentive} = (\hat{a}_{M'} - a_M > a_{thr} + a_{bias} + p * (a_F + a_{F'} - \hat{a}_F - \hat{a}_{F'})) \tag{3.4}$$

avec p le facteur de politesse situé (allant de 0 pour un véhicule totalement égoïste à 1 pour un véhicule totalement altruiste), a_{thr} le gain d'accélération minimum et a_{bias} la valeur d'accélération de biais (permettant de gérer les changements de voie obligatoires et les règles européennes du code de la route).

Ainsi, pour un véhicule donné, lorsque les deux critères de sécurité (*safety*) et de motivation (*incentive*) exprimées par les deux équations 3.3 et 3.4 sont vérifiés, le véhicule change de voie instantanément.

3.1.4 Indicateurs de performance

Les différentes lois de poursuites et modèles de changement de voie précédemment présentés ont été évalués par leurs auteurs grâce à différents indicateurs de performance. Ces indicateurs sont essentiels pour évaluer les modèles selon des critères choisis. Pour la plupart, les auteurs d'articles scientifiques sur les Systèmes de Transport Intelligents utilisent leurs propres critères et termes pour évaluer leurs modèles. Cela rend toute comparaison difficile voir impossible. De plus, la définition des termes diffère d'un article à un autre. C'est la raison pour laquelle des auteurs ont classifié ces critères et termes afin de normaliser un maximum les significations et les notations [16].

Toutefois, les éléments cités répondent essentiellement à la problématique de fluidité. Pour notre part, l'objectif est de définir des critères permettant de répondre à la fois à des problématiques de sécurité, de fluidité et de confort. Nous avons donc

	Critère	Variable	Description
Sécurité	*Time-To-Collision*	$TTC_{i,j}(t)$	Durée au pas de temps t avant laquelle le véhicule i va entrer en collision avec le véhicule j (situé devant) s'ils conservent les mêmes vitesses, en [s]
	Distance minimale de sécurité	$SD_{i,j}(t)$	Distance minimale nécessaire au pas de temps t entre le véhicule i, et le véhicule j (situé devant) en [m]
Efficacité	Vitesse	$v_i(t)$	Vitesse du véhicule i au pas de temps t, en [$m.s^{-1}$].
	Hétérogénéité locale de la vitesse	$HLV_i^c(t)$	Variance des vitesses par rapport à un ensemble de véhicules environnants c (définie par rapport au point de vue du véhicule i) au pas de temps t
Confort	Accélération	$a_i(t)$	Accélération du véhicule i au pas de temps t, en [$m.s^{-2}$]

TABLEAU 3.1: Indicateurs de performance individuels.

défini un ensemble de critères pour caractériser et évaluer nos simulations de la manière la plus précise possible. Ces critères peuvent s'observer de façon statique ou dynamique (observables dès l'initialisation du modèle ou au cours du temps), de façon individuelle ou groupée. Selon les cas, le critère observé peut varier et avoir une signification différente.

Les paramètres statiques vont représenter les différents paramètres initiaux du modèle permettant d'observer à quel point un scénario est complexe, à savoir :

— le type de route ;

— le nombre de voies ;

— le flux de véhicules entrant dans chaque voie ;

— la proportion de véhicules autonomes sur la route (donnée à partir du flux de véhicules entrant) ;

— la taille de chaque route (sa largeur et sa longueur si elle n'est pas considérée comme infinie) ;

— le pas de temps de simulation ;

— le pas de temps de décision pour chaque véhicule.

Les autres éléments étudiés sont des indicateurs dynamiques et la mise à jour se fait selon le pas de temps de simulation fixé précédemment. Nous avons regroupé les différents indicateurs et leurs valeurs associées pour exprimer les critères de sécurité, d'efficacité et de confort dans le tableau 3.1. Pour l'aspect sécurité, il est possible d'observer la valeur du *Time-To-Collision* [93] (TTC), qui mesure la durée avant laquelle deux véhicules entrent en collision s'ils conservent les mêmes vitesses. On peut considérer qu'il y a un danger potentiel quand cette valeur est inférieure à 3 secondes [74]. La valeur du TTC est donnée par la formule suivante :

$$TTC_{i,j}(t) = \frac{x_j(t) - x_i(t) - \frac{1}{2}(l_i + l_j)}{v_i(t) - v_j(t)} \; \forall v_i(t) > v_j(t) \tag{3.5}$$

où $x_i(t)$ est la position de i au pas de temps t, $v_i(t)$ la vitesse de i au pas de temps t et l la longueur des véhicules.

Il est également possible d'observer la distance de sécurité, généralement définie à partir des lois de poursuite. Parmi elles, celle de la société *Intel* qui a défini une distance minimale de sécurité pour les véhicules sans conducteur [76] dont la formule est la suivante :

$$SD_{i,j}(t) = l + T_j \left[v_j(t) - v_i(t) + \rho_i.(\alpha + \beta)\right] - \frac{\rho_i^2.\beta}{2} + \frac{(T_i - T_j)(v_j(t) + \rho_i.\alpha - (T_j - \rho_i).\beta)}{2} \tag{3.6}$$

avec :

$v_i(t)$: vitesse du véhicule i au temps t,

ρ_i : temps de réaction du véhicule i,

α : accélération maximale des véhicules,

β : freinage maximal des véhicules,

T_i : temps pour que le véhicule i soit à l'arrêt en appliquant un freinage β,

T_j : temps pour que le véhicule j soit à l'arrêt en appliquant une accélération α pendant le temps de réaction ρ_i puis un freinage β après.

Enfin, afin de mesurer l'hétérogénéité autour de soi, on définit l'hétérogénéité locale d'un véhicule i comme la variance des vitesses des véhicules environnants c (définie manuellement selon la perception de la route au véhicule i), correspondant

	Critère	Variable	Description
Sécurité	Nombre de véhicules ne respectant pas un TTC minimal	$nbTTC_x^c(t)$	Parmi un ensemble c de véhicules au pas de temps t, le nombre ayant une valeur du TTC inférieure à x secondes
	Nombre de véhicules ne respectant pas une SD minimale	$nbSD^c(t)$	Parmi un ensemble c de véhicules au pas de temps t, le nombre ne respectant pas la distance de sécurité
Efficacité	Vitesse moyenne	$mV^c(t)$	La vitesse moyenne d'un ensemble de véhicules c au pas de temps t
	Hétérogénéité globale de la vitesse	$HGV^c(t)$	Variance des vitesses par rapport à un ensemble de véhicules c au pas de temps t
	Travel Time moyen	$mTT^c(t)$	Temps moyen pour chaque véhicule de c pour atteindre une position ou répondre à un événement donné au pas de temps t [en s]
	Waiting Time moyen	$mWT^c(t)$	Temps moyen d'attente pour chaque véhicule de c au pas de temps t (temps à l'arrêt) [en s]
Confort	Nombre de véhicules ne respectant pas une accélération minimale	$nbA_x^c(t)$	Parmi un ensemble c de véhicules au pas de temps t, le nombre ayant une valeur d'accélération inférieure à x

TABLEAU 3.2: Indicateurs de performance de groupe.

à la formule suivante :

$$HLV_i^c(t) = \mathop{\mathbb{E}}_{x \in c}[(v_x(t) - \mathop{\mathbb{E}}_{y \in c}[v_y(t)])^2] \tag{3.7}$$

où $\mathbb{E}[X]$ représente l'espérance d'un ensemble X de valeurs (ici l'ensemble des vitesses v_x appartenant à c).

On a donc un ensemble d'indicateurs permettant d'évaluer certains aspects à tout pas de temps de simulation. Toutefois, ils permettent d'évaluer uniquement chaque critère séparément et il est difficile d'établir une relation entre-eux. En effet, un véhicule peut être sécurisé mais ne pas respecter le critère d'efficacité (véhicule à l'arrêt dans un embouteillage). De même, un véhicule efficace peut ne pas être confortable (accélérations / freinages successifs proches de la vitesse limite). Enfin,

un véhicule confortable n'est pas pour autant toujours sécurisé (véhicule à vitesse constante à un mètre d'un véhicule).

Ainsi, les indicateurs choisis permettent de construire une mesure de la sécurité (Équation 3.8), de l'efficacité (Équation 3.9) et du confort (Équation 3.10). Tous ces indicateurs ont des valeurs réelles comprises entre 0 et 1. La mesure de la sécurité se traduit par la valeur du *TTC* (le facteur vaut 0 lorsque le *TTC* est inférieur ou égal à 1 seconde). La mesure de l'efficacité observe la vitesse du véhicule par rapport à sa vitesse limite. Enfin, la mesure du confort va analyser les k dernières accélérations du véhicule (le facteur vaut 1 si les k dernières accélérations sont supérieures à $0\ m.s^{-2}$). Lorsqu'une des k dernières accélérations est négative, elle va impacter directement la valeur de confort de façon exponentielle (plus la décélération est élevée, plus l'impact sur la mesure du confort va avoir une influence négative).

$$safety_{i,j}(t) = 1 - \frac{1}{max(1, TTC_{i,j}(t))} \tag{3.8}$$

$$efficiency_i(t) = \frac{v_i(t)}{v_{lim}} \tag{3.9}$$

$$comfort_{i,k}(t) = \underset{(t+1-k)\leq y\leq t}{\mathbb{E}} \left[max(1, e^{a_i(y)}) \right] \tag{3.10}$$

Afin d'établir un lien entre ces trois indicateurs, nous avons imaginé un nouvel indicateur regroupant les trois critères à la fois. Appelée *HPN* (*HaPiNness*) ou "satisfaction", elle permet de représenter le niveau de satisfaction d'un passager dans un véhicule qui souhaite l'ensemble des trois critères respectés à la fois (voir Équation 3.11).

$$HPN_{i,j,k}(t) = \left(safety_{i,j}(t)\right)^{\alpha} \ . \ \left(efficiency_i(t)\right)^{\beta} \ . \ \left(comfort_{i,k}(t)\right)^{\gamma} \tag{3.11}$$

Il est possible d'ajouter des coefficients d'exigence α, β et γ si l'on souhaite établir un poids pour chaque critère. Afin d'analyser le comportement de cet indicateur ainsi que l'influence de ses coefficients d'exigence, nous avons effectué une étude de cas sur une route à une voie à sens unique constituée de deux véhicules. Le premier véhicule est contrôlé par la loi de poursuite *IDM*, initialement à une vitesse nulle et se dirigeant vers le second véhicule, à l'arrêt, 500 mètres plus loin.

Dans la Figure 3.3, nous avons observé l'évolution de la valeur du nouvel indicateur en posant chaque coefficient à 1 (courbe "Global" pleine). Cette valeur est comparée à l'évolution de chacun des critères observés séparément (sécurité, efficacité, confort). Enfin, nous avons proposé des coefficients pour chacun des critères : 2 pour la sécurité, 1 pour l'efficacité, et 0.25 pour le confort (courbe "Global" en pointillés).

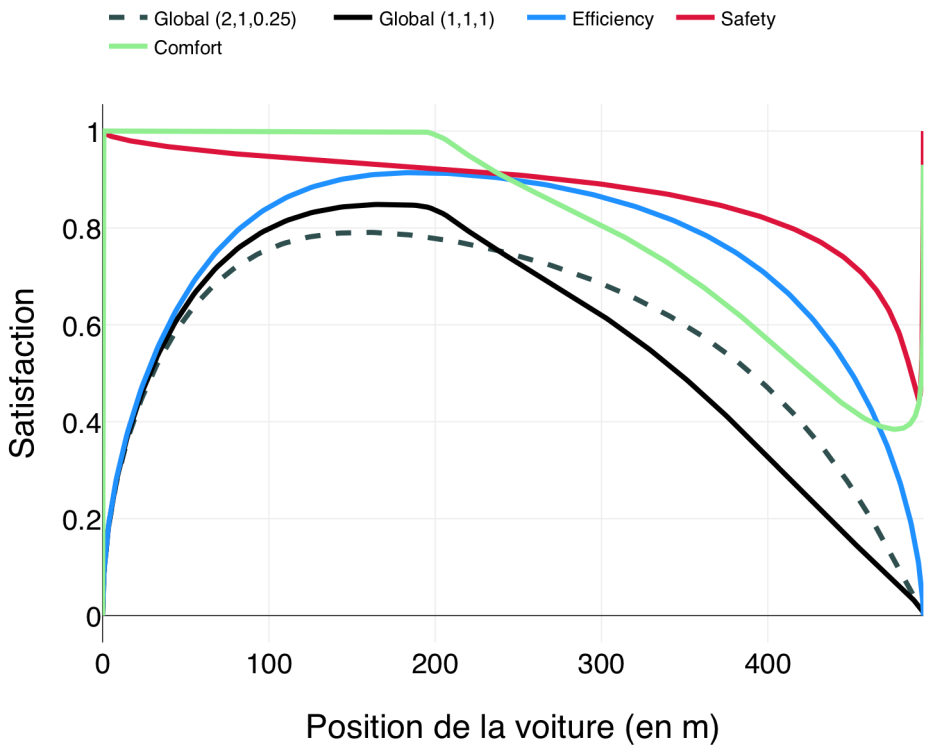


FIGURE 3.3: Évolution de la valeur de satisfaction (HPN) en fonction de la position du véhicule à partir de plusieurs indicateurs de performance.

On remarque tout d'abord que l'indicateur de confort a une influence assez forte sur l'indicateur global (1,1,1). En effet, lorsque le véhicule décélère (à partir de 200 mètres), l'indicateur global chute de la même façon que le valeur de confort seule. L'indicateur de sécurité seul (*safety*) diminue vers la fin de la simulation, n'ayant pas d'influence étant donnée la vitesse faible du véhicule. Par rapport à l'indicateur d'efficacité seul (*efficiency*), la valeur globale donne des valeurs plus faibles de satisfaction, rendant son exigence plus forte.

Si l'on compare l'indicateur global à la version affinée (2,1,0.25), on remarque que ce dernier réduit l'exigence due au confort lors de la décélération et accentue l'exigence avant la décélération en raison d'une valeur de sécurité qui n'est

pas maximale. De plus, l'allure de la courbe se rapproche de l'indicateur d'efficacité seule (*efficiency*) donc l'exigence reste toujours plus grande. Elle permet alors de représenter un véhicule intelligent dont l'exigence première est la sécurité, puis l'efficacité et enfin le confort, sans pour autant la négliger.

D'une façon plus générale, cet indicateur paraît donc pertinent et permet d'avoir une analyse individuelle plus précise de chaque véhicule. De plus, il permet de définir des exigences propre à une exigence d'un passager ou d'un conducteur humain. Ainsi, certains conducteurs préfèrent aller le plus vite possible au risque de négliger la sécurité et d'autres vont préférer être dans une situation de confort tout en gardant une vitesse raisonnable. Au vu des résultats obtenus en Figure 3.3, l'indicateur proposant les coefficients $\alpha = 2$, $\beta = 1$ et $\gamma = 0.25$ va être utilisé pour observer l'évolution de la satisfaction sur les véhicules intelligents tout au long de ce chapitre.

3.2 Analyse du changement de voie $MOBIL$

3.2.1 Limitations du modèle

Au vu des lois de poursuite et des modèles de changement de voie existants, on se pose la question de l'intérêt de concevoir un nouveau modèle de décision pour les véhicules. Toutefois, lorsqu'on observe le modèle $MOBIL$, on s'aperçoit que son modèle décisionnel diffère selon le type de changement de voie. En effet, on remarque assez rapidement que $MOBIL$ gère bien les changements de voie impliquant un avantage de vitesse (DLC), notamment en évitant d'impacter négativement l'accélération des véhicules situés à l'arrière. Cependant, les règles diffèrent pour les changements de voie obligatoires (MLC). Ces changements de voie apparaissent dans des cas tels que la voie d'insertion, la sortie d'autoroute ou encore la rétrécissement de voie. Si nous prenons l'exemple de la voie d'insertion, à partir de plusieurs simulations, on remarque que le véhicule a des difficultés pour assurer un équilibre entre l'obligation (je dois changer de voie) et la courtoisie (je dois essayer de ne pas gêner les autres véhicules). De plus, en supposant que le véhicule soit sur la voie d'insertion (par exemple pour entrer sur l'autoroute), on peut imaginer le changement de voie $MOBIL$ couplé à IDM de deux façons différentes (voir figure 3.4) :

— Le véhicule A imagine un véhicule à l'arrêt à la fin de la voie d'insertion afin d'éviter toute sortie de route (cf. Figure 3.4 en haut). Cela paraît très sécurisé mais cela ne respecte pas le fonctionnement même d'une voie d'insertion où le véhicule doit accélérer le plus possible afin de s'insérer à une vitesse proche de la voie principale. Les concepteurs de l'outil de simulation *MovSim* (également les concepteurs de IDM et $MOBIL$), l'ont implémenté de cette façon [88] ;

— Le véhicule A ne voit pas de véhicule sur la voie d'insertion (cf. Figure 3.4 en bas). Il respecte donc le fonctionnement de la voie d'insertion en accélérant tout du long (car IDM ne détecte pas de véhicule). Toutefois, lorsque le véhicule A atteint une certaine limite de sécurité (l'obligeant à freiner à sa limite maximale de décélération), le véhicule A voit un véhicule à la fin de la voie, le forçant à freiner brusquement afin d'éviter une éventuelle sortie de route. Cette limite correspond à l'instant avant lequel le véhicule A devrait freiner avec une accélération inférieure à la valeur b_{safe} de $MOBIL$.

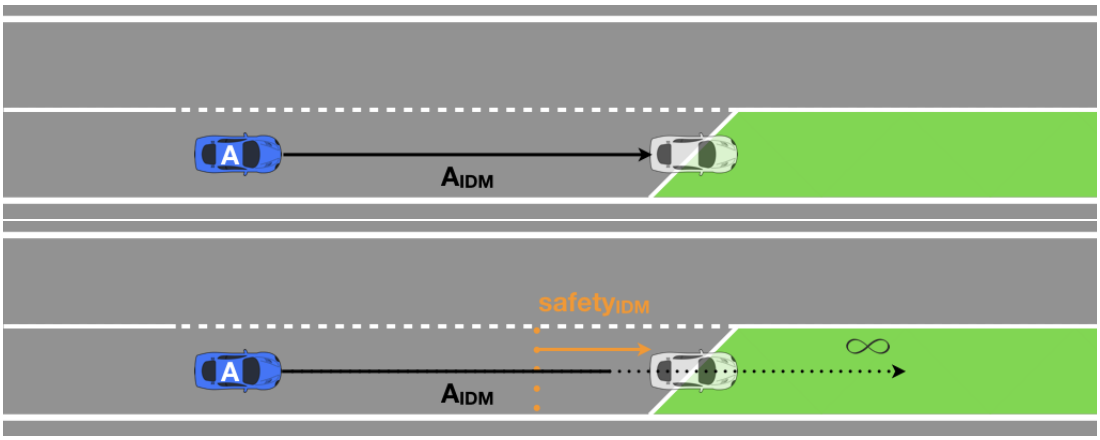


FIGURE 3.4: En haut, le cas où le véhicule A voit un véhicule virtuel en fin de voie. En bas, le cas où le véhicule A ne voit ce véhicule virtuel qu'à partir d'une certaine limite (en orange).

Dans les deux cas cités précédemment, le contrôle du véhicule proposé par le couple IDM / $MOBIL$ ne paraît pas idéal et l'on pourrait montrer que c'est le cas également dans d'autres situations où le changement de voie est obligatoire. Afin d'évaluer ce modèle, et plus précisément de mettre en évidence ses principales limitations, nous avons décidé de simuler la première approche sur GAMA. Pour cela, nous avons généré un flux de véhicules sur la voie principale et sur la voie d'insertion et observé ceux qui respectent au mieux les critères de sécurité, de fluidité et

de confort. Nous allons donc observer ces critères à partir de l'indicateur $HPN_{i,j,k}(t)$ dans un diagramme espace-temps à partir des conditions initiales définies dans la Table 3.3. Ce diagramme est constitué d'un ensemble de points représentant chacun une position d'un véhicule (en ordonnée) à un pas de temps de simulation (en abscisse). La valeur de l'indicateur se fait grâce à la coloration de chacun des points. Ainsi, on peut observer la "trace" de chaque véhicule et l'évolution de son niveau de "satisfaction".

	Paramètres	Unité	Valeurs		
Simulation	Temps réel de simulation	*min*	10		
	Pas de temps	*s*	0.1		
	Pas de décision	*s*	0.1		
Route	Taille voie principale	*m*	650		
	Taille voie d'insertion complète	*m*	550		
	Taille de la portion d'insertion	*m*	350		
	Vitesse limite de la route	$km.h^{-1}$	110		
	Flux entrant sur la voie principale	$veh.h^{-1}$	Entre 500 et 2500		
	Flux entrant sur la voie d'insertion	$veh.h^{-1}$	Entre 500 et 2500		
Véhicule autonome			Trafic non mixte	Trafic mixte	
				Véhicules légers	Véhicules lourds
Véhicules	Taille des véhicules	*m*	4	Entre 3.5 et 5	Entre 6 et 8
	Proportion véhicules	*%*	-	Environ 80	Environ 20
IDM	a	$m.s^{-2}$	3	Entre 3 et 5	Entre 0.5 et 3
	b	$m.s^{-2}$	1	Entre 1 et 3	2
	δ	-	4	4	4
	s_0	*m*	2	Entre 2 et 5	Entre 2 et 5
	v_{lim}	$km.h^{-1}$	110	Entre 100 et 110	Entre 70 et 90
	T	*s*	0.7	0.7	0.7
MOBIL	a_{thr}	$m.s^{-2}$	0.3		
	a_{bias}	$m.s^{-2}$	0.2		
	b_{safe}	$m.s^{-2}$	4		
	p	-	1		

TABLEAU 3.3: Paramètres globaux pour chaque simulation ainsi que les paramètres spécifiques à un trafic mixte et non mixte. Dans le cas du trafic mixte, on définit deux types de véhicules : les véhicules légers et les véhicules lourds.

