



HAL
open science

Planification de mouvements dynamiques appliquée à la conception de la liaison au sol

Fabrice Boyer

► **To cite this version:**

Fabrice Boyer. Planification de mouvements dynamiques appliquée à la conception de la liaison au sol. Robotique [cs.RO]. Institut National Polytechnique de Toulouse - INPT, 2007. Français. NNT : . tel-00949966

HAL Id: tel-00949966

<https://theses.hal.science/tel-00949966>

Submitted on 20 Feb 2014

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.

N° d'ordre : 2538

THESE

présentée

pour obtenir

LE TITRE DE DOCTEUR DE L'UNIVERSITE DE TOULOUSE

DELIVRE PAR « L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE »

École doctorale : Systèmes

Spécialité : Systèmes Automatiques

Par M. Boyer Fabrice

Titre de la thèse

Planification de mouvements dynamiques
appliquée à la conception de la liaison au sol

Soutenue le 13 Novembre 2007 à 11h devant le jury composé de :

M. Philippe Martinet

Président

MM. Florent Lamiraux

Directeur de thèse

Christian Laugier

Rapporteur

Pierre Ferbach

Membre

Thèse

préparée au
Laboratoire d'Analyse et d'Architecture des Systèmes
Société PSA Peugeot Citroën

en vue de l'obtention du titre de
Docteur de L'Université de Toulouse

Délivré par «**l'Institut National Polytechnique de Toulouse** »
Spécialité Systèmes Automatiques

PLANIFICATION DE MOUVEMENTS
DYNAMIQUES APPLIQUÉE À LA CONCEPTION
DE LA LIAISON AU SOL

Fabrice Boyer

13 Novembre 2007

Soutenu le 13 Novembre 2007 à 11h devant le jury

Philippe Martinet
Florent Lamiroux
Christian Laugier
Pierre Ferbach

*«Ohana» means «Family»
«Family» means «No one gets left behind»
— Lilo & Stitch*

Résumé :

L'objectif de cette thèse est d'étendre l'utilisation des modèles de calcul de la dynamique du véhicule en proposant de nouvelles méthodologies inspirées du domaine de la planification de mouvements. Les modèles de dynamique du véhicule que nous considérons sont de nature industrielle : ils sont complexes, non linéaires et disponibles en règle générale sous forme de boîte noire uniquement. Ils sont utilisés traditionnellement avec des techniques de simulations qui partent d'une définition précise d'un état de départ et des sollicitations venant du conducteur. Les méthodes que nous proposons visent à prendre en compte une description plus "réelle" de la manœuvre ou du test qu'effectue le véhicule : un certain domaine initial dont part le véhicule, un couloir à suivre sans sortir des limites, et un domaine d'arrivée. Outre la recherche d'une solution à ce problème de réalisation d'une certaine manœuvre, on traite également les deux problèmes suivants : produire un échantillon représentatif de l'ensemble des manières de réaliser la manœuvre ; trouver la valeur limite d'un paramètre (par exemple la vitesse initiale) au-delà de laquelle il n'y a plus de solution. Différentes techniques ont été mises en œuvre avec succès, citons notamment : des méthodes exploratoires, un outil de contrôle optimal et un algorithme modifié de déformation de trajectoire. Ces outils génériques sont tous capables de s'adapter sans peine à toutes sortes de véhicules ou d'obstacles. Chacun présente cependant des avantages et des contraintes spécifiques. Ces méthodes ont été appliquées aux cas particuliers de manœuvres standardisées de véhicules de tourisme. Les méthodes proposées permettent de déterminer de manière robuste et cohérente les limites physiques des véhicules sur ces tests.

Mots clés : planification de mouvements, optimisation de trajectoire, déformation de trajectoire, méthodes exploratoires globales, modèle dynamique de véhicule

Abstract :

The main goal of this PhD is to extend the use of vehicle dynamics computation models by proposing new methodologies inspired from the motion planning area. The vehicle dynamics models which are being considered are industrial grade : they are complex, non linear and most of the time only available as a black box. They are traditionally used with simulation techniques requiring a precise initial state and inputs coming from a driver. The methods we are proposing take into account a more realistic description of the current test : an initial state the vehicle is starting from, a corridor to be followed and an arrival area. Additionally to solving the problem of finding such a solution maneuver, two other questions are being addressed : finding a representative set of the numerous solutions ; finding the limit of a parameter (for instance the initial speed) after which no solution is found. Several techniques have been successfully tested : global exploration methods, optimal control and a modified trajectory deformation algorithm. These generic tools are able to easily adapt to new vehicles or obstacles. Each one has specific pros and cons. These methods have been applied to the example of standardized drive maneuvers. The physical limits of a vehicle passing through these tests have been robustly discovered.

Keywords : motion planning, trajectory optimisation, trajectory deformation, global exploration methods, vehicle dynamics model

Remerciements

Je tiens à remercier chaleureusement mon directeur de thèse Florent Lamiraux ainsi que Pierre Ferbach mon tuteur chez PSA. Grâce à leurs compétences et leurs conseils, ils ont su me guider efficacement tout en me laissant la liberté d'explorer mes propres pistes. Je tiens également à remercier Christian Laugier et Philippe Martinet pour avoir accepté d'être les rapporteurs de ma thèse. Je suis également reconnaissant envers Jean Chillon et Patrick Boutard pour leur accueil à PSA, ainsi que Jean-Paul Laumond et Raja Chatilla pour leur hospitalité au LAAS. Un grand merci à tous ceux qui ont contribué de près ou de loin à la réussite de cette expérience tant au LAAS que chez PSA :

- Olivier Lefebvre, David Bonnafous et Matthieu Hillion pour leur aide sur la déformation de trajectoires.
- Matthieu Gallien, Sylvain Joyeux et Matthieu Herbb pour leurs grandes compétences en informatique et tout ce qu'ils ont su m'apporter dans ce domaine.
- Josselyn Gal, Jérémie Deak, Gary Lamarque et Béatrice Bourgeteau pour leurs compétences et leur disponibilité en toutes occasions.
- Saïd Daalaoui, Olivier Fontanel, Pascal de Saint Palais, Ludovic Provost, Benoît Berthe, Vincent Deschamps, Franck Moiroit et Emmanuel Callot pour leur aide sur toutes les questions de modélisation et de dynamique du véhicule.
- Frédéric Monnerie et Benoît Parmentier pour les réponses qu'ils ont su apporter à tous mes problèmes avec Matlab.
- Jorge Villagra, Dominique Marc Bernède, Yacine Chitour, Vincent Boyer et Michel Courdresses pour leur aide sur les questions d'optimisation.
- Laurent Chiche, Xavier Rochereau, Thomas Clément et Cécile Lemaire pour les essais et mesures sur piste.
- Christophe Monteil, Bruno Sachetat, Nicolas Briand, Dominique Dignac, Patrick Rayneau, Michael Venet et Gregory Caillet pour les essais en simulation.
- Thomas Chapron, Jean-Pierre Colinot, Laure Bruel et Emmanuel Ryckebusch pour leur aide lors de la mise en œuvre du simulateur SHERPA.
- Tous ceux que j'ai pu oublier ...

Table des matières

1	Introduction	17
1.1	Contexte général	17
1.2	Objectifs de la thèse	17
1.3	Contraintes	18
1.4	Modélisation des systèmes dynamiques non linéaires	19
1.5	Les solutions industrielles préexistantes	20
1.5.1	Simulation en boucle ouverte	20
1.5.2	Les automates de conduite	22
1.5.3	Optimisation empirique d'un angle volant paramétré	23
1.6	La planification de mouvements	23
1.6.1	Notions essentielles	23
1.6.2	Système cinématique	24
1.6.3	Les algorithmes classiques de la robotique	26
1.7	Plan de l'étude	27
2	Éléments de dynamique du véhicule	29
2.1	Généralités sur la liaison au sol	29
2.1.1	Notions de dérive et de sur-braquage	31
2.1.2	Positionnement du plan de roue	32
2.1.3	La caisse	34
2.1.4	Les pneumatiques	34
2.1.5	Ressorts, amortisseurs et barres anti-dévers	35
2.1.6	Electronic Stability Program (ESP)	37
2.1.7	La répartition de freinage	37
2.2	Les véhicules de référence de cette étude	39
2.3	Sécurité active et tests standardisés	39
2.3.1	La notion de sécurité active	39
2.3.2	Le test d'évitement dit "VDA"	40
2.3.3	Le test d'évitement "ville"	47
2.3.4	Freinage en courbe	48
2.3.5	Vent latéral	50
2.4	Modélisation informatique	50
2.4.1	Modèles organiques	50
2.4.2	Modèle AMESim/Car	51
2.4.3	Modèle SimulinkCar	51
2.4.4	Extensions du modèle SimulinkCar	52
2.4.5	Les fichiers FDV	53
2.4.6	Les sources d'erreur	53
2.4.7	Modélisation des obstacles et détection de collisions	53

3	Les méthodes exploratoires	55
3.1	Introduction	55
3.2	Algorithmes disponibles	56
3.2.1	Forward Dynamic Programming (FDP)	56
3.2.2	Rapidly exploring Random Trees (RRT)	59
3.2.3	Resolution Complete RRT (RCRRT)	60
3.2.4	Kino Dynamic Planner (KDP)	61
3.3	Limitations	61
4	Commande Optimale	67
4.1	Objectifs et Définitions	67
4.2	Techniques existantes	68
4.2.1	Résolution analytique	68
4.2.2	Méthodes numériques directes et indirectes	68
4.2.3	BVP et Méthodes de tir	69
4.3	Principe de fonctionnement	69
4.4	Algorithme utilisé	70
4.5	Paramétrage de l'entrée	70
4.6	Choix du critère	71
4.7	Choix des contraintes	71
4.7.1	Calcul de la distance de pénétration	72
4.7.2	Contraintes heuristiques	77
4.8	Paramétrage de l'algorithme	77
4.9	Résultats	78
4.9.1	Recherche du \mathcal{P} limite	78
4.9.2	Convergence	79
4.9.3	Robustesse	81
4.10	Limitations	81
5	La déformation de trajectoire	83
5.1	Introduction	83
5.2	Principe de fonctionnement	83
5.3	Théorie de la déformation de trajectoire	85
5.3.1	Idée initiale	85
5.3.2	Décomposition de la perturbation	86
5.3.3	La fonction de potentiel	87
5.3.4	Perturbation des entrées	89
5.3.5	Résumé de l'algorithme	89
5.4	Calcul des forces	89
5.4.1	Distance aux obstacles	90
5.4.2	Distance de pénétration	90
5.4.3	Forces heuristiques	90
5.5	Choix de la base de décomposition $\{\mathbf{e}_i\}_{i=[1,p]}$	92
5.6	Calcul numérique des \mathbf{E}_i	92
5.7	Conditions aux limites et orthonormalisation	95
5.8	Prise en compte de contraintes sur la commande	97
5.9	Implémentation et paramétrage	97
5.10	Résultats	100
5.10.1	Recherche des limites de \mathcal{P}	100

5.10.2	Convergence	105
5.10.3	Respect de l'hypothèse de linéarité locale	105
5.10.4	Robustesse	105
5.11	Limitations	112
5.12	Synthèse	113
6	Recherche d'un ensemble de solutions	115
6.1	Approche par les méthodes exploratoires	115
6.2	Approche par l'optimisation	117
6.3	Approche par la déformation de trajectoire	119
6.3.1	Perturbation des entrées	119
6.3.2	Projection et orthonormalisation	119
6.3.3	Ajustement de l'amplitude des perturbations	119
6.3.4	Résultats et robustesse	121
6.4	Approche avec des essayeurs humains	124
6.4.1	Description du simulateur de conduite	124
6.4.2	Protocole de mesure	126
6.4.3	Résultats	126
6.5	Synthèse	128
7	Conclusion	131
7.1	Synthèse technique	131
7.2	Utilisation industrielle	132
7.3	Perspectives	132
A	Interface Matlab pour la Motion Strategy Library	135
	Bibliographie	139

Notations

- \mathcal{C} : espace des configurations
- \mathcal{C}_{obst} : espace des collisions
- \mathcal{C}_{free} : espace libre
- \mathcal{C}_{zci} : zone de collision inévitable
- s : temps
- \mathbf{q} : vecteur d'état du système
- \mathbf{u} : vecteur d'entrée
- $\boldsymbol{\eta}$: direction de déformation
- $\boldsymbol{\eta}_{max}$: norme du déplacement maximum de la trajectoire entre deux itérations
- \mathbf{v} : perturbation
- n : nombre de points de la trajectoire discrétisée
- p : taille de la base \mathbf{e}
- m : dimensions de l'espace d'état
- \mathcal{U} : ensemble des entrées \mathbf{u} possibles
- \mathbf{E}_i : direction de déformation élémentaire
- \mathbf{e}_i : perturbation élémentaire
- Φ : une trajectoire dans l'espace d'état
- V : potentiel global d'une trajectoire
- U : potentiel d'une configuration
- $\boldsymbol{\lambda}$: vecteur coordonnées dans les bases \mathbf{e} et \mathbf{E}
- τ : coefficient d'ajustement de l'amplitude des déformations
- PD^g : distance de pénétration généralisée
- V_{max} : vitesse de passage maximale dans le test en cours
- $P_{freinage}$: pression de freinage
- γ_{lat} : accélération latérale
- γ_{longi} : accélération longitudinale
- E_{AV}, E_{AR} : empattements avant/arrière
- θ, ϕ, ψ : angles de roulis, tangage, lacet
- δ_{volant} : braquage des roues AV créé par la rotation du volant
- $\delta_{AV}, \delta_{AR}, \delta_{CDG}$: dérives avant, arrière et au centre de gravité
- F_z : effort vertical appliqué sur une roue
- \mathcal{P} : paramètre d'un test que l'on peut faire varier

Abréviations

- ABS : AntiBlockierSystem (système anti-blocage de roue)
- AFU : Aide au Freinage d’Urgence
- AV/AR : AVant/ARrière
- BAD : Barre Anti-Dévers
- BCV : Banc de Caractérisation Véhicule
- BFGS : Broyden-Fletcher-Goldfarb-Shanno
- BVP : Boundary Value Problem
- CDG : Centre De Gravité
- CGC : Contrôle Global de Chassis
- CI : Conditions Initiales
- DAE : Direction Assistée Électrique
- DAEH : Direction Assistée Électro-Hydraulique
- DDL : Degrés De Liberté
- DIRCOL : DIRect COLlocation
- DSC : Dynamic Stability Control
- DT : Déformation de Trajectoire
- ESP : Electronic Stability Program
- FDP : Forward Dynamic Programming
- FDV : Fichier de description de Dynamique Véhicule
- GA : Genetic Algorithms
- GPC : General Polygon Clipper
- ICRA : International Conference on Robotics and Automation
- IEEE : Institute of Electrical and Electronics Engineers
- ISO : International Organization for Standardization
- KDP : KinoDynamic Planner
- KKT : Karusch Kuhn Tucker
- LAS : Liaison Au Sol
- MNS : Masse Non Suspendue
- MSL : Motion Strategy Library
- MTAC : Masse Totale Admissible en Charge
- NHTSA : National Highway Traffic Safety Administration (sécurité routière américaine)
- NLSQP : Non Linear Sequential Quadratic Programming
- ODE : Ordinary Differential Equation
- PD : Programmation Dynamique
- PID : Proportionnel Intégral Dérivée
- PM : Planification de Mouvements
- PNL : Programmation Non Linéaire
- PQP : Proximity Query Package
- PRM : Probabilistic RoadMaps

- PSA : Peugeot Société Anonyme
- RCRRT : Resolution Complete Rapidly exploring Random Trees
- REF : Répartition Électronique de Freinage
- RK : Runge-Kutta
- RRT : Rapidly exploring Random Trees
- SAE : Society of Automotive Engineers
- SB : Sur-Braquage
- SBD : Sur-Braquage Dynamique
- SHERPA : Simulateur de conduite
- SQP : Sequential Quadratic Programming
- TSB : Taux de Sur-Braquage
- VDA : Verband Der Automobilindustrie (union de l'industrie automobile)

Chapitre 1

Introduction

1.1 Contexte général

Le comportement routier des véhicules de tourisme est couramment simulé par les ingénieurs des constructeurs automobiles à l'aide de modèles dynamiques. Ceux-ci sont souvent complexes et fortement non linéaires. Ils sont sollicités de manières diverses afin d'évaluer les qualités du véhicule sur certaines prestations bien définies. Un des objectifs de ces simulations est d'être représentatif des essais physiques réalisés par les essayeurs. Ces manœuvres consistent par exemple à partir d'une situation initiale puis à arriver dans un domaine objectif tout en respectant certaines contraintes. Les contraintes sur la trajectoire se présentent généralement sous la forme d'un couloir à l'intérieur duquel le véhicule doit rester. Ce couloir est modélisé par un ensemble d'obstacles.

Dans le domaine de la simulation des systèmes dynamiques, les méthodes appliquées pour exploiter les modèles et simuler ces manœuvres sont limitées. Elles se résument le plus souvent à de la boucle ouverte ou fermée (voir chapitre 1.5).

La "planification de mouvements dynamiques" se définit comme le calcul de mouvements solutions pour un système qui part d'une situation initiale et doit rejoindre une situation objectif tout en évitant des obstacles et en respectant certaines contraintes. On aperçoit bien ici la similitude du problème et l'intérêt qu'il y a à utiliser ces techniques sur les modèles de dynamique du véhicule. Cependant, les comportements dynamiques non linéaires, les contraintes industrielles et surtout la difficulté même qu'il y a à résoudre ces problèmes de planification nous amèneront à développer plusieurs techniques différentes suivant les cas considérés.

1.2 Objectifs de la thèse

L'objectif de cette thèse est de proposer de nouvelles méthodologies de calcul permettant de simuler des manœuvres de dynamique du véhicule. Les éléments disponibles sont les mêmes que ceux dont dispose un essayeur sur une piste, à savoir :

- Un véhicule : Etant donné que l'on travaille en simulation, un modèle informatique représentant le plus fidèlement possible le comportement dynamique du véhicule est utilisé. Ce modèle fournira un ensemble de contraintes sur les trajectoires possibles du modèle.

- Une piste d’essai : Elle est modélisée par un couloir plus éventuellement un ensemble d’obstacles décrits par des polygones. La définition géométrique de la carrosserie du véhicule doit également être connue pour la détection de collisions.
- Des conditions initiales : Le véhicule commence la manœuvre avec certaines conditions initiales de position et de vitesse.
- Des contraintes sur l’entrée : Le conducteur étant un être humain, il ne peut appliquer que des vitesses ou des couples au volant bornés. Il faut donc prendre en compte ces contraintes afin d’obtenir des solutions réalistes.

On introduit ici la variable \mathcal{P} qui peut être un paramètre quelconque de la manœuvre ou des conditions initiales. Cette variable est différente pour chaque manœuvre (voir chapitre 2.3). Dans le cas d’un test VDA, il pourra s’agir de la vitesse de passage, pour un freinage en courbe \mathcal{P} pourra être la pression de freinage, etc. Les procédures de calcul qui seront développées dans la suite de cette thèse poursuivent plusieurs objectifs.

En premier lieu, elles doivent permettre d’obtenir une solution quelconque au problème. Pour cela, on attend en sortie de l’algorithme une trajectoire géométriquement sans collision avec les obstacles. On attend également la valeur du paramètre \mathcal{P} après laquelle il n’existe plus de solution. Les résultats de la recherche de limite devront être robustes et pertinents par rapport à des modifications de paramétrage du modèle (masse, pneu, etc.) connues pour impacter cette limite. On souhaite également aller plus loin et déterminer un ensemble de trajectoires solutions.

Les *trois problèmes principaux* traités dans ce document sont donc :

- **Problème 1** : Détermination d’une solution quelconque pour une manœuvre à \mathcal{P} donné.
- **Problème 2** : Calcul de la valeur limite du paramètre \mathcal{P} au delà de laquelle il n’existe plus de solution.
- **Problème 3** : Détermination d’un ensemble de solutions aussi représentatif que possible à \mathcal{P} donné.

1.3 Contraintes

L’ensemble de ces calculs devra pouvoir s’effectuer en des temps compatibles avec le rythme des études effectuées en bureau d’étude. La durée des simulations a été arbitrairement fixée à une nuit au maximum. L’obtention de ces résultats est également contrainte par la volonté d’utilisation industrielle des outils de cette thèse. La plus importante limitation est l’emploi de modèles de véhicules utilisés par les ingénieurs et représentatifs d’un comportement réel. Ceci exclut de se limiter à des modèles analytiques simplifiés, si souvent utilisés dans la recherche sur ces questions.

Afin de rester au plus proche de la réalité, la prise en compte exacte de la géométrie du véhicule et des obstacles via un système de détection de collisions s’avère également nécessaire. Les obstacles décrivant le couloir pourront ainsi être adaptés sans avoir à modifier les outils de résolution mis en œuvre. Ces contraintes sont volontairement surdimensionnées car il est évident que l’on pourrait avoir accès aux équations du modèle utilisé ou à un modèle analytique des obstacles. Mais alors, une simple modification de la manœuvre obligerait à retoucher un code peu familier aux ingénieurs. Lors de cette thèse, le choix a été fait de considérer le modèle de véhicule comme un système boîte noire fermé. Un tel système se définit comme une boîte acceptant des sollicitations en entrée et fournissant des sorties en retour, mais dont le fonctionnement interne est totalement inconnu et inaccessible. Certaines hypothèses de continuité et de dérivabilité des

trajectoires obtenues ont tout de même été retenues afin de faciliter les calculs. Ces hypothèses sont peu choquantes dans le cadre de la modélisation du comportement dynamique de véhicules.