Nous avons défini deux cas d'étude, un flux non mixte (homogène) de véhicules ayant les mêmes caractéristiques globales et un trafic mixte (hétérogène) avec des caractéristiques globales différentes (vitesse maximale, accélération, taille du véhi-

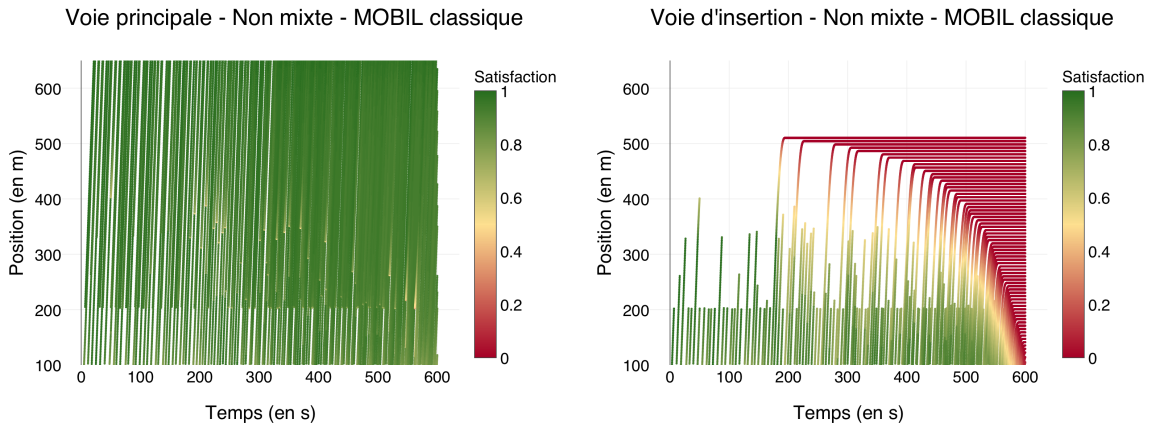


FIGURE 3.5: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL**.

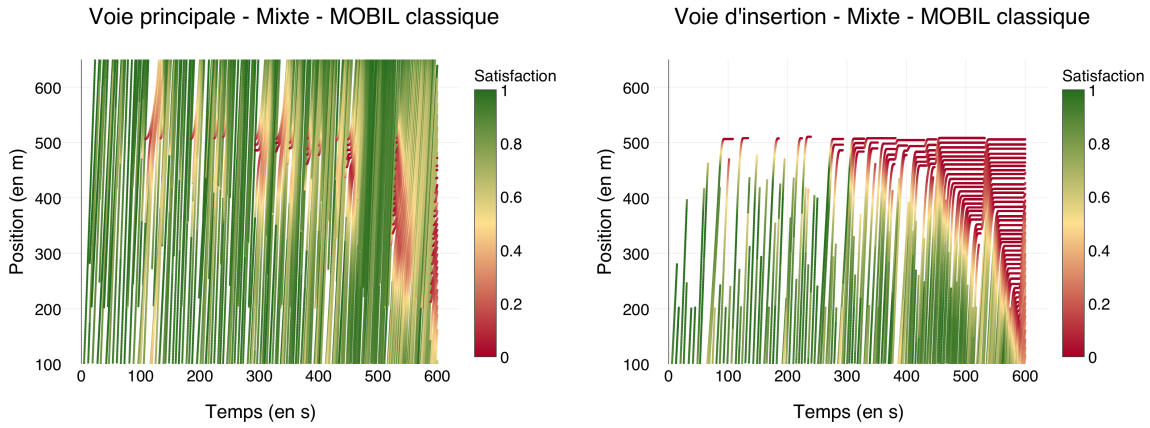


FIGURE 3.6: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL**.

cule...). Afin d'élargir l'éventail de situations, nous allons générer un flux de véhicules pendant 10 minutes de simulation dont le nombre augmente progressivement jusqu'à arriver à un flux très élevé (où le phénomène de congestion est presque inévitable). Nous avons observé les flux sur la voie principale et la voie d'insertion dans les deux cas d'étude et constaté les choses suivantes :

— Dans un trafic non mixte (Figure 3.5), les véhicules situés sur la voie d'insertion changent de voie dès que possible lorsque le trafic est faible. Toutefois, lorsqu'il y a plus de véhicules présents sur la route, les véhicules ont beaucoup de difficultés à changer de voie provoquant de la congestion sur la voie d'insertion (à 200 secondes de simulation environ). Ainsi, le nombre de véhicules sur la

voie principale sera un peu plus faible et moins perturbé ne provoquant pas de problèmes particuliers sur celle-ci ;

— Dans un trafic mixte (Figure 3.6), les véhicules ont beaucoup plus de difficultés à s'insérer directement (premiers arrêts dès 100 secondes de simulation). D'ailleurs, le phénomène de congestion sur la voie d'insertion apparaît légèrement plus tard (à 280 secondes de simulation environ). Certains véhicules sont plus lents, facilitant l'insertion de véhicules. Cependant, les conséquences sur la voie principale se ressentent (zones rouges sur le diagramme de la voie principale).

On se rend aussi compte que, bien que le couple IDM / $MOBIL$ paraisse adapté pour des situations où le changement de voie est facultatif, ce n'est pas toujours le cas pour les changements de voie obligatoires. En effet, cela est dû au fait qu'il se contente d'analyser les propriétés des autres véhicules et d'effectuer des prédictions sur le court terme. Ainsi, il ne prend pas en compte l'intention exacte du véhicule et ne communique pas avec celui-ci pour s'insérer plus facilement. La notion de courtoisie est donc toute relative, et à pour conséquence de ne pas compliquer l'insertion obligatoire.

3.2.2 Améliorations du modèle

L'objectif actuel est de proposer un modèle permettant d'éviter le plus possible une insertion à l'arrêt en bout de voie, responsable en partie des phénomènes d'embouteillages dans ce type de route. Comme nous l'avons vu précédemment, les conditions du couple IDM / $MOBIL$ paraissent trop fortes pour pouvoir gérer l'éventail de situations possibles dans une voie d'insertion. Afin de s'assurer de cela, nous avons décidé de retirer la condition *incentive* (Equation 3.4) de $MOBIL$ et ainsi, prendre en compte uniquement la condition *safety* (Equation 3.3) du modèle. Cela va permettre de faciliter le changement de voie tout en restant sécurisé. Toutefois, le véhicule sera moins courtois avec les véhicules de la voie principale. De plus, cette simplification permettrait d'imaginer un modèle décisionnel plus adapté pour ce type de route tout en conservant l'aspect sécuritaire.

Afin d'évaluer ce nouveau modèle que l'on appellera $MOBIL_{safe}$, nous allons mettre en place les deux études de cas mises en place précédemment (trafic non

mixte / mixte) tout en comparant les deux modèles $MOBIL$ et $MOBIL_{safe}$. Après analyse des Figures 3.7 (pour le trafic non mixte) et 3.8 (pour le trafic mixte), on constate les éléments suivants :

— Dans un trafic non mixte (voir Figure 3.7), on remarque que $MOBIL_{safe}$ facilite l'insertion du véhicule et conserve globalement une valeur haute du HPN par rapport à $MOBIL$. De plus, le phénomène de congestion arrive plus tardivement (à 450 secondes de simulation environ), bien qu'il arrive que certains véhicules restent bloqués en voie d'insertion ;

— Dans un trafic mixte (voir Figure 3.8), les différences entre $MOBIL$ et $MOBIL_{safe}$ sont plus légères au niveau de la voie d'insertion (phénomène de congestion arrivant à 300 secondes de simulation environ). Tout comme avec $MOBIL$, la présence de véhicules lents facilite l'insertion.

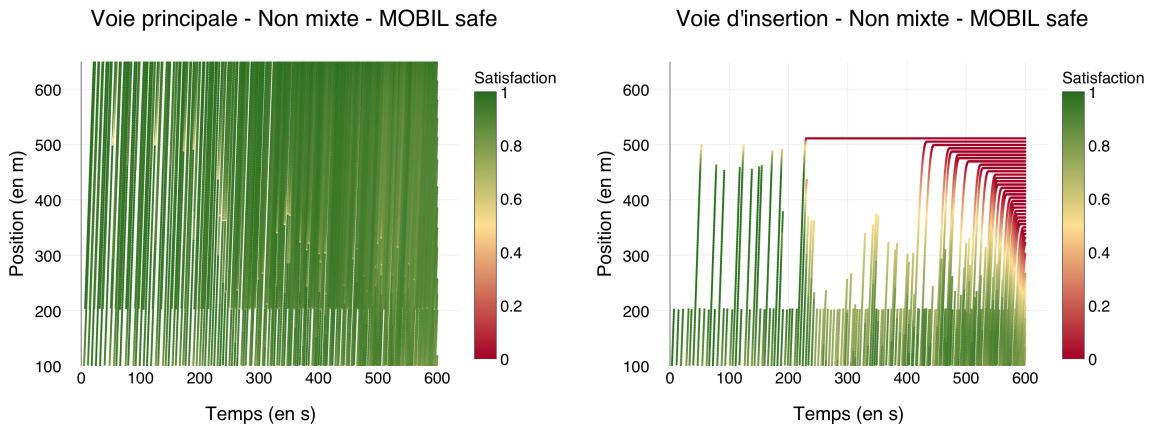


FIGURE 3.7: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL safe**.

Finalement, en retirant l'aspect *incentive*, on se retrouve avec un modèle faisant des changements de voies plus facilement, évitant de se retrouver avec un véhicule au bout de la voie d'insertion. Évidemment ce n'est pas suffisant car on observe toujours des problèmes d'insertion en raison du manque de courtoisie imposant aux véhicules de la voie principale de freiner plus brusquement. Toutefois, il permet un changement de voie sécurisé et nous laisse une certaine liberté pour définir un protocole de coopération pour améliorer la décision entre les véhicules.

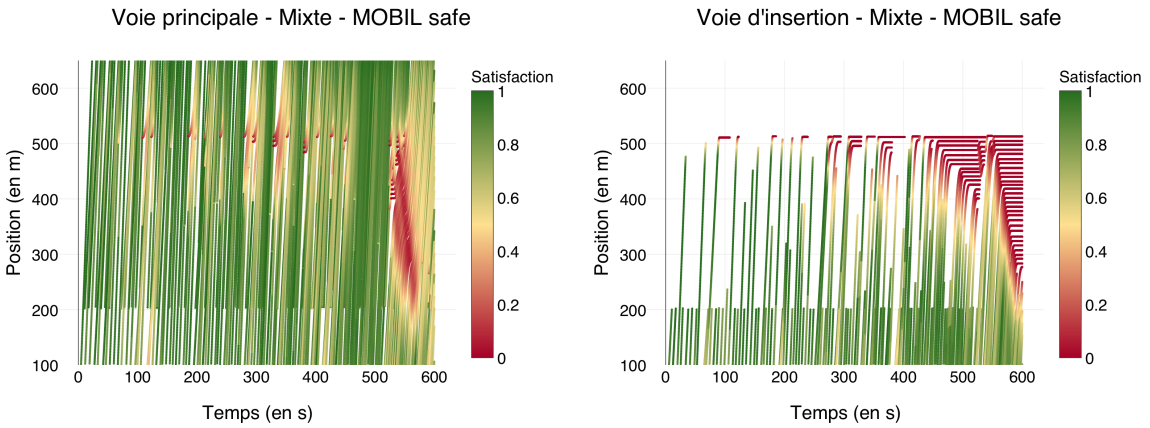


FIGURE 3.8: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL safe**.

3.3 Un protocole de coopération entre véhicules

3.3.1 Objectifs et modèle de décision

L'objectif est de concevoir un protocole de coopération entre véhicules communicants, permettant ainsi de faciliter un changement de voie obligatoire tout en prenant en compte le critère de courtoisie. Plusieurs modèles réalistes de véhicules ont été conçus pour étudier l'impact d'un environnement mixte [62] voir des solutions pour s'adapter dans un tel trafic [51].

Nous avons décidé de mettre en évidence le protocole de coopération à partir d'un scénario d'une voie d'insertion associée à une route principale à une voie. En effet, plusieurs cas de changement de voie obligatoire ne contiennent pas de route principale à plusieurs voies où l'on peut se déporter sur la gauche pour faciliter une insertion. Ces routes sont plus dangereuses car les décisions possibles à un instant donné sont plus restreintes (un véhicule sur la voie principale pourra uniquement accélérer, ralentir ou conserver sa vitesse).

La difficulté est de concevoir un véhicule capable d'anticiper un potentiel danger sans impacter négativement ses buts initiaux (atteindre le bout de la route le plus rapidement possible tout en étant sécurisé et confortable). De plus, bien que l'on souhaite concevoir un protocole de coopération, il peut arriver que l'information n'ait pas été transmise correctement d'une entité à une autre. On pourrait donc imaginer une communication "indirecte" où les véhicules vont se contenter de détecter les

autres véhicules avec qui ils vont convenir d'un accord commun à la manière de la théorie des jeux [66], lorsqu'on cherche un équilibre de Nash jugeant que les joueurs ont chacun intérêt à effectuer une stratégie particulière, déterminée par une valeur d'utilité [84].

Pour cela, nous allons déterminer trois points importants à définir pour le bon fonctionnement du protocole de coopération :

— la perception (*Qui doit communiquer avec qui ?*) : il faut définir dynamiquement quelles entités vont communiquer entre elles par rapport à leurs positions et/ou vitesses à un instant donné ;

— la décision (*Que dois-je faire à cet instant ?*) : à partir des caractéristiques des véhicules environnants et de ses propres caractéristiques, quelle décision il faut prendre ;

— l'action (*Comment le faire ?*) : après avoir déterminé la décision que le véhicule souhaite appliquer, comment l'exprimer au sein de celui-ci.

Ainsi, il est possible de définir le protocole de coopération comme une surcouche de la loi de poursuite IDM et du modèle de changement de voie simplifié $MOBIL_{safe}$ déjà défini. De plus, à partir de ce modèle de perception-décision-action, il pourra être implémenté facilement sur GAMA, tout comme sur d'autres outils de simulation.

3.3.2 Formalisation du protocole

Afin de formaliser ce protocole, il convient de définir différentes variables que l'on utilisera tout au long de cette partie, soient :

— $R = \{r_1, \ldots, r_n\}$ l'ensemble des routes observables pendant la simulation (avec n le nombre de routes) et r_i l'ensemble des véhicules se trouvant sur cette route ;

— c_i^k le véhicule ayant l'identifiant i se trouvant sur la route r_k, d'où $i \in r_k$. Chaque véhicule c_i^k possède une taille de longueur l_i, une position longitudinale x_i, une vitesse v_i et une accélération a_i ;

— $prev_i^k$ le premier véhicule situé derrière le véhicule i sur la route k ;

— $next_i^k$ le premier véhicule situé devant le véhicule i sur la route k ;

— $view_i$ l'ensemble des identifiants des véhicules que le véhicule i voit dans sa zone de perception ;

— $accIDM_{i,j}$ l'accélération calculée par IDM entre un véhicule i et un véhicule j à partir de leurs positions longitudinales (x_i et x_j) et leurs vitesses longitudinales (v_i et v_j).

— par_i^j la valeur d'un des paramètres fixes j du modèle IDM dans le véhicule i (avec $j \in \{a, b, s_0, T, v_{lim}, \delta\}$).

Définissons la route R observable pendant la simulation telle que $R = \{r_1, r_2\}$ avec r_1 comme étant la route permettant l'insertion et r_2 la route principale constituée d'une voie.

3.3.2.1 Perception

Il s'agit ici de déterminer à chaque pas de temps de simulation, quels sont les véhicules qui vont communiquer entre eux. Comme nous l'avons dit précédemment, l'objectif est de proposer un protocole qui peut se combiner aux modèles $IDM/MOBIL_{safe}$, sans les impacter directement. Ainsi, le modèle de changement de voie initial va se charger de garantir une décélération maximale autorisée pour le véhicule s'insérant. Notre objectif pour ce véhicule est, à la manière du critère *incentive* de $MOBIL$, d'impacter le moins possible le véhicule situé derrière lui sur la voie principale tout en facilitant le changement de voie imposé par $MOBIL_{safe}$.

Pour cela, nous allons tout d'abord définir une zone où les véhicules pourront communiquer (définie dynamiquement à partir du moment où les véhicules de chaque voie peuvent s'observer). Ensuite, nous allons établir une priorité sur les véhicules présents sur la voie d'insertion à partir de cette zone (r_1) : le véhicule le plus proche de la fin de voie est prioritaire pour communiquer. Celui-ci va communiquer avec un véhicule présent sur la voie principale (r_2) s'il y en a un. Selon la perception observée à un instant donnée, on va définir différentes règles :

— on va d'abord observer le véhicule de la voie principale situé juste devant le véhicule qui s'insère. Si les deux véhicules sont proches au point d'entrer en collision au prochain pas de temps si un changement de voie était fait à ce moment là (véhicules superposés), alors les véhicules s'associent et on considère que c'est une communication "proche" ($close$) ;

— si le véhicule s'insérant observe un véhicule situé derrière lui sur la voie principale, alors les véhicules s'associent et on considère que c'est une communication "classique" ($normal$).

L'algorithme de perception (détaillé dans l'Algorithme 1) détermine les couples de véhicules qui vont communiquer entre-eux à chaque pas de temps de simulation. Il fonctionne de la façon suivante : pour chaque véhicule présent sur la voie d'insertion, on associe, si possible, un véhicule de la voie principale qui pourra communiquer uniquement avec ce véhicule. Si on détecte deux véhicules "proches" non associés, on les associe ensemble, sinon on associe deux véhicules "classiques" non utilisés. Si aucune des deux conditions n'est respectée, le véhicule s'insérant ne communiquera pas (voir exemple sur la Figure 3.9).

Algorithm 1 Perception()

Require: $R \leftarrow \{r_1, r_2\}$ ▷ r1 = voie d'insertion / r2 = voie principale
Ensure: $\forall i \in r_1$, r_1 sorted by x_i descending ▷ On trie les véhicules de r1 par ordre d'arrivée sur la voie

$onCouple, listClose, listNormal \leftarrow \{\}$
for all i of r_1 **do**
  if $\exists j \in r_2 \mid (next_i^2 = j)\ \ (j \in view_i)\ \ (j \notin onCouple)\ (x_j - x_i) < \frac{1}{2}(l_j + l_i)$ **then** ▷ On vérifie si on peut trouver un véhicule "proche"
    $onCouple \leftarrow^+ \{j\}$
    $listClose \leftarrow^+ \{i, j\}$
  else if $\exists j \in r_2 \mid (prev_i^2 = j)\ \ (j \in view_i)\ \ (j \notin onCouple)$ **then** ▷ On vérifie si on peut trouver un véhicule "classique"
    $onCouple \leftarrow^+ \{j\}$
    $listNormal \leftarrow^+ \{i, j\}$
  end if
end for
return $listClose, listNormal$ ▷ Renvoie la liste des couples "proches" et "classiques"

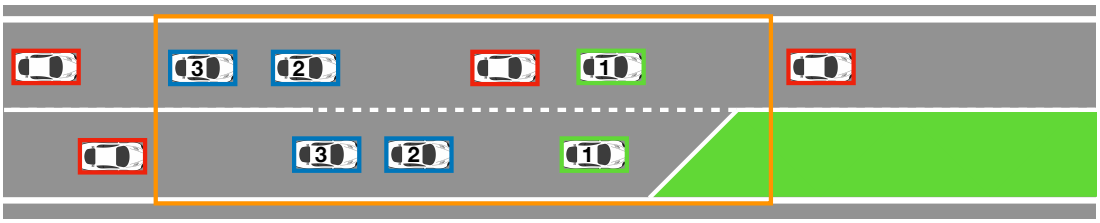


FIGURE 3.9: Application de l'algorithme de perception dans la zone orange. En vert, les couples "proches" (ici, un seul). En bleu, les couples "classiques" (ici, deux). En rouge, les véhicules ne communiquant pas.

3.3.2.2 Décision

Maintenant que la perception est définie dynamiquement, l'objectif est de déterminer les différentes décisions à prendre pour chaque véhicule de chaque couple associé dans le but de faciliter la possibilité au véhicule de s'insérer grâce à l'action de $MOBIL_{safe}$. Soient c_i^1 et c_j^2 deux véhicules communicants. On définit des règles selon la perception obtenue :

— si les deux véhicules sont "proches", le véhicule s'insérant c_i^1 va décélérer ($dec(i)$) afin qu'il puisse communiquer avec un véhicule "classique" et c_j^2 va accélérer ($acc(j)$) pour faciliter la séparation entre les deux véhicules ;

— si les deux véhicules sont "classiques", on va calculer deux accélérations à partir de la loi de poursuite IDM (voir Figure 3.10) : une première calculant l'accélération entre les deux véhicules communicant et une seconde calculant l'accélération entre le véhicule s'insérant et le premier véhicule situé juste devant lui sur la voie principale.

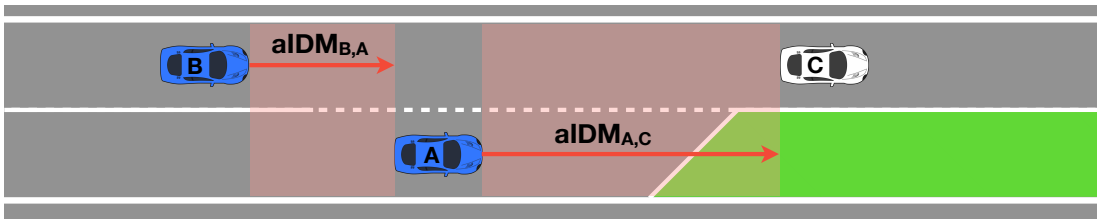


FIGURE 3.10: Accélérations observées lorsque deux véhicules sont considérés comme "classiques". Le véhicule A détermine l'accélération calculée par IDM avec le véhicule C ($aIDM_{A,C}$). Le véhicule B détermine l'accélération calculée par IDM avec le véhicule A ($aIDM_{B,A}$)

Si c_i^1 et c_j^2 sont deux véhicules "classiques", on va donc définir les deux formules 3.12 et 3.13 suivantes.

$$u_1 = aIDM_{i,next_i^2} + b_{safe} \tag{3.12}$$

avec : b_{safe} la valeur positive de décélération limite conseillée dans le critère de sécurité de $MOBIL$ (en $m.s^{-2}$).

$$u_2 = aIDM_{j,i} + b_{max} \tag{3.13}$$

avec : b_{max} la valeur positive de décélération maximale autorisée pour un véhicule (en $m.s^{-2}$).

u_1 représente le seuil de l'accélération calculée par IDM que la voiture située sur la voie principale souhaite atteindre. u_2 représente alors le seuil du véhicule s'insérant avec le véhicule situé devant lui s'il change de voie. Selon les valeurs de u_1 et u_2 obtenues, les décisions diffèrent pour chaque véhicule. Comme on peut le voir sur la Table 3.4 :

— si $u_1 > 0$, c_i^1 accélère ($acc(i)$) car sa distance de sécurité avec le véhicule situé devant lui sur la voie principale est suffisante. Sinon, il ralentit ($dec(i)$) ;

— si $u_2 > 0$, c_j^2 conserve sa vitesse ($keep(j)$) car il a atteint une vitesse suffisante pour faciliter le changement de voie de c_i^1. Sinon, il ralentit ($dec(j)$).

Si pour un trafic non mixte, la valeur de u_1 convient, cela ce complique dans le cas d'un trafic mixte. En effet, chaque véhicule ayant ses propres paramètres de la loi de poursuite IDM, la valeur finale calculée par celle-ci peut être faussée et provoquer des incohérences au niveau des décisions. Afin de remédier à ce problème, nous reprenons le terme $\frac{s^*(v_i,\Delta v_i)}{s}$, pour lequel nous établissons les paramètres de IDM au pire cas. Ce qui donne la valeur $u_1{}'$ donnée à la Formule 3.14. Ainsi, pour les cas de trafic non mixte, on utilisera u_1. Sinon, on utilisera $u_1{}'$. u_2 étant directement relié au véhicule étudié, il conserve l'accélération calculée par IDM.

$$u_1{}' = \frac{s^*_{worst}(v_i, \Delta v_i)}{s} \tag{3.14}$$

avec s^*_{worst} la valeur maximale que peut prendre s^* pour des valeurs au pire cas de a, b, s_0 et T donnés.

	$u_1 > 0$	$u_1 \leq 0$
$u_2 > 0$	$acc(i), keep(j)$	$dec(i), keep(j)$
$u_2 \leq 0$	$acc(i), dec(j)$	$dec(i), dec(j)$

TABLEAU 3.4: Décisions à prendre dans le cas où les deux véhicules c_i^1 et c_j^2 communiquent et sont "classiques".

Ainsi, on obtient un algorithme de décision (détaillé dans l'Algorithme 2). A chaque pas de temps de simulation, on va vérifier si chaque véhicule souhaitant s'insérer peut changer de voie grâce à l'action du critère de sécurité (*safety*) de $MOBIL$

(Equation 3.3). Si ce n'est pas le cas, après application de l'algorithme de perception, si les deux véhicules sont "proches", le véhicule souhaitant s'insérer ralentit, permettant de faciliter son changement de voie ou de communiquer avec un véhicule "classique". Dans ce cas, les deux véhicules appliquent les décisions relatives à la Table 3.4.

3.3.2.3 Action

Maintenant que chaque véhicule communicant sait quelle décision il doit prendre, il faut définir les différentes décisions possibles. En effet, $acc()$, $dec()$ ne sont pas des valeurs définies. Il est possible de déterminer dynamiquement une valeur par rapport à la position du véhicule et sa position par rapport à la fin de la voie d'insertion ($dynamicAcc()$). Ainsi, l'équation 3.15 donne les valeurs pour un véhicule i de $dynamicAcc(i)$, $acc(i)$ et $dec(i)$ suivantes ($keep(i)$ étant égal à $0\ m.s^{-2}$) :

$$\begin{aligned} dynamicAcc(i) &= \frac{par_i^b \times dLim}{\mathsf{max}(x_{onramp} - x_i, dLim)} \\ dec(i) &= -dynamicAcc(i) \\ acc(i) &= \mathsf{min}(par_i^a, dynamicAcc(i)) \end{aligned} \tag{3.15}$$

avec :

— par_i^b : la valeur de décélération conseillée définie sur IDM pour le véhicule i (en $m.s^{-2}$)

— par_i^a : la valeur d'accélération maximale définie sur IDM pour le véhicule i (en $m.s^{-2}$)

— $dLim$: les $dLim$ mètres avant lesquels il est dangereux se trouver sur la voie d'insertion impliquant d'appliquer la plus grande accélération / décélération possible (que l'on fixera à 50 mètres par défaut) ;

— x_{onramp} : la position en abscisse de la fin de la voie d'insertion.

Ainsi, grâce à ce modèle de perception-décision-action, il est possible de définir un modèle de décision centralisé (via une borne sur la route) ou décentralisé (modèle pour chaque véhicule) dans des situations similaires à la voie d'insertion où un changement de voie obligatoire est nécessaire.

Algorithm 2 Decision()

Require: $R \leftarrow \{r_1, r_2\}$ ▷ r1 = voie d'insertion / r2 = voie principale

$listDecisions \leftarrow \{\}$
for all i of R **do**
  if $i \in r_1$ $MOBIL_{safe}$ **then** ▷ On change de voie si le modèle MOBIL safe le permet
    $listDecisions \leftarrow^{+} \{i, changeLane(i)\}$
  else
    $listClose$ of $Perception()$
    $listNormal$ of $Perception()$
    if $\exists j \mid (\{i,j\} \in listClose \ \ \{j,i\} \in listClose$ **then** ▷ Si le véhicule est dans un couple "proche"
      if $i \in r_1$ **then**
        $listDecisions \leftarrow^{+} \{i, dec(i)\}$
      else
        $listDecisions \leftarrow^{+} \{i, acc(i)\}$
      end if
    else if $\exists j \mid (\{i,j\} \in listNormal \ \ \{j,i\} \in listNormal$ **then** ▷ Si le véhicule est dans un couple "classique"
      if $i \in r_1$ **then**
        $u_1 \leftarrow accIDM_{i,next_i^2} + b_{safe}$ ▷ Calcul par IDM de l'accélération du véhicule s'insérant avec le véhicule situé devant lui sur la voie principale
        if $u_1 > 0$ **then** ▷ On vérifie si le seuil d'accélération est atteint
          $listDecisions \leftarrow^{+} \{i, acc(i)\}$
        else
          $listDecisions \leftarrow^{+} \{i, dec(i)\}$
        end if
      else
        $u_2 \leftarrow accIDM_{i,j} + b_{max}$ ▷ Calcul par IDM de l'accélération du véhicule situé sur la voie principale avec le véhicule s'insérant
        if $u_2 > 0$ **then** ▷ On vérifie si le seuil d'accélération est atteint
          $listDecisions \leftarrow^{+} \{i, keep(i)\}$
        else
          $listDecisions \leftarrow^{+} \{i, dec(i)\}$
        end if
      end if
    else ▷ Si le véhicule se trouve dans aucun couple, il applique le modèle longitudinal IDM
      if $i \in r_1$ **then**
        $listDecisions \leftarrow^{+} \{i, accIDM_{i,next_i^1}\}$
      else
        $listDecisions \leftarrow^{+} \{i, accIDM_{i,next_i^2}\}$
      end if
    end if
  end if
end for
return $listDecisions$ ▷ Retourne la liste des décisions pour chaque véhicule présent sur la voie d'insertion

3.3.3 Résultats et comparaisons avec $MOBIL$ et $MOBIL_{safe}$

Maintenant que nous avons un protocole de coopération défini (que l'on appellera $MOBIL_{coop}$), il convient de le comparer avec les modèles $MOBIL$ et $MOBIL_{safe}$ définis précédemment afin de déterminer ses potentiels apports et limitations. Pour cela, nous allons reprendre le même scénario utilisé pour comparer $MOBIL$ et $MOBIL_{safe}$ (voir tableau 3.3).

Nous avons observé les flux sur la voie principale et la voie d'insertion dans les deux cas d'étude (trafic mixte et non mixte) et constaté les choses suivantes :

— Dans un trafic non mixte (voir Figure 3.11), les véhicules s'insèrent globalement plus facilement lorsque le trafic est faible. Toutefois, leurs valeurs de satisfaction (HPN) sont globalement plus faibles lorsque l'insertion de ne se fait pas instantanément en raison des potentielles décélérations effectuées. Mais la valeur de satisfaction n'atteint pas des valeurs inférieures à 0.2 contrairement aux modèles $MOBIL$ et $MOBIL_{safe}$. Lorsque le trafic est plus élevé (notamment en fin de simulation), les véhicules commencent à avoir des difficultés à s'insérer mais ils y arrivent avec beaucoup plus de facilité qu'avec $MOBIL$ et $MOBIL_{safe}$ où l'on observe des embouteillages en voie d'insertion. En fin de simulation, la valeur d'HPN sur la voie principale est légèrement plus faible, expliquée par la présence d'un plus grand nombre de véhicules ayant réussi à s'insérer pour $MOBIL_{coop}$;

— Dans un trafic mixte (voir 3.12), on observe globalement les mêmes phénomènes. Si dans un trafic faible, on n'observe pas de différences particulières, dans un trafic élevé, les difficultés d'insertion se font un peu plus nombreuses. En effet, un début d'embouteillage se forme en fin de simulation (580 secondes), mais cela reste plus efficace que les modèles $MOBIL_{safe}$ et $MOBIL$. En fin de simulation, sur $MOBIL_{coop}$, on observe des zones oranges où la valeur de satisfaction est plus faible. Cela est dû à des insertions plus difficiles faites à faible vitesse. Cela augmente la congestion sur la voie principale sans pour autant toutefois former un embouteillage tout de suite.

On remarque donc que l'ajout du protocole de coopération facilite grandement le changement de voie tout en gardant une notion de courtoisie entre les véhicules. Il est évident que le modèle est limité lorsque le nombre de véhicules est trop grand

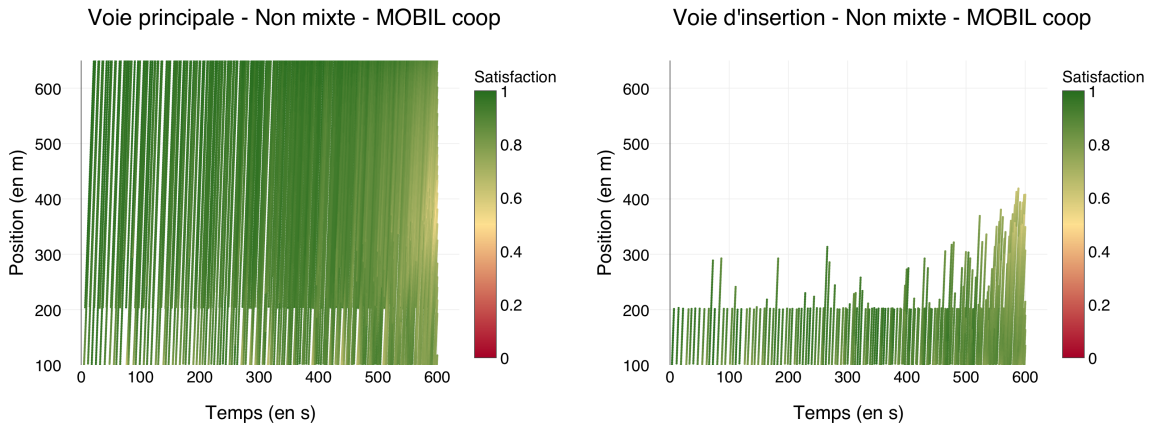


FIGURE 3.11: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL coop**.

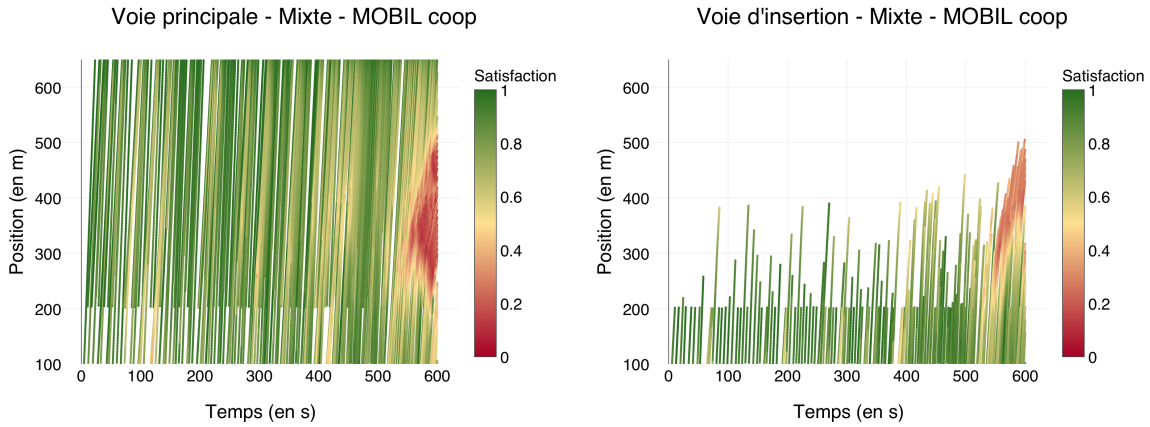


FIGURE 3.12: Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL coop**.

avec des caractéristiques différentes. Toutefois, il en reste un modèle intéressant pour les raisons suivantes :

— c'est un modèle qui s'intègre directement au modèle du changement de voie sécurisé $MOBIL_{safe}$ sans l'impacter directement ;

— il peut s'adapter à toute loi de poursuite et tout modèle de changement de voie ;

— il permet un changement de voie sans nécessité de faire de prédictions sur la durée ;

— il peut s'adapter à des environnements plus complexes où le nombre de décisions possibles est plus grand (route principale à plusieurs voies) ;

— c'est un modèle de décision qui peut être à la fois centralisé (via une unité RSU présente sur la route avec une perception globale très bonne) et décentralisé (au sein d'un véhicule avec une perception plus limitée) ;

— il est possible d'imaginer ce protocole sur un véhicule autonome non communicant effectuant les décisions à partir d'une simple analyse de l'environnement. En effet, il peut calculer une accélération au pire cas de chaque véhicule avec lequel il va "s'associer" afin de faciliter l'insertion des véhicules ;

— ce modèle de décision reprend le principe de courtoisie lors d'un changement de voie obligatoire ;

— il est capable de s'adapter à tout type de trafic (mixte/non mixte, faible/élevé).

Ces deux derniers points sont intéressants car ils rendent possible l'adaptation du protocole aux routes actuelles constituées de véhicules ni autonomes ni communicants. C'est sur ce point que nous allons nous concentrer dans la partie 3.4 en se posant la question suivante : un tel véhicule peut-il s'adapter dans un environnement constitué de véhicules non communicants ?

3.4 Prise en compte de véhicules non communicants sur le protocole

3.4.1 Motivations

Dans la section 3.3, nous avons montré par simulation qu'un véhicule équipé de notre protocole de coopération était capable de s'adapter dans toute densité de trafic, même si les véhicules ont des caractéristiques qui leur sont propres (accélération, taille du véhicule, vitesse maximale...). On peut alors supposer qu'un tel véhicule puisse d'adapter dans un environnement constitué de véhicules non communicants. Plusieurs modèles existants ont proposé de solutions de coopération en prenant en compte de tels véhicules. On peut citer la mise en place d'un modèle bilatéral basé sur la confiance observant l'impact des véhicules communicants dans un environnement constitué de véhicules non communicants [43] à partir de plusieurs scénarios dont une démonstration a été mise en place [42]. On peut également citer un modèle de loi de poursuite prenant en compte des comportements timides et

agressifs de véhicules pouvant être associés à des conducteurs humains [67].

Ces véhicules non communicants peuvent être de deux types :

— *des véhicules non communicants autonomes* : ils possèdent un modèle longitudinal et latéral représentés respectivement par IDM et $MOBIL_{safe}$. Ils possèdent toutefois des caractéristiques proches des véhicules autonomes communicants (temps de réaction faible et accélération maximale élevée) ;

— *des véhicules non communicants et non autonomes* : ils possèdent un modèle longitudinal et latéral représentés respectivement et uniquement par IDM et $MOBIL$. Contrairement à l'autre type, ils peuvent posséder des caractéristiques très différentes (temps de réaction, distances de sécurité. . .). Ils peuvent représenter un véhicule conduit par un être humain rationnel (dans le sens où son intention n'est pas de provoquer un accident).

Dans ce cas des êtres humains rationnels, il est important de définir des paramètres permettant de définir différentes conduites. Cela est possible en changeant les paramètres d'IDM et de $MOBIL$, pour obtenir trois profils différents :

— *des véhicules classiques* : ce sont les véhicules que l'on voit le plus sur les routes actuelles, respectant le code de la route (ne dépassent pas la vitesse limite de la route) et pouvant être altruistes par moments (paramètre p de $MOBIL$) ;

— *des véhicules lents* : tout comme les véhicules classiques, ils respectent le code de la route mais leur temps de réaction est plus élevé et leur vitesse maximale plus faible ;

— *des véhicules agressifs* : contrairement aux deux autres types, ils peuvent ne pas respecter les limitations de vitesse de la route et accélèrent et décélèrent plus brusquement. Du fait qu'ils agissent rapidement, le temps de réaction de ces véhicules sera généralement faible.

L'objectif est d'observer avant tout l'influence des véhicules autonomes communicants sur des environnements actuels (êtres humains) ou futurs sur un court terme (véhicules autonomes de niveau 3). Pour cela, nous définissons deux études de cas : une première ajoutant des véhicules uniquement autonomes dans l'environnement et une autre ajoutant des véhicules conduits par des êtres humains. Dans ces

études de cas, nous faisons varier le pourcentage de véhicules non communicants présents et d'observer s'il existe un seuil où le protocole apporte positivement et/ou négativement quelque chose dans l'environnement global.

Pour observer l'apport du protocole sur un tel environnement, nous reprenons la voie d'insertion avec les paramètres du tableau 3.3. Tout comme dans la section 3.2, nous simulons un trafic fluide qui se congestionne au fur et à mesure du temps. A partir de ces simulations, nous observons les critères suivants en comparant les valeurs obtenues pour les véhicules communicants et non communicants :

— *Waiting time* moyen en voie d'insertion : cela permet d'observer si des véhicules se retrouvent à l'arrêt en voie d'insertion (impliquant un temps de parcours plus élevé) ;

— *Travel time* moyen des véhicules générés en voie principale : cela permet notamment d'observer l'impact des véhicules présents en voie d'insertion sur les véhicules générés en voie principale.

3.4.2 Étude de cas 1 : Ajout de véhicules autonomes non communicants dans l'environnement

Nous avons effectué plusieurs simulations pour différents pourcentages de véhicules communicants et établi une moyenne sur les valeurs obtenues. Les résultats obtenus pour le temps d'attente moyen en voie d'insertion, présents sur la Figure 3.13, permettent d'apporter les remarques suivantes :

— grâce au protocole de coopération, le temps d'attente des véhicules en voie d'insertion est beaucoup plus faible que dans l'environnement non coopératif. On observe ces améliorations à partir de 20% de véhicules communicants ;

— dans l'environnement coopératif, le protocole de coopération peut avoir un impact négatif sur le temps d'attente des véhicules autonomes non communicants, lorsque le nombre de véhicules communicants est inférieur à 60%. Toutefois, le temps d'attente augmente seulement de 15% environ par rapport à l'environnement non coopératif (environ 1 seconde de plus en moyenne) ;

— malgré ce dernier point, le temps d'attente moyen des véhicules est globalement plus faible à partir de 20% de véhicules communicants.