Cette modélisation boîte noire permet d'utiliser facilement des environnements de modélisation classiques tels que Matlab/Simulink et autorise la modification du véhicule dans son interface graphique habituelle. Il est par exemple possible de rajouter des systèmes de contrôle mécatroniques avancés qui modifient profondément le comportement du véhicule (voir l'ESP au chapitre 2.1.6).

On aperçoit ici l'orientation forte vers les contraintes industrielles d'utilisation par des ingénieurs peu ou pas initiés aux techniques développées dans la suite de ce document. La généralité des techniques employées autorise évidemment leur réutilisation aisée pour des modèles académiques souvent plus simples (cinématiques) mais pas pour autant triviaux à résoudre dans le cadre de la planification de trajectoire.

La planification de mouvements dynamiques est un problème dit "NP-complet". Ceci signifie que l'on sait qu'il est impossible à résoudre en temps polynomial. Le nombre de situations à explorer explose très rapidement et il faudra obligatoirement faire des approximations pour arriver à résoudre le problème en un temps raisonnable (discrétisation, exploration locale, etc.). La résolution numérique ou théorique exacte est donc inenvisageable pour cette catégorie de problèmes.

1.4 Modélisation des systèmes dynamiques non linéaires

Au sein de la branche dédiée à la modélisation et aux calculs de dynamique route chez PSA, des ingénieurs sont chargés de créer, paramétrer, maintenir et corréler des modèles de comportement dynamique des véhicules. Ces modèles non linéaires peuvent être vus comme un système entièrement boîte noire prenant en entrée certaines commandes fonction du temps (angle volant, accélération, freinage, etc.) et fournissant d'autres variables en sortie (trajectoire, roulis, lacet, etc.). Suivant la complexité du modèle, le système possède un nombre plus ou moins élevé de variables d'état qui décrivent le comportement interne du système. Ces variables d'état \mathbf{q} représentent le minimum d'information à connaître sur le système pour pouvoir décrire son comportement futur. Les systèmes physiques sont classiquement modélisés par un ensemble d'équations différentielles fournissant la dérivée de l'état à partir de la seule donnée de l'état courant $\mathbf{q}(s)$, de l'entrée $\mathbf{u}(s)$ et du temps s :

$$\dot{\mathbf{q}}(s) = f(\mathbf{q}(s), \mathbf{u}(s)) \quad (1.1)$$

La notation $\dot{\mathbf{q}}$ représentera à partir de maintenant la dérivée de \mathbf{q} par rapport au temps s . On appelle cette modélisation une formulation explicite. Cette équation très simple et générique permet d'utiliser directement toutes les méthodes de résolution numérique classiques (Euler, Runge-Kutta, Adams-Moulton, etc.).

Le temps a été noté s et non pas t en raison de l'origine des algorithmes utilisés par la suite. La plupart des robots sont modélisés par des systèmes purement cinématiques qui ne font pas intervenir explicitement le temps dans leur formulation, mais l'abscisse curviligne s . Cette thèse s'inscrivant dans la continuité de ces travaux de robotique, les notations sont reprises afin de conserver une homogénéité dans les équations. Et cela, même si le temps s intervient explicitement

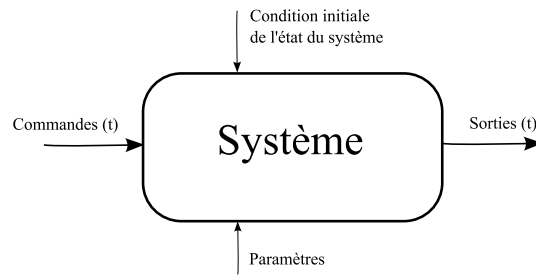


FIG. 1.1 – Simulation d’un modèle en boucle ouverte

dans la modélisation du système et qu’il n’est pas nécessaire de paramétrer temporellement les trajectoires générées.

Le monde de l’automatique effectue une distinction entre l’état interne d’un modèle et ses sorties qui sont calculées à partir de l’état via l’équation $\mathbf{y} = g(\mathbf{q}(s), \mathbf{u}(s))$. Sauf mention contraire, nous considérerons que la sortie \mathbf{y} du modèle est une image exacte de son état interne \mathbf{q} .

Un système dynamique est décrit à un instant $s \in [0, S]$ par la valeur de l’ensemble de ses variables d’état. Ces variables sont contenues dans le vecteur configuration $\mathbf{q}(s)$, celui-ci contient toute l’information nécessaire à la reprise d’une simulation depuis cet instant s . L’ensemble des configurations possibles pour un système (i.e. l’ensemble des valeurs potentiellement accessibles par le vecteur d’état du système) est appelé espace des configurations et est nommé \mathcal{C} . Le symbole \mathcal{U} représente l’ensemble des entrées \mathbf{u} possibles. La figure 1.1 illustre le concept du système dynamique prenant $\mathbf{u}(s)$ en entrée et fournissant $\mathbf{q}(s)$ en sortie.

Dans la pratique, une modélisation suffisamment générique de la base mécanique passive d’un véhicule type a été définie. Celle-ci permet de représenter le comportement routier de la plupart des véhicules particuliers commercialisés aujourd’hui. Les ingénieurs n’ont alors plus qu’à identifier le paramétrage de ces modèles à partir de mesures ou de données physiques. Chez PSA, l’ensemble de ces grandeurs est condensé dans un fichier unique appelé Fichier de Dynamique Véhicule (FDV). Les principaux concepts de modélisation associés à ce paramétrage seront détaillés dans le chapitre 2. Le vecteur \mathbf{q} contient beaucoup de variables en raison de la prise en compte de nombreux degrés de libertés sur le modèle. C’est l’intégration des positions et surtout des vitesses à l’intérieur du vecteur \mathbf{q} qui permet la modélisation d’effets dynamiques via l’équation (1.1). Le modèle que nous utilisons possède un comportement complexe et beaucoup de variables d’état. Cependant, le faible nombre d’entrées utilisées (1 ou 2) simplifie grandement le problème et autorise les approches développées dans la suite du document.

1.5 Les solutions industrielles préexistantes

Dans le domaine de la dynamique route, diverses solutions sont déjà utilisées afin de simuler des manœuvres quelconques. La grande diversité des techniques mises en œuvre montre le fort intérêt de la communauté vis-à-vis de ces questions.

1.5.1 Simulation en boucle ouverte

L’entrée \mathbf{u} d’un modèle de véhicule peut être alimentée par un angle volant issu d’une mesure sur véhicule (voir figure 1.1), ou par son approximation. Ces sollicitations prédéfinies permettent

d'obtenir un certain nombre d'informations sur le comportement du système. Cependant, elles ne garantissent pas que la trajectoire soit une solution du problème simulé, et elles n'essaient pas de s'adapter au comportement du véhicule. Les limites en \mathcal{P} de la manœuvre en cours ne peuvent en aucun cas être déterminées ainsi. On rappelle que \mathcal{P} est un paramètre de la manœuvre en cours (vitesse initiale, intensité de freinage, etc.) que l'on peut faire évoluer. C'est la variation de ce paramètre qui permet de pousser le véhicule vers ses limites physiques lors du test considéré.

La simulation brute d'angles volant mesurés ne fonctionne généralement pas immédiatement en raison des écarts entre modélisation et réalité. Cette manière de procéder peut conduire à la déstabilisation du véhicule simulé qui effectue un tête-à-queue. Les ingénieurs sont donc parfois amenés à modifier les angles volant pour les rendre acceptables par le système modélisé.

Les études théoriques de commandabilité (globale ou en temps fini) ou d'observabilité, fréquemment abordées en automatique classique sont difficilement réalisables dans le cas d'un système dynamique boîte noire non linéaire. La définition de ce dernier peut fortement varier du fait de l'ajout de systèmes électroniques et de la grande diversité des paramétrages possibles. Elles ne seront donc pas abordées ici, mais l'article [Hermann et Krener, 1977] permet d'en approfondir les notions fondamentales. Il est question dans cette thèse du comportement dynamique non linéaire d'un véhicule soumis à des sollicitations très exigeantes et non du comportement purement cinématique de celui-ci lors de manœuvres de type parking. Cela complexifie grandement le problème.

De manière générale, on peut considérer que pour les problèmes 1 et 2 présentés dans le chapitre 1.2, le système est très instable. En effet, lorsque l'on rajoute une petite perturbation à un angle volant, la trajectoire résultante peut grandement varier. La figure 1.2 illustre bien ce phénomène. Une tolérance d'un demi-degré sur l'angle volant suffit à décaler la trajectoire de presque 1 m. Une telle déviation crée inévitablement de nombreuses collisions. C'est cette sensibilité qui explique en grande partie l'échec des méthodes exploratoires se basant sur une discrétisation de l'entrée \mathbf{u} (voir chapitre 3).

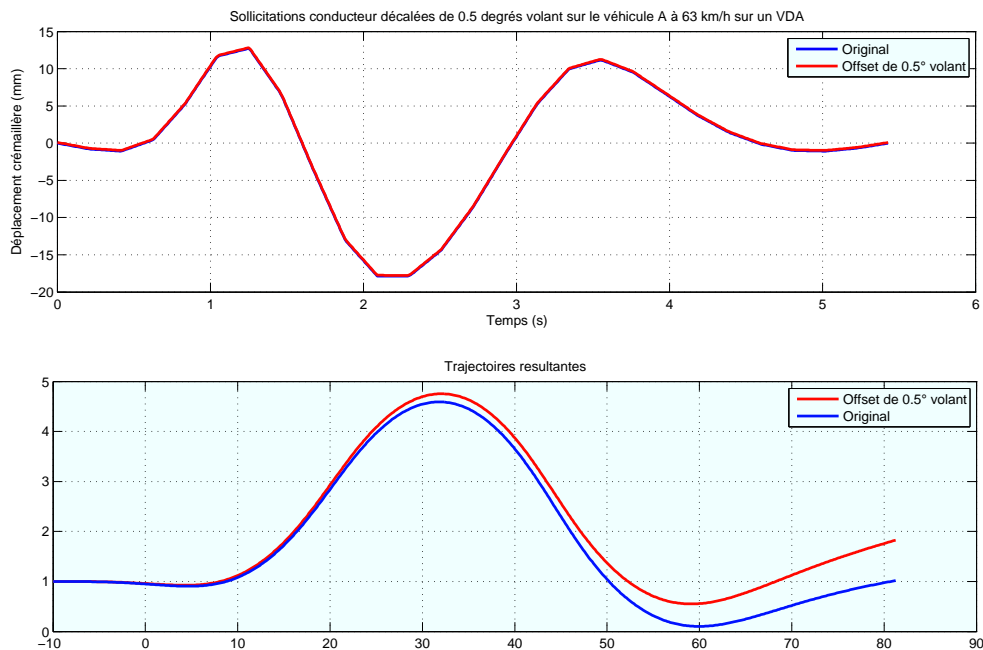


FIG. 1.2 – Une petite perturbation sur l'angle volant peut entraîner de grandes répercussions sur la trajectoire résultante

1.5.2 Les automates de conduite

Une autre possibilité pour approcher la résolution des problèmes 1 et 2 du chapitre 1.2 est d'asservir le modèle en utilisant une boucle de rétroaction. Ce système comporte le plus souvent un comparateur, un régulateur et le modèle de véhicule (voir figure 1.3). L'entrée est alors une trajectoire de référence sur laquelle le véhicule s'asservit. Les résultats obtenus dépendent fortement de la trajectoire de référence choisie et des performances du régulateur. La trajectoire de référence ne s'adapte pas pour utiliser au mieux le potentiel du véhicule, et l'on ne garantit pas l'absence de collisions. La trajectoire de référence n'est d'ailleurs jamais parfaitement suivie. D'autres grandeurs telles qu'une mesure d'accélération latérale sur véhicule peuvent servir de référence, mais les limitations sont les mêmes. Le régulateur le plus simple, et le plus utilisé en pratique est l'action Proportionnelle Intégrale Dérivée (PID). Sa mise en œuvre et son réglage sont connus depuis longtemps des automaticiens.

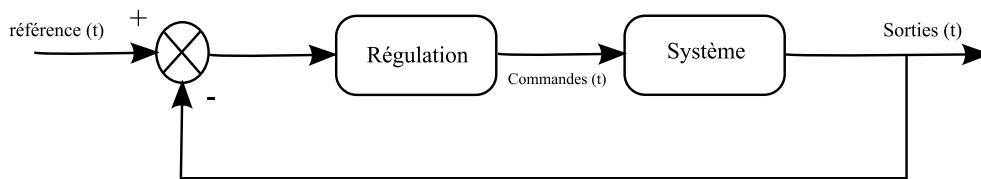


FIG. 1.3 – Schéma général d'une régulation en boucle fermée asservie sur une trajectoire de référence

Ces méthodes conviennent parfaitement au suivi de trajectoires relativement peu contraignantes où le véhicule a un comportement linéaire suffisamment prédictible (accélérations latérales $\gamma_{lat} \leq 3 \text{ m/s}^2$). Elles ne sont par contre pas aptes à exploiter au mieux le potentiel du véhicule qui se cache dans ses non linéarités ($3 \text{ m/s}^3 \leq \gamma_{lat} \leq 10 \text{ m/s}^2$). Un exemple d'utilisation de ces techniques est le "roulage numérique" qui vise à calculer le fonctionnement des trains en endurance. La régulation est alors utilisée pour faire suivre au modèle une trajectoire de plusieurs kilomètres. Les signaux obtenus sur le modèle au niveau des différents organes sont réalistes et adaptés au véhicule.

Dans le cadre de la régulation en boucle fermée, toutes les autres méthodes de l'automatique linéaire ou non linéaire sont utilisables. Cependant, des développements spécifiques au problème considéré ont été réalisés afin d'assimiler le régulateur à un modèle de conducteur humain. De très nombreux travaux visent à représenter un conducteur sous forme d'une fonction de transfert, citons notamment [Zhong *et al.*, 2000] [Pascali *et al.*, 2002], [Naudé et Steyn, 1993], [Renski, 1998], [Allen et J.P.Christos, 2002], [Day et Metz, 2000]. Les paramètres typiques d'un modèle de conducteur sont des temps d'anticipation, des distances de visibilité ou des gains. Ces techniques sont très répandues dans l'industrie aujourd'hui. D'autres techniques utilisent la logique floue [Hajjaji et Ouladsine, 2001, Uzunsoy et O.Olatunbosun, 2003], la commande prédictive [Prokop et Günther, 2001, Garrott *et al.*, 1982], la commande adaptative [Post *et al.*, 1997] ou les réseaux de neurones [Fujioka *et al.*, 1991] pour modéliser un conducteur.

Le paramétrage de ces modèles est toujours quelque peu empirique et ne permet pas de garantir la stabilité ou la robustesse de ces techniques. De manière plus théorique, on sait qu'il n'existe aucune loi de commande stabilisante générique (stabilizing feedback) pour le problème qui nous intéresse. Des techniques d'optimisation ou de contrôle optimal ont été utilisées pour guider des véhicules de course le long de circuits complets tels que cela est décrit par [Muehlmeier et Mueller, 2002, Vögel *et al.*, 2003]. L'approche généralement privilégiée est de minimiser l'écart entre la trajectoire en cours et une trajectoire de référence qui doit toujours être

fournie par avance. On verra au chapitre 4 qu'il est possible d'adapter ces algorithmes pour déterminer directement une trajectoire solution au problème qui nous intéresse.

Il est possible d'utiliser la ligne médiane d'un circuit de plusieurs kilomètres comme référence pour ces techniques. Par contre, une succession de couloirs présente un centre géométrique beaucoup moins régulier et difficile à utiliser comme référence. La figure 1.4 présente l'allure de la ligne médiane d'un couloir calculé à l'aide de l'algorithme de Voronoï (détaillé dans [Aurenhammer, 1991]). Pour pallier à ce problème, il est possible de faire appel à de nombreuses

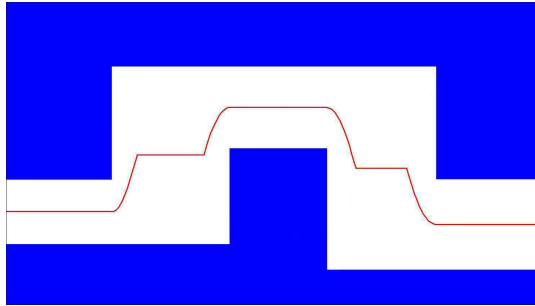


FIG. 1.4 – Centre géométrique d'un couloir calculé à l'aide de l'algorithme de Voronoï

techniques capables de générer des trajectoires de référence. Les automates vus précédemment pourront ainsi s'y asservir. On peut par exemple utiliser des techniques de planification de trajectoires avec un modèle simplifié (voir chapitre 1.6.3).

Le point commun de toutes ces techniques est d'utiliser une entrée de référence arbitraire, ce qui les rend toutes inutilisables pour résoudre les problèmes 1 et 2 du chapitre 1.2.

1.5.3 Optimisation empirique d'un angle volant paramétré

Une des solutions déjà mise en œuvre chez PSA pour essayer de simuler un test a été de paramétrer un angle volant à l'aide de morceaux de courbes (droites, demi-cercles) afin d'obtenir une sollicitation relativement représentative (voir figure 1.5). Une optimisation est ensuite lancée : les paramètres de l'angle volant sont ajustés en fonction des cônes percutés par le véhicule jusqu'à résolution des collisions. La manière dont s'adapte l'angle volant en fonction des cônes renversés est entièrement issue de l'expérience des essayeurs. Cette méthode ad-hoc présente des problèmes de convergence et ne s'adapte pas facilement à de nouveaux modèles ou à de nouvelles manœuvres.

1.6 La planification de mouvements

1.6.1 Notions essentielles

Comme cela a été vu au chapitre 1.4, un système dynamique peut être représenté dans son intégralité à chaque instant s grâce à la donnée du vecteur configuration \mathbf{q} . Le temps s appartient à l'intervalle $I = [0, S]$ où S est la durée totale de la simulation. On définit une trajectoire comme étant une application Φ telle que :

$$\begin{aligned} \Phi : I &\mapsto \mathcal{C} \\ s &\mapsto \Phi(s) = \mathbf{q}(s) \end{aligned} \quad (1.2)$$

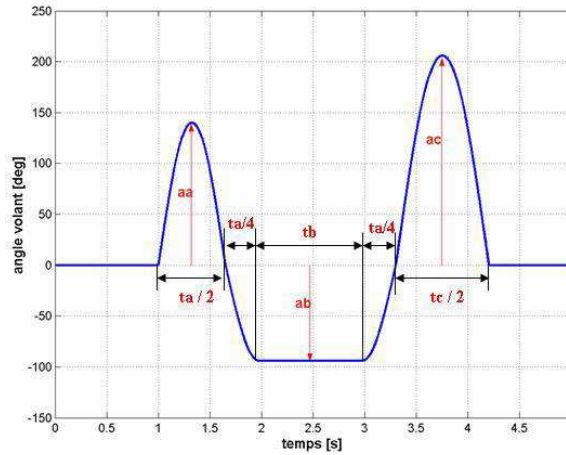


FIG. 1.5 – Paramétrage simplifié d'un angle volant à l'aide d'arcs de cercle et de segments de droite

L'espace \mathcal{C} de l'ensemble des configurations physiquement atteignables par le système est contraint par la condition supplémentaire de non-collision avec les obstacles du monde physique (on pourrait aussi y intégrer d'autres contraintes portant sur d'autres variables d'état). On définit \mathcal{C}_{coll} comme étant le sous-ensemble de \mathcal{C} tel que :

$$\forall \mathbf{q} \in \mathcal{C}_{coll}, \mathbf{q} \text{ est en collision avec un obstacle}$$

Dans le cas d'une manœuvre en couloir effectuée par un véhicule non ponctuel, la figure 1.6 représente l'évolution de \mathcal{C}_{coll} pour les variables d'état x, y . Chaque figure est calculée pour un angle de lacet du véhicule ψ différent. On définit l'espace libre \mathcal{C}_{free} comme étant $\mathcal{C} - \mathcal{C}_{coll}$. Cet espace représente le couloir géométrique à l'intérieur duquel le véhicule peut évoluer librement.

Le calcul explicite de ces différentes zones est souvent très difficile, voire impossible. Il est cependant possible d'obtenir une approximation de l'espace libre à l'aide de différences de Minkowski comme cela a été fait au chapitre 4.7.1 si le véhicule est convexe.

1.6.2 Système cinématique

L'exemple canonique du système purement cinématique utilisé en robotique, est un robot cubique actionné par deux roues indépendantes situées de part et d'autre du centre de son axe longitudinal (voir figure 1.7). Ce modèle est inspiré du robot *Hilare 2* qui a été développé par le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) et décrit dans [Jacobs *et al.*, 1991, Lamiroux *et al.*, 1999]. Il servira par la suite de modèle simplifié facilement accessible et rapide à simuler afin de mettre au point les algorithmes.