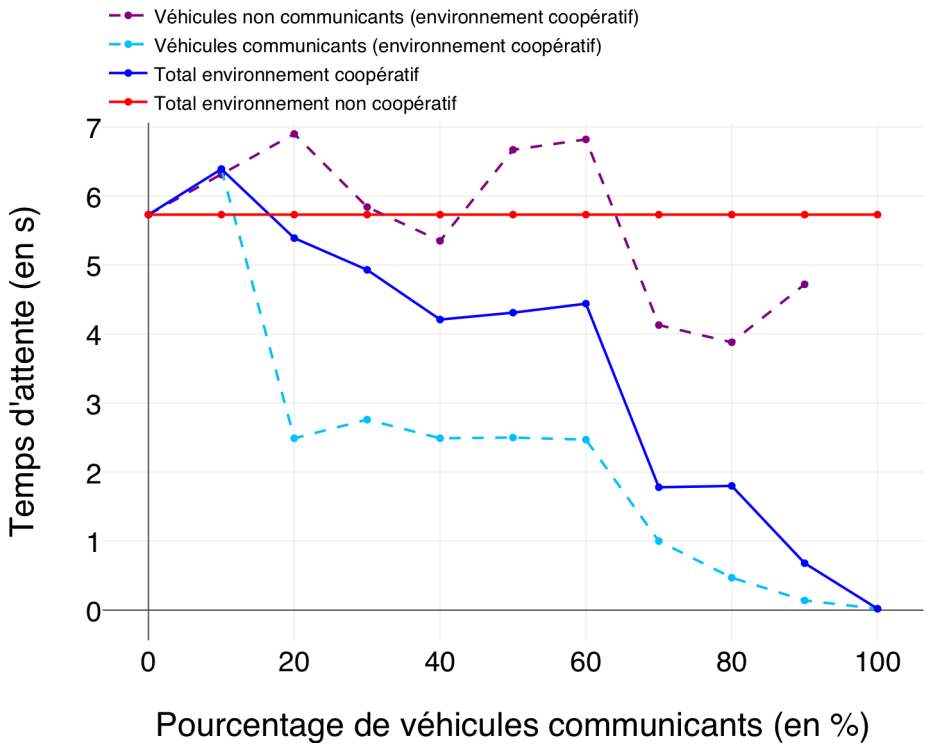


FIGURE 3.13: **Temps d'attente moyen** des véhicules (*Waiting Time*) en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules autonomes non communicants**).

Lorsqu'on observe les valeurs du temps de parcours des véhicules dans chaque voie (voir Figure 3.14), on remarque les éléments suivants :

— les véhicules dans l'environnement coopératif ont un temps de parcours généralement inférieur au temps de parcours obtenu dans un milieu intégralement non coopératif ;

— dans l'environnement coopératif, le gain est globalement négligeable (8% environ). Mais les véhicules coopératifs n'ont pas réellement d'impact négatif sur les véhicules non coopératifs ;

Ainsi, le protocole paraît s'adapter sans problème dans un environnement mixte constitué de véhicules autonomes non communicants. Le temps d'attente en voie d'insertion est réduit et l'impact sur la voie principale est plutôt faible. Toutefois, l'impact sur les véhicules autonomes non communicants est plus grand car le protocole priorise les véhicules communicants lors de l'insertion.

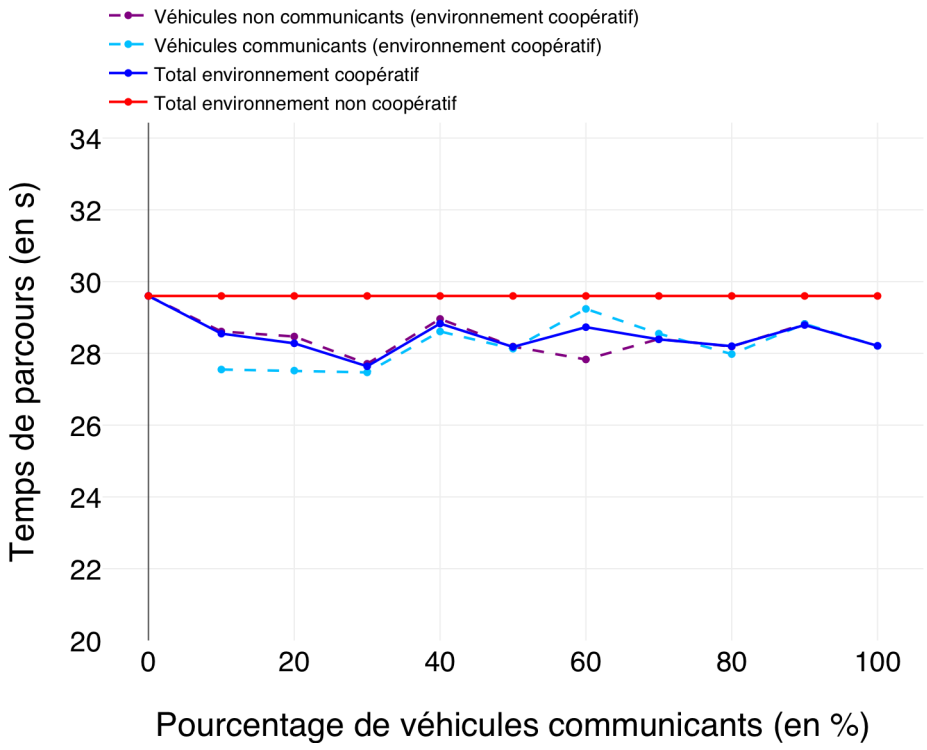


FIGURE 3.14: **Temps de parcours moyen** (*Travel Time*) des véhicules générés en voie principale en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules autonomes non communicants**).

3.4.3 Étude de cas 2 : Ajout de véhicules non autonomes et non communicants dans l'environnement

Tout comme la première étude de cas, nous avons effectué plusieurs simulations pour différents pourcentages de véhicules communicants et établi une moyenne sur les valeurs obtenues. Les résultats obtenus pour le temps d'attente moyen en voie d'insertion, présents sur la Figure 3.15, permettent d'apporter les remarques suivantes :

— grâce au protocole de coopération, le temps d'attente des véhicules en voie d'insertion est beaucoup plus faible que dans un environnement non coopératif. On observe de grandes améliorations à partir de 30% de véhicules communicants ;

— le protocole a un impact négatif sur le temps d'attente des véhicules non communicants, mais ils gardent un temps d'attente globalement plus faible que dans un milieu non coopératif ;

— les véhicules communicants réduisent de près de 60% le temps d'attente par rapport à des véhicules présents dans un milieu non coopératif.

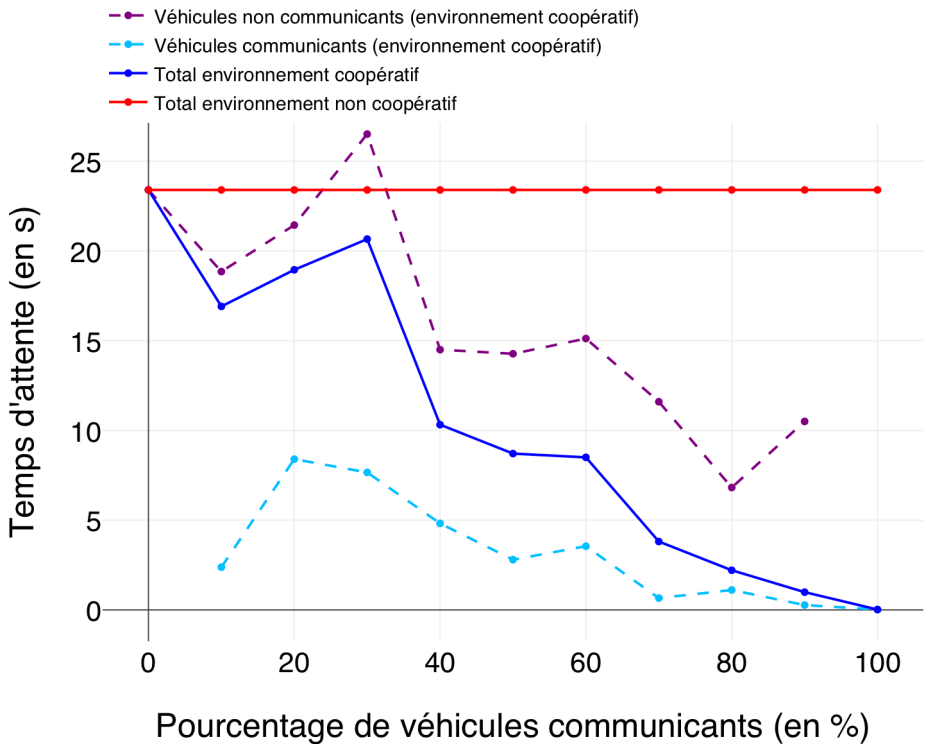


FIGURE 3.15: **Temps d'attente moyen** des véhicules (*Waiting Time*) en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules non autonomes et non communicants**).

Lorsqu'on observe les valeurs du temps de parcours des véhicules dans chaque voie (voir Figure 3.16), on remarque les éléments suivants :

— à partir de 60% de véhicules communicants, on obtient une amélioration du temps de parcours de l'ordre de 5% comparé à un environnement non coopératif ;

— en dessous de ces 60%, le temps reste plus élevé mais reste toutefois négligeable (augmentation de l'ordre de 7%).

Finalement, le constat avec des véhicules non autonomes et non communicants est proche de l'étude de cas précédente bien que les temps d'attente et de parcours soient globalement plus élevés en raison du comportement plus agressifs des véhicules conduits par des êtres humains. Il reste toutefois difficile de définir des comportements humains très réalistes pouvant empêcher le protocole de fonctionner (véhicules contre-coopératifs par exemple). Une autre explication peut venir

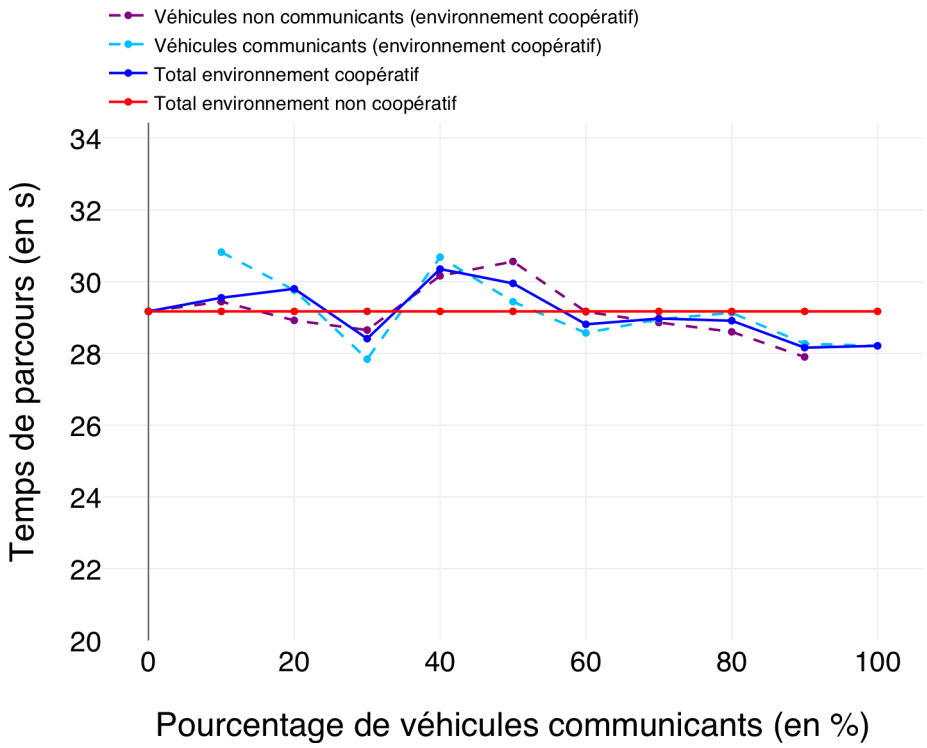


FIGURE 3.16: **Temps de parcours moyen** (*Travel Time*) des véhicules générés en voie principale en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules non autonomes et non communicants**).

du fait d'utiliser le modèle $MOBIL$ plutôt que $MOBIL_{safe}$ pour les véhicules conduits par des êtres humains, bien que cela montre que le protocole peut s'adapter dans un environnement plus perturbé. C'est sur ce point qu'il va falloir insister pour mieux étudier l'impact du protocole dans un tel environnement, notamment en ajustant le modèle de décision d'un véhicule conduit par un être humain.

3.5 Synthèse

Nous avons déterminé le niveau d'abstraction de notre modèle en l'associant à des modèles existants grâce à des lois de poursuite comme IDM et à des modèles de changement de voie comme $MOBIL$, représentant respectivement la dimension longitudinale et latérale de la route. L'association de ces deux modèles permet d'obtenir un véhicule ayant une certaine autonomie. Toutefois, on remarque assez rapidement que $MOBIL$ souffre de quelques limitations lors de changements de voie obligatoires. Nous l'avons alors redéfini puis associé à un protocole de coopération pour faciliter le changement de voie, rendant le véhicule plus courtois.

Ce protocole de coopération donne de bonnes améliorations dans un trafic non mixte comme dans un trafic mixte. Il s'adapte également facilement à un environnement constitué de véhicules n'appliquant pas ce protocole. La prise en compte de conducteurs humains rationnels montre des difficultés dans la conception de ce type de comportement en utilisant uniquement les modèles IDM et $MOBIL$. Le simple changement des paramètres ne suffit pas pour concevoir des véhicules régissant aux actions des autres (être contre-coopératif, moins courtois...).

Grâce à la mise en place de ce protocole, nous avons proposé une première méthode de conception du modèle de décision à partir d'un système de perception, décision, action. L'aspect décisionnel tel qu'il a été conçu permet d'imaginer un ensemble de fonctionnalités indépendantes. Ainsi, le protocole de coopération n'a pas d'impact sur le modèle de changement de voie associé par exemple.

A présent, nous avons donc un modèle paramétrable simulé répondant à un ensemble d'exigences. L'objectif est à présent de déterminer comment ces exigences peuvent être garanties. C'est ce que nous allons voir dans le chapitre 4 où une méthodologie va être appliquée pour assurer un lien entre le modèle de simulation et le modèle de vérification formelle à partir d'une étude de cas sur la loi de poursuite IDM. Après l'application de cette méthodologie, nous allons définir les éléments nécessaires pour combiner le modèle final dans un simulateur de véhicules ayant une dynamique et un environnement plus réalistes.

4 | Méthodologie de conception de véhicules intelligents

4.1 Analyse de nouvelles propriétés du modèle par vérification formelle . . 94

4.1.1 Motivations et méthodologie 94

4.1.2 Vérification formelle 96

4.1.3 Comparaisons avec le modèle de simulation 99

4.1.4 Application de la méthode 101

4.1.5 Synthèse 104

4.2 Interfaçage du modèle avec des modèles plus réalistes 104

4.2.1 Motivations 104

4.2.2 Différentes représentations de l'aspect latéral 105

4.2.3 Comparaison entre une trajectoire simplifiée et réaliste pour des véhicules autonomes non communicants 108

4.2.4 Intérêt d'une trajectoire instantanée de changement de voie dans le cas de véhicules autonomes communicants 111

4.2.5 Synthèse 114

4.3 Synthèse : proposition d'une méthodologie 114

Au chapitre 3, nous avons défini un protocole de coopération montrant tout son intérêt pour des véhicules intelligents. Toutefois, ce protocole a été validé uniquement par simulation, ne garantissant pas les propriétés observées. Nous allons donc utiliser la vérification formelle afin de permettre l'analyse de ces propriétés à partir d'une étude de cas basée sur le modèle longitudinal IDM.

Après cette étape, l'objectif est de déterminer comment le modèle ainsi obtenu peut s'implanter sur un véhicule réel. En effet, actuellement le modèle du véhicule est défini à un haut niveau d'abstraction qui peut paraître au premier coup d'œil assez éloigné de la réalité, notamment par la présence d'un changement de voie instantané. Pour cela, nous allons comparer plusieurs trajectoires de changement de voie de précisions différentes. A partir d'une étude de cas, nous allons déterminer comment assurer une compatibilité entre notre modèle actuel et un modèle réaliste.

Enfin, en regroupant tous les outils et méthodes détaillés jusqu'ici, nous allons proposer une méthodologie complète de conception de véhicules intelligents.

4.1 Analyse de nouvelles propriétés du modèle par vérification formelle

4.1.1 Motivations et méthodologie

La simulation permet actuellement de modéliser les comportements de véhicules dans un environnement choisi dont plusieurs types de scénarios peuvent être testés dans un temps plus court. Toutefois, quand les véhicules présentent des comportements non déterministes (comportements proposant plusieurs actions possibles à un instant donné), les outils de simulation ne sont généralement pas exhaustifs. En effet, chaque simulation correspond à un seul chemin dans le graphe de l'ensemble de tous les comportements possibles.

Dans le but d'améliorer ce dernier point, la technique de la vérification formelle (*model-checking*), qui est une méthode exhaustive, peut être utilisée. Les outils de model-checking permettent d'obtenir automatiquement des réponses binaires (oui

ou non) à des questions concernant le comportement dynamique des véhicules. Pour faire cela, des modèles spécifiques de véhicules dans un environnement donné doivent être fournis. Cependant, l'utilisation de ces outils est généralement limité, car faisant face au phénomène bien connu de l'explosion de l'espace des états.

Nous visons à proposer une méthode basée sur la combinaison des approches de simulation et de vérification formelle dans le but d'en tirer les avantages de chacun dans des cas non-déterministes. Le niveau d'abstraction doit être soigneusement choisi afin de garantir une représentation convaincante du comportement des véhicules obtenue en simulation. A cause de l'explosion de l'espace des états, le même niveau d'abstraction ne peut pas être utilisé lors de la vérification formelle. De plus, on doit aussi discrétiser la plupart des grandeurs, ce qui peut conduire à des erreurs de calculs et à des cumuls de ces erreurs, induisant un certain écart par rapport à la simulation. Pour cette raison, nous devons vérifier que les résultats obtenus sont relativement similaires.

Pour définir les niveaux d'abstraction souhaités, il est nécessaire de déterminer les principales propriétés d'intérêt à étudier avec ces outils. Ici, nous nous concentrons sur deux propriétés principales :

— la sécurité : elle va garantir que le véhicule respecte toujours les distances de sécurité et minimise le risque d'accident, qui peut être exprimé par la valeur du *Time-To-Collision* (TTC) définie dans la partie 3.1.4 ;

— la fluidité du trafic : elle va optimiser la vitesse de chaque véhicule pour ainsi réduire le phénomène de stop-and-go (embouteillages) et peut être exprimée par la valeur du *Travel Time*, également définie dans la partie 3.1.4.

Notre méthode est décrite en Figure 4.1. La première étape (étape 0 sur la figure) consiste à produire des données à l'aide de la simulation. Sur la base de telles données, un expert humain émet des hypothèses sur le comportement des véhicules autonomes pour une situation initiale donnée. Cela pourrait être par exemple : "tous les véhicules sont en sécurité tout le temps". Ce type de requête nécessite une vérification exhaustive dans le cas d'une évolution non déterministe du système. L'étape 1 consiste à exprimer la requête en logique temporelle (la traduction des propriétés en formules de logique temporelle peut être partiellement automatisée à l'aide d'un ensemble prédéfini de requêtes) et de vérifier formellement le modèle. Si le véri-

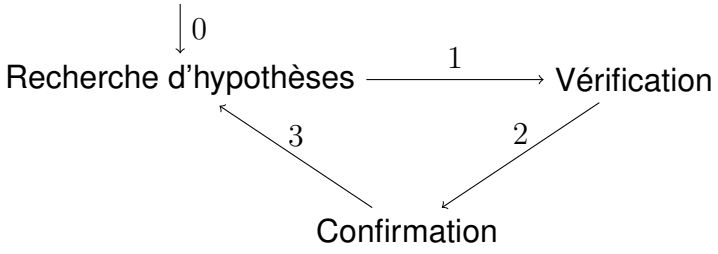


FIGURE 4.1: Étapes de la méthode de recherche de propriétés via la vérification formelle.

ficateur de modèle trouve une exécution menant à un contre-exemple, l'exécution correspondante est ensuite confirmée par la simulation, pour assurer que ce n'est pas un faux positif dû aux abstractions (étape 2). Dans le cas où il est confirmé que certains contre-exemples invalident l'hypothèse, le processus peut continuer avec une nouvelle hypothèse, grâce à l'enrichissement des données à travers le processus de vérification (étape 3). Plus précisément, la méthode nous permet d'ajouter des exécutions particulières aux données qui donneront des informations critiques afin de mieux comprendre le comportement d'un véhicule et ses causes.

Pour illustrer notre méthode, nous proposons une étude de cas, à savoir un modèle utilisant la loi de poursuite IDM dans une situation donnée, ce qui est plutôt simple pour la simulation, mais où la vérification du modèle doit faire face à une potentielle accumulation d'erreurs due à la discrétisation nécessaire des équations à nombres réels. En particulier, nous nous intéressons aux questions suivantes :

— Quelles sont les conditions requises pour qu'un véhicule se trouve dans une situation dangereuse ?

— Existe-t-il des exécutions où un véhicule est toujours en sécurité ?

— Est-il possible d'arrêter la voiture sans dépasser un seuil de décélération ?

4.1.2 Vérification formelle

La vérification formelle (ou *model-checking*) est une méthode permettant de résoudre des problèmes de décision complexes. Plus spécifiquement, il permet de vérifier les propriétés comportementales d'un système donné et fournit des exemples de comportements qui respectent ou violent la propriété vérifiée. Un outil de vérification de modèles peut être vu comme un opérateur qui utilise un modèle (par exemple un ensemble de comportements possibles de véhicules sur une portion

de route) et une propriété (par exemple la possibilité d'une collision entre les véhicules) et donne un résultat binaire (vrai ou faux). Cela nécessite généralement de modéliser le système comme une machine à états finis avec la formalisation d'une propriété comportementale sous forme de logique temporelle. Le résultat de la vérification est ensuite obtenu par l'inspection automatisée de tous les états du modèle, ce qui signifie que tous les futurs possibles d'une situation initiale donnée sont pris en compte pour évaluer la propriété. L'atout principal de cette méthode est qu'elle gère le non-déterminisme tout en garantissant l'exhaustivité. Toutefois, obtenir une réponse peut être difficile en raison du nombre d'états du modèle. En effet, comme une formalisation des caractéristiques du système est nécessaire, le modèle qui en résulte est souvent composé d'un très grand nombre d'états (souvent plusieurs milliards).

Par exemple, dans le cas de la modélisation de véhicules sur une portion de route, il faut prendre en considération :

— les variables nécessaires pour exprimer l'état d'un véhicule (position, vitesse, accélération, direction . . .) ;

— le coût de la prise de décision non déterministe (à quel point le non déterminisme va avoir un impact sur l'explosion de l'espace des états) ;

— le nombre de véhicules, augmentant de façon exponentielle le nombre d'états.

L'utilisation des abstractions permet de traiter le problème de l'explosion d'état, tout en impactant sur la fiabilité du système grâce à la discrétisation nécessaire. Actuellement, l'accumulation d'erreurs dues à la discrétisation et le calcul des nombres finis peut conduire à un comportement totalement différent de celui attendu. Cela dit, la vérification du modèle peut être intéressante dans la résolution de problèmes complexes, mais le modèle doit présenter un compromis satisfaisant entre réalisme et calculabilité [9].

Dans notre analyse combinée, nous utilisons une version légèrement modifiée (sans communication ni mouvement latéral) d'un modèle existant initialement consacré à l'évaluation de la robustesse des véhicules [10]. Il tourne sur le *model-checker* UPPAAL [91] [56], qui permet la vérification de propriétés exprimées dans un sous-ensemble de CTL (*Computationnal Tree Logic*) [30]. L'expressivité de ce langage de requête est généralement suffisant pour le type d'informations que nous voulons

obtenir. Le modèle considère un tronçon routier composé de plusieurs voies unidirectionnelles avec plusieurs agents (véhicules) sur chacune d'entre elles. Cet environnement est représenté par une structure de données contenant l'état de chaque véhicule avec un ensemble de contraintes sur leurs actions possibles. Un état du véhicule est conservé comme un ensemble de valeurs, y compris la position, la vitesse, la direction, la connaissance de l'environnement... La position actuelle est exprimée en utilisant des valeurs discrètes, mais avec suffisamment de précision pour modéliser la progression du véhicule sans entraîner un comportement anormal dû à la perte d'informations. Plus précisément, la position d'un véhicule est considérée comme un point sur une grille orthogonale bidimensionnelle. À une fréquence donnée, chaque véhicule utilise les informations dont il dispose pour prendre une décision sur l'action immédiate à effectuer (il agit sur l'accélération et la direction). Il effectue une action sur sa propre fréquence, c'est-à-dire que les véhicules peuvent être vus comme des agents indépendants. Les vitesses et les positions des véhicules sont mises à jour de manière simultanée, ce qui signifie que l'observation du système est indépendante de la fréquence de prise de décision des véhicules.

Dans l'étude de cas que nous allons effectuer dans les parties 4.1.3 et 4.1.4, nous supposons que chaque véhicule connaît la position et la vitesse exacte du véhicule qui le précède chaque fois qu'une décision est prise, ce qui signifie que les informations provenant des capteurs sont considérées comme parfaites. Cependant, le défi de cette étude de cas est d'exprimer la fonction continue d'une loi de poursuite d'un véhicule en utilisant un réglage discret, tout en atteignant une précision satisfaisante et un espace suffisamment petit pour être analysé dans un délai raisonnable. Pour ce faire, nous avons mis à l'échelle les variables avant la division, implémenté les fonctions d'arrondi, de puissance et de racine carrée (qui ne sont pas supportées par UPPAAL parce qu'elles ne sont généralement pas nécessaires pour la vérification des modèles) et avons judicieusement choisi la granularité des variables utilisées dans le modèle. Nous avons obtenu un modèle satisfaisant aux exigences de précision et qui peut être analysé par un *model-checker* en quelques minutes au maximum pour quelques véhicules sur une portion de route de quelques centaines de mètres. Pour tous les scénarios présentés dans les deux études de cas qui vont suivre, la portion de route observée fait 200 mètres de long, la longueur de chaque véhicule est exactement de 5 mètres et leur processus de prise de décision

se produit tous les 100 millisecondes.

4.1.3 Comparaisons avec le modèle de simulation

Nous voulons d'abord vérifier la différence de comportement entre la simulation et les approches de vérification formelle en comparant le temps de parcours à la fin de la portion de route et la position des véhicules à un moment donné (trois secondes après le début du scénario). Le premier scénario pose trois véhicules contrôlés par IDM sur une route à une voie à sens unique. La Figure 4.2 illustre ce scénario. Au départ :

— le véhicule A est en position $0\ m$ et sa vitesse est de $30\ m.s^{-1}$;

— le véhicule B est en position $50\ m$ et sa vitesse est de $25\ m.s^{-1}$;

— le véhicule C est à la position $100\ m$ et sa vitesse est de $20\ m.s^{-1}$.

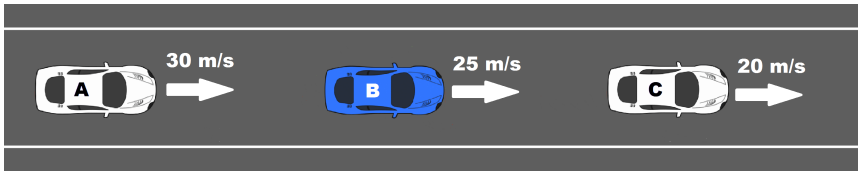


FIGURE 4.2: Situation initiale du premier scénario avec les vitesses respectives de chaque véhicule.

Ici, les véhicules A et B doivent adapter leur vitesse pour éviter une collision, ce qui semble être un bon exemple pour détecter une éventuelle accumulation d'erreurs. Les résultats de la comparaison sont présentés dans le tableau 4.1 et montrent que la position et le temps de parcours des véhicules (*travel time*) sont très proches, avec le pourcentage d'erreur le plus élevé étant sur le temps de parcours de B (0.52%). Ces résultats montrent que le comportement moyen obtenu pour les deux outils sont suffisamment similaires pour utiliser la position et le temps de déplacement comme des indicateurs fiables.

Critère	Temps de parcours [en s]			Position à 3 s [en m]		
	Véhicule A	Véhicule B	Véhicule C	Véhicule A	Véhicule B	Véhicule C
Simulation	$7.38\ s$	$5.68\ s$	$3.92\ s$	$80.51\ m$	$125.69\ m$	$174.03\ m$
Model-checking	$7.39\ s$	$5.71\ s$	$3.92\ s$	$80.74\ m$	$125.26\ m$	$173.99\ m$

TABLEAU 4.1: Comparaison du temps de parcours et de la position dans un temps donné pour les trois véhicules du premier scénario.

Ensuite, nous vérifions le comportement des indicateurs plus sensibles tels que le TTC ou la valeur d'accélération. Le scénario 2 comprend deux véhicules comme illustré dans la Figure 4.3 :

— un véhicule A contrôlé par IDM, initialement à la position $0\ m$ avec la vitesse de $20\ m.s^{-1}$;

— un véhicule B initialement à la position $50\ m$ avec la vitesse de $30\ m.s^{-1}$, contrôlé avec la règle suivante : B commence par décélérer à $-7\ m.s^{-2}$ pendant une seconde, puis accélère à $5\ m.s^{-2}$ pendant une seconde, puis décélère à nouveau à $-7\ m.s^{-2}$ jusqu'à ce qu'il s'arrête.

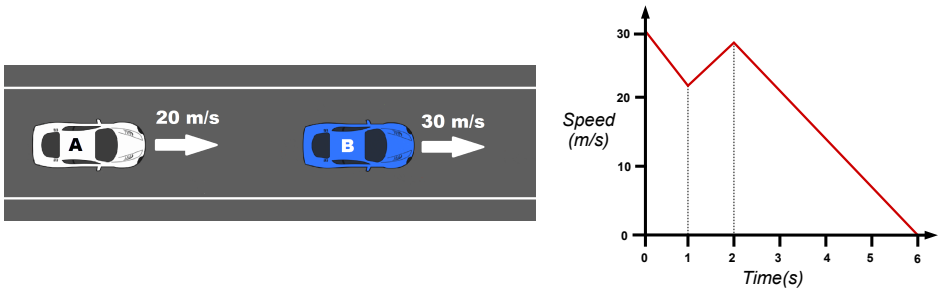


FIGURE 4.3: Situation initiale du scénario 2 (à gauche) avec l'évolution de la vitesse du véhicule B (à droite).

On peut remarquer que la situation initiale est sans danger pour le véhicule A (c'est-à-dire qu'il respecte une certaine distance de sécurité). Sur ce scénario, nous vérifions les valeurs et l'heure d'apparition des deux pires valeurs possibles du TTC et de la valeur d'accélération minimale (c'est-à-dire la décélération maximale) pour le véhicule A. À partir du même véhicule, nous vérifions également le moment où sa valeur d'accélération devient négative pour la première fois.

Critère	Min TTC (valeur) [en s]	TTC minimal (temps) [en s]	Accélération minimale (valeur) [en $m.s^{-2}$]	Accélération minimale (temps) [en s]	Première décélération (temps) [en s]
Simulation	$1.78\ s$	$6.0\ s$	$-7.36\ m.s^{-2}$	$6.0\ s$	$2.70\ s$
Model checking	$1.76\ s$	$6.0\ s$	$-9.40\ m.s^{-2}$	$6.0\ s$	$2.70\ s$

TABLEAU 4.2: Comparaison par rapport aux critères basés sur l'accélération et le TTC du véhicule A dans le scénario 2.

Les résultats sont présentés dans le tableau 4.2 et montrent que pour les deux outils, tous les événements se produisent dans les mêmes unités de temps. Aussi, pour le TTC, les valeurs sont très proches avec seulement $0.02\ s$ de différence

(1.13% d'erreur), ce qui signifie que nous pouvons utiliser le TTC comme un indicateur avec une confiance raisonnable. Par contre, la valeur de l'accélération minimale est assez différente avec plus de $2\ m.s^{-2}$ de différence donnant $27,71\%$ d'erreur. Ceci est dû à la propagation des erreurs de calcul de la valeur discrète lors de la prise de décision avec la loi de poursuite IDM. En raison du niveau d'abstraction différent, l'accélération calculée à un certain stade dans le *model-checker* est légèrement différent de celle obtenue en simulation.

Si l'accélération calculée par l'équation IDM avec le domaine discret du *model-checker* est supérieure à l'accélération réelle, cela implique que la vitesse à l'étape suivante sera plus grande que nécessaire et le temps de collision plus bas que prévu. En conséquence, la prochaine accélération calculée avec IDM sera inférieure à l'accélération réelle, ce qui compense cette différence. On peut donc observer des différences localement importantes sur l'accélération, même si les valeurs moyennes sont proches.

4.1.4 Application de la méthode

Dans cette section, nous allons mettre en place un exemple non déterministe dans lequel nous pouvons appliquer la méthodologie décrite dans la partie 4.1.1. Nous définissons un troisième scénario comme étant une variante non déterministe du scénario 2 fonctionnant de la manière suivante :

1. Le véhicule B commence avec une accélération de $-7\ m.s^{-2}$
2. A un moment donné (à un événement e_1), son accélération change et passe à $5\ m.s^{-2}$
3. A un autre instant (à un événement e_2), son accélération repasse à une accélération de $-7\ m.s^{-2}$

Nous définissons $\alpha = e_2 - e_1$ comme la durée pendant laquelle le véhicule B à une accélération positive. Ces valeurs sont illustrées dans la Figure 4.4. Nous précisons que si e_2 n'arrive jamais avant que le véhicule sorte de la portion de route observée, alors on considère la valeur de α comme étant infinie.

Tout d'abord, nous lançons plusieurs exécutions aléatoires à partir de l'outil de simulation afin d'obtenir à la fois la valeur de TTC minimale et les valeurs d'accélération. A partir de ces données, nous faisons les trois hypothèses suivantes :

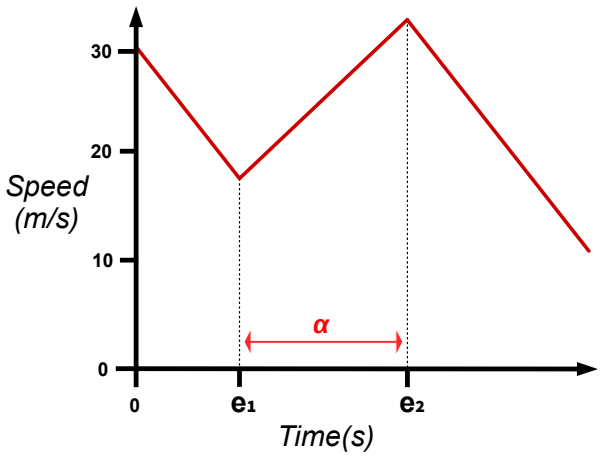


FIGURE 4.4: Valeurs de e_1, e_2 et α sur un comportement possible du véhicule B.

1. Il existe une durée γ telle que si $e_1 \leq \gamma$, alors le TTC ne sera jamais inférieur à 1.7 s
2. Dans le but d'avoir un TTC inférieur à 1.7 s, α doit être compris entre $e_1 * 0.5$ et e_1
3. Il n'y a pas d'exécution possible où le véhicule B s'arrête et sa valeur d'accélération n'est jamais plus petit que deux fois la décélération désirée $-b$ (fixée à $-6\ m.s^{-2}$)

Cela nous conduit aux requêtes de logique temporelle suivantes (en logique CTL, $\Diamond$ exprime "finalement" et $\Box$ exprime "globalement") :

1. $\neg\exists \Diamond (TTC < 1.7 \wedge e_1 \leq \gamma)$
2. $\forall \Box (TTC < 1.7 \Rightarrow (\alpha < e_1 \wedge \alpha > e_1/2))$
3. $\neg\exists \Diamond(a_A \geq -6 \wedge on_the_road(A))$, où $on_the_road(A)$ est une variable booléenne, valant $Vrai$ si le véhicule A n'a pas encore atteint la fin de la portion de route observée.

La première hypothèse se confirme aisément, mais elle est plus intéressante pour trouver un maximum pour γ. En utilisant le *model-checker*, nous trouvons par dichotomie une valeur de $2.3\ s$, mais il faut garder à l'esprit qu'il y a peut être une légère erreur dans la valeur du TTC calculée. Ensuite, nous explorons le voisinage de cette valeur par simulation afin de l'affiner. La simulation indique une exécution avec $e_1 = 2.2s$, où un TTC inférieur à $1.7\ s$ est trouvé, mais aucun à $e_1 = 2.1\ s$. On peut donc supposer que l'actuel maximum pour γ est 2.1.

La seconde hypothèse semble être erronée. Actuellement, le *model-checker* trouve une exécution violant une telle propriété pour les deux bornes supérieures (avec $e_1 = 2.4$ et $e_1 = 4.9$) et inférieures (avec $e_1 = 3.3$ et $e_1 = 4.4$). Nous affinons ensuite les bornes inférieures et supérieures afin d'obtenir un résultat, où $\alpha \in [e_1 * 0.3, e_1 * 1.1]$. L'exploration de ces limites par simulation ne montre aucun contre-exemple.

Enfin, la troisième requête donne un résultat positif, mais nous devons nous rappeler que, comme nous l'avons montré précédemment, la valeur de l'accélération peut être très différente entre l'outil de vérification et de simulation. La vérification de la même requête avec une valeur de $-7\ m.s^{-2}$ au lieu de $-6\ m.s^{-2}$ donne toujours un résultat positif ajoutant une certaine confiance au résultat.

Le processus de vérification a pris moins de 10 secondes pour chaque requête malgré la complexité du système en raison de la taille des plages de variables et du grand nombre de positions dans lesquelles chacun des agents peut être. Cependant, le système n'est pas encore si complexe en termes de non déterminisme, car nous avons limité le nombre d'actions possibles que le véhicule *leader* peut effectuer à un moment donné. Ce choix a été fait pour les besoins de l'étude de cas, car il était plus facile de vérifier la cohérence de notre résultat dans un tel cas. Pour cette raison, on pourrait soutenir qu'il aurait été possible de vérifier tous les comportements possibles avec la simulation seule dans un délai relativement raisonnable, ce qui, dans ce cas particulier, pourrait effectivement fonctionner. Cependant, dans le cas de comportements plus complexes, cela ne semble pas être une méthode raisonnable. En outre, bien que le temps de vérification augmente également avec la taille et la complexité du système, il est toujours possible, dans une certaine mesure, de vérifier de manière exhaustive, dans un délai raisonnable, des hypothèses sur des systèmes non déterministes complexes (moins de quelques heures). Nous remarquons que la performance dépend principalement du niveau de non déterminisme. Par exemple, l'ajout d'autres véhicules déterministes sur un scénario aura un coût très faible, tandis que l'ajout de comportements plus possibles peut avoir de graves conséquences sur les performances.

4.1.5 Synthèse

A partir d'un modèle simulé de véhicule donnant une représentation satisfaisante de la réalité, nous avons montré qu'il était possible de modéliser avec une bonne précision le comportement des véhicules sur l'outil de vérification formelle, dont la description nécessitait la discrétisation d'un grand nombre de variables. Ensuite, nous avons expérimenté et comparé les deux modèles à partir d'indicateurs appropriés, et ainsi montré que les valeurs obtenues coïncidaient. Motivés par la complémentarité de ces techniques, nous avons proposé une méthode les combinant afin d'accroître la confiance dans les résultats. Finalement, grâce à cette modélisation et à une telle méthodologie, nous avons fourni un cas d'étude montrant qu'il était possible d'obtenir efficacement des informations utiles sur le comportement des véhicules modélisés. En particulier, l'analyse sur la loi de poursuite IDM a mis en évidence certains comportements non triviaux [11].

A l'état actuel, il est donc possible de concevoir un modèle de décision du véhicule, de le tester par simulation, et de l'enrichir en garantissant un ensemble de propriétés par vérification formelle. A partir de ce modèle, l'objectif est à présent d'assurer une compatibilité avec un niveau d'abstraction plus bas, pour l'instancier dans un véhicule réel.

4.2 Interfaçage du modèle avec des modèles plus réalistes

4.2.1 Motivations

Le modèle de décision actuel se base sur deux aspects principaux : l'aspect longitudinal via la loi de poursuite IDM et l'aspect latéral via le modèle de changement de voie MOBIL. Bien que le modèle IDM puisse s'apparenter à un système ACC (*Adaptive Cruise Control*) simplifié, il est plutôt simple de l'imaginer sur un véhicule réel (modèle continu grâce à une équation différentielle). Toutefois, c'est moins le cas pour le modèle MOBIL. En effet, contrairement à IDM, c'est un modèle discret où le changement de voie se fait dès que le modèle a décidé de le faire (passage

instantané d'une voie vers une autre).

Au niveau d'abstraction le plus bas, les modèles très réalistes définissent généralement le changement de voie soit avec une loi de contrôle de trajectoire définie comme une succession de points de cheminement que le véhicule doit suivre [40], soit en tant que modèle mathématique de référence variant selon la vitesse du véhicule et qui est chargé de consigne à un contrôleur embarqué dans un véhicule [59]. Actuellement, plusieurs solutions sont proposées pour déterminer des trajectoires d'un véhicule intelligent selon les situations [98], notamment des véhicules communicants dans un parking automatisé [53] mais aussi l'étude du contrôle latéral proposant ainsi une trajectoire réaliste lors d'un changement de voie [71] [52].

Finalement, avec un faible niveau d'abstraction, on se rapproche de la physique réelle du véhicule afin de concevoir les trajectoires les plus précises tout en prenant en compte des altérations possibles de l'environnement (météo, dégradation de la route...).

Un niveau d'abstraction plus élevé permet de nous concentrer sur le moment d'intention du véhicule pour des manœuvres particulières (changer de voie, tourner à gauche...), de pouvoir observer plus d'éléments dans un temps plus restreint et de vérifier des propriétés, liées par exemple à la sécurité du système, plus efficacement, grâce à une réduction de l'espace des états.

La question est alors la suivante : si les propriétés données sont établies en utilisant des modèles avec un haut niveau d'abstraction, ces propriétés sont-elles toujours valables lorsque nous revenons à des niveaux d'abstraction plus faibles ? Autrement dit, qu'avons-nous perdu en gravissant les niveaux d'abstraction ? L'objectif est tout d'abord de définir différentes représentations du modèle latéral. Ces différents modèles seront analysés et comparés via deux études de cas afin de répondre aux questions ci-dessus.

4.2.2 Différentes représentations de l'aspect latéral

Il est possible de modéliser un changement de voie de plusieurs façons. Nous avons choisi d'en analyser trois, à différents niveaux d'abstraction. Nous déterminerons les avantages et les inconvénients respectifs de chacun de ces modèles pour déterminer dans quel contexte ils peuvent apporter une contribution intéres-

sante. Chaque véhicule effectuant le changement de voie applique la loi de poursuite IDM avant et après celui-ci. Pendant le changement, le véhicule applique une des trois trajectoires présentées ci-dessous. Afin de comparer les différentes trajectoires entre elles, on applique manuellement le changement de voie du véhicule.

4.2.2.1 Trajectoire réaliste

Au niveau le plus fin, ce type de trajectoire consiste en une succession de points de passage du véhicule. Nous avons en entrée une matrice qui représente toutes les positions longitudinales / latérales et la vitesse à chaque pas de temps. C'est une trajectoire très réaliste mais il est difficile de la définir dynamiquement pour chaque cas lors d'un changement de voie. Elle est conçue de façon à respecter les limites de confort, de sécurité et respectant la légalité en choisissant notamment une accélération latérale maximale inférieure à $1.2\ m.s^{-2}$ et une vitesse longitudinale inférieure à la vitesse limite autorisée. Par ailleurs, la trajectoire est conçue de manière suffisamment lisse, en limitant le jerk (dérivée de l'accélération) longitudinal à $3\ m.s^{-3}$ pour limiter les phénomènes de secousses dans l'axe longitudinal du véhicule. L'exemple donné sur la Figure 4.5 montre un exemple de trajectoire réaliste avec un pas de temps $0.1\ s$ et une vitesse variant entre 70 et $90\ km.h^{-1}$. L'accélération longitudinale associée, en entrée du modèle de dynamique du véhicule, a été testée sur un modèle couplant les mouvements longitudinal et latéral, celui-ci permettant de tenir compte de la variation de la vitesse longitudinale lors d'un changement de voie comparable à une manœuvre dans un virage. Ce test permet de vérifier les limites de confort et les variables dynamiques comme la vitesse longitudinale et la position longitudinale pour vérifier le réalisme de la trajectoire donnée. Les ensembles de trajectoires peuvent être conçus pour diverses vitesses initiales et finales et pour une accélération latérale et une position le long d'un horizon temporel donné, mais il est difficile de la définir dynamiquement pour chaque cas lors d'un changement de voie.