Un tel véhicule peut se modéliser grâce à la définition de sa vitesse longitudinale v et de sa vitesse de rotation $\omega = \dot{\psi}$ autour de son axe vertical (vitesse de lacet). Son vecteur d'entrée est donc $\mathbf{u} = [v, \omega]$. Ses paramètres d'état sont sa position x, y et son angle de lacet ψ . Dans ce cas-là, on a $\mathbf{q} = [x, y, \psi]$. Si l'on suppose que le véhicule roule sans glisser, son vecteur vitesse doit en permanence être colinéaire à son axe longitudinal. L'équation modélisant le comportement

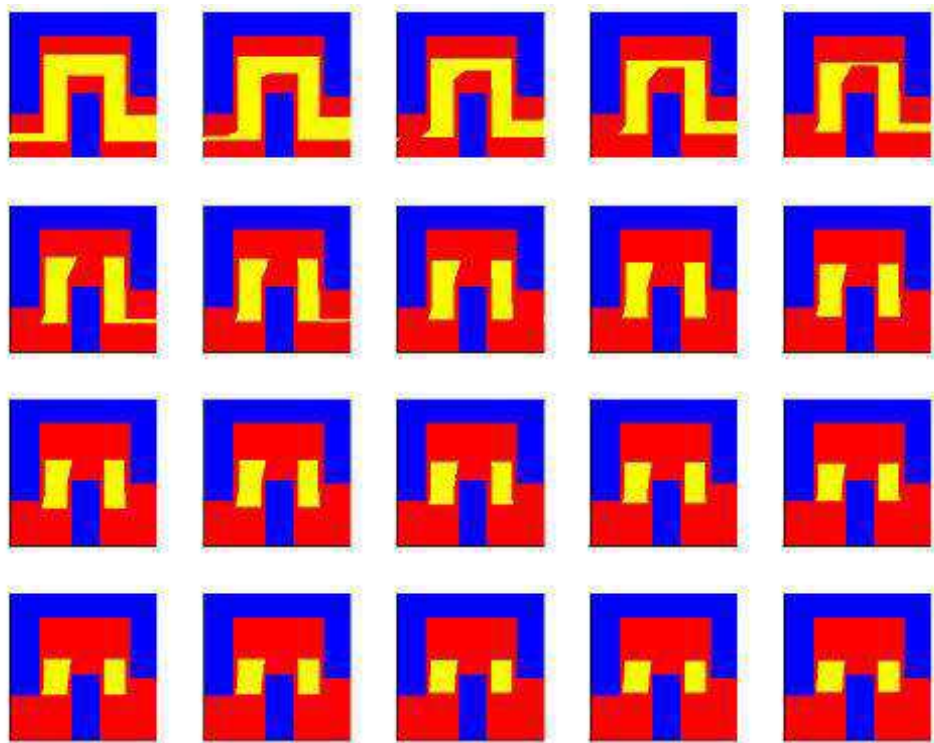


FIG. 1.6 – Représentation graphique de l'espace libre (en jaune) d'un test en couloir pour ψ variant entre 0 et $\pi/2$

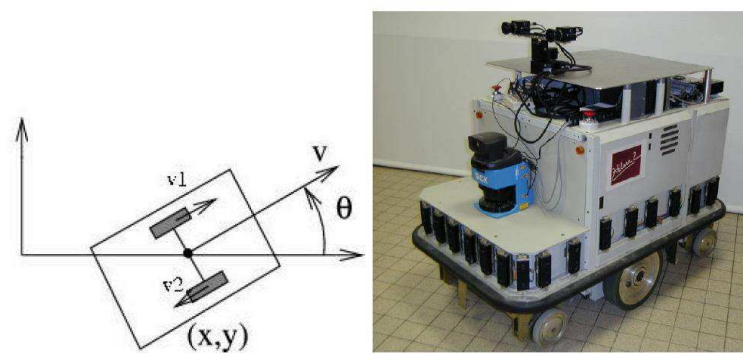


FIG. 1.7 – Modélisation simple d'un robot type *Hilare 2*

cinématique de ce robot est alors :

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} v \cdot \cos(\psi) \\ v \cdot \sin(\psi) \\ \omega \end{pmatrix} \quad (1.3)$$

Ce type de robot est dit non-holonome, car il possède plus de degrés de liberté que d'actionneurs. Il faut donc que celui-ci effectue des créneaux pour se déplacer latéralement. C'est ce genre de comportement, relativement proche de ceux des voitures, qui rend ces robots intéressants en tant que modèle simplifié de véhicule. Si le véhicule est purement cinématique, on ne parle plus de planification de mouvements (PM) mais de planification de trajectoires. Il est à noter que le terme anglais "Path Planning" englobe souvent les deux notions. Le modèle de véhicule PSA est évidemment non-holonome du fait de la prise en compte des phénomènes dynamiques.

1.6.3 Les algorithmes classiques de la robotique

La généralité des techniques de planification de mouvements permet la résolution des problèmes 1 et 2 présentés au chapitre 1.2, sans avoir à effectuer d'adaptations ou de réglages spécifiques à la manœuvre considérée. Elles s'adaptent naturellement à tout type de contraintes (modèles, obstacles, etc.). L'exemple typique de leur utilisation industrielle est la détermination de trajectoires géométriques pour l'assemblage et la manutention en environnement encombré dans une usine ou sur une chaîne de montage. On peut ainsi savoir dès la phase de conception amont si la planche de bord d'une automobile peut être montée en passant par la portière ou le pare-brise. Une excellente introduction à l'ensemble des techniques de planification de mouvements se trouve dans [LaValle, 2006], qui constitue une référence récente du domaine.

Les méthodes de planification les plus utilisées en robotique sont basées sur des algorithmes dits de Probabilistic RoadMaps (PRM). Ce sont eux qui fournissent les résultats les plus probants dans ce domaine. Énormément de travaux ont été publiés autour de ces algorithmes afin de les adapter à de nombreux cas spécifiques ou d'optimiser leur fonctionnement (graphe de visibilité, etc.). Des détails sur le fonctionnement de ces algorithmes peuvent être trouvés dans [Kavraki *et al.*, 1996] et [Svestka et Overmars, 1998, Barraquand *et al.*, 1997].

L'idée fondatrice des PRM est d'échantillonner aléatoirement l'espace libre et d'essayer de connecter entre eux les différents états à l'aide d'une méthode locale. Cette dernière permet de connecter deux configurations quelconques à coup sûr, à condition qu'il n'y ait pas d'obstacles. Du fait de la présence d'obstacles, toutes les configurations ne pourront être connectées entre elles. Le résultat de cette opération est une structure de graphe (RoadMap). Différentes simplifications peuvent être appliquées pour réduire la taille de cette structure. Une fois cette roadmap construite, il est possible d'effectuer très rapidement des requêtes de planification de mouvements différentes. Chaque requête permet de connecter des états finaux et initiaux quelconques grâce à des algorithmes de parcours de graphe bien connus (Dijkstra [Dijkstra, 1959], A*, etc.). La figure 1.8 illustre le fonctionnement de cet algorithme.

La nécessité de disposer d'une méthode locale pour le modèle de véhicule est la principale limitation de cet algorithme par ailleurs très performant. En effet, on ne connaît des méthodes locales que pour certaines catégories de systèmes relativement simples (linéaire, unicycle, modèle cinématique de véhicule, etc.) et plus généralement pour les systèmes différentiellement plats sans dérive (voir [Lamiroux et Laumond, 1997]). Les méthodes locales les plus connues sont les courbes de Reeds and Shepp pour les modèles cinématiques de véhicule ([Reeds et Shepp, 1990]). En l'absence d'obstacles, il existe toujours une combinaison

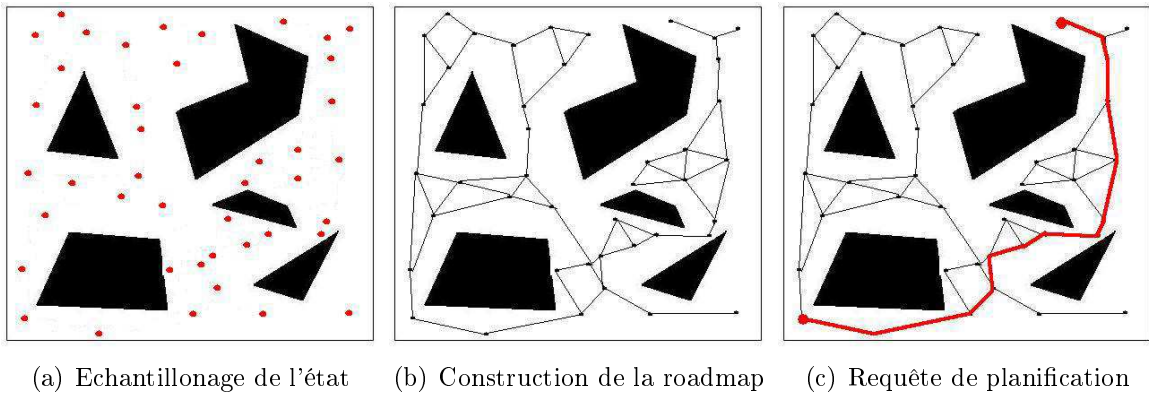


FIG. 1.8 – Illustration du fonctionnement d'un algorithme à base de probabilistic roadmaps

de ces courbes qui permet de relier deux configurations quelconques. Il existe des jeux de courbes plus complexes (clothoïdes) pour des modèles de véhicule à courbure continue (voir [Fraichard et Scheuer, 2004, Scheuer et Fraichard, 1997, Fraichard et Scheuer, 1994]). Il a été prouvé que l'algorithme PRM converge vers une solution lorsque le nombre d'états échantillonnés tend vers l'infini et que cette solution existe. Ces courbes sont très spécifiques à un modèle donné et ne peuvent être utilisées dans le cas où le modèle de véhicule est quelconque.

De nombreux autres algorithmes de planification de mouvements basés sur l'existence de cette méthode locale ne pourront être utilisés pour la même raison. Citons par exemple la méthode du fil d'Ariane [Ahuactzin, 1994] ou les méthodes à base de décomposition cellulaire. Les méthodes à base de champs de potentiel ([Rimon et Koditschek, 1992] et [Barraquand *et al.*, 1992, Khatib et Chatila, 1995]) sont plus adaptées aux robots ponctuels et aux grands espaces de configuration.

Dans le cas considéré dans cette thèse (système boîte noire au comportement dynamique non linéaire), il n'existe évidemment pas de méthode locale générique permettant de connecter deux états quelconques. Par contre, l'utilisation de cette technique sur un modèle simplifié de véhicule permet de générer des références de trajectoires qui pourront servir d'entrée aux automates ou aux modèles de conducteur du chapitre 1.5.2. Une autre limitation de ces méthodes est qu'elles ne génèrent pas des mouvements, mais des chemins géométriques possibles avec la cinématique donnée. Pour remonter aux entrées à partir de ce chemin, il faut paramétrer temporellement le chemin de manière à ce que la dynamique soit acceptable. C'est ce que l'on appelle le passage aux références dynamiques. Cette phase n'est pas nécessaire avec tous les algorithmes qui seront développés dans la suite de la thèse.

Les algorithmes utilisant des méthodes locales étant inutilisables, il faudra donc se tourner vers d'autres outils de planification de mouvements capables de gérer les systèmes dynamiques quelconques et uniquement disponibles sous forme de boîte noire. De par la complexité du problème, les algorithmes possibles sont bien moins nombreux et imposent certaines limitations.

1.7 Plan de l'étude

Le chapitre 2 présentera quelques éléments de dynamique du véhicule ainsi que l'implémentation informatique des modèles de calcul. Les chapitres 3, 4, 5 introduiront le fonctionnement

de trois algorithmes capables de résoudre les problèmes 1 et 2 (solution quelconque et limite de \mathcal{P}), ainsi que les résultats associés :

- Le chapitre 3 utilise des méthodes globales de planification de mouvements qui explorent exhaustivement tout l'espace d'état.
- Le chapitre 4 illustre la manière dont des outils de contrôle optimal ont été utilisés pour résoudre les questions 1 et 2.
- Le chapitre 5 utilise une méthode de déformation de trajectoire locale hybride entre planification et optimisation.

Le chapitre 6 présentera la résolution du problème numéro 3 (ensemble de solutions) par les trois algorithmes précédents. Finalement, le chapitre 7 synthétisera les différents résultats obtenus et les perspectives d'utilisation de ces outils.

Chapitre 2

Éléments de dynamique du véhicule

L'évaluation des qualités des algorithmes développés dans les chapitres suivants devra notamment se faire par la confrontation entre des résultats simulés et des mesures effectuées sur piste. L'objectif de ce chapitre est de présenter sommairement les éléments de physique et de technologie automobile qui seront nécessaires pour comprendre la finalité de cette étude et bien appréhender la suite (grandeurs comparées, robustesse, etc.). Une partie modélisation informatique détaillera la manière dont tous ces éléments ont été retranscrits logiciellement. Pour mémoire, on cite ici quelques références classiques du domaine de la physique et de la technologie de la Liaison Au Sol (LAS) : [Gillespie, 1992] , [Reimpell *et al.*, 2001], [Halconrui, 1995]. Ces références seront probablement utiles à qui souhaite explorer plus en profondeur ce domaine.

2.1 Généralités sur la liaison au sol

La liaison au sol (LAS) se définit comme la réunion de l'ensemble des systèmes permettant d'assurer le contrôle de la trajectoire du véhicule. Ce périmètre assez vaste comprend l'ensemble des organes suivants :

- Le système de direction
- Le système de freinage
- Les essieux (voir figure 2.1 et chapitre 2.1.2)
- Les pneumatiques et les roues métalliques (voir chapitre 2.1.4)
- Les amortisseurs, ressorts, butées, barres anti-dévers, etc. (voir chapitre 2.1.5)
- Les systèmes actifs : ESP, anti-dévers actif, suspension pilotées, etc. (voir chapitre 2.1.6)

Les grandeurs fréquemment rencontrées en dynamique du véhicule sont :

- L'angle volant A_{volant} ou le déplacement de la crémaillère y_n
- L'accélération latérale γ_{lat} et longitudinale γ_{longi}
- Les angles de roulis, tangage, lacet : θ, ϕ, ψ (voir figure 2.2)
- Les chargements verticaux appliqués par les pneus sur le sol : $F_{z1}, F_{z2}, F_{z3}, F_{z4}$
- Les pressions de freinage sur chaque roue : P_1, P_2, P_3, P_4
- Les dérives AV, AR et au centre de gravité (cdg) : $\delta_{AV}, \delta_{AR}, \delta_{cdg}$ (voir chapitre 2.1.1)
- La dérive des pneus δ_{pneu} (voir chapitre 2.1.4)

La figure 2.2 présente la définition des angles SAE de roulis θ , tangage ϕ , lacet ψ utilisés dans l'automobile.

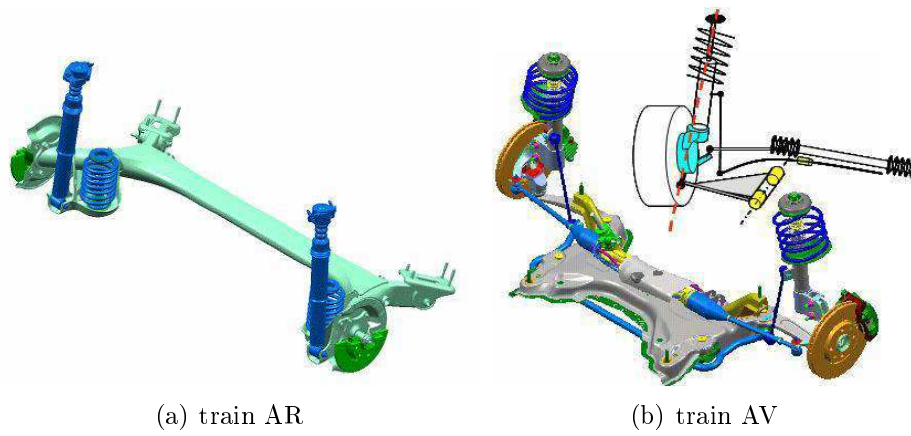


FIG. 2.1 – Exemple de train roulant à base de Pseudo Mac Pherson à l'avant et de traverse déformable à l'arrière

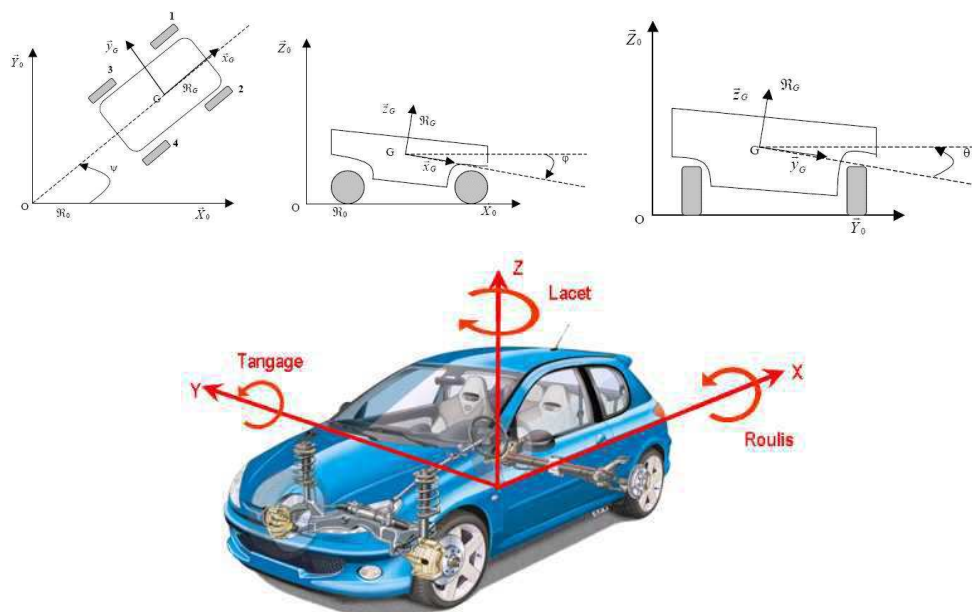


FIG. 2.2 – Définition des angles SAE (roulis θ , tangage ϕ , lacet ψ) utilisés dans l'automobile

2.1.1 Notions de dérive et de sur-braquage

Dérive

Dans le domaine de l'automobile, la dérive au centre de gravité (cdg) désigne l'angle formé entre le vecteur vitesse d'un véhicule et son axe longitudinal (voir figure 2.3). Cette définition est similaire pour les pneumatiques ou les trains du véhicule. Les dérives aux trains avant (AV) et arrière (AR) sont définies de la même manière, mais en un point situé au centre de chaque essieu. Ce sont des grandeurs particulièrement importantes dans la mesure où elles définissent le caractère sous-vireur ou sur-vireur d'un véhicule.

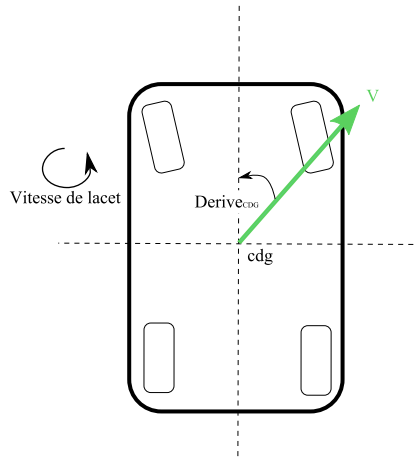


FIG. 2.3 – La dérive au cdg est l'angle entre le vecteur vitesse et l'axe longitudinal du véhicule

Soit R le rayon d'un cercle sur lequel est inscrit le véhicule à la vitesse V constante. On a alors $V = R \cdot \dot{\psi}$ et $\gamma_{lat} = \frac{V^2}{R}$. Les empattements E_{AV} et E_{AR} sont définis comme sur la figure 2.4 et $\dot{\psi}$ représente la vitesse de lacet du véhicule autour de l'axe vertical passant par son centre de gravité (cdg). Les dérives statiques aux trains AV, AR et au cdg sont alors liées par les relations suivantes :

$$\delta_{AV} = \delta_{cdg} - \frac{\dot{\psi} \cdot E_{AV}}{V} \quad (2.1)$$

$$\delta_{AR} = \delta_{cdg} + \frac{\dot{\psi} \cdot E_{AR}}{V} \quad (2.2)$$

On définit δ_{volant} comme étant le braquage des roues avant créé par la rotation du volant. δ_{Ack} est l'angle de braquage du train avant pour qu'un véhicule ne dérive ni de l'avant ni de l'arrière puisse suivre le cercle de rayon R à la vitesse V . Il a pour expression :

$$\delta_{Ack} = \frac{E_{AV} + E_{AR}}{R} = \frac{\dot{\psi} E}{V} = \delta_{AR} - \delta_{AV} \quad (2.3)$$

$\delta_{volant} = \delta_{Ack}$ lorsque la vitesse V tend vers 0. Dans ce cas là, le véhicule ne dérive plus et se comporte cinématiquement.

Sur-braquage statique

Le sur-braquage SB est défini comme étant :

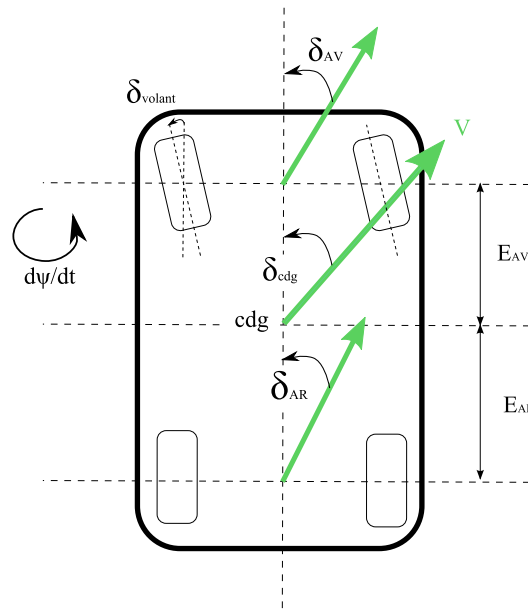


FIG. 2.4 – Les dérives aux trains AV et AR sont fonction de l’empattement et de la vitesse de lacet

$$SB = \delta_{volant} - \delta_{Ack} = \delta_{volant} + \delta_{AV} - \delta_{AR} \quad (2.4)$$

Si l’angle volant à appliquer pour suivre un cercle de rayon R à la vitesse V est supérieur à δ_{Ack} , le véhicule est dit sous-vireur. Dans le cas contraire, le véhicule est sur-vireur. Il est neutre si $\delta_{volant} = \delta_{Ack}$ quelque soit V . La figure 2.5 illustre bien cette problématique. Le sur-braquage est bien le braquage volant *supplémentaire* qu’il faut rajouter par rapport à un véhicule cinématique afin de rester sur le cercle de rayon R à vitesse V . Le caractère sur- ou sous-vireur d’un véhicule contribue à ce que l’on appelle son "typage".