4.2.2.2 Trajectoire simplifiée

Lorsqu'un véhicule souhaite changer de voie, on peut imaginer qu'il choisisse une position dans la voie cible. On définit un point cible qui devient l'objectif à atteindre pour le véhicule. Lors d'un changement de voie, si ce point est défini sur la

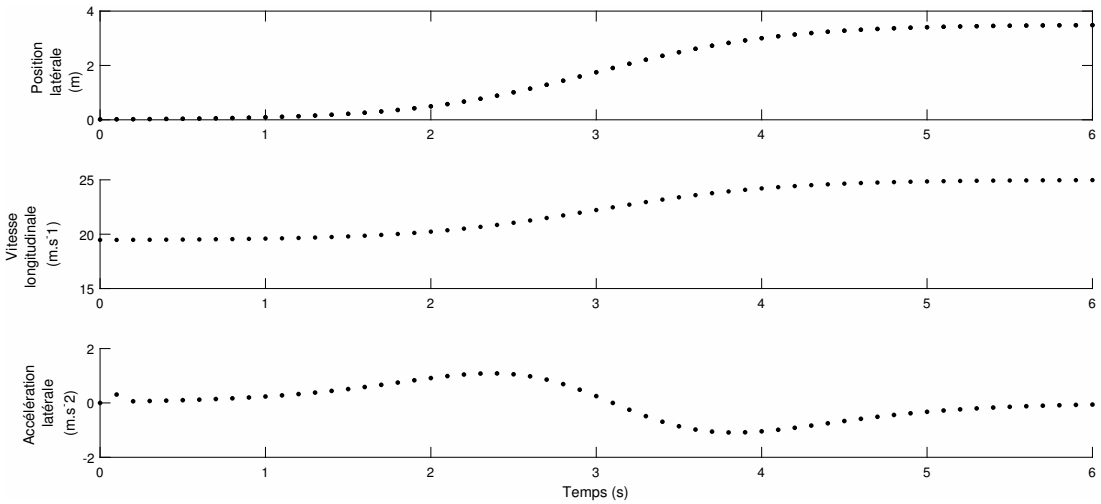


FIGURE 4.5: Trajectoire de changement de voie réaliste

voie cible, le véhicule effectuera une trajectoire en ligne droite jusqu'à atteindre le point cible. Le véhicule continuera ensuite à avancer sur la nouvelle voie (voir la Figure 4.6). L'avantage de ce modèle est qu'il est facile à implémenter et que l'on peut définir dynamiquement une trajectoire pour chaque cas. L'inconvénient est qu'il est moins réaliste que le modèle précédent. De plus, un véhicule réel ne pourrait pas suivre une telle trajectoire à cause de la discontinuité de la dérivée de la position (c'est-à-dire, de la vitesse).

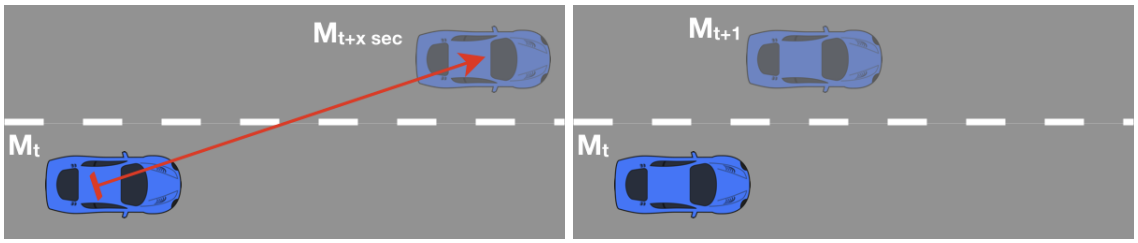


FIGURE 4.6: A gauche, la trajectoire simplifiée du changement de voie. A droite, la version instantanée de la trajectoire.

4.2.2.3 Trajectoire instantanée

Dans beaucoup de modèles de décision de changement de voie comme MOBIL, une hypothèse est faite sur la trajectoire du véhicule pendant la manœuvre. En effet, si l'on suppose qu'au temps t le véhicule est dans sa voie réelle, à l'instant $t+1$, le véhicule est sur la voie cible (voir la Figure 4.6). Il s'agit d'un changement de voie instantané dans lequel nous ne considérons donc plus une trajectoire continue

comme dans les deux premiers modèles. C'est une trajectoire très facile à mettre en œuvre mais, bien sûr, ce n'est pas un modèle de changement de voie réaliste. Cependant, il peut être très utile de définir un modèle comme celui-ci afin de réduire l'espace des états si l'on veut en vérifier les propriétés en utilisant le model-checking ce qui permet de limiter l'explosion combinatoire lors de l'exploration exhaustive de l'espace des états (aspect détaillé en Section 4.1).

4.2.3 Comparaison entre une trajectoire simplifiée et réaliste pour des véhicules autonomes non communicants

Nous allons maintenant développer deux études de cas, permettant de mettre en évidence l'apport d'un modèle de changement de voie avec un niveau d'abstraction plus élevé, par rapport à une trajectoire réaliste. Nous définissons les hypothèses suivantes qui s'appliquent aux deux études de cas :

— pendant la simulation, nous faisons des observations sur un pas de temps de $0.1s$;

— tous les véhicules utilisent le modèle *IDM* sauf lors du changement de voie ;

— nous n'observerons que des situations de conduite où la valeur de l'accélération longitudinale n'est jamais inférieure à la valeur limite de $-g\ m.s^{-2}$ (environ $-10\ m.s^{-2}$) ;

— nous définirons des trajectoires différentes avec une durée de changement de voie variable et des variations de vitesse entre le début et la fin de ce changement de voie.

Comme nous l'avons mentionné précédemment, il est utile d'avoir une trajectoire simplifiée du fait de sa simplicité de mise en œuvre et de la possibilité de la définir dynamiquement. Cependant, bien que les conditions initiales et finales soient réalistes, ce n'est pas le cas pour la trajectoire qui n'est pas réalisable par un véhicule réel. Ainsi, le problème suivant se pose : quelles sont les différences en termes de dynamique du véhicule entre une trajectoire simplifiée et une trajectoire réaliste, et ces différences sont-elles problématiques ?

Pour déterminer si c'est le cas, nous avons décidé de créer un scénario de voie d'accélération (il est tout à fait possible d'imaginer d'autres scénarios comme une route à plusieurs voies ou un rétrécissement de voie par exemple). Comme on peut

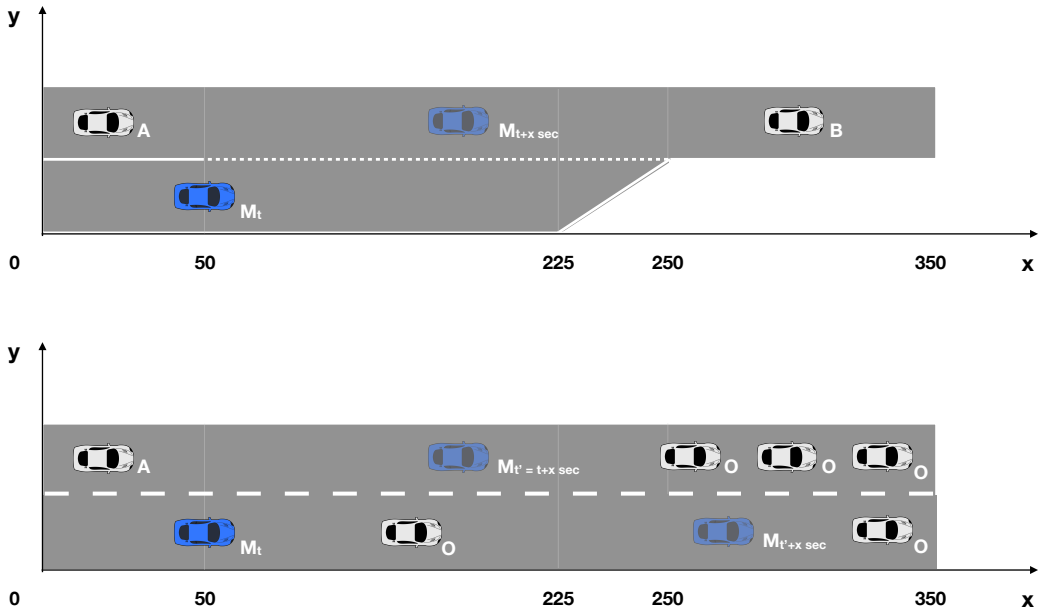


FIGURE 4.7: Scenarii des deux études de cas. En haut, la première étude de cas de l'insertion d'un véhicule à partir d'une voie d'accélération. En bas, la seconde étude de cas représentant le dépassement d'un véhicule.

le voir sur la Figure 4.7, nous posons un véhicule *M* qui va effectuer un changement de voie vers la voie principale. Afin d'augmenter le panel de situations, nous avons positionné un véhicule statique *B* qui représente le début d'un embouteillage, forçant le véhicule *M* à décélérer. De plus, nous ajoutons un véhicule *A* situé sur la voie principale derrière le véhicule *M*, afin de voir l'influence d'une telle trajectoire de changement de voie sur les autres véhicules. Enfin, nous admettons que le véhicule *A* voit le véhicule *M* dès qu'il atteint la limite de la rampe d'entrée. L'hypothèse posée ici est purement indicative et n'aura pas de grande influence sur les résultats obtenus.

Afin d'avoir des résultats représentatifs dans ce type de scénario, nous allons faire varier la position et la vitesse de *A* ainsi que la position de *B*. Pour estimer l'erreur possible, nous comparerons les deux modèles en fonction des critères suivants :

— la différence de position d'un même véhicule entre les deux modèles de trajectoire. On observe les véhicules *A* et *M* ;

— la différence de vitesse d'un même véhicule entre les deux modèles de trajectoire. On observe les véhicules *A* et *M* ;

— les différences de perception (quand *A* voit le véhicule *M* sur la voie principale pour la première fois) permettant d'observer le potentiel retard (ou avance) de la trajectoire simplifiée sur la trajectoire réaliste lors du changement de voie.

	Perception (en s)			Position de M (en m)			Vitesse de M (en m.s^{-1})		
	70 km.h^{-1} à 90 km.h^{-1} en 6 s	70 km.h^{-1} à 90 km.h^{-1} en 3 s	70 km.h^{-1} à 70 km.h^{-1} en 3 s	70 km.h^{-1} à 90 km.h^{-1} en 6 s	70 km.h^{-1} à 90 km.h^{-1} en 3 s	70 km.h^{-1} à 70 km.h^{-1} en 3 s	70 km.h^{-1} à 90 km.h^{-1} en 6 s	70 km.h^{-1} à 90 km.h^{-1} en 3 s	70 km.h^{-1} à 70 km.h^{-1} en 3 s
Trajectoire réaliste	3,00	1,50	1,50	120,46	90,43	89,17	21,97	21,81	19,44
Trajectoire simplifiée	3,40	1,80	1,70	129,14	95,62	91,95	22,47	22,52	19,44
Erreur	***0,40***	***0,30***	***0,20***	***8,68***	***5,19***	***2,78***	***0,50***	***0,72***	***0,00***

TABLEAU 4.3: Différence de valeur (erreur) pour chaque critère entre les deux modèles sur trois exemples de trajectoires.

On remarque dans la Table 4.3, au niveau de la trajectoire simplifiée, que le véhicule *M* est vu par *A* un peu plus tard que dans la trajectoire réaliste (entre deux et quatre pas de temps), ce qui implique que la voiture se trouve plus loin avec une vitesse plus élevée. Nous pouvons également voir que lorsque nous décidons de définir des trajectoires avec un temps de changement de voie plus court, l'erreur pour les positions est plus faible. De plus, nous avons la valeur la plus faible lorsque nous définissons une trajectoire sans accélération lors du changement de voie. Donc, on peut en déduire que la perception de la voiture avec une trajectoire simplifiée est correcte par rapport à une trajectoire réaliste.

Les Figures 4.8 et 4.9 présentent la situation où nous avons l'erreur la plus élevée sur les positions et les vitesses pour les véhicules *A* et *M*. Ainsi, on obtient une erreur sur la position d'environ 2.0 mètres pour *M* (environ 1.1 mètre pour *A*) et environ $1.1\ m.s^{-1}$ pour la vitesse de *M* ($1.9\ m.s^{-1}$ pour *A*). Ce pic est atteint lorsque le véhicule *M* atteint la nouvelle voie. Ensuite, cette erreur se réduit grâce au comportement d'IDM dont l'objectif est d'atteindre une vitesse maximale avec une inter-distance suffisante. Là encore, la différence de position et de vitesse de toutes les voitures semble satisfaisante (l'erreur la plus élevée est inférieure à la longueur moyenne d'un véhicule).

Nous pouvons conclure que l'étude d'une trajectoire simplifiée n'aura pas d'influence sur le comportement global du véhicule lors d'un changement de voie, bien que l'on observe des faibles erreurs sur la dynamique du véhicule. Toutefois, l'erreur

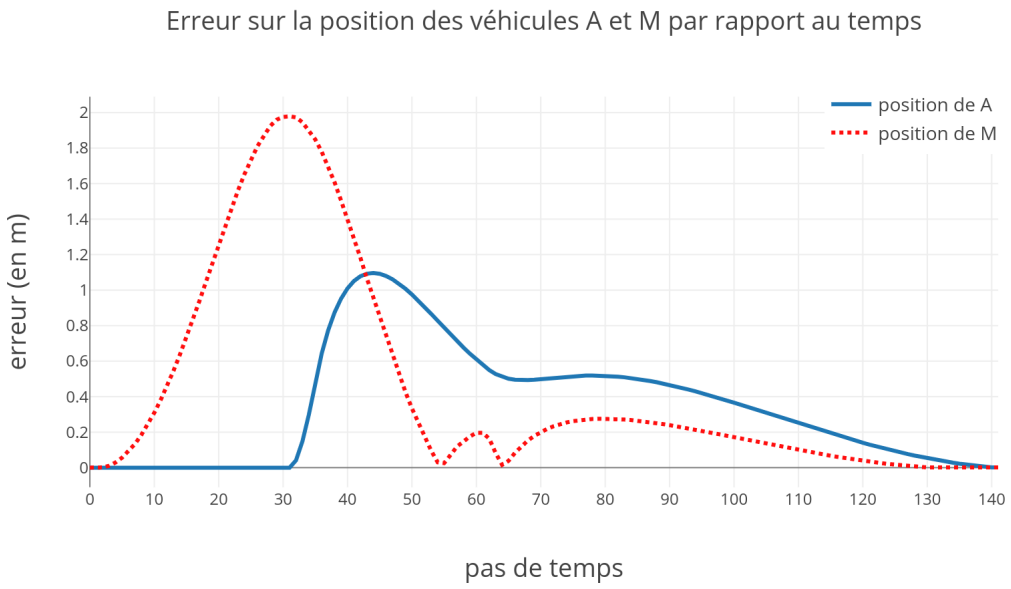


FIGURE 4.8: Erreur des positions des véhicules *A* et *M*

constatée peut s'amplifier sur les véhicules situés à l'arrière. On peut donc supposer que l'on n'utilise la trajectoire simplifiée que dans des cas locaux (influençant la dynamique des véhicules environnants uniquement).

4.2.4 Intérêt d'une trajectoire instantanée de changement de voie dans le cas de véhicules autonomes communicants

Nous avons observé qu'il était assez facile de faire des comparaisons entre une trajectoire réaliste et une trajectoire simplifiée. Cependant, il existe des différences plus importantes dans la perception des véhicules. On pourrait donc proposer que ces véhicules communiquent entre eux afin de s'informer mutuellement lorsqu'un véhicule a l'intention de changer de voie, permettant ainsi d'anticiper un danger potentiel. On peut supposer que le modèle de trajectoire de changement de voie discret peut prendre ce rôle. Pour cela, on peut imaginer que lorsqu'un véhicule a l'intention de changer de voie, il envoie une représentation virtuelle d'un véhicule correspondant au modèle discret du changement de voie. La question suivante se pose alors : quel serait l'impact lié à l'introduction d'un tel véhicule virtuel, et l'abstraction ainsi introduite permet-elle d'étudier de nouvelles propriétés sur le système ?

Pour déterminer l'impact du véhicule virtuel, nous avons décidé de créer un scé-

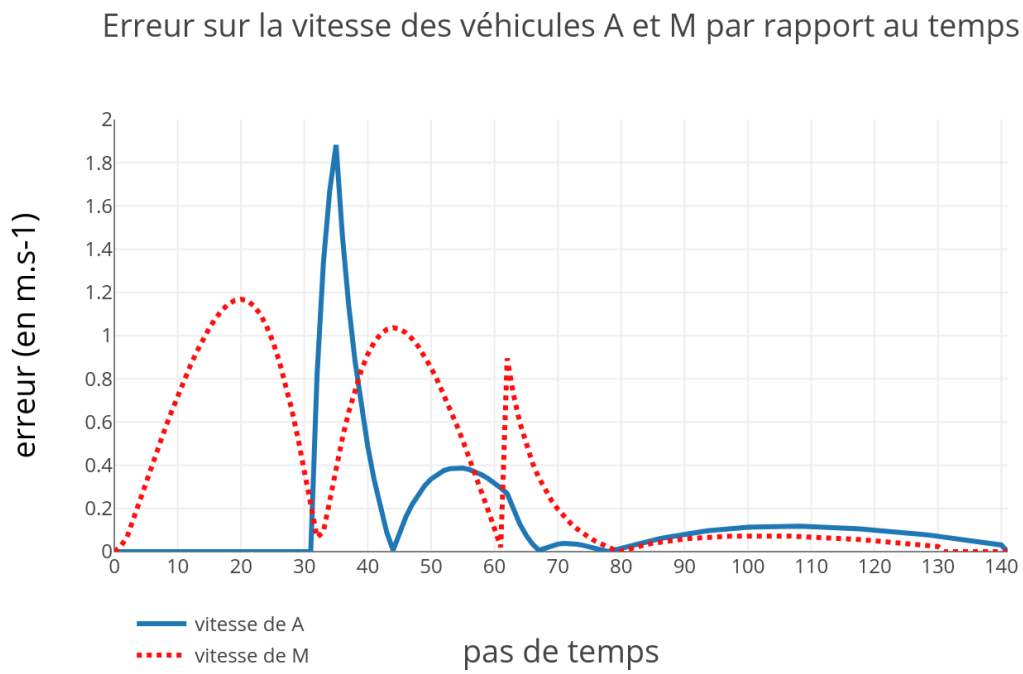


FIGURE 4.9: Erreur des vitesses des véhicules *A* et *M*

nario de route à deux voies. Comme on peut le voir sur la Figure 4.7, nous plaçons un véhicule *M* qui fait un changement de voie vers la voie de gauche afin d'éviter un obstacle *O*. Ensuite, le véhicule change à nouveau de voie en direction de la voie de droite afin d'éviter un embouteillage. Comme dans le premier scénario, nous ajoutons un véhicule *A* situé sur la voie de gauche derrière le véhicule *M*. Enfin, pour la trajectoire réaliste, on admet que le véhicule *A* voit *M* dès qu'il atteint la limite entre les deux voies. Une fois de plus, l'hypothèse posée ici est purement indicative et n'aura pas de grande influence sur les résultats obtenus.

Comme dans le premier scénario, afin d'avoir des résultats représentatifs dans ce type de scénario, nous faisons varier la position et la vitesse de *A*. Afin de définir les propriétés, nous analyserons les critères suivants :

— la position du véhicule *A* et *M* ;

— la valeur longitudinale du *time-to-collision* (TTC) [93].