Si le sur-braquage est positif, alors le véhicule est sous-vireur (sur-vireur si négatif). Un véhicule qui dérive plus du train avant que du train arrière est donc sous-vireur. En pratique, on évalue ce typage en comparant l’allure des courbes de dérive AV et AR. Afin que cette comparaison ait un sens, on reporte la valeur du braquage des roues AV δ_{volant} sur la valeur de dérive AV δ_{AV} . On obtient de cette manière deux courbes aisément comparables. Si la courbe de dérive AV est au-dessus de l’AR le véhicule est sous-vireur.

Le Taux de Sur-braquage (TSB) se définit comme étant :

$$TSB = \frac{SB}{\gamma^{lat}} \quad (2.5)$$

De la même manière, on peut représenter la valeur du rapport vitesse de lacet $\dot{\psi}$ sur braquage volant δ_{volant} en fonction de la vitesse sur le cercle et observer les différents typages de véhicule (voir figure 2.6).

2.1.2 Positionnement du plan de roue

Le plan de la roue possède une position et une orientation dans l’espace qui est fondamentale pour le comportement routier du véhicule. Les petites variations de son orientation conditionnent

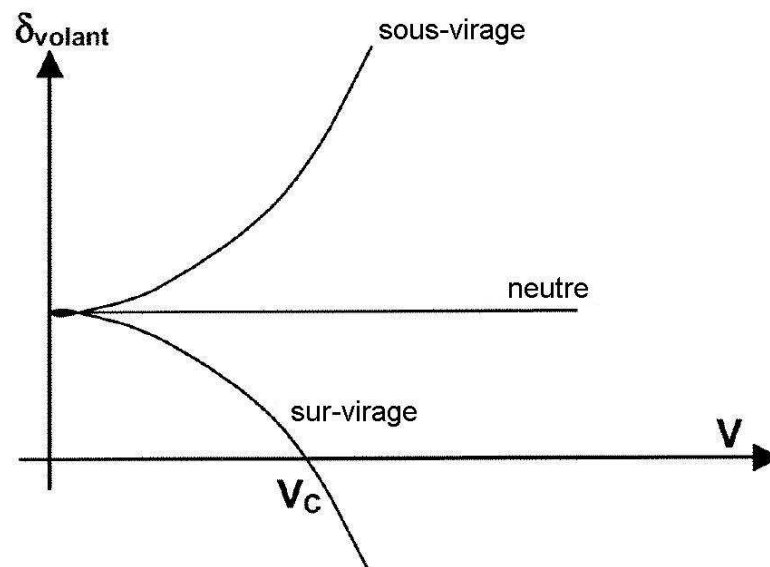


FIG. 2.5 – Braquage des roues permettant de rester sur un cercle de rayon R en fonction de la vitesse V du véhicule

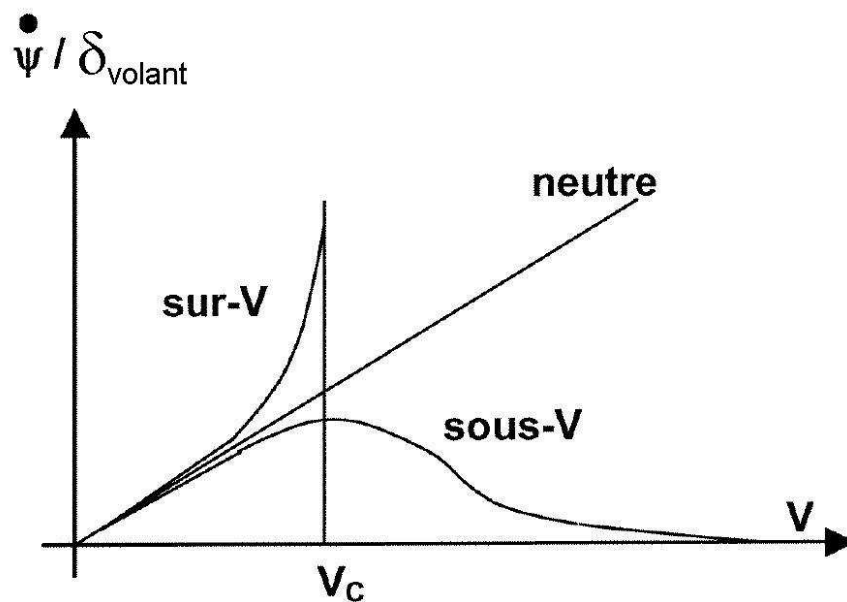


FIG. 2.6 – Détermination du type d'un véhicule via l'évolution du rapport $\dot{\psi} / \delta_{\text{volant}}$ en fonction de la vitesse V du véhicule sur le cercle de rayon R fixe

fortement la stabilité et le comportement du véhicule. La pince décrit la rotation des roues vers l'intérieur suivant l'axe vertical. Le carrossage est la rotation du plan de roue suivant l'axe longitudinal du véhicule (voir figure 2.7).

Cette position est fonction des débattements des suspensions et du déplacement de la crémaillère : ce sont les épures cinématiques. L'orientation de ce plan de roue est également fonction des efforts appliqués par le contact pneu-route sur la roue. Des déformations élastiques interviennent et déplacent ce plan. On regroupe généralement les deux notions sous le terme d'élasto-cinématique.

Ces variations influencent beaucoup le comportement du véhicule. Par exemple : augmenter la pince AV tend à favoriser la vivacité du véhicule tandis qu'une réduction de pince AV (ouverture) tend plutôt à le stabiliser. Les épures avant sont généralement conçues de manière à ouvrir au freinage pour sécuriser cette phase. De même, le comportement en virage est impacté par ces variations.

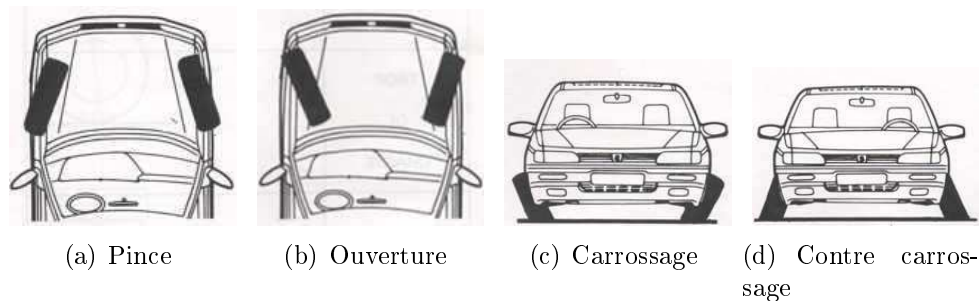


FIG. 2.7 – Définitions de la pince/ouverture et du carrossage positif/négatif

Dans une modélisation fonctionnelle (voir chapitre 2.4), ces mouvements du plan de roue sont décrits par des cartographies issues de calculs organiques, de mesures sur banc ou de choix de conception. Par exemple, le braquage cinématique d'une roue avant est décrit comme étant une fonction de la position crémaillère, du déplacement vertical de la roue courante ainsi que de celui de la roue opposée.

2.1.3 La caisse

La caisse a une influence très importante par ses paramètres de masse et d'inertie. Cette masse est appelée masse suspendue, car elle est en quelque sorte "portée" par la liaison au sol (LAS). La masse de la LAS est appelée masse non suspendue (MNS). Ces grandeurs conditionnent fortement la vitesse de lacet maximum ainsi que les différents reports de charge. De par la position actuelle des moteurs thermiques, le centre de gravité des véhicules du commerce est généralement positionné vers l'avant. La masse présente sur les pneus AV est plus élevée qu'à l'AR. Cela tend à rendre les véhicules plus sous-vireurs et donc plus sécurisants pour les conducteurs inexpérimentés.

2.1.4 Les pneumatiques

Si un véhicule dérive en virage, c'est parce que ses pneus dérivent. L'angle entre l'axe longitudinal du pneumatique et son vecteur vitesse est non nul : il y a un dérapage latéral. Cette

dérive génère une force transversale qui permet au véhicule de tourner (voir figure 2.9). Le comportement d'un pneumatique est modélisé à l'aide d'un ensemble d'équations analytiques communément dénommées "Formules magiques de Pacejka" (voir [Pacejka, 2002]). Ces formules permettent de modéliser ce comportement en longitudinal, transversal et en sollicitations couplées. A titre d'exemple, la formule donnant la force latérale pure F_y générée par le pneu en fonction de la dérive δ , de l'effort vertical F_z et du carrossage γ est :

$$F_y(\delta, F_z, \gamma) = D_y \cdot \sin[C_y \cdot \arctg(B_y \cdot \Phi_y)] + SV_y \quad (2.6)$$

avec :

$$\text{Valeur Pic : } D_y(F_z) = (a_1 \cdot F_z^2 + a_2 \cdot F_z)$$

$$\text{Facteur de forme : } C_y = a_0$$

$$\text{Rigidité de dérive : } BCD_y(F_z, \gamma) = [a_3 \cdot \sin(2 \cdot \arctg(F_z/a_4))] \cdot (1 - a_5 \cdot |\gamma|)$$

$$B_y(F_z, \gamma) = \frac{BCD_y}{C_y \cdot D_y}$$

$$\text{Facteur de rigidification : } \Phi_y(\delta, F_z, \gamma) = (1 - E_y) \cdot (\delta + SH_y) + E_y/B_y \cdot \arctg(B_y \cdot (\delta + SH_y))$$

$$\text{Facteur de courbure : } E_y(F_z) = \min((a_6 \cdot F_z + a_7); 1)$$

$$\text{Offset horizontal : } SH_y(F_z) = (a_8 \cdot \gamma + a_9 \cdot F_z + a_{10})$$

$$\text{Offset vertical : } SV_y(F_z, \gamma) = a_{12} \cdot F_z + a_{13} + (a_{112} \cdot F_z^2 + a_{111} \cdot F_z) \cdot \gamma$$

Ces formules sont issues d'une caractérisation expérimentale et ne proposent pas une modélisation physique du comportement du pneu. A quelques exceptions près, les paramètres a_{xxx} de ces formules décrivent uniquement l'allure des courbes représentatives des efforts et moments renvoyés par le pneumatique. Ils n'ont pas de sens physique.

Ces paramètres sont identifiés à partir de mesures effectuées sur un banc dédié (FlatTrac). Les équations du modèle de Pacejka sont nombreuses, complexes et peu intuitives, mais ce modèle représente une bonne approximation du comportement du pneumatique dans tous les domaines de la dynamique route.

La figure 2.9 montre l'existence d'une saturation de la force transversale F_y lorsque la dérive δ augmente. Ces phénomènes de saturation expliquent en partie l'existence même d'une limite à \mathcal{P} pour beaucoup de tests. On appelle rigidité de dérive K_y , la pente à l'origine de la courbe $F_y = f(\delta)$. Ce paramètre décrit l'intensité avec laquelle le pneu génère la force qui permet de tourner (voir figure 2.9). L'adhérence maximum définit quant à elle le maximum de force que peut créer le pneu. La rigidité de dérive K_y et l'adhérence maximale sont fonction de la charge verticale F_z appliquée aux pneumatiques (voir figure 2.8).

Les efforts au pneu sont donc fonction des différents reports de charge qui peuvent avoir lieu lorsqu'un véhicule manœuvre. Les phénomènes transitoires sont modélisés via l'introduction des longueurs de relaxation τ_x et τ_y .

2.1.5 Ressorts, amortisseurs et barres anti-dévers

Les ressorts et amortisseurs sont modélisés de manière fonctionnelle par la donnée de leur loi effort/déplacement ou loi effort/vitesse. Les barres anti-dévers permettent de contrer le roulis en virage et assurent ainsi un meilleur confort aux passagers. En contrepartie, des variations de charge parfois défavorables viennent affecter la tenue de route. Les raideurs torsionnelles de ces barres influent sur les reports de charge et donc sur la dynamique route.

A somme constante, la répartition des raideurs entre l'AV et l'AR influe sur le comportement sous-vireur ou sur-vireur d'un véhicule. Il est généralement admis que plus un véhicule est raide

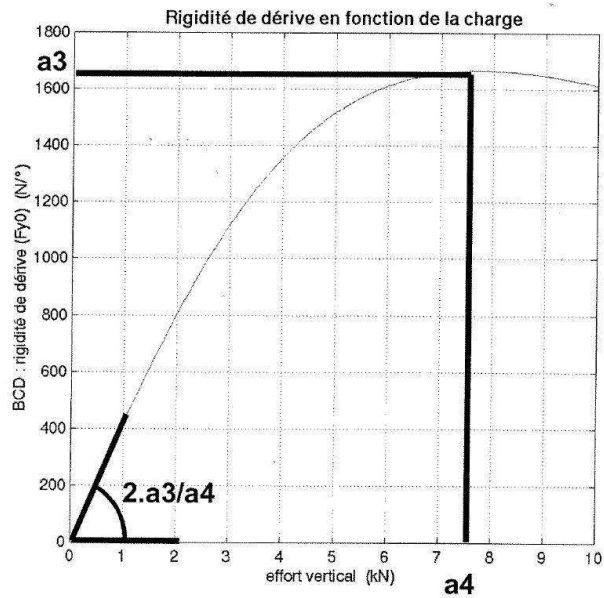


FIG. 2.8 – Evolution de la rigidité de dérive K_y d'un pneu en fonction de la charge verticale F_z

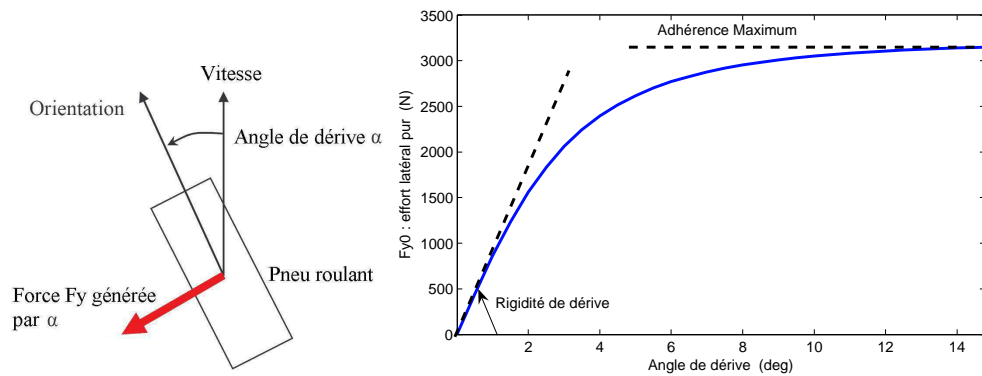


FIG. 2.9 – Effort transversal F_y généré par le pneu à partir de sa dérive δ_{pneu}

à l'AV, plus il glissera de l'AV et plus il sera sous-vireur (voir définitions au chapitre 2.1.1). De même, raidir la barre anti-dévers à l'AR d'un véhicule revient à accentuer son caractère sur-vireur. Il en va de même pour les raideurs des ressorts. Il est ainsi possible de changer le typage d'un véhicule en jouant sur ces grandeurs.

2.1.6 Electronic Stability Program (ESP)

L'Electronic Stability Program (ESP), aussi appelé ESC pour Electronic Stability Control est un équipement de sécurité active destiné à améliorer le contrôle de trajectoire d'un véhicule automobile. L'équipementier automobile allemand Bosch fut le premier à développer le système ESP qui fut utilisé d'abord par Mercedes-Benz et BMW en 1995 avant de se généraliser sur d'autres marques et de descendre sur d'autres gammes ([Bosch, 2001]). Depuis la première version, qui ne comprenait qu'un système de contrôle dynamique de trajectoire, l'ESP a beaucoup évolué. Ce terme regroupe souvent l'ensemble des systèmes électroniques de gestion (anti-blocage ABS, anti-patinage ASR, aide au freinage d'urgence AFU, contrôle dynamique de stabilité DSC, répartition électronique de freinage REF, etc.)

Le contrôle dynamique de trajectoire qui est une des fonctions du bloc ESP permet de corriger la trajectoire en agissant sur le système de freinage et l'injection d'essence au moteur. Son fonctionnement est simple ; le véhicule lors d'un virage pris à trop grande vitesse peut perdre de l'adhérence :

- Soit sous-virer : le train avant glisse et l'automobile va tout droit au lieu de suivre la courbe.
- Soit sur-virer : le train arrière décroche et si la trajectoire n'est pas rétablie la voiture part en tête-à-queue.

Ces phénomènes sont décelés par des capteurs placés au niveau des trains roulants et de la direction. Des calculateurs vérifient que la vitesse des roues avant est bien la même que celle des roues arrière. Ils s'occupent aussi de comparer que la direction voulue par le conducteur (volant) est en adéquation avec celle du véhicule (capteur de roue). Donc si une situation anormale se produit, la trajectoire peut être corrigée par une action ciblée sur l'injection et sur le freinage.

La figure 2.10 présente le schéma général de fonctionnement d'un ESP avec l'ensemble des capteurs et actionneurs nécessaires à la réalisation de la fonction de correction de trajectoire d'un ESP. Les figures 2.11 et 2.12 présentent un exemple de correction appliquée sur chaque roue par l'ESP afin de stabiliser le véhicule. Les pressions de freinage appliquées sont loin d'être négligeables et influent fortement sur le comportement dynamique du véhicule (alors même que le conducteur ne touche pas à la pédale de frein).

2.1.7 La répartition de freinage

En freinage, on cherche souvent à faire en sorte que les pneus avant et arrière travaillent avec la même adhérence en dépit du report de charge ΔF_z (de l'AR vers l'AV). Celui-ci vaut $\Delta F_z = \frac{M \cdot \gamma_{longi} \cdot H}{E}$ où M est la masse, γ_{longi} l'accélération longitudinale, H la hauteur du cdg et E l'empattement du véhicule. La puissance du freinage AV doit donc être supérieure à celle de l'arrière sous peine de déstabiliser le véhicule même lors d'un freinage en ligne droite. Cette fonction de répartition était souvent assurée par des systèmes mécaniques. Aujourd'hui c'est l'ESP qui assure cette fonction électroniquement (via le pilotage de valves). Ces systèmes électroniques plus fins permettent de répartir le freinage non seulement entre l'AV et l'AR, mais individuellement sur chaque roue afin de stabiliser le véhicule notamment lors d'un freinage en courbe.

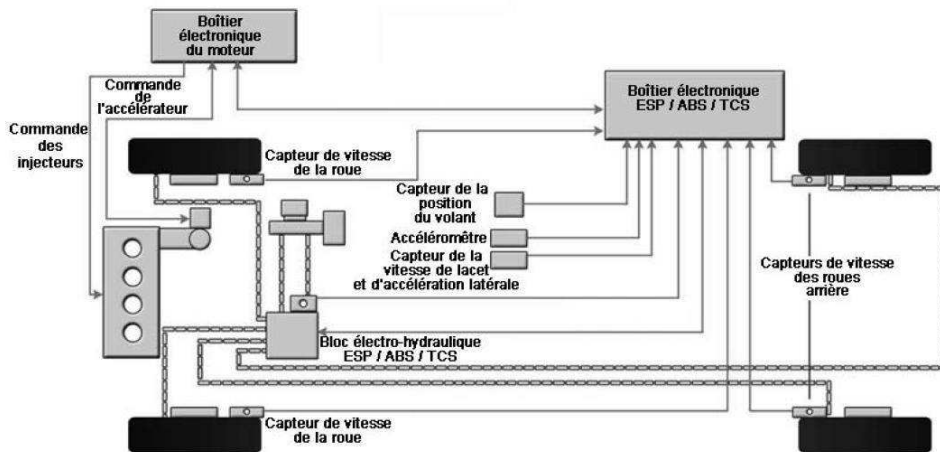


FIG. 2.10 – Schéma général des actionneurs et capteurs de l'ESP

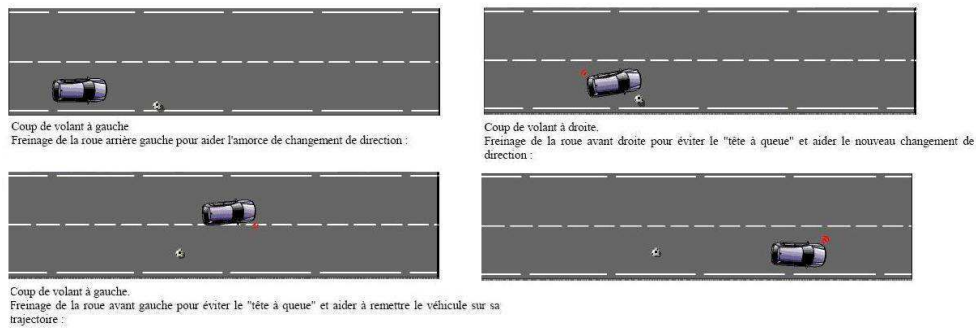


FIG. 2.11 – Action de l'ESP lors d'un évitement

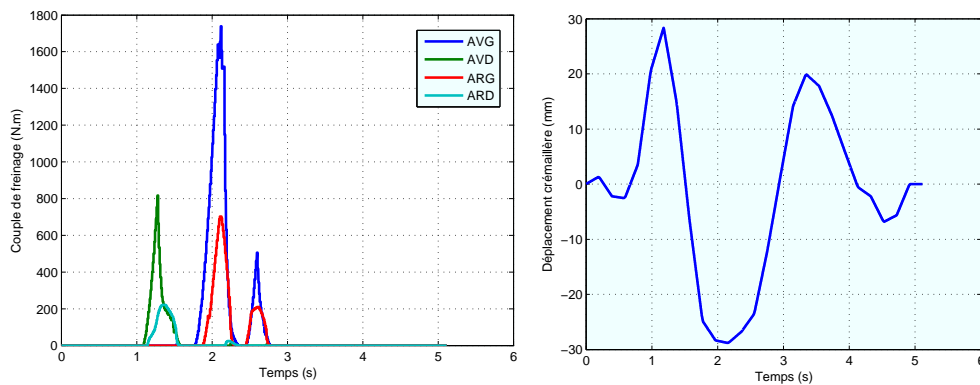


FIG. 2.12 – Exemple de pressions de freinage fournies par un ESP lors d'un test à 50 km/h

2.2 Les véhicules de référence de cette étude

Les caractéristiques des différents véhicules qui serviront de référence tout au long de cet ouvrage sont détaillées ci-dessous. La figure 2.13 illustre les différents types de véhicules présents ainsi que leurs dimensions. Certains des éléments présentés ici pourront être modifiés afin de réaliser les études de sensibilité et de robustesse des chapitres suivants. Ces véhicules ont été choisis de manière à couvrir un certain éventail de véhicules légers du groupe PSA Peugeot Citroën :

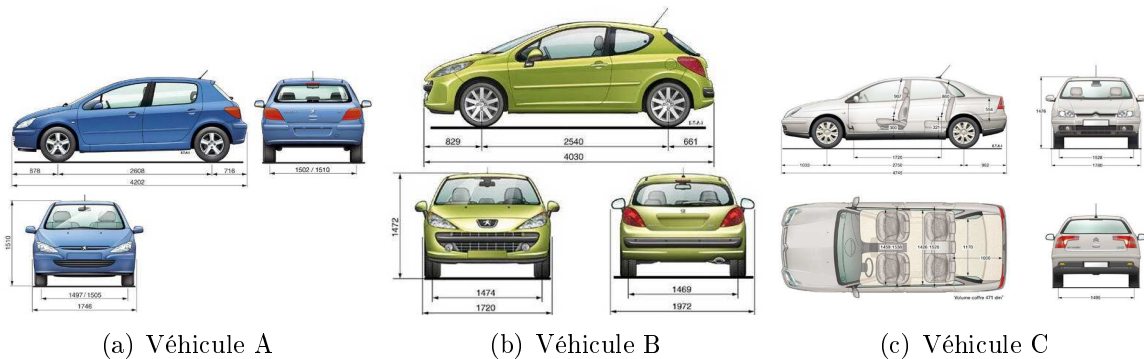


FIG. 2.13 – Les véhicules de référence de cette étude (307, 207, C5)

Véhicule A : Le véhicule A est un véhicule de milieu de gamme type Peugeot 307. La version de référence est une 5 portes 2 l essence. La masse de la caisse avec moteur est de 1250 kg pour un chargement dit 2P14 (2 personnes à l'avant de 70 kg et 14 kg de bagages dans le coffre). Les dimensions de sa carrosserie hors rétroviseurs sont de 4.2 m × 1.73 m.

Véhicule B : Le véhicule B est un véhicule plus petit à vocation citadine type Peugeot 207. La version de référence est la 5 portes 1.6 l HDi. Ses dimensions un peu plus réduites sont de 4.005 m × 1.708 m. La masse de la caisse est de 1275 kg en chargement 2P14. Les déclinaisons Diesel, Essence et Sport sont disponibles et seront étudiées par la suite.

Véhicule C : Le véhicule C est une berline familiale plus lourde et plus grande de type Citroën C5. C'est le seul véhicule pour lequel un modèle d'ESP soit disponible. Sa carrosserie plus imposante mesure 4.775 m × 1.83 m. La masse de la caisse est de 1504 kg.

2.3 Sécurité active et tests standardisés

2.3.1 La notion de sécurité active

Dans le domaine automobile, la sécurité active (ou primaire) est définie comme étant l'ensemble des éléments mis en jeu dans un véhicule afin de permettre à un conducteur d'éviter l'accident. Cette notion inclut tous les dispositifs électroniques d'assistance à la conduite tels que l'ESP (décrit au chapitre 2.1.6) ou les dispositifs d'avertissement (franchissement de ligne blanche, présence d'un véhicule dans l'angle mort, freinage du véhicule précédent, etc.). Cependant, c'est la base mécanique passive du véhicule qui joue le plus grand rôle pour cette prestation. Celle-ci doit fournir un comportement routier sain au véhicule pour permettre au conducteur de garder la maîtrise de sa trajectoire en toutes circonstances.

La sécurité passive (ou secondaire) se définit comme étant l'ensemble des moyens mis en œuvre permettant de minimiser les conséquences d'un accident une fois que celui-ci a eu lieu (châssis déformable, airbags, etc.). Ces performances sont évaluées au cours de crash tests tels que ceux pratiqués par l'EuroNCAP. Cette notion de sécurité passive peut être mesurée objectivement grâce à des tests standards issus des données d'accidentologie de la sécurité routière. Ces tests sont purement mécaniques et aisément reproductibles. Ils permettent une notation cohérente et objective du comportement du véhicule en cas de choc.

L'évaluation de la sécurité active présente quant à elle un caractère singulièrement différent, dans la mesure où une trajectoire dépend à la fois des qualités dynamiques du véhicule et du comportement du conducteur. Afin de tenter d'obtenir une notation cohérente des qualités de sécurité active de différents véhicules, des tests inspirés de situations de vie réelles ont été imaginés et mis en place par des essayeurs. Les situations retenues simulent par exemple un évitement en urgence d'un obstacle arrivant inopinément sur la route ou un freinage effectué lors d'un virage. L'interprétation de ces tests est toujours délicate.

Au fil du temps, certains de ces tests ont fini par servir de référence et ont été standardisés par l'International Organization for Standardization (ISO). Il existe plusieurs catégories de tests effectués par les essayeurs suivant la prestation que l'on cherche à évaluer ou le défaut que l'on souhaite mettre en valeur. Les tests les plus classiques sont les tests d'évitement. Ils simulent une manœuvre d'évasion effectuée en urgence suite à l'apparition d'un obstacle imprévu sur la trajectoire. Les dimensions ont été fixées d'après les retours d'expériences et les données d'accidentologie (voir [Fenaux, 2005]). Leur définition a une grande importance sur les résultats. Par exemple, la prise en compte ou non des paramètres de largeur/longueur du véhicule est fondamentale vis-à-vis des résultats qui sont attendus.

La sécurité active se préoccupe de permettre à un véhicule de rester sur la route ou d'éviter des obstacles imprévus. Les exemples d'application des algorithmes décrits dans la suite seront naturellement tirés de ce domaine privilégié.

2.3.2 Le test d'évitement dit "VDA"

Définition

Le test d'évitement VDA est le plus classique et le plus médiatisé des tests d'évitement. Il consiste en une succession de trois couloirs décalés les uns par rapport aux autres (voir figure 2.14). Le but est de traverser ce test à la vitesse la plus élevée possible sans toucher les plots délimitant les couloirs. Les vitesses de passage maximum V_{max} relevées avec la plupart des voitures particulières de grande série se situent entre 60 *km/h* et 75 *km/h*. Dans le cadre des tests d'évitement, on a $\mathcal{P} = \text{Vitesse initiale}$.

Cet essai est parfois dénommé "test de l'élan" en référence à son utilisation historique en Suède. Dans ce pays, cette manœuvre avait pour but d'évaluer la capacité d'un véhicule à éviter en urgence un élan (l'animal) se trouvant sur une route. Le conducteur entre dans le premier couloir à une vitesse V donnée en levant le pied de l'accélérateur. Par conséquent, la seule commande disponible pour l'essayeur au cours du test est l'angle volant qu'il doit ajuster au mieux afin de passer le test.

La conséquence directe de ce type de procédure est une certaine dépendance des résultats vis-à-vis des compétences de l'essayeur. Seuls des essayeurs expérimentés et spécialisés dans

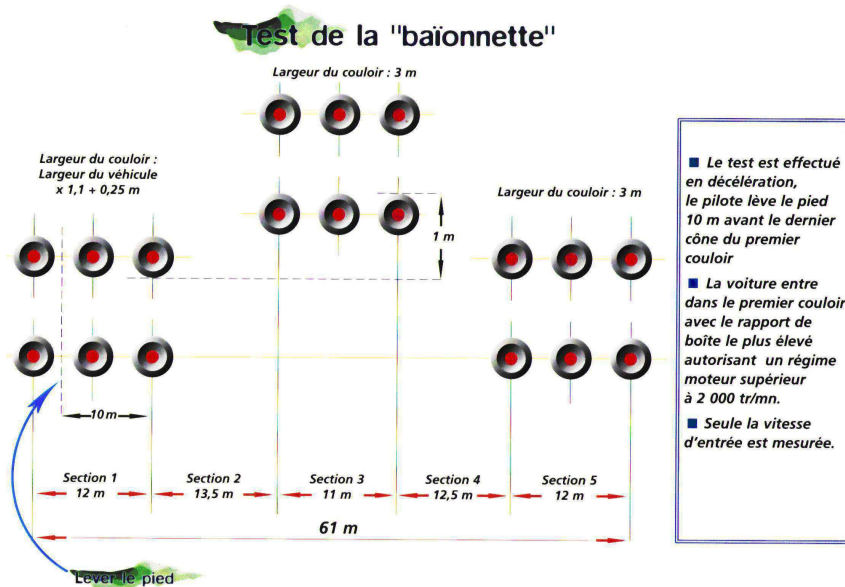


FIG. 2.14 – Description géométrique du test VDA standard (L=largeur du véhicule hors rétroviseurs)

le domaine savent vraiment utiliser le potentiel d'une voiture. Ils doivent également pouvoir reproduire trois fois de suite les solutions trouvées de manière à valider la vitesse de passage.

La figure 2.17 donne l'allure d'un angle volant solution pour un test VDA. Les amplitudes maximum relevées par les essayeurs sur un VDA sont de l'ordre de ± 200 deg pour l'angle volant, ± 800 deg/s pour la vitesse volant et ± 5000 deg/s² pour l'accélération volant. Les couples volant mesurés sont au maximum de 20 ou 30 N.m. La limite physiologique de vitesse volant pour un être humain est de l'ordre de 1500 deg/s en pic mais ces vitesses ne sont généralement atteintes que lors d'un évitement ville (voir chapitre 2.3.3).

L'angle volant de la figure 2.17 présente trois pics caractéristiques qui correspondent aux trois coups de volant nécessaires pour :

- Lancer le véhicule dans le deuxième couloir (1^{er} pic)
- Stabiliser le véhicule pour franchir le deuxième couloir (2^{me} pic)
- Ramener le véhicule dans le troisième couloir (3^{me} pic)
- Puis retour au neutre pour stabiliser le véhicule dans le troisième couloir et sortir du test en ligne droite

La figure 2.15 présente quelques exemples d'évolution de grandeurs physiques caractéristiques lors d'un test VDA (vitesse de lacet, accélération latérale).

Les détails de la procédure peuvent être trouvés dans [ISO, 1999] et [ISO, 2002]. Les conditions de passage dans le VDA sont très strictement encadrées par cette norme (chargement, plein d'essence, température des pneus, etc.). Différents cas de charge du véhicule sont essayés pour chaque véhicule. Ce test sollicite très fortement le véhicule en transversal et vise à le pousser à ses limites dans ce domaine. Il peut donc s'avérer relativement dangereux de par les fortes accélérations transversales subies. Les essayeurs s'équipent d'un casque pour se protéger en cas d'éventuel retournement du véhicule. La figure 2.16 illustre bien l'amplitude des phénomènes dynamiques présents lors du franchissement correct d'un test VDA.

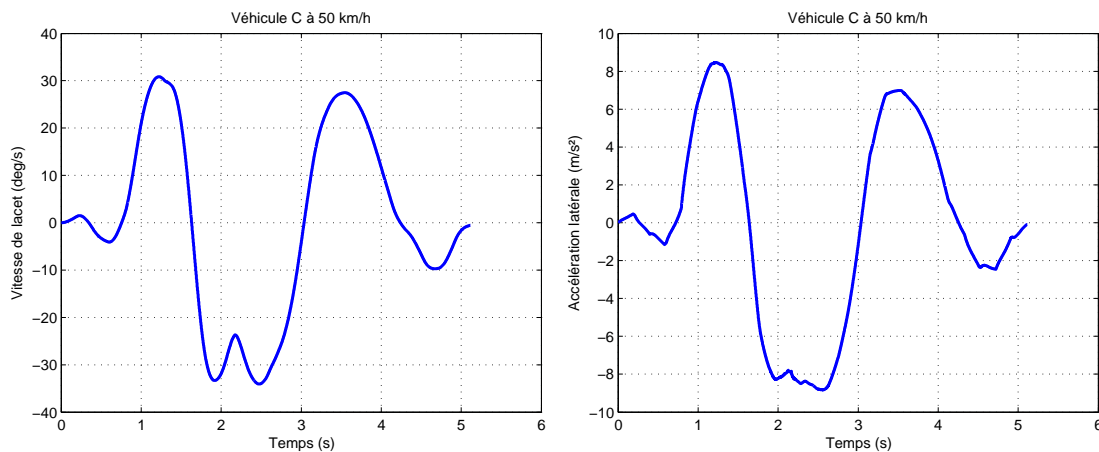


FIG. 2.15 – Exemple d'évolution de grandeurs physiques classiques lors d'un VDA effectué avec le véhicule C

La norme prévoit que la largeur des différents couloirs du test VDA s'adapte à la largeur L du véhicule (hors rétroviseurs). Par exemple, la largeur du premier couloir est de $1.1 \times L + 0.25 \text{ m}$. Le détail des autres couloirs se trouve sur la figure 2.14. D'autres tests consommateurs prévoient d'adapter également la géométrie des couloirs à la longueur du véhicule.

Il existe de nombreuses autres variantes de tests d'évitement qui diffèrent suivant les pays, les constructeurs et les organismes les définissant. Les procédures de test sont assez variées. Elles se distinguent par leurs dimensions (et donc l'ordre de grandeur de la vitesse d'entrée dans le test), leur manière de s'adapter à la géométrie du véhicule et l'objectif poursuivi par le test (stabilité par rapport au retournement pour les 4X4, capacité d'évitement, etc.).

Parmi tous ces tests d'évitement, citons notamment le Consumer Union Short Course, le Ford PCL Lane Change et le Open-Loop Pseudo Double Lane Change. L'organisme NHTSA aux Etats-Unis est très impliqué dans la définition et la validation de ces procédures. Chacun de ces tests présente des qualités et des défauts assez variables en termes de réalisme, de répétabilité, d'objectivité et de capacité de discrimination. Une comparaison de ces différents tests est disponible dans [O'Hara et Robert, 2005]. Chez le constructeur PSA Peugeot Citroën, le test de référence est le test VDA. C'est principalement à lui que nous nous intéresserons au cours de cette thèse même si la généralité des techniques employées permet de traiter facilement tout autre test.

A titre indicatif, la figure 2.18 présente un exemple de grille d'évaluation des performances d'un véhicule sur un test VDA en fonction de sa vitesse de passage maximale. La figure 2.19 présente l'évolution des angles volant lors de la réalisation de la procédure de recherche de vitesse de passage maximale. Il sera par la suite intéressant de comparer ces mesures aux résultats que nous obtiendrons en simulation. Les grandeurs analysées lors d'un test VDA sont principalement la vitesse de passage maximale, le pic de vitesse de lacet, les dérives aux essieux et le sur-braquage dynamique. La définition de cette dernière grandeur est explicitée dans le paragraphe suivant

Sur-braquage dynamique

Le sur-braquage statique vu au chapitre 2.1.1 n'a un sens que pour un véhicule en virage stabilisé. Afin de généraliser cette notion sur le test VDA, on utilise ce que l'on appelle le sur-

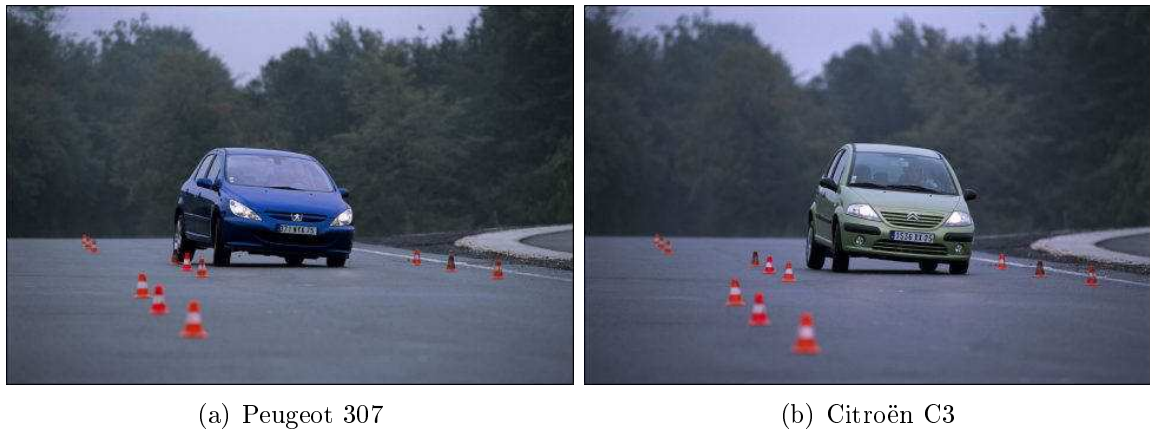


FIG. 2.16 – Les sollicitations dynamiques lors d'un test VDA à 70 km/h sont fortes

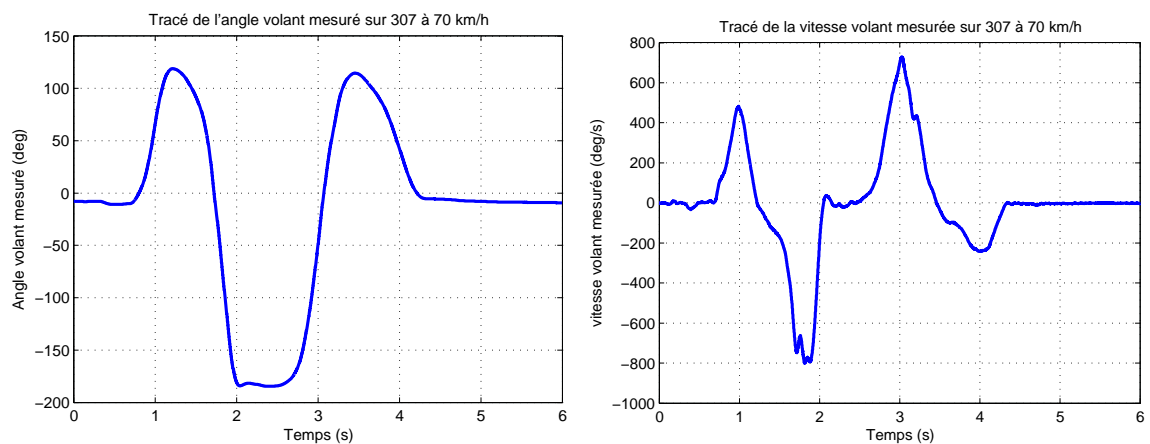


FIG. 2.17 – Allure générale de l'angle et de la vitesse volant permettant à un essayeur de passer un test VDA avec le véhicule A à 70 km/h



FIG. 2.18 – Evaluation des performances d'un véhicule au test VDA en fonction de sa vitesse de passage

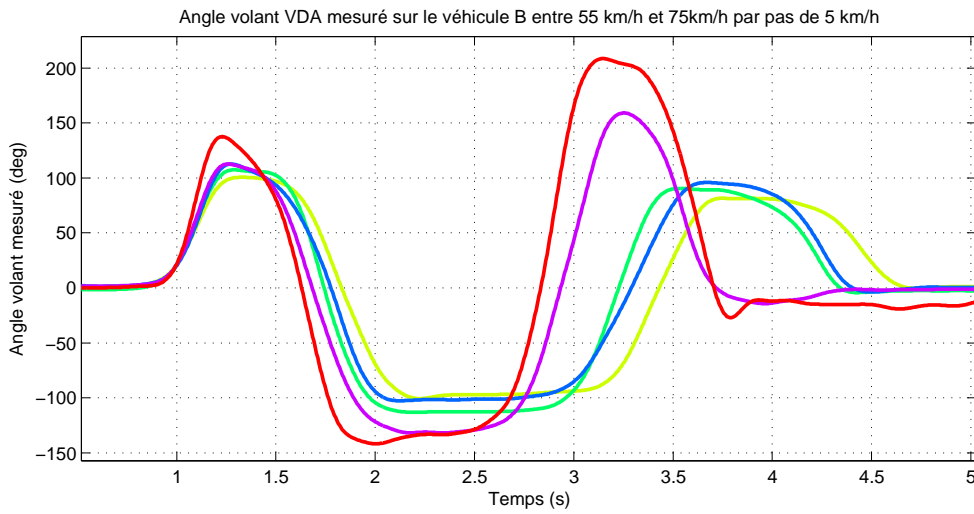


FIG. 2.19 – Angles volant mesurés sur le véhicule B lors de la procédure de recherche de vitesse maximum

braquage dynamique. Cette grandeur représente la différence entre les maxima de dérives AV et AR qui sont pris à des instants différents. La figure 2.20 illustre ce calcul. Un même véhicule peut-être dynamiquement sur ou sous-vireur suivant la vitesse de passage considérée dans le VDA, tandis qu'il est toujours sous-vireur statiquement. On rappelle bien que sur ces figures, le braquage dû à la rotation des roues avant δ_{volant} est reporté sur la valeur de l'angle de dérive AV δ_{AV} .