Grâce à l'abstraction introduite par le changement de voie instantané, on a pu utiliser des techniques de *model-checking* qui ont permis de déterminer la validité de certaines propriétés. Pour cela, on vérifie que cette propriété est vérifiée quelles que soient les conditions initiales des véhicules environnants. Ainsi, nous pouvons

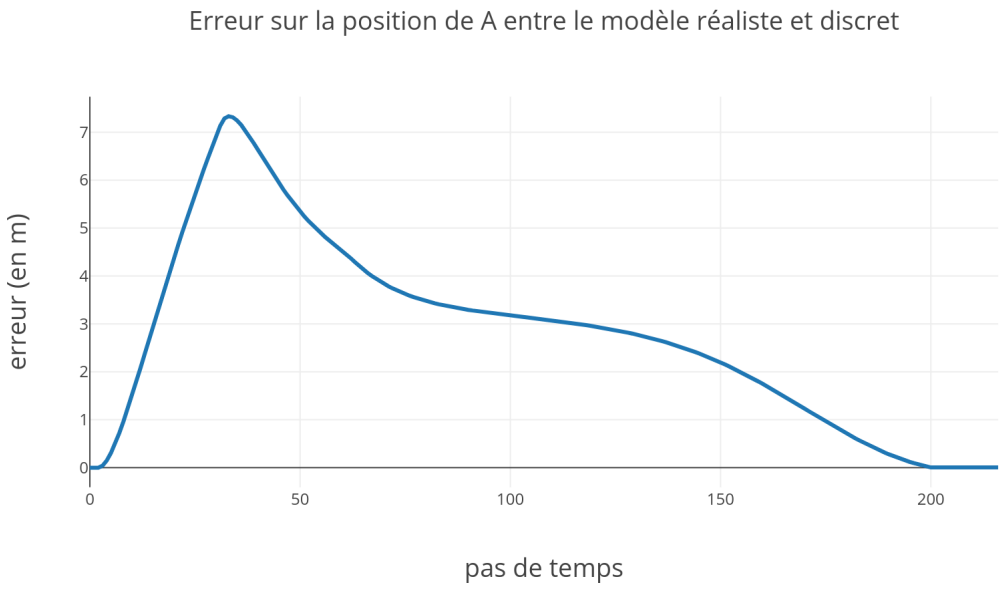


FIGURE 4.10: Différence de position entre les véhicules *A* des deux modèles ($realistic.x_A - discrete.x_A$) en fonction du temps dans une des situations possibles.

définir, pour toutes les situations, les propriétés suivantes :

— la position du véhicule *A* sur le modèle discret est toujours inférieure ou égale à la position de *A* sur le modèle réaliste ($discrete.x_A \leq realistic.x_A$) comme on peut le voir sur un exemple de situation en Figure 4.10 ;

— la valeur TTC minimum de *M* du modèle discret est toujours supérieure à la valeur TTC minimum de *M* du modèle réaliste ($min(discrete.TTC_M) \geq min(realistic.TTC_M)$).

On peut aussi dire qu'à chaque pas de temps où *M* change de voie, la valeur du TTC de *M* du modèle discret est toujours supérieure à la valeur du TTC de *M* du modèle réaliste ($discrete.TTC_M \geq realistic.TTC_M$).

Finalement, on peut observer que le modèle discret à tendance à positionner le véhicule virtuel derrière le véhicule utilisant le modèle réaliste. Ainsi, lorsque le véhicule suiveur reçoit le véhicule virtuel, l'inter-distance calculée est réduite impliquant une valeur de l'accélération calculée par IDM plus faible. Il a donc tendance à augmenter la valeur du TTC dans certains cas, ce qui permet au véhicule suiveur d'anticiper un peu plus. On peut donc dire que, pour ce type de scénario, le véhicule est plus sécurisé en augmentant l'inter-distance entre les deux véhicules.

4.2.5 Synthèse

Bien que les modèles réalistes soient utiles parce qu'ils sont directement transposables dans les systèmes automobiles réels, ils induisent un niveau de complexité plus élevé des simulateurs qui peut ne pas être nécessaire pour certaines propriétés. En effet, l'objectif n'est pas de se rapprocher de la réalité mais plutôt de l'approximer afin de pouvoir observer les mêmes phénomènes (embouteillages, collision proche...), sans pour autant avoir une exactitude en termes de dynamique du véhicule. Ainsi, nous avons montré que les modèles simplifiés ne modifient pas beaucoup la dynamique des véhicules et qu'ils ne modifient pas la dynamique du système dans son ensemble. De plus, les changements de voie instantanés peuvent être considérés comme des signaux d'avertissement envoyés aux autres véhicules, ce qui leur permet de mieux anticiper et ainsi de rendre le véhicule plus confortable pour ses passagers et de fluidifier le trafic.

A partir de cette étude, nous avons proposé une première méthodologie liée à la trajectoire du véhicule, représentant l'action finale de celui-ci [80]. Elle fait suite au modèle de décision du véhicule basé sur une observation de l'environnement et de potentielles interactions. Elle permet de refermer la boucle nécessaire à la conception complète du véhicule intelligent dont nous allons proposer à présent une méthodologie globale.

4.3 Synthèse : proposition d'une méthodologie

Après avoir parcouru les différentes étapes de la méthodologie proposée initialement dans la partie introductive du manuscrit, l'objectif est d'en proposer une version affinée. Cette version est disponible sur la Figure 4.11 et fonctionne de la manière suivante :

1. Concevoir un modèle de véhicule en déterminant tous les éléments nécessaires à son fonctionnement (caractéristiques, actions possible...).
2. Choisir un outil de simulation répondant à un même niveau d'abstraction pour assurer la compatibilité entre les deux éléments.

3. Simuler le modèle et s'assurer que le modèle réponde bien aux exigences de départ. Si ce n'est pas le cas, affiner le modèle. Dans le temps, définir un ensemble d'hypothèses permettant de déterminer un ensemble de propriétés.
4. Vérifier formellement les propriétés déterminées à partir des hypothèses définies précédemment. Confirmer la validité de ces propriétés par simulation. Si le modèle contient des faux positifs, l'affiner pour les retirer.
5. A partir du modèle obtenu, déterminer s'il est possible de concevoir une trajectoire réaliste correspondant à l'action souhaitée, permettant d'assurer une compatibilité avec un simulateur plus réaliste. Si ce n'est pas le cas, revenir au modèle pour faire les changements nécessaires.

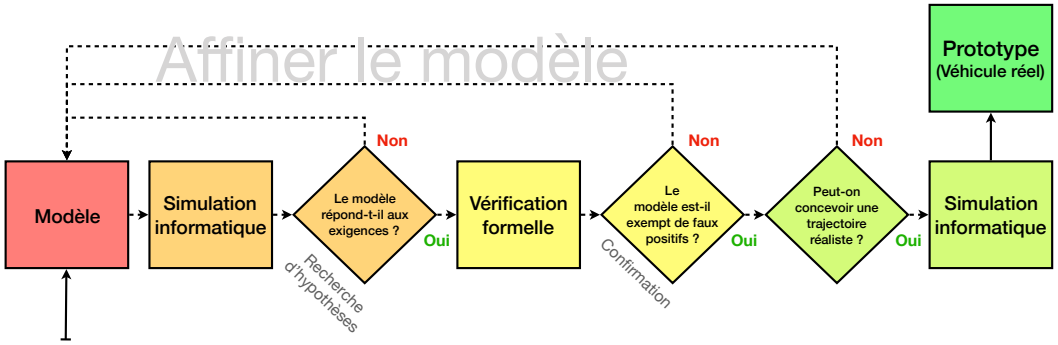


FIGURE 4.11: Méthodologie de conception affinée d'un modèle de véhicule.

L'ensemble de ces étapes permet d'arriver à un modèle pouvant être implanté dans un véhicule réel. Afin d'enrichir cette méthodologie et de faciliter le passage de chacune des étapes, nous avons proposé une méthodologie pour la conception du modèle initial d'un véhicule coopératif détaillé en chapitre 3. Elle est inspiré des principes d'architecture objet [17] et d'architecture de subsomption de Brooks [20]. Elle comprend les parties suivantes (voir la Figure 4.12) :

— une partie perception où l'on insère dans une boîte noire l'ensemble des éléments observables et où l'on détermine les véhicules qui vont communiquer avec le véhicule étudié. Dans le cadre de notre protocole de coopération, on a déterminé les couples de véhicules pouvant coopérer entre la voie d'insertion et la voie principale dans le but de faciliter l'insertion ;

— une partie décision appliquant des modèles de décision indépendants par ordre de priorité. Ainsi sur la Figure 4.12, le bloc situé le plus en haut est

prioritaire. Dans cette conception, deux modèles initiaux sont proposés répondant à une autonomie longitudinale et latérale du véhicule. Dans le cadre de la conception du protocole de coopération, nous avons utilisé la loi de poursuite IDM et le modèle de changement de voie MOBIL. Si ce dernier n'autorise pas le changement de voie, alors on applique le protocole de coopération qui donne une décision et y associe une action ;

— dans la partie action, on récupère l'accélération du bloc de décision global. Cette valeur est envoyée au modèle de véhicule générique, qui met à jour sa position et sa vitesse à partir de cette valeur.

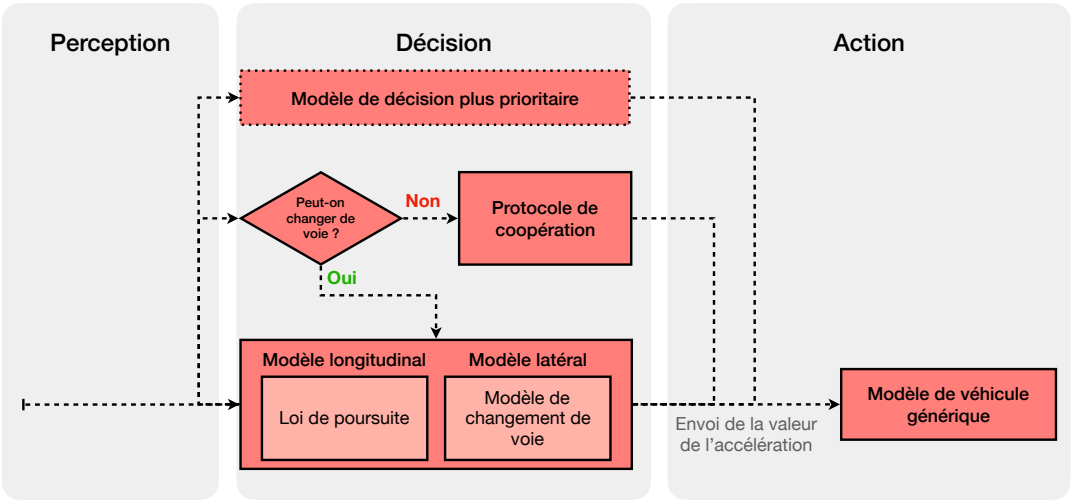


FIGURE 4.12: Proposition d'une méthodologie pour la conception du modèle de véhicule.

Grâce à cette méthodologie, il devient plus facile et intuitif de concevoir des Systèmes de Transport Intelligents Coopératifs tout en conservant un haut niveau de sécurité et de réalisme.

Conclusion

Aujourd'hui très médiatisé, le véhicule autonome est au cœur de toutes les attentions et représente un énorme défi pour les entreprises en terme d'innovation technologique. Imaginer un véhicule capable de conduire sans intervention humaine n'est plus utopique aujourd'hui. Toutefois, nous avons observé qu'un tel véhicule souffrait encore de quelques limitations matérielles d'une part, algorithmiques d'autre part. En effet, de nombreuses questions se posent encore sur l'ajout de ce type de véhicule dans les routes actuelles.

Pour imaginer de nouvelles solutions, nous avons conçu un prototype d'environnement véhiculaire dans le simulateur agent GAMA, permettant la conception rapide de simulations de véhicules sur différents types de route. De plus, il est possible de définir et de modifier facilement son propre modèle de décision, ce qui à l'avantage de pouvoir créer de nombreuses situations en mélangeant des véhicules aux comportements différents.

Cet outil, nous l'avons utilisé pour mettre en place un premier protocole de coopération dans des cas de changement de voie obligatoire comme la voie d'insertion. A partir de modèles longitudinaux et latéraux existants, nous avons réussi à intégrer un module de communication permettant de répondre au critère de courtoisie sans négliger les critères de sécurité, d'efficacité et de confort. Ce protocole peut fonctionner dans des cas centralisés comme décentralisés, on peut donc l'imaginer au sein de la route (via une infrastructure connectée) comme au sein d'un véhicule (associée au modèle de décision initial). De plus, nous avons montré par simulation que ce protocole s'adapte très bien dans un environnement mixte mais aussi dans un environnement constitué de véhicules non communicants, qu'ils soient autonomes ou conduits par des êtres humains, même à faible proportion.

A partir du modèle obtenu via simulation, nous avons proposé une méthode vi-

sant à déterminer des propriétés de sécurité par vérification formelle à partir d'une étude de cas sur la loi de poursuite IDM. Ainsi, à partir de ce modèle, nous avons montré qu'il est possible d'interfacer ce modèle avec un modèle plus réaliste notamment sur l'aspect latéral, en étudiant différentes trajectoires.

De manière plus générale, ce travail a été effectué dans le but de proposer une méthodologie pour concevoir des véhicules intelligents. Ainsi, il est tout à fait possible avec l'ensemble des outils et méthodes proposés, de partir d'un modèle de décision initial couplé à un modèle générique d'un véhicule autonome. Celui-ci est simulé pour rechercher un ensemble d'hypothèses que l'on vérifie formellement. Après confirmation, on associe la décision du modèle à une action, représentée par une trajectoire réaliste. On atteint donc l'ensemble du processus de conception d'un prototype de véhicule intelligent.

Perspectives et discussion

En plus des contributions effectuées, le travail de thèse ouvre de nombreuses perspectives en raison des possibilités d'ouverture qu'offrent chacune d'entre-elles.

Dans le chapitre 2, l'outil mis en place sur le simulateur agent *GAMA* ne fait l'objet que d'un prototype pour le moment. L'objectif premier sera donc de l'intégrer définitivement sous *GAMA* sous forme d'une extension. Comme une extension de trafic routier est déjà présente, l'idée est de combiner les deux extensions afin d'imaginer une extension de trafic plus complète, plus basée sur l'analyse comportementale des véhicules. L'environnement routier peut être également plus étoffé en ajoutant des nouveaux types de route comme le carrefour, le carrefour giratoire (rond-point) ou encore le passage à niveau d'un train. Quant aux comportements des véhicules, on pourra ajouter d'autres modèles de loi de poursuite et de changement de voie existants ce qui permettra d'obtenir à terme une extension de trafic plus complète couplée à une conception basé sur les agents, afin de reproduire un large éventail de situations actuelles et futures.

Dans le chapitre 3, le protocole de communication obtenu montre un réel intérêt de faire communiquer les véhicules à la manière des conducteurs humains qui se soumettent à un accord commun. Dans un premier temps, l'objectif est d'ajouter de nouvelles voie sur la route principale afin de permettre aux véhicules de se déporter sur la gauche, et ainsi, faciliter l'insertion des véhicules. L'algorithme à concevoir va se combiner avec celui déjà conçu (à la manière de notre protocole de communication avec *MOBIL*). Bien que le protocole de communication ait été conçu pour la voie d'insertion sur autoroute, il est tout à fait possible de l'imaginer dans un éventail de situations où le changement de voie est obligatoire. Parmi les situations proches, on retrouve le rétrécissement de voie ou encore un véhicule à l'arrêt sur la route qu'il faut contourner. Toutefois, d'autres situations peuvent être envisagées comme

le carrefour giratoire ou l'insertion d'un véhicule à partir d'une voie rapide (céder le passage). Seul l'algorithme de perception est à modifier en raison de la forme différente de ces environnements. Concernant le modèle de décision, l'idée est de partir d'autres solutions existantes comme la théorie des jeux ou la négociation par arguments pour proposer des communications plus riches. Enfin, il est important de considérer le comportement des conducteurs humains afin de proposer une simulation de l'environnement actuel plus réaliste. Pour cela, deux pistes sont envisagées. La première consiste à reproduire au plus proche de la réalité le comportement d'un véhicule dans un environnement donné à partir d'un ensemble de données en entrée de véhicules. La seconde consiste à ajouter un modèle d'émotions pour chaque agent évoluant au cours du temps. Par exemple, la décision d'un véhicule autonome peut mettre en colère un être humain qui va avoir un comportement plus agressif, voire contre-coopératif.

Le chapitre 4 a montré la possibilité de garantir certaines propriétés par vérification formelle sans avoir une explosion de l'espace des états. L'objectif principal est de l'étendre aux changements de voie et aux communications, l'outil de vérification permettant déjà d'analyser ces situations. Pour cela, nous allons mettre en place le protocole de communication sur cet outil pour ainsi garantir des propriétés de sécurité, tout en obtenant des résultats similaires sur l'outil de simulation. L'outil de vérification proposant très peu d'erreurs sur les positions et les vitesses, il est tout à fait possible d'imaginer le protocole de communication avec le critère prenant en compte uniquement ces paramètres. A un niveau d'abstraction plus bas, nous allons évaluer le protocole en assurant son réalisme en permettant un changement de voie réaliste sur la durée. L'ensemble du travail effectué permettra de proposer une méthodologie plus complète sur la conception d'un modèle de décision d'un véhicule à la fois sécurisé et réaliste. De plus, l'outil de vérification prend en charge des environnements plus complexes que ceux étudiés dans l'étude de cas du chapitre 4.

Il est possible encore d'affiner la méthodologie globale en déterminant les éléments nécessaires permettant de passer d'une décision à un ensemble de trajectoires réalistes. Ainsi, une nouvelle étape de conception peut permettre de choisir la trajectoire la plus convaincante par rapport à une observation de l'environnement.

L'introduction de véhicules intelligents sur les routes actuelles est révolutionnaire

car elle permettra la réduction du nombre d'accidents sur la route. Toutefois, elle pose dès aujourd'hui de grands problèmes d'éthique. En cas de risque d'accident sur la route, un dilemme de choix peut se poser : qui le véhicule va-t-il sauver ? Le passager du véhicule qui a payé pour l'accès, voire l'achat de celui-ci ? Ou le piéton qui traverse la route ?

De même, imaginer des véhicules communicants offre de nombreux avantages, notamment sur l'anticipation et sur la connaissance globale des autres véhicules. Elle pose toutefois des questions autour de la fiabilité de ce type de système. En effet, il n'est pas impossible d'imaginer un véhicule recevant des communications malveillantes pouvant prendre le contrôle d'un véhicule. C'est pour ces raisons qu'il est indispensable de concevoir un système à la fois sécurisé et sûr.

La mise à disposition de ces véhicules pour le grand public approche à grand pas. Il faudra alors faire face à un enjeu majeur : ne pas faire peur aux usagers de la route avec cette nouvelle technologie. Cela veut dire proposer un véhicule fiable, capable d'assurer une sécurité optimale pour ses passagers et les autres usagers de la route, de réagir en temps réel aux évolutions de la route, de s'adapter à tous types de contextes et de situations, de coopérer avec les autres véhicules qu'ils soient autonomes ou non, et d'avoir en toutes circonstances un comportement qui puisse être perçu comme prévisible et rationnel. La méthodologie et les outils présentés dans ce manuscrit constituent selon nous des briques importantes pour permettre la conception de tels véhicules, et ainsi assurer leur acceptation par le plus grand nombre.

Bibliographie

[1] Archives darpa. http://archive.darpa.mil/grandchallenge/overview.html.

[2] Site officiel de via (senozon). http://www.senozon.com/products/via.

[3] *Microscopic Modeling of Traffic Flow : Investigation of Collision Free Vehicle Dynamics*. PhD thesis, Universität zu Köln, 1998.

[4] Site web officiel de oktal. http://www.oktal.fr/fr/, 2018.

[5] Kazi Iftekhar Ahmed. Modeling Drivers ' Acceleration and Lane Changing Behavior. *Transportation*, Ph.D :189, 1999.

[6] Rob Allan. Survey of Agent Based Modelling and Simulation Tools. *Engineering*, 501(October) :57–72, 2009.

[7] Waymo (Alphabet). Waymo. https://waymo.com/, 2018.

[8] Hasnaâ Aniss. Overview of an ITS project : SCOOP@F. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9669, pages 131–135, 2016.

[9] Johan Arcile, Raymond Devillers, Hanna Klaudel, Witold Klaudel, and Bożena Woźna-Szcześniak. Modeling and checking robustness of communicating autonomous vehicles. In *Advances in Intelligent Systems and Computing*, volume 620, pages 173–180, 2018.

[10] Johan Arcile, Raymond R. Devillers, Hanna Klaudel, Witold Klaudel, and Bozena Wozna-Szczesniak. Modeling and checking robustness of communicating autonomous vehicles. In *Distributed Computing and Artificial Intelligence, 14th International Conference, DCAI 2017, Porto, Portugal, 21-23 June, 2017*, pages 173–180, 2017.

[11] Johan Arcile, Jeremy Sobieraj, Hanna Klaudel, and Guillaume Hutzler. Combination of simulation and model-checking for the analysis of autonomous vehicles' behaviours : a case study. *EUMAS 2017 - EUropean Conference of Multi-Agent Systems, à paraître*, 2017.

[12] M Balmer, K Meister, M Rieser, K Nagel, and K W Axhausen. Agent-based simulation of travel demand : Structure and computational performance of MATSim-T. *2nd TRB Conference on Innovations in Travel Modeling Portland Juni 2008*, (June) :1–33, 2008.

[13] L. Barbosa. Studies on traffic flow models, 1961.

[14] Ana L. C. Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(May) :1–29, 2013.

[15] M Beer, M D'Inverno, N R Jennings, M Luck, C Preist, and M Schroeder. Negotiation in multi-agent systems. *The Knowledge Engineering Review*, 14(3) :285–289, 1999.

[16] Robbin J. Blokpoel, Daniel Krajzewicz, and Ronald Nippold. Unambiguous metrics for evaluation of traffic networks. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 1277–1282, 2010.

[17] Grady Booch. Object-Oriented Development. *IEEE Transactions on Software Engineering*, SE-12(2) :211–221, 1986.

[18] Andrei Borshchev. Getting Started with AnyLogic - a Multi-Method Simulation Modeling Tool , 2010.

[19] Francois Bousquet, Innocent Bakam, Hubert Proton, and Christophe Le Page. Cormas : Common-pool resources and multi-agent systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1416, pages 826–837, 1998.

[20] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1-3) :139–159, 1991.

[21] Gordon D. B. Cameron and Gordon I. D. Duncan. PARAMICS Parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, 10 :25–53, 1996.

[22] Jordi Casas, Jaime L. Ferrer, David Garcia, Josep Perarnau, and Alex Torday. Traffic simulation with Aimsun. In *International Series in Operations Research and Management Science*, volume 145, pages 173–232. 2010.