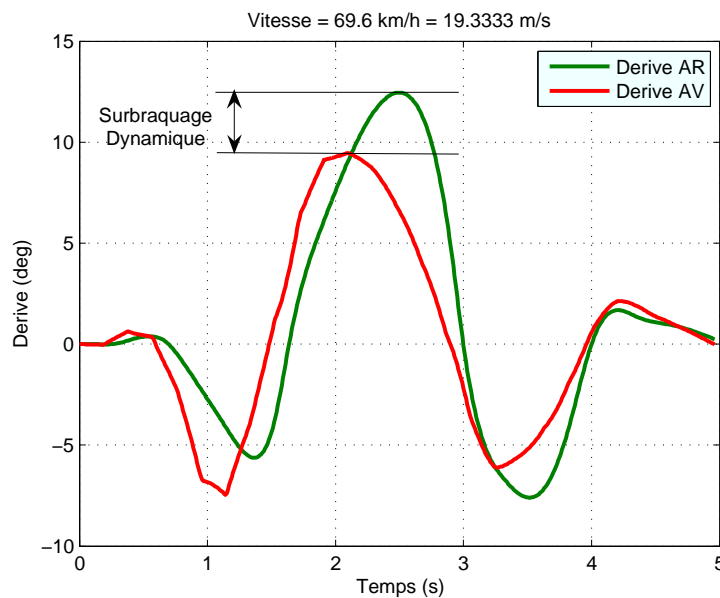


FIG. 2.20 – Calcul du sur-braquage dynamique

Afin de voir l'allure générale des dérives, les figures 2.21, 2.22, 2.23 présentent différents exemples de mesures effectuées avec le véhicule B au cours d'un test VDA réel. Les trois cas sur-vireur/sous-vireur/neutre y sont visualisables. Ces courbes présentent les trois pics corres-

pendants aux trois couloirs du test VDA, le deuxième pic présente généralement les amplitudes les plus fortes.

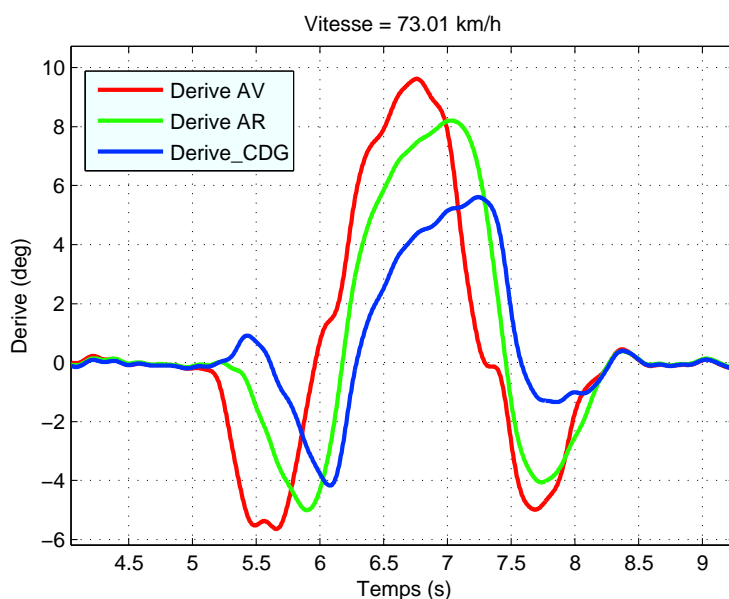


FIG. 2.21 – Mesures des dérives (AV, AR et CDG) au cours d'un test d'évitement VDA sur le véhicule B à 73 km/h. Ce passage est sous-vireur (l'AV dérive plus que l'AR sur le deuxième pic)

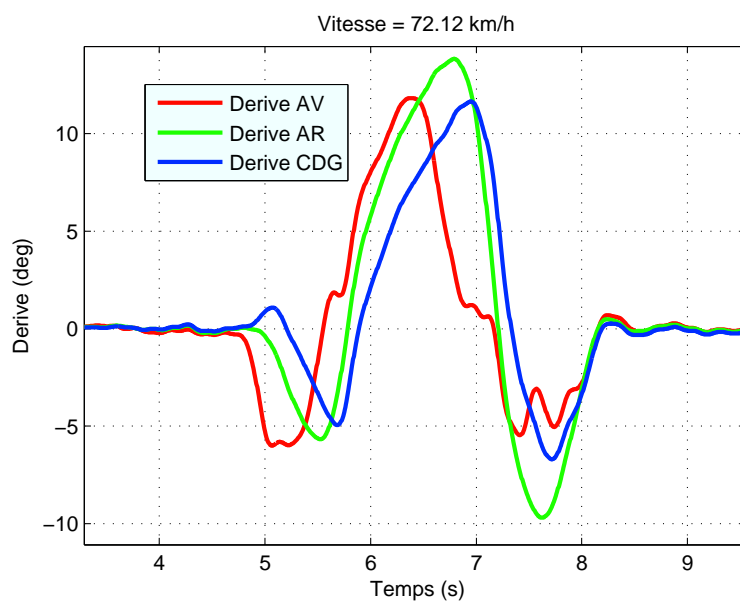


FIG. 2.22 – Mesures des dérives (AV, AR et CDG) au cours d'un test d'évitement VDA sur le véhicule B à 72 km/h. Ce passage est sur-vireur (l'AR dérive plus que l'AV sur le deuxième pic)

La figure 2.24 présente le résultat de mesures de sur-braquage dynamique effectuées par des essayeurs sur le véhicule B et où l'on constate l'existence d'un "mur" de surbraquage dynamique qui empêche le franchissement du test VDA à des vitesses supérieures. Suivant les caractéristiques

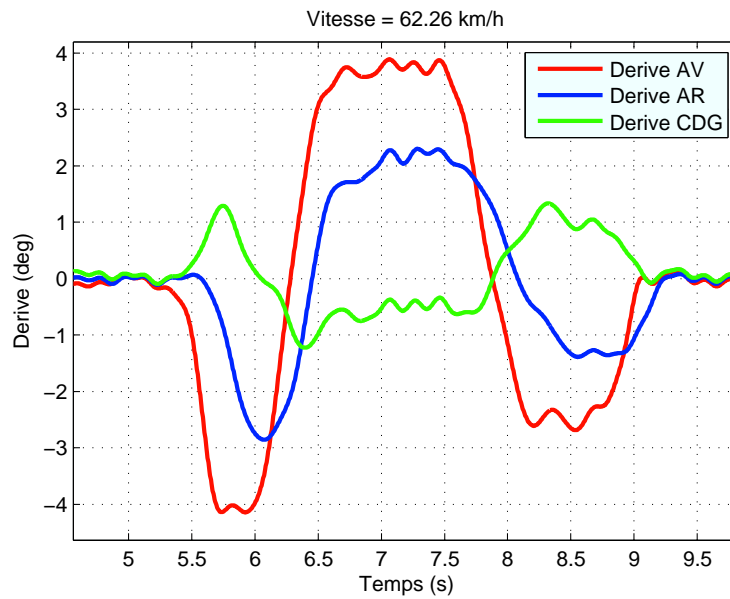


FIG. 2.23 – Mesures des dérives (AV, AR et CDG) au cours d'un test d'évitement VDA sur le véhicule B à faible vitesse (62 km/h). Les dérives sont en phase et de faibles amplitudes

du véhicule, ce "mur" peut être orienté vers le haut (sous-virage dynamique car $\delta_{AV} > \delta_{AR}$ sur le deuxième pic) ou comme ici vers le bas (sur-virage dynamique car $\delta_{AR} > \delta_{AV}$ sur le deuxième pic).

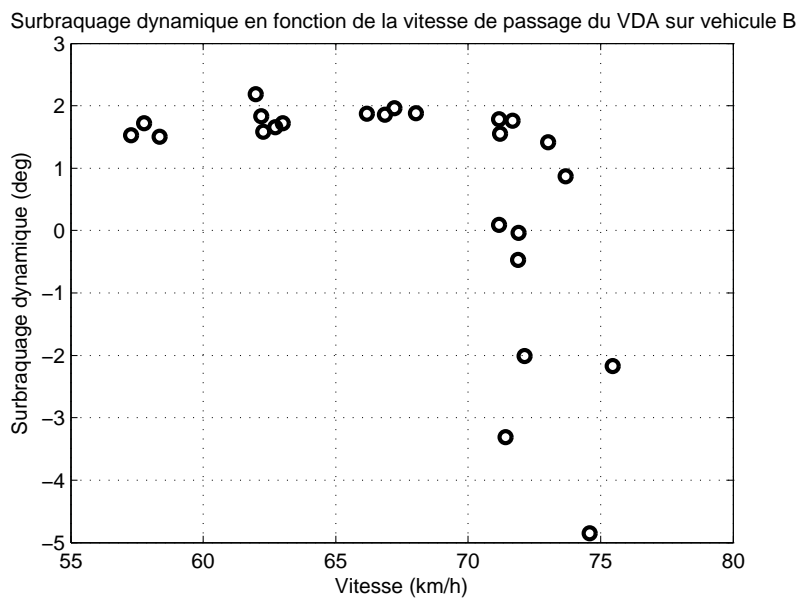


FIG. 2.24 – Mesures réelles de sur-braquage dynamique lors de manœuvres VDA réalisées avec le véhicule B

La comparaison de différents véhicules présente une diversité de comportements en termes de SB dynamique. La figure 2.25 illustre bien cet état de fait. Le "mur" du sur-braquage est croissant ou décroissant suivant le "typage dynamique" du véhicule.

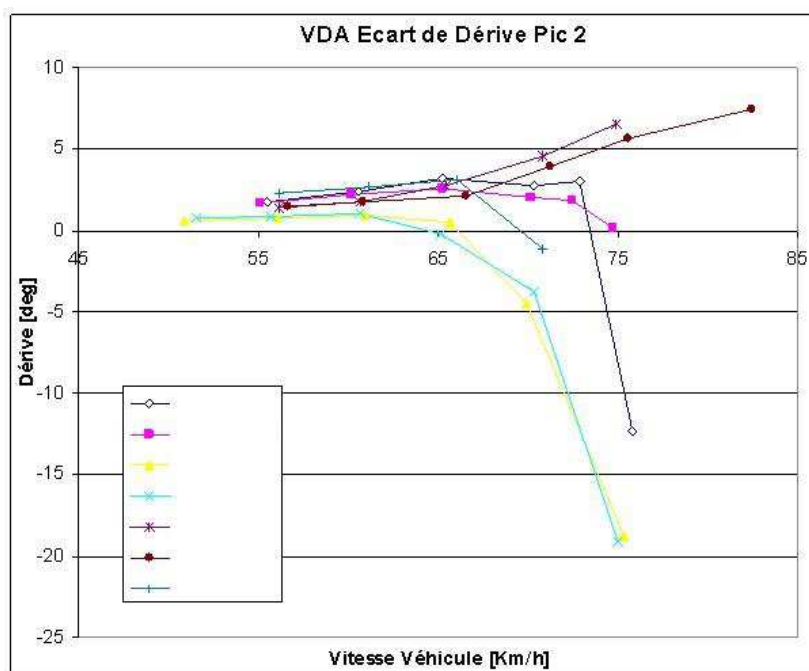


FIG. 2.25 – Comparaisons de mesures de sur-braquage dynamique pour différents modèles de véhicules

2.3.3 Le test d'évitement "ville"

Ce test est similaire au VDA de par sa géométrie et sa procédure d'essai. L'idée est de simuler un véhicule en milieu urbain en train d'éviter un obstacle de faibles dimensions (piéton traversant inopinément la route, portière qui s'ouvre). La vitesse et les cotes des couloirs ont été adaptées en conséquence (voir figure 2.26). L'ordre de grandeur de la vitesse de passage est entre 35 et 60 *km/h*. La figure 2.27 présente l'allure d'un angle volant typique lors d'un évitement ville.

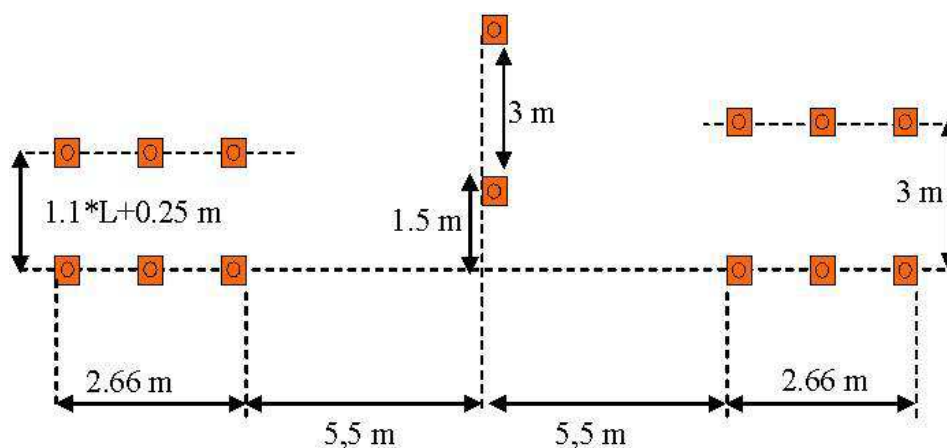


FIG. 2.26 – Description géométrique du test de l'évitement ville (L = largeur du véhicule hors rétroviseurs)

La principale utilisation de ce test est la mise au point des systèmes de direction, car les vitesses de rotation volant y sont très élevées (jusqu'à 1500 *deg/s*).

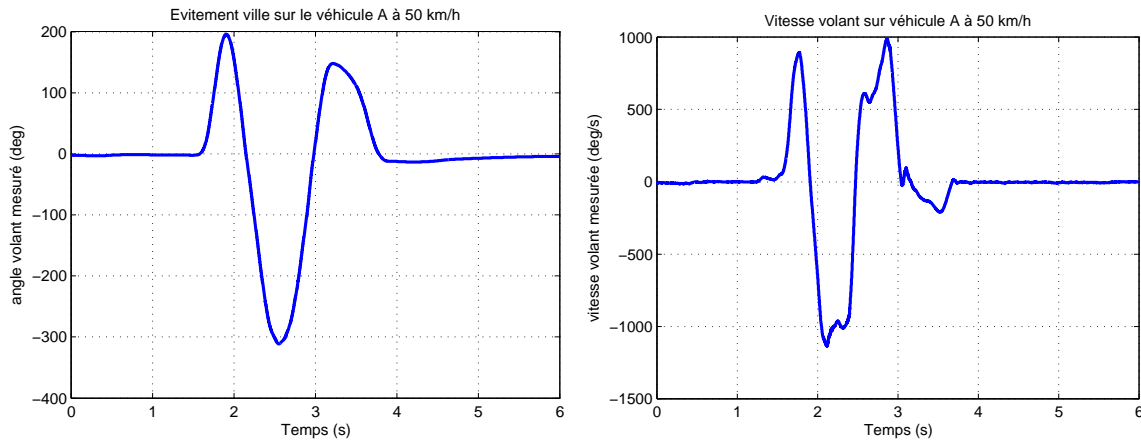


FIG. 2.27 – Angle volant et vitesse volant mesurés lors du franchissement d'un test d'évitement ville à 50 km/h sur une Peugeot 307

2.3.4 Freinage en courbe

Lors de ce test, le véhicule est lancé à 125 km/h sur un cercle de 180 m de rayon et doit effectuer un freinage d'une seconde (voir figure 2.29). L'essayeur doit maintenir un angle volant constant correspondant à l'état stabilisé du véhicule sur le cercle (soit une accélération latérale d'environ 6.7 m/s^2). Différents passages sont effectués avec une intensité de freinage $\mathcal{P} = P_{\text{freinage}}$ croissante, jusqu'au blocage d'une des roues ou jusqu'à intervention de l'ABS. Les détails de la procédure peuvent être trouvés dans le document [ISO, 2006]. Ce type de freinage a pour conséquence quasi inévitable une forte déstabilisation du véhicule qui sur-virera ou sous-virera lors du test. Les essayeurs regardent alors la variation de rayon de la trajectoire induite par le freinage, le pic de vitesse de lacet et le comportement du véhicule (sur ou sous-vireur). Un autre test très proche de ce dernier est le test dit du "lever de pied en courbe". Celui-ci consiste à lâcher brutalement l'accélérateur dans un virage ce qui revient à effectuer un freinage en courbe plus léger et déstabilise le véhicule. Pour réaliser ces tests par le calcul, la simulation en boucle ouverte qui a été vue au chapitre 1.5.1 est largement suffisante, car l'angle volant est maintenu constant durant le freinage. On choisit néanmoins de s'intéresser aussi aux corrections à apporter au volant pour rester sur la route sans partir en tête-à-queue.

L'ellipse de la figure 2.28 présente le domaine d'utilisation possible d'un véhicule en termes d'accélération transversale/longitudinale. Le domaine réellement utilisé par les conducteurs est beaucoup plus restreint, le potentiel d'adhérence du véhicule est donc sous-exploité en temps normal. Cependant, suite à un événement, le véhicule peut être placé n'importe où sur le cercle, et là, le véhicule doit être stable et permettre au conducteur de s'en sortir.

Ces tests consommateurs, même s'ils permettent d'évaluer le potentiel d'évitement d'un véhicule, ne donnent pas pour autant le pouvoir à un conducteur de l'exploiter correctement. Un conducteur en situation d'urgence avec un véhicule parfaitement capable d'éviter l'obstacle peut avoir un accident simplement par manque de formation ou de connaissance des limites de son véhicule. Le domaine des fortes accélérations à la fois transversales et longitudinales est particulièrement méconnu, comme cela est bien visible sur ce schéma. Par contre, par temps de pluie ou de neige, l'adhérence diminue fortement (l'ellipse se rétracte) et un conducteur restant dans son domaine d'utilisation normal peut conduire aux limites sans s'en rendre réellement compte.

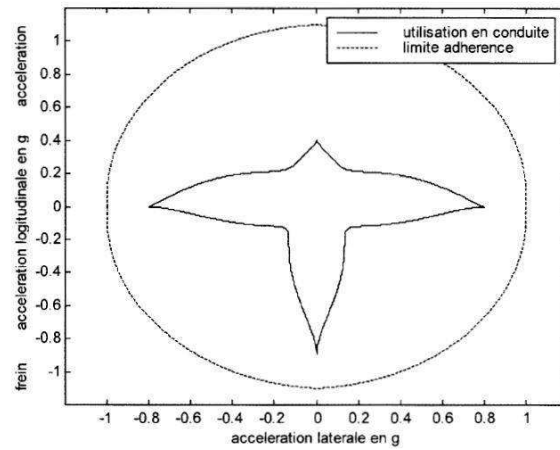


FIG. 2.28 – Domaine courant d'utilisation d'un véhicule et domaine réellement exploité par les conducteurs

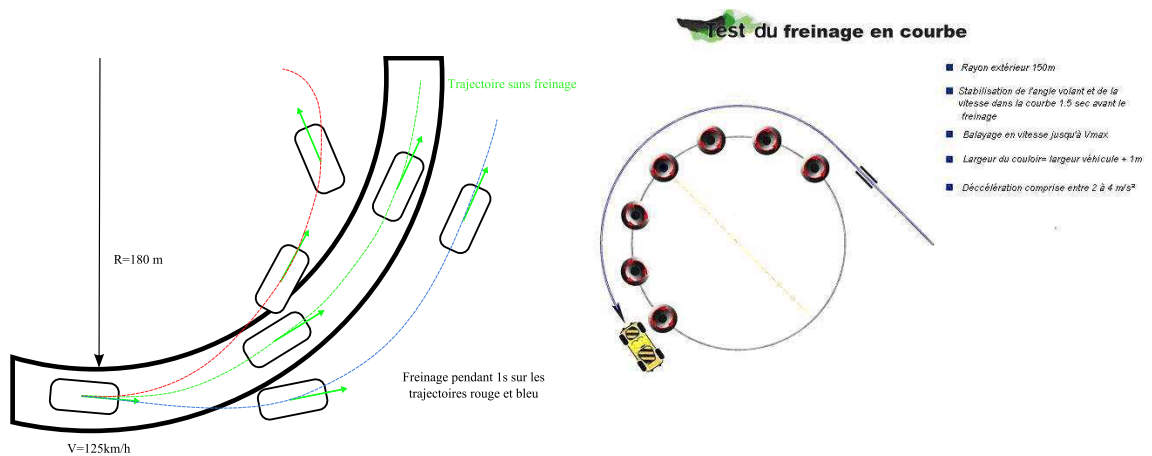


FIG. 2.29 – Description géométrique du test du freinage en courbe

2.3.5 Vent latéral

Lors de ce test, le véhicule entre à la vitesse de 120 km/h dans un couloir de 36 m de longueur où souffle un vent latéral de 65 km/h (voir figure 2.30). Le véhicule subit alors une déviation lorsqu'il traverse le couloir volant lâché. Cette déviation est mesurée à l'aide d'une pissette d'eau solidaire du véhicule. Celle-ci permet de laisser une trace au sol qui identifie clairement la trajectoire suivie par le véhicule. Les détails de la procédure se trouvent dans le document [ISO, 1996]. Des passages volant tenu sont également effectués et notés subjectivement afin d'évaluer la facilité à suivre une trajectoire rectiligne lors de la traversée de la veine de vent. Le conducteur doit agir au volant pour rester dans sa ligne, il ne doit pas anticiper la claque de vent, mais se rapprocher du comportement d'un client surpris par une rafale. Calculer par la simulation l'importance de la correction volant à apporter pourrait être intéressant lors de la conception des véhicules.