[23] D Chen. V2V & V2I for connected vehicle. In *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, page 1, 2018.

[24] R. Coppola and M. Morisio. Connected car : Technologies, issues, future trends. *ACM Computing Surveys*, 49(3), 2016.

[25] C. Cseh. Architecture of the dedicated short-range communications (DSRC) protocol. *VTC '98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No.98CH36151)*, 3 :2095–2099, 1998.

[26] Alfred Daniel, Anand Paul, Awais Ahmad, and Seungmin Rho. Cooperative Intelligence of Vehicles for Intelligent Transportation Systems (ITS). *Wireless Personal Communications*, 87(2) :461–484, 2016.

[27] Ernst D. Dickmanns. *Dynamic vision for perception and control of motion*. Springer, 2007.

[28] Karl-Heinz Dietsche and Dietrich Kuhlgatz. History of the automobile. In *Fundamentals of Automotive and Engine Technology*, pages 1–7. 2014.

[29] Arnaud Doniec, René Mandiau, Sylvain Piechowiak, and Stéphane Espié. A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*, 21(8) :1443–1454, 2008.

[30] E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited : on branching versus linear time temporal logic. *J. ACM*, 1(33) :151–178, 1986.

[31] ETSI. *Intelligent Transport Systems (ITS) ; European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*, volume 0. 2010.

[32] Eurostat. Volume of passenger transport relative to gdp. `http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=tran_hv_pstra&lang=en`, July 2018.

[33] Martin Fellendorf and Peter Vortisch. Microscopic traffic flow simulator VISSIM. In *International Series in Operations Research and Management Science*, volume 145, pages 63–93. 2010.

[34] Jacques Ferber. *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*, volume 222. 1999.

[35] Foundation For and Intelligent Physical. FIPA Communicative Act Library Specification. *Architecture*, 2000(SC00037J), 2002.

[36] Martin Fowler and Bob Mcwhirter. *The Definitive ANTLR Reference*, volume 67. 2007.

[37] Commissariat général au développement durable. Chiffres clés du transport - édition 2018. `http://www.statistiques.developpement-durable.gouv.fr/publications/p/2764/873/chiffres-cles-transport-edition-2018.html`, March 2018.

[38] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B*, 15(2) :105–111, 1981.

[39] P. G. Gipps. A model for the structure of lane-changing decisions. *Transportation Research Part B*, 20(5) :403–414, 1986.

[40] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3) :589–606, 2010.

[41] Glassfish Kenai and Migration Admin. Java Compiler Compiler (JavaCC), 2010.

[42] Maxime Gu, Romain Billot, Nour-eddin El Faouzi, De Lyon, and Salima Hassas. Multi-Agent Dynamic Coupling for Cooperative Vehicles Modeling. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 4276–4277, 2015.

[43] Maxime Guériau, Fréderic Armetta, Salima Hassas, Romain Billot, and Nour Eddin El Faouzi. A constructivist approach for a self-Adaptive decision-making system : Application to road traffic control. In *Proceedings - 2016 IEEE 28th International Conference on Tools with Artificial Intelligence, ICTAI 2016*, pages 670–677, 2017.

[44] Maxime Gueriau, Romain Billot, Salima Hassas, Frederic Armetta, and Nour Eddin El Faouzi. An extension of MovSim for multi-agent cooperative vehicles modeling. In *2014 International Conference on Connected Vehicles and Expo, ICCVE 2014 - Proceedings*, pages 859–860, 2015.

[45] Peter Hancock. Are autonomous cars really safer than human drivers ? https://theconversation.com/are-autonomous-cars-really-safer-than-human-drivers-90202, February 2018.

[46] Tobias Haubrich, Sven Seele, Rainer Herpers, Martin E. Muller, and Peter Becker. A Semantic Road Network Model for traffic simulations in virtual environments : Generation and integration. In *2014 IEEE 7th Workshop on Software Engineering and Architectures for Realtime Interactive Systems, SEARIS 2014*, pages 43–50, 2014.

[47] Andrew J. Hawkins. California green lights fully driverless cars for testing on public roads, 2018.

[48] W. Helly. Simulation of bottlenecks in single-lane traffic flow. In *Traffic Flow Theory Symposium*, pages 207–238, 1959.

[49] Jin Yan Hsu, Kun Lung Ku, Tong Kai Jhang, Hung Pang Lin, and Chih Jung Yeh. Integration and implementation of a lane keeping system with vehicle dynamics control. In *2017 International Automatic Control Conference, CACS 2017*, volume 2017-November, pages 1–6, 2018.

[50] Institute of Electrical and Electronics Engineers. 802.11P-2010, 2010.

[51] Venkatesan Kanagaraj, Karthik K. Srinivasan, R. Sivanandan, and Gowri Asaithambi. Study of unique merging behavior under mixed traffic conditions. *Transportation Research Part F : Traffic Psychology and Behaviour*, 29 :98–112, 2015.

[52] Alexander Katriniok, Jan P Maschuw, Lutz Eckstein, Dirk Abel, and A Motivation. Optimal Vehicle Dynamics Control for Combined Longitudinal and Lateral Autonomous Vehicle Guidance. *Control Conference (ECC), 2013 European*, pages 974–979, 2013.

[53] Tobias Kessler and Alois Knoll. Multi vehicle trajectory coordination for automated parking. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 661–666, 2017.

[54] Arne Kesting, Martin Treiber, and Dirk Helbing. General Lane-Changing Model MOBIL for Car-Following Models. *Transportation Research Record : Journal of Transportation Research Board*, 1999(1) :86–94, 2007.

[55] Daniel Krajzewicz and G Hertkorn. SUMO (Simulation of Urban MObility) An open-source traffic simulation. *. . . Symposium on Simulation . . .*, pages 63–68, 2002.

[56] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1-2) :134–152, Oct 1997.

[57] John R Levine, Tony Mason, and Doug Brown. *Lex And Yacc*. 1992.

[58] Patrick Lin. Who's at Fault in Uber's Fatal Collision ? *IEE Spectrum*, 2018.

[59] S. Mammar, V. B. Baghdassarian, and L. Nouveliere. Speed scheduled vehicle lateral control. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pages 80–85, 1999.

[60] Rafael Math, Angela Mahr, Mohammad Mehdi Moniri, and Christian Müller. OpenDS : A new open-source driving simulator for research. *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Appilcations*, pages 7–8, 2012.

[61] Kai Nagel and Michael Schneckenberg. Nagel, Schreckenberg 1992 2221. *Journal de Physique I*, 2(12) :2221–2229, 1992.

[62] D. Ngoduy. Effect of the car-following combinations on the instability of heterogeneous traffic flow. *Transportmetrica B*, 3(1) :44–58, 2015.

[63] NHTSA. Federal Automated Vehicles Policy. *International Energy Agency (IEA)*, (September) :116, 2016.

[64] OICA. Automated driving - definition for levels of automation. `http://www.oica.net/wp-content/uploads/WP-29-162-20-OICA-automated-driving.pdf`, March 2014.

[65] Larry E. Owen, Yunlong Zhang, Lei Rao, and Gene McHale. Traffic flow simulation using CORSIM. *Winter Simulation Conference Proceedings*, 2 :1143–1147, 2000.

[66] Simon Parsons and Michael Wooldridge. Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, pages 1–14, 2000.

[67] Guanghan Peng, Hongdi He, and Wei Zhen Lu. A new car-following model with the consideration of incorporating timid and aggressive driving behaviors. *Physica A : Statistical Mechanics and its Applications*, 442 :197–202, 2016.

[68] L.A. Pipes. An Operational Analysis of Traffic Dynamics. *Journal of Applied Physics*, 24(3) :274–281, 1953.

[69] Mizanur Rahman, Mashrur Chowdhury, Yuanchang Xie, and Yiming He. Review of microscopic lane-changing models and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems*, 14(4) :1942–1956, 2013.

[70] Rajesh Rajamani. Adaptive Cruise Control. In *Encyclopedia of Systems and Control*, pages 13–19. 2015.

[71] Christian Rathgeber, Franz Winkler, Dirk Odenthal, and Steffen Muller. Lateral trajectory tracking control for autonomous vehicles. In *2014 European Control Conference (ECC)*, pages 1024–1029, 2014.

[72] A. Reuschel. Fahrzeugbewegungen in der kolonne. *Österr. Ingen.-Archiv 4, 193-215*, 1950.

[73] Tom Robinson, Eric Chan, and Erik Coelingh. Operating Platoons On Public Motorways : An Introduction To The SARTRE Platooning Programme. In *Proceedings of the 17th ITS World Congress*, 2010.

[74] Mahmoud Saffarzadeh, Navid Nadimi, Saber Naseralavi, and Amir Reza Mamdoohi. A general formulation for time-to-collision safety indicator. *Proceedings of the Institution of Civil Engineers - Transport*, 166(5) :294–304, 2013.

[75] Wouter Schakel, Victor Knoop, and Bart van Arem. Integrated Lane Change Model with Relaxation and Synchronization. *Transportation Research Record : Journal of the Transportation Research Board*, 2316 :47–57, 2012.

[76] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. 2017.

[77] Wen Shi and Ya-Ping Zhang. Decision analysis of lane change based on fuzzy logic. In *Applied Mechanics and Materials*, volume 419, pages 790–794, 2013.

[78] Reid G. Smith. Correction to “The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver”, 1981.

[79] Jérémy Sobieraj, Guillaume Hutzler, and Hanna Klaudel. Modélisation et simulation de la coopération dans les STI : un comparatif. In *Cohésion : fondement ou propriété émergente - JFSMA 17 - Vingt-cinquièmes journées francophones sur les systèmes multi-agents, Caen, France, July 4-6, 2017.*, pages 183–192, 2017.

[80] Jeremy Sobieraj, Lydie Nouvelière, Guillaume Hutzler, and Hanna Klaudel. Modélisation du changement de voie de véhicules autonomes à différents niveaux d'abstraction. *JFSMA 18 - Journées francophones sur les systèmes multi-agents 2018, à paraître*, 2018.

[81] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1) :3–15, January 2011.

[82] Patrick Taillandier. Traffic simulation with the GAMA platform. *International Workshop on Agents in Traffic and Transportation*, 2014.

[83] Patrick Taillandier, Duc An Vo, Edouard Amouroux, and Alexis Drogoul. GAMA : A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7057 LNAI, pages 242–258, 2012.

[84] Geert Jan Olsder Tamer Basar. *Dynamic Noncooperative Game Theory*. 1999.

[85] Sebastian Thrun. What we're driving at. `http://googleblog.blogspot.fr/2010/10/what-were-driving-at.html`, October 2010.

[86] Tomer Toledo, Haris Koutsopoulos, and Moshe Ben-Akiva. Modeling Integrated Lane-Changing Behavior. *Transportation Research Record : Journal of the Transportation Research Board*, 1857 :30–38, 2003.

[87] Ranjeet Singh Tomar and Shekhar Verma. Lane change trajectory prediction using artificial neural network. *International Journal of Vehicle Safety*, 6(3) :213, 2013.

[88] Martin Treiber and Ame Kesting. An open-source microscopic traffic simulator. *IEEE Intelligent Transportation Systems Magazine*, 2(3) :6–13, 2010.

[89] Martin Treiber and Arne Kesting. Elementary Car-Following Models. In *Traffic Flow Dynamics*, pages 157–180. 2013.

[90] Martin Treiber and Arne Kesting. *Traffic Flow Dynamics*. 2013.

[91] Uppaal. `http://www.uppaal.org/`.

[92] VEDECOM. Plateforme d'expérimentation et de test de communication entre véhicules. Présentation dans le cadre de Mobilité@VEDECOM 2018, April 2018.

[93] Katja Vogel. A comparison of headway and time to collision as safety indicators. *Accident Analysis & Prevention*, 35(3) :427 – 433, 2003.

[94] U Wilensky. Netlogo. *Center for Connected Learning and ComputerBased Modeling Northwestern University Evanston IL*, 2009(26.02.2009) :Evanston, IL, 1999.

[95] M. Williams. PROMETHEUS-The European research programme for optimising the road transport system in Europe. In *IEE Colloquium on Driver Information*, pages 1–9, 1998.

[96] Kristie L. Young and Paul M. Salmon. Examining the relationship between driver distraction and driving errors : A discussion of theory, studies and methods, 2012.

[97] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (VANETS) : status, results, and challenges. *Telecommunication Systems*, 50(4) :217–241, 2012.

[98] Sumin Zhang, Weiwen Deng, Qingrong Zhao, Hao Sun, and Bakhtiar Litkouhi. Dynamic trajectory planning for vehicle autonomous driving. In *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, pages 4161–4166, 2013.

Liste des figures

1 Méthodologie de conception d'un modèle de véhicule intelligent. En dessous, le niveau d'abstraction pour chaque étape. 3

1.1 Types de communications entre véhicules communicants. En jaune, les communications entre véhicules (V2V). En violet, les communications avec l'infrastructure (V2I ou I2V). 12

1.2 Perception du lidar des véhicules autonomes de Waymo (© Google). 18

2.1 Squelette d'un fichier GAML sur GAMA avec un début de simulation de véhicules. 38

2.2 Analyse de trois véhicules se dirigeant vers le point cible au début de la simulation (à gauche) et à la fin (à droite). 39

2.3 En haut, le langage définissant trois routes. En bas, la traduction du langage en un fichier XML contenant toutes les informations relatives au cahier des charges de l'utilisateur. 45

2.4 Positionnement des routes à partir du cahier des charges spécifié par l'utilisateur grâce au langage. 46

2.5 Fonctionnement global de la prise de décision. 48

2.6 Diagramme de classes de l'extension. 49

2.7 Temps de simulation pour les trois différents modèles de décision sur une route de **400 mètres**. 52

2.8 Temps de simulation pour les trois différents modèles de décision sur une route de **2000 mètres**. 53

3.1 Principe d'IDM : à chaque pas de temps, le véhicule i adapte sa vitesse par rapport au véhicule $i-1$ 60

3.2 Positionnement des variables pour $MOBIL$. 61

3.3 Évolution de la valeur de satisfaction (HPN) en fonction de la position du véhicule à partir de plusieurs indicateurs de performance. 67
3.4 En haut, le cas où le véhicule A voit un véhicule virtuel en fin de voie. En bas, le cas où le véhicule A ne voit ce véhicule virtuel qu'à partir d'une certaine limite (en orange). 69
3.5 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL**. . . 71
3.6 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL**. 71
3.7 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL safe**. 73
3.8 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL safe**. . . 74
3.9 Application de l'algorithme de perception dans la zone orange. En vert, les couples "proches" (ici, un seul). En bleu, les couples "classiques" (ici, deux). En rouge, les véhicules ne communiquant pas. . . 77
3.10 Accélérations observées lorsque deux véhicules sont considérés comme "classiques". Le véhicule A détermine l'accélération calculée par IDM avec le véhicule C ($aIDM_{A,C}$). Le véhicule B détermine l'accélération calculée par IDM avec le véhicule A ($aIDM_{B,A}$) 78
3.11 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic non mixte** avec le **modèle MOBIL coop**. 83
3.12 Diagrammes spatio-temporels de la valeur de satisfaction (HPN) des véhicules de la voie principale (à gauche) et de la voie d'insertion (à droite) dans le cas d'un **trafic mixte** avec le **modèle MOBIL coop**. . . 83
3.13 **Temps d'attente moyen** des véhicules (*Waiting Time*) en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules autonomes non communicants**). 87

3.14 **Temps de parcours moyen** (*Travel Time*) des véhicules générés en voie principale en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules autonomes non communicants**). 88
3.15 **Temps d'attente moyen** des véhicules (*Waiting Time*) en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules non autonomes et non communicants**). . . 89
3.16 **Temps de parcours moyen** (*Travel Time*) des véhicules générés en voie principale en fonction du pourcentage de véhicules communicants (dans un environnement composé de **véhicules non autonomes et non communicants**). 90

4.1 Étapes de la méthode de recherche de propriétés via la vérification formelle. 96
4.2 Situation initiale du premier scénario avec les vitesses respectives de chaque véhicule. 99
4.3 Situation initiale du scénario 2 (à gauche) avec l'évolution de la vitesse du véhicule B (à droite). 100
4.4 Valeurs de e_1, e_2 et α sur un comportement possible du véhicule B. . . 102
4.5 Trajectoire de changement de voie réaliste 107
4.6 A gauche, la trajectoire simplifiée du changement de voie. A droite, la version instantanée de la trajectoire. 107
4.7 Scenarii des deux études de cas. En haut, la première étude de cas de l'insertion d'un véhicule à partir d'une voie d'accélération. En bas, la seconde étude de cas représentant le dépassement d'un véhicule. 109
4.8 Erreur des positions des véhicules *A* et *M* 111
4.9 Erreur des vitesses des véhicules *A* et *M* 112
4.10 Différence de position entre les véhicules *A* des deux modèles ($realistic.x_A - discrete.x_A$) en fonction du temps dans une des situations possibles. . 113
4.11 Méthodologie de conception affinée d'un modèle de véhicule. 115
4.12 Proposition d'une méthodologie pour la conception du modèle de véhicule. 116

Liste des tableaux

2.1 Tableau comparatif des simulateurs présentés [79]. 36

3.1 Indicateurs de performance individuels. 63
3.2 Indicateurs de performance de groupe. 65
3.3 Paramètres globaux pour chaque simulation ainsi que les paramètres spécifiques à un trafic mixte et non mixte. Dans le cas du trafic mixte, on définit deux types de véhicules : les véhicules légers et les véhicules lourds. 70
3.4 Décisions à prendre dans le cas où les deux véhicules c_i^1 et c_j^2 communiquent et sont "classiques". 79

4.1 Comparaison du temps de parcours et de la position dans un temps donné pour les trois véhicules du premier scénario. 99
4.2 Comparaison par rapport aux critères basés sur l'accélération et le TTC du véhicule A dans le scénario 2. 100
4.3 Différence de valeur (erreur) pour chaque critère entre les deux modèles sur trois exemples de trajectoires. 110

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Méthodes et outils pour la conception de Systèmes de Transport Intelligents Coopératifs

Mots clés : *Systèmes Multi-Agents ; Systèmes de Transport Intelligents ; Simulation ; Coopération*

Résumé : La voiture est le mode de transport le plus utilisé en Europe et en Amérique du Nord. Aujourd'hui, il est de plus en plus sécurisé notamment grâce à des systèmes d'assistance à la conduite. Toutefois, il représente encore la plus grande part d'accidents sur la route en France dont 90 % seraient de cause humaine. A partir de 2020, de nouveaux types de véhicules vont apparaître sur la route : ce sont des véhicules dont la décision ne dépendra plus uniquement du conducteur humain, appelés couramment véhicules autonomes.
Pour concevoir de tels systèmes, trois grandes exigences doivent être respectées à la fois : la sécurité (respect du code de la route), l'efficacité (aller le plus vite possible) et le confort (ne pas se sentir en danger dans le véhicule). En plus d'imaginer un véhicule où seuls les capteurs embarqués fournissent les informations nécessaires pour conduire, on peut ajouter la possibilité de communiquer avec d'autres véhicules ou l'infrastructure de la route. Ce dernier point doit prendre en compte une quatrième exigence, la courtoisie (ne pas avoir un impact négatif sur les véhicules environnants). Il peut permettre également de gérer une situation proche où les véhicules conduits par des êtres humains et les véhicules autonomes vont se trouver dans un même environnement.
Pour étudier ces comportements, la simulation informatique peut être une bonne solution pour mettre en place un éventail de scénarios possibles dans différents environnements. Toutefois, cela impose un niveau d'abstraction pouvant affecter le niveau de réalisme du modèle.
Dans ce manuscrit, nous avons défini des méthodes et outils afin de définir une méthodologie permettant de concevoir des Systèmes de Transport Intelligents Coopératifs. A partir d'un modèle de véhicule, nous allons le simuler, puis le vérifier formellement. A partir du modèle obtenu, on assure une compatibilité avec un simulateur plus réaliste. De plus, à partir d'un outil de simulation, nous avons mis en place un protocole de coopération permettant aux véhicules de s'adapter plus facilement à des environnements routiers actuels.

Title : Methods and tools for the design of Cooperative Intelligent Transportation Systems

Keywords : *Multi-Agent Systems ; Intelligent Transportation Systems ; Simulation ; Cooperation*

Abstract : The car is the most used mode of transport in Europe and North America. Today, it is increasingly secure thanks to driver assistance systems. However, it is still the largest share of road accidents in France, 90% of which are caused by humans. From 2020, new types of vehicles will appear on the road : they are vehicles whose decision will no longer depend only on the human driver, commonly called autonomous vehicles.
To design such systems, three main requirements must be respected at the same time : the safety (respect of traffic laws), the efficiency (go as fast as possible) and the comfort (not to feel in danger in the vehicle). In addition to imagine a vehicle where only on-board sensors provide the necessary information to drive, one can add the ability to communicate with other vehicles or the road infrastructure. This last point have to take into account a fourth requirement, courtesy (does not have a negative impact on surrounding vehicles). It can also help to manage a similar situation where vehicles driven by human beings and autonomous vehicles will be in the same environment.
To study these behaviors, computer simulation can be a good solution to set up a range of possible scenarios in different environments. However, this imposes a level of abstraction that can affect the level of realism of the model.
In this PhD thesis, we defined methods and tools to define a methodology for designing Cooperative Intelligent Transport Systems. From a vehicle model, we have shown that we can simulate it and then formally check it. From the model obtained, compatibility with a more realistic simulator is ensured. In addition, based on a simulation tool, we have implemented a cooperation protocol that allows vehicles to adapt more easily to current road environments.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France