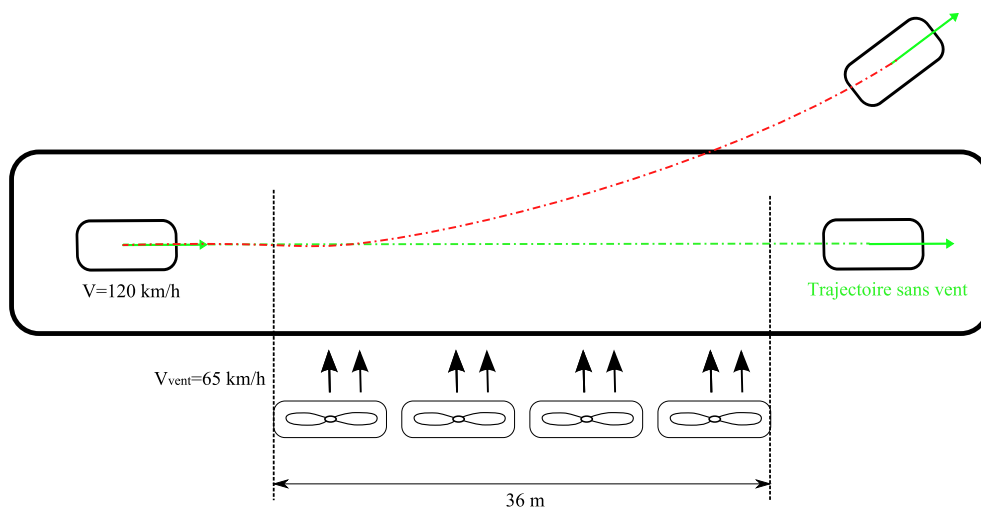


FIG. 2.30 – Description géométrique du test de sensibilité au vent latéral effectué volant lâché

2.4 Modélisation informatique

Il existe différents niveaux de modélisation d'un véhicule qui vont de la modélisation exhaustive de l'ensemble des organes mécaniques de la liaison au sol (modélisation organique), jusqu'à la réduction du système à un ensemble de cartographies représentant l'évolution de certaines grandeurs en fonction de certaines autres (modélisation fonctionnelle). En pratique, il existe toujours nombre d'exceptions et de modélisations intermédiaires. Aucun modèle n'est réellement purement fonctionnel ou organique. Les modèles de véhicules sont généralement caractérisés par un faible nombre d'entrées (angle volant, freinage), mais un grand nombre de variables d'état et de paramètres.

2.4.1 Modèles organiques

Les modèles organiques sont basés sur des modèles dynamiques multi-corps ADAMS qui décrivent précisément le comportement de chaque élément mécanique de la LAS. Les pièces mé-

talliques déformables peuvent être modélisées à l'aide de super-éléments issus de calculs éléments finis. Ce modèle est le plus précis qui soit disponible. Les temps de calcul sont longs mais les modèles peuvent servir à générer les tables cinématiques nécessaires aux modèles fonctionnels plus rapides.

2.4.2 Modèle AMESim/Car

Ce modèle utilise le logiciel AMESim de la société IMAGINE comme support pour la simulation. Il se base sur une modélisation fonctionnelle des organes, et est bien adapté aux calculs de dynamique route. Le temps d'exécution est d'environ 30 s de calcul pour 5 s de simulation (6 fois plus lent que le temps réel). L'interface avec le monde extérieur se fait avec des lectures/écritures de fichiers qui pénalisent les temps de calcul et rendent ce modèle difficilement exploitable avec les algorithmes de planification gourmands en calcul. Le modèle contient une centaine de variables d'état, représentant finement les dynamiques de chaque sous-système.

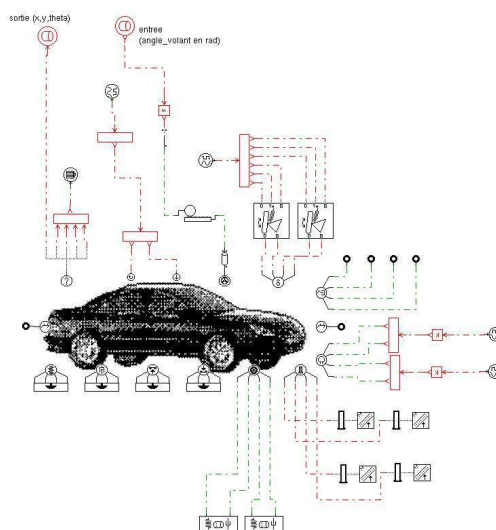


FIG. 2.31 – Modèle de comportement dynamique de véhicule FDV/Car utilisé dans le logiciel AMESim

2.4.3 Modèle SimulinkCar

Ce modèle est implémenté sur la plate-forme logicielle Matlab/Simulink. Sa vocation est de servir de base à des simulations pour le monde de l'automatique. La précision des calculs n'est pas tout à fait du même niveau que pour les modèles ADAMS ou AMESim, mais la rapidité du calcul (temps réel) nous oblige à nous concentrer sur ce modèle.

Le système est modélisé avec quatorze degrés de liberté (DDL) :

- les rotations des roues (4)
- le débattement vertical des roues (4)
- le mouvement libre de la caisse dans l'espace (6)

A ces quatorze DDL correspondent vingt-quatre variables d'état :

- la position/rotation libre de la caisse dans l'espace (6)
- les dérivées de ces six grandeurs (6)

- la vitesse de rotation des roues (4)
- le débattement vertical des roues (4)
- la vitesse de débattement des roues (4)

En reprenant les notations vues précédemment on a $\mathbf{u} = \mathbf{A}_{\text{volant}}$ et

$$\mathbf{q} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \theta, \phi, \psi, \dot{\theta}, \dot{\phi}, \dot{\psi}, w_1, w_2, w_3, w_4, z_1, z_2, z_3, z_4, \dot{z}_1, \dot{z}_2, \dot{z}_3, \dot{z}_4)$$

Le vecteur \mathbf{q} contient les données de position/orientation ainsi que toutes leurs dérivées afin de représenter correctement les phénomènes dynamiques. Le solveur utilisé est une méthode d'Euler simple avec un pas de 1 ms qui est suffisant pour assurer rapidité, stabilité et précision dans le domaine d'étude considéré.

Les paramètres utilisés correspondent aux éléments détaillés au chapitre 2.1, ils sont beaucoup trop nombreux pour être tous explicités ici. L'ensemble de ce paramétrage est regroupé à l'intérieur d'un fichier FDV (Fichier Dynamique Véhicule) qui permet de décrire complètement toute la base passive du véhicule. Dans la suite de ce document, seuls les résultats issus de simulations SimulinkCar seront présentés. Cependant tout ce qui est fait avec Simulink peut être également réalisé avec AMESim/Car. Les résultats obtenus sont généralement assez similaires.

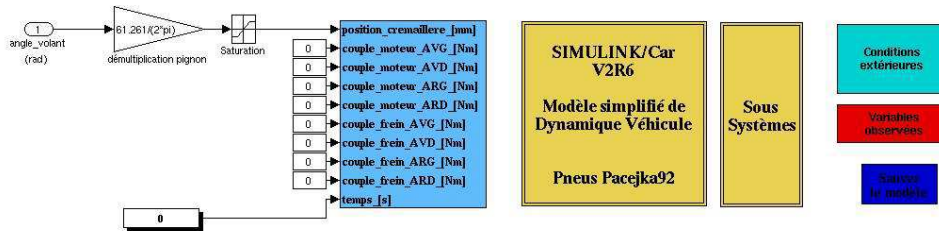


FIG. 2.32 – Le modèle de comportement dynamique de véhicule SimulinkCar dans l'environnement Matlab/Simulink

2.4.4 Extensions du modèle SimulinkCar

Le modèle SimulinkCar a pour vocation la simulation de la seule base mécanique passive du véhicule. Cependant, l'utilisation de la plate-forme Simulink permet de rajouter facilement des extensions au modèle afin de simuler le comportement de sous-systèmes. Ces rajouts augmentent souvent le temps de calcul de manière importante. Les exemples les plus classiques de sous-systèmes disponibles sous Simulink sont :

- Modèle de direction assistée : l'échec de certains tests peut être causé par une direction assistée mise en défaut. Il est donc crucial de pouvoir intégrer l'assistance de direction (parfois électrique aujourd'hui) au modèle initial. Les modèles de Direction Assistée Électrique (DAE) sont souvent donnés par les fournisseurs sous forme de boîtes noires que l'on intègre au modèle de véhicule AMESim ou Simulink.
- Modèle d'ESP : de même, le système ESP dont le fonctionnement est décrit au chapitre 2.1.6n influe grandement sur le comportement dynamique du véhicule. Les fournisseurs toujours soucieux de préserver leur propriété intellectuelle ne fournissent généralement ces modèles que sous forme de boîte noire compilée. Le modèle d'ESP étudié est issu d'une collaboration avec un fournisseur et est disponible uniquement sous forme de S-fonction Simulink. Le modèle fourni ne peut être modifié et est lié à une configuration spécifique du véhicule C.

2.4.5 Les fichiers FDV

Ce fichier condense en un seul endroit toutes les différentes cartographies (épures) et paramètres de la liaison au sol. Toutes les grandeurs sur lesquelles il est possible de jouer y sont présentes. Un importateur est fourni avec le modèle SimulinkCar pour lire aisément ce format de données. Ces fichiers peuvent être directement exportés depuis le modèle organique d'ADAMS ou générés à partir de mesures standardisées sur banc (Banc véhicule BCV, banc suspension, banc pneu FlatTrack).

2.4.6 Les sources d'erreur

Lors de la planification de mouvements, on va pousser le modèle de véhicule aux limites des possibilités physiques. Ce faisant, il est probable que l'on puisse également exploiter les défaillances de la modélisation. Il faut donc faire très attention lors de l'interprétation des résultats de calcul. Les erreurs sont présentes tout au long de la chaîne du calcul :

- Modélisation : équations simplifiées, phénomènes négligés
- Paramétrage : incertitudes sur des paramètres, tolérances de fabrication
- Résolution : approximations faites par le solveur
- Planification de mouvements : discrétisation, minima locaux

Il faudra considérer les résultats obtenus plus en termes qualitatifs que quantitatifs. Cela signifie que l'on va s'intéresser prioritairement à l'impact de variations paramétriques sur la limite en \mathcal{P} .

2.4.7 Modélisation des obstacles et détection de collisions

Les couloirs des différents tests n'ont pas été modélisés sous forme d'une série de cônes comme cela aurait pu paraître naturel, mais à l'aide d'obstacles polygonaux. L'objectif de cette représentation est de conserver une continuité et une régularité aussi grande que possible sur les contraintes. Les tests sont rigoureusement aux bonnes dimensions, mais tous les cas où une petite partie de la carrosserie du véhicule vient se loger entre deux cônes sont exclus.

La prise en compte des contraintes géométriques se fait pour tous les algorithmes des chapitres suivants grâce à un calcul de détection de collisions très précis. La librairie de détection utilisée est le Proximity Query Package (PQP) de l'université de Caroline du Nord. Ces routines utilisent les méthodes de Bounding Box et de Swept Sphere décrites en [Larsen *et al.*, 1999] et [Lin *et al.*, 1997] pour accélérer les calculs. On peut ainsi déterminer aisément si oui ou non le véhicule est en collision avec les obstacles. En cas de non collision, il est possible d'obtenir la plus petite distance entre le véhicule et les obstacles. En cas de collision, il est possible d'utiliser la librairie GPC qui fournit le polygone d'intersection entre les obstacles et le véhicule.

L'utilisation de ces bibliothèques requiert une description de la géométrie du véhicule et des obstacles sous forme de triangles tridimensionnels (la troisième dimension étant superflue dans le cas présent). Les algorithmes de Chazelle [Chazelle, 1991, Chazelle, 1987] et de Seidel [Seidel, 1991] permettent de décomposer sous forme de triangles n'importe quel polygone même concave. Le résultat de ce calcul dans le cas particulier d'un test VDA est présenté en figure 2.33. Sur cette modélisation, le couloir final a été volontairement rallongé afin de supprimer le cas problématique d'une sortie en tête-à-queue sans collision. Ces algorithmes qui ont été portés sous Matlab permettent de trianguler les polygones décrivant n'importe quel test et permettent

ainsi d'adapter facilement toutes les techniques des chapitres suivants à d'éventuelles autres manœuvres.

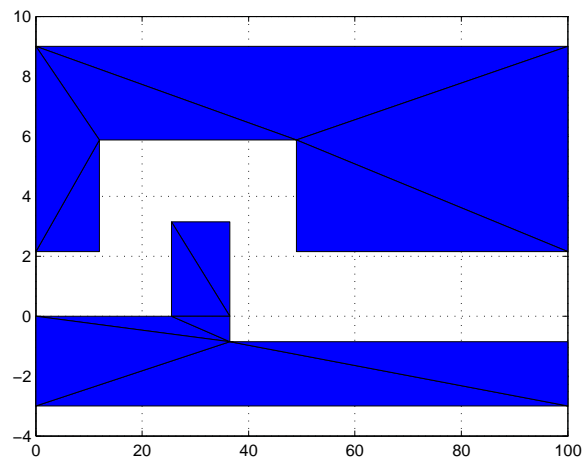


FIG. 2.33 – Modélisation informatique des 3 couloirs d'un test VDA sous forme d'obstacles polygonaux décomposés en triangles (les axes ne sont pas normés)

Chapitre 3

Les méthodes exploratoires

3.1 Introduction

Comme cela a été vu au chapitre 1.6.3, l'utilisation de techniques exploratoires basées sur l'existence de méthodes locales est impossible pour résoudre les problèmes demandés. Il existe cependant une autre classe d'algorithmes de planification de type exploratoire qui permettent d'adresser le cas des systèmes dynamiques complexes (voir [Donald *et al.*, 1993]). Ces algorithmes sont basés sur un échantillonnage des entrées et permettent de couvrir rapidement l'espace d'état à partir d'une configuration initiale via la création d'un arbre de recherche (voir figure 3.1).

Chaque arbre correspond à une unique requête de planification partant d'un état initial vers un état final. Cet arbre ne peut être réutilisé pour d'autres conditions initiales/finales comme cela est possible avec les roadmaps des PRM. Afin d'éviter une explosion inévitable du nombre de combinaisons à calculer, de nombreux types de filtrages des entrées et/ou des états sont utilisés.

C'est à ce niveau que se situent les principales différences entre les algorithmes exploratoires. Le principal intérêt de ces méthodes est d'être globales. C'est à dire qu'elles explorent exhaustivement tout l'espace d'état avec la certitude de ne pas s'arrêter dans un minimum local. Cette généralité se paye évidemment par une moindre précision issue de la discrétisation.

Tous les algorithmes présentés ci-dessous ont été testés avec le modèle de véhicule Simulink-Car. La librairie Motion Strategy Library (MSL) développée par Lavelle et Kuffner (disponible à l'adresse <http://msl.cs.uiuc.edu/msl/>) a servi de base pour mettre en œuvre ces algorithmes exploratoires. Le modèle de référence étant uniquement disponible sous Matlab, une interface a été développée afin de pouvoir utiliser la MSL depuis cet environnement (voir annexe A). Nombre des algorithmes décrits ci-dessous sont déjà disponibles dans cette librairie, les autres ont été ajoutés manuellement en dérivant les classes disponibles. Leur implémentation en langage C++ permet l'utilisation de structures bien plus efficaces et rapides que celles disponibles sous Matlab. Ce dernier se chargeant de la partie simulation du modèle qui lui échoit naturellement ainsi que du pré- et post-traitement des données.

Les seuls pré-requis de ces algorithmes sont les éléments suivants :

- Un état initial \mathbf{q}_{init}
- Une zone d'arrivée
- Des limites sur la dynamique de l'entrée
- Un modèle explicite de véhicule capable de simuler la trajectoire du véhicule à partir de \mathbf{q}_{init} et d'une entrée $\mathbf{u}(t)$ pendant un temps Δt

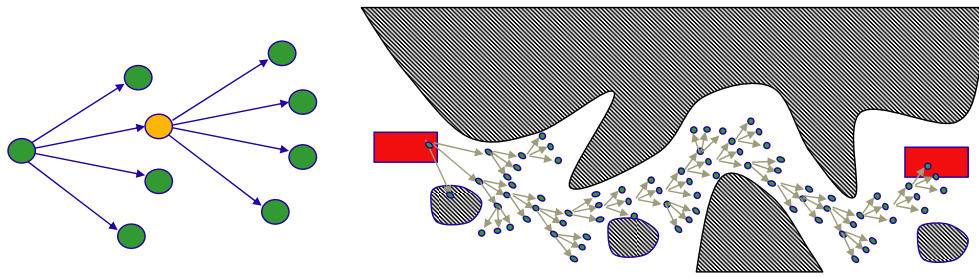


FIG. 3.1 – Les algorithmes exploratoires peuvent trouver des solutions à des problèmes de planification de mouvements dynamiques de faible dimension et ne requérant pas une trop grande précision

- Une fonction capable de déterminer l'existence d'une collision entre une configuration quelconque \mathbf{q} du véhicule et les obstacles

La plupart des méthodes développées dans la suite du chapitre se basent sur la création d'un arbre contenant en chacun de ses nœuds une configuration du système. La connaissance de cette configuration permet de reprendre la simulation à partir de n'importe quel endroit de l'arbre en écrasant la configuration initiale du modèle par la valeur contenue dans le nœud. Il faut donc que la sortie du modèle représente exactement son état interne. Le modèle peut toujours être boîte noire (fonctionnement inconnu) mais il faut pouvoir imposer un état quelconque au système (et donc connaître au minimum la dimension du vecteur d'état \mathbf{q}).

Cette condition est satisfaite pour tous les modèles Simulink sauf ceux comportant un modèle d'ESP où l'état de celui-ci est inconnu et inaccessible. Ces derniers ne pourront donc être testés dans ce chapitre. Les méthodes des chapitres 4 et 5 n'ont pas cette exigence particulière.

3.2 Algorithmes disponibles

3.2.1 Forward Dynamic Programming (FDP)

La méthode Forward Dynamic Programming (FDP) a été initialement introduite par Latombe dans [Barraquand et Latombe, 1993]. Comme son nom l'indique, c'est une variante issue du cadre plus général de la programmation dynamique (PD) initiée par Bellman (voir [Bellman, 1957, Bertsekas, 2001]).

L'algorithme part d'un état initial \mathbf{q}_{init} , échantillonne régulièrement les entrées possibles \mathbf{u} du système parmi \mathcal{U} . Ensuite, il simule chacune de ces entrées pendant une durée Δt et place les configurations résultantes dans un arbre. Les nœuds en collision ne sont pas rajoutés dans l'arbre. Le filtrage des états résultants est ensuite assuré par une grille issue de la discrétisation de l'espace d'état. Chaque case de la grille ne peut contenir qu'une seule et unique configuration (voir figure 3.2). Si une configuration tombe dans une case déjà occupée, celle-ci est tout simplement supprimée de l'arbre. Un coût est associé à chaque état, et c'est l'état ayant le coût le plus faible qui est choisi pour servir de point de départ à la phase d'expansion suivante.

L'utilisation de ce système occupe beaucoup de place en mémoire à cause de la discrétisation relativement fine de l'espace d'état qui est nécessaire. Dans le cas qui nous intéresse, les cases ne sont pas présentes sur l'ensemble des états du système, mais seulement sur les états correspon-

dants à la position géométrique du véhicule x, y, ψ où se trouvent les obstacles. Une illustration graphique du fonctionnement de cet algorithme est présentée sur la figure 3.2.

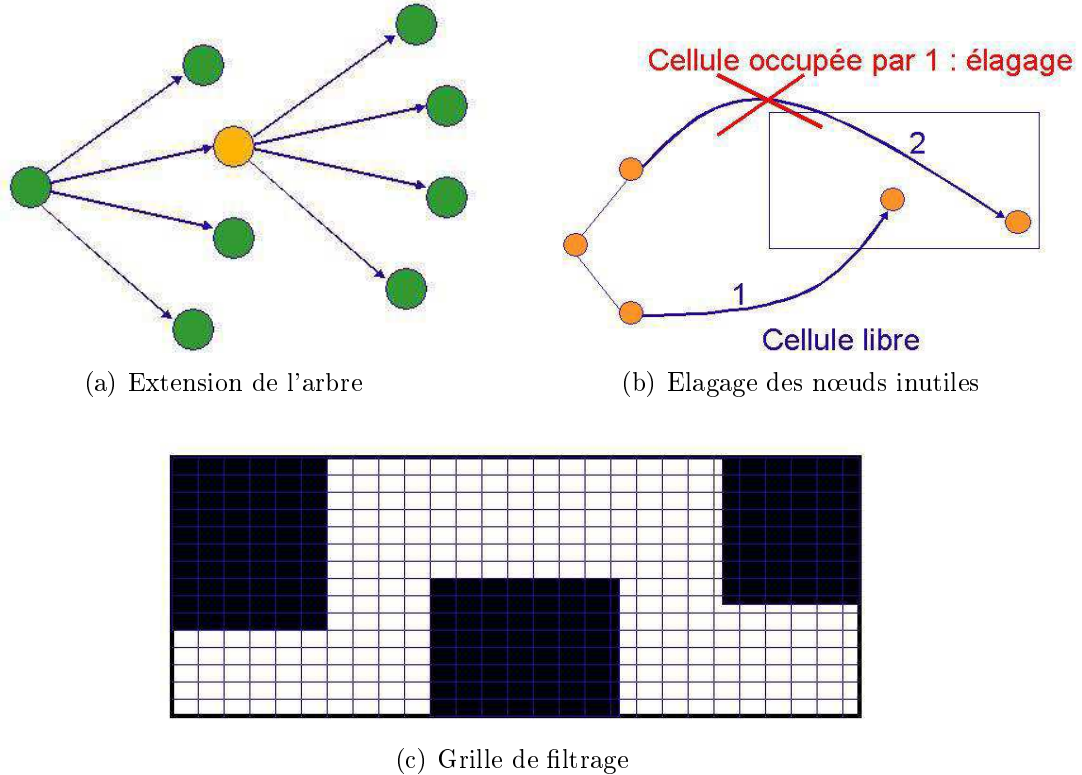


FIG. 3.2 – Illustration graphique du fonctionnement de l'algorithme FDP

L'ensemble des entrées \mathcal{U} échantillonné peut être choisi en fonction des entrées précédentes afin de respecter les contraintes de vitesse et d'accélération angulaire au volant. L'algorithme s'arrête lorsqu'il entre dans une zone définie comme étant le but à atteindre ou que le nombre maximum de nœuds dans l'arbre est épuisé.

Les axes de toutes les figures représentant les trajectoires des véhicules sont volontairement non normés afin de permettre une visualisation plus aisée. La distance maximum suivant l'axe horizontal des x est de 61 m , tandis que suivant l'axe vertical la distance maximum est d'environ 6 m . Ceci explique l'aspect inhabituel des véhicules (rectangles en tirets verts) qui sont pourtant de dimension $4.3 \times 1.73\text{ m}$ dans le sens de la longueur. Les obstacles sont représentés par les rectangles noirs qui symbolisent les zones interdites. Une trajectoire solution est tracée en rouge par-dessus l'arbre de recherche bleu clair qui montre les zones explorées par l'algorithme. Chaque nœud de la trajectoire discrétisée est représenté par un petit cercle. La figure 3.3 résume l'ensemble des éléments à appréhender sur ces tracés.

Un exemple de résultat obtenu avec cette méthode est présenté sur la figure 3.4, dans le cas particulier d'un test VDA avec le véhicule A à 40 km/h . On observe que l'algorithme FDP fournit une solution acceptable au problème. L'angle volant présente un profil un peu différent de celui présenté en figure 2.17, car la faible vitesse oblige à effectuer un retour en position neutre du volant sur le deuxième pic afin de stabiliser le véhicule dans la ligne droite du deuxième couloir.

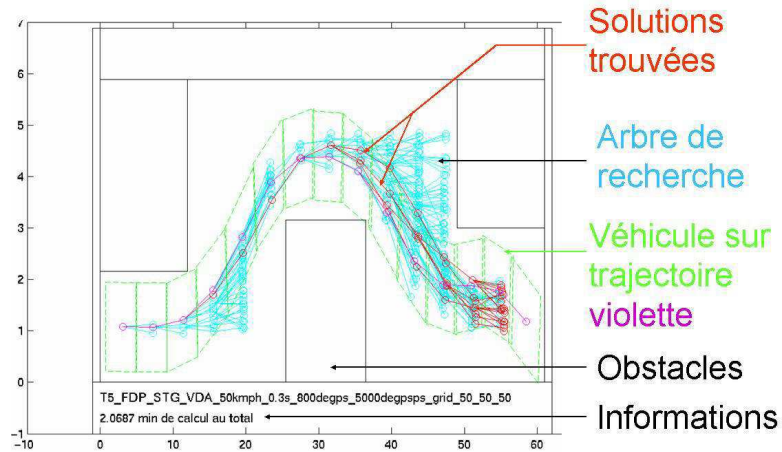


FIG. 3.3 – Décryptage des éléments présents sur les graphiques des méthodes exploratoires

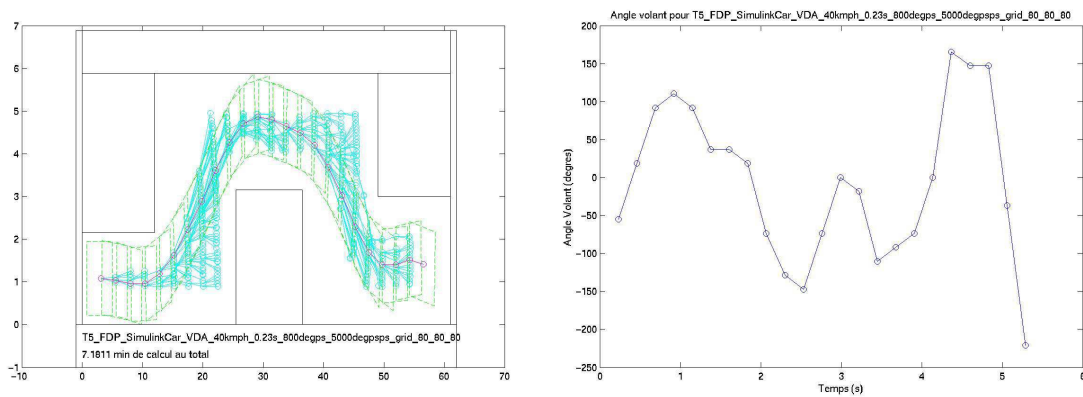


FIG. 3.4 – Solution FDP avec SimulinkCar pour un test VDA effectué avec le véhicule A à 40 km/h

La figure 3.5 présente le même calcul pour le franchissement du test dit de l'évitement ville (voir chapitre 2.3.3). On constate que les amplitudes obtenues sont beaucoup augmentées et sont proches de celles réalisées expérimentalement par les essayeurs en figure 2.27.

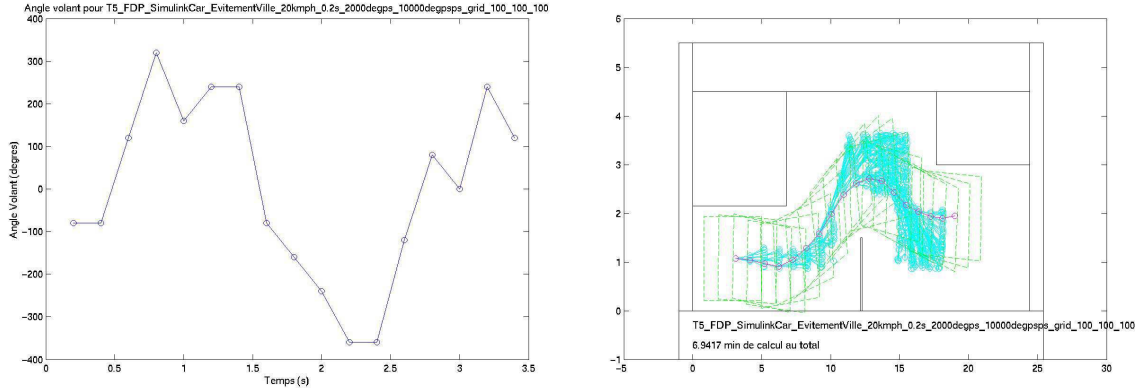


FIG. 3.5 – Solution FDP avec SimulinkCar pour un évitement ville effectué avec le véhicule A

De nombreuses variations de cet algorithme sont possibles grâce à l'opportunité de choisir des fonctions coût différentes. Il est par exemple possible de chercher à minimiser le temps de parcours vers le but, la vitesse volant ou la distance aux obstacles. Plus simplement, il suffit de ne rien minimiser et de laisser l'algorithme trouver une solution quelconque parmi l'infinité possible.

3.2.2 Rapidly exploring Random Trees (RRT)

Cette méthode qui a été développée par LaValle et Kuffner dans [LaValle et Kuffner, 2000, LaValle et Kuffner, 1999] constitue en quelque sorte l'état de l'art pour les méthodes de planification de mouvements dynamiques. Le principe est de choisir un état \mathbf{q}_{rand} au hasard dans l'espace libre \mathcal{C}_{free} , puis de sélectionner le nœud de l'arbre \mathbf{q}_{near} le plus proche suivant une métrique définie (l'arbre est initialisé par \mathbf{q}_{init}). A partir de ce nœud de l'arbre, on choisit dans l'ensemble \mathcal{U} des entrées possibles, l'entrée \mathbf{u} qui permet de se rapprocher le plus du nœud \mathbf{q}_{rand} pour une simulation élémentaire de durée Δt . Le nœud \mathbf{q}_{new} issu de ce calcul est alors intégré à l'arbre. L'algorithme s'arrête lorsqu'il a atteint la zone d'arrivée, ou après un nombre prédéfini de nœuds dans l'arbre. Il suffit alors de remonter l'arbre de la fin vers le début pour obtenir la solution du problème.

Une illustration graphique de ce fonctionnement est disponible sur la figure 3.6. La principale qualité de cet algorithme est de s'étendre rapidement et efficacement en couvrant le plus possible l'espace d'état. Les zones vides sont très rapidement explorées grâce au guidage induit par l'échantillonnage aléatoire de l'espace d'état. Les bornes de vitesse et d'accélération volant sont bien respectées, car seules les entrées permettant de rester dans les limites acceptables sont testées. Il est possible de biaiser le choix aléatoire du nœud \mathbf{q}_{rand} en le remplaçant par moment par un état de la zone d'arrivée (avec une certaine probabilité). L'exploration est ainsi guidée vers l'objectif sans pour autant tomber dans des minima locaux.

Le problème du calcul d'une distance (métrique) entre deux configurations n'est pas anodin, car l'espace d'état n'est généralement pas euclidien. Les angles et les distances qui sont mélangés

dans l'état \mathbf{q} ne constituent pas des grandeurs homogènes. La parade consiste généralement à rajouter un coefficient sur les angles pour les homogénéiser à des distances.

```

BUILD_RRT( $x_{init}$ )
1  $\mathcal{T}.init(x_{init});$ 
2 for  $k = 1$  to  $K$  do
3    $x_{rand} \leftarrow \text{RANDOM.STATE}();$ 
4    $\text{EXTEND}(\mathcal{T}, x_{rand});$ 
5 Return  $\mathcal{T}$ 

```

```

EXTEND( $\mathcal{T}, x$ )
1  $x_{near} \leftarrow \text{NEAREST\_NEIGHBOR}(x, \mathcal{T});$ 
2 if  $\text{NEW\_STATE}(x, x_{near}, x_{new}, u_{new})$  then
3    $\mathcal{T}.add\_vertex(x_{new});$ 
4    $\mathcal{T}.add\_edge(x_{near}, x_{new}, u_{new});$ 
5   if  $x_{new} = x$  then
6     Return Reached;
7   else
8     Return Advanced;
9 Return Trapped;

```

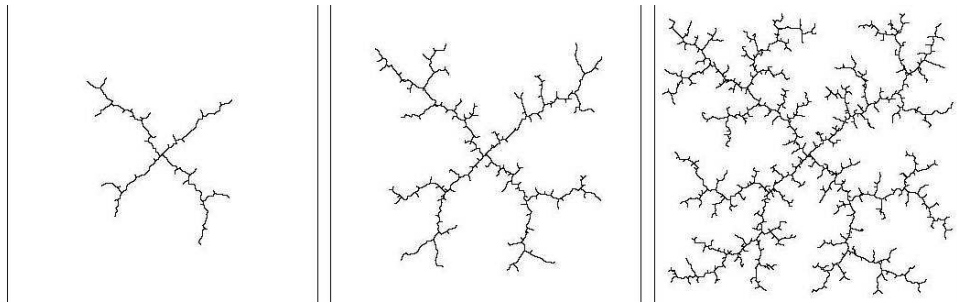
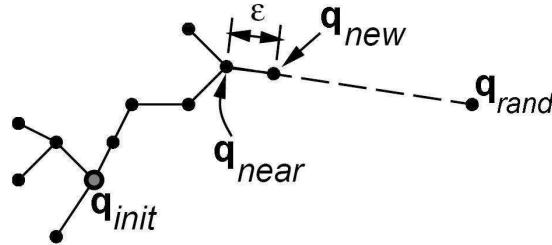


FIG. 3.6 – Illustration du fonctionnement de l'algorithme RRT

Des résultats ont pu être obtenus avec cette technique, mais ceux obtenus avec sa variante RC-RRT présentée en 3.2.3 sont bien meilleurs tant en termes de vitesse de calcul que de qualité de la solution obtenue. Par conséquent, la section suivante ne détaillera que les résultats de simulation de cette variante.

3.2.3 Resolution Complete RRT (RCRRT)

Le sigle RCRRT signifie "Resolution Complete Rapidly-Exploring Random Trees". Cette méthode dérivée de l'algorithme RRT a été spécialement adaptée par Cheng et Lavelle dans [Cheng et LaValle, 2001] et [Cheng, 2005] afin de réduire la sensibilité des algorithmes RRT aux problèmes de mesure de distance entre deux configurations. Cet algorithme est particulièrement bien adapté aux systèmes dynamiques complexes.

Un exemple est d'ailleurs traité dans cet article avec le passage d'un test "Consumer Union Short Course" équivalent du test VDA. Le modèle utilisé est un modèle analytique à 9 dimensions. Les résultats sont intéressants (voir figure 3.7) mais le problème de la détermination de V_{max} et de la robustesse paramétrique n'est pas abordé. La simplicité du modèle utilisé ne permet pas l'obtention de résultats réellement utilisables industriellement.

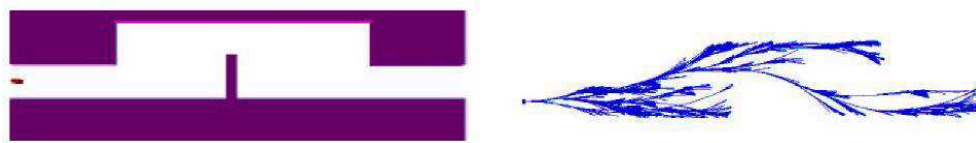


FIG. 3.7 – Franchissement du test Consumer Union Short Course par un modèle de véhicule à 9 variables d'état à l'aide de l'algorithme RC-RRT

L'idée de cet algorithme est d'associer à chaque nœud de l'arbre RRT une probabilité de collision en fonction des résultats des explorations issues de ce nœud. Lors de la phase de choix du nœud le plus proche de l'algorithme RRT, ceux avec la plus faible probabilité de collisions sont favorisés. Un exemple de calcul appliqué au cas particulier du test VDA est disponible sur la figure 3.8. L'allure de l'angle volant et la trajectoire associée sont acceptables.

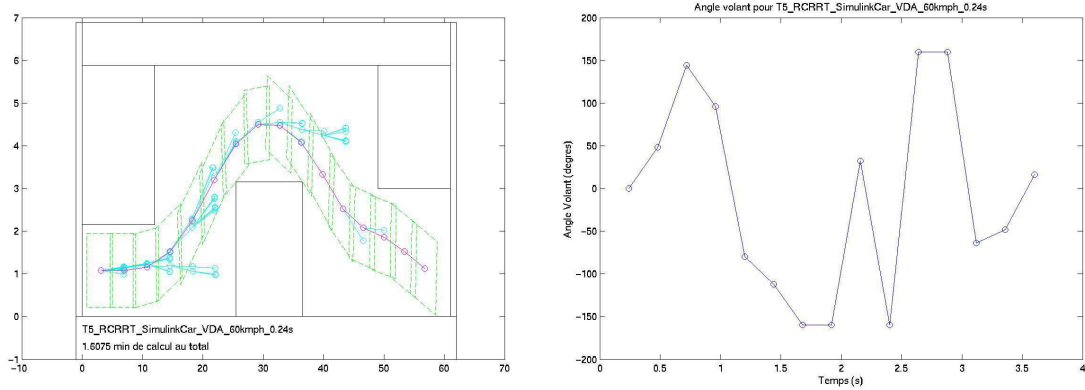


FIG. 3.8 – Passage d'un test VDA avec le modèle SimulinkCar à 60 km/h avec l'algorithme RCRRT

3.2.4 Kino Dynamic Planner (KDP)

Cette méthode aléatoire est issue des travaux de thèse de Hsu [Hsu, 2000] et se base aussi sur la création d'un arbre de recherche. Cette technique a été spécifiquement développée afin de gérer les problèmes où les obstacles et le véhicule sont mouvants. Cependant, elle reste parfaitement utilisable dans le cas qui nous concerne. Le nœud de départ est choisi au hasard dans l'arbre, de même que l'entrée et le temps d'intégration. Un système de grille permet de contenir l'explosion combinatoire et de guider l'exploration vers les zones peu couvertes.

L'algorithme travaille dans l'espace $Configuration \times Temps$ en sélectionnant aléatoirement : l'entrée \mathbf{u} qui va être simulée parmi \mathcal{U} , le nœud de l'arbre qui va être étendu et la durée Δt de la simulation avec cette entrée. Une grille discrétisant l'espace d'état est associée à cet algorithme. Celle-ci mémorise pour chacune de ses cases (BIN), le nombre de nœuds qui y sont tombés. L'algorithme guide ensuite les explorations suivantes en biaisant les probabilités lors du choix du nœud à étendre. L'objectif est d'améliorer la rapidité, la convergence et la couverture de cet algorithme. Une illustration plus graphique du fonctionnement de cet algorithme est présentée sur la figure 3.9.

Un exemple de calcul appliqué au cas du test VDA est présenté en figure 3.10. Le modèle SimulinkCar du véhicule A passe avec succès à 50 km/h . La discrétisation temporelle est ici fixe et vaut $\Delta t = 150 ms$.

3.3 Limitations

Ces algorithmes ne permettent la planification qu'avec un paramètre \mathcal{P} donné, la détermination des limites du test s'effectue par incrémentation progressive de \mathcal{P} après chaque planification

Algorithm 1

1. Initialize T with m_b ; $i \leftarrow 1$
2. **repeat**
3. Pick a milestone m from T with probability $\pi_T(m)$
4. Pick a control function u from \mathcal{U}_ℓ uniformly at random
5. $m' \leftarrow \text{PROPAGATE}(m, u)$
6. **if** $m' \neq \text{nil}$ **then**
7. Add m' to T ; $i \leftarrow i + 1$
8. Create an arc e from m to m' ; store u with e
9. **if** $m' \in \text{ENDGAME}$ **then** **exit** with SUCCESS
10. **if** $i = N$ **then** **exit** with FAILURE

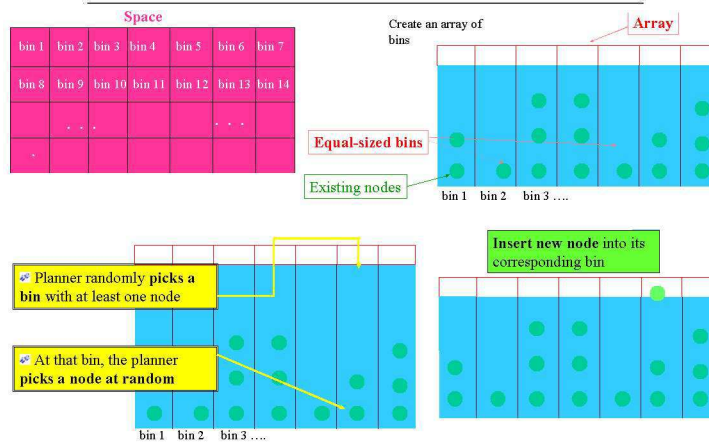


FIG. 3.9 – Illustration du fonctionnement de l’algorithme KDP

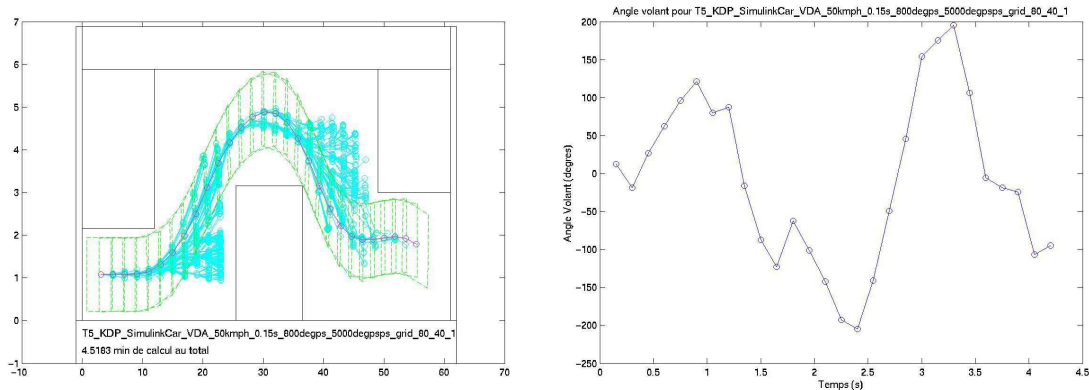


FIG. 3.10 – Solution obtenue avec l’algorithme KDP sur le modèle SimulinkCar du véhicule A

réussie. Les premiers tests s'effectuent loin des limites afin d'être sûr de trouver facilement une solution. Dans le cas du test VDA, les vitesses de passage maximum V_{max} qui ont pu être atteintes avec ces techniques au bout d'une nuit de calcul sont au mieux de l'ordre de 60 km/h . Ces valeurs sont loin des valeurs réellement obtenues en essais et des résultats des autres techniques de simulation développées dans les chapitres 4 et 5.

La principale faiblesse de ces techniques est leur manque intrinsèque de précision ce qui peut expliquer ces résultats surprenants. La raison en est la nécessaire discrétisation des entrées et du pas de temps. En dépit des différentes techniques de filtrage détaillées précédemment, cette discrétisation n'a jamais pu être suffisante pour aller chercher les limites du système en un temps raisonnable. Un compromis a donc dû être trouvé entre temps de calcul et précision de l'exploration.

Lorsqu'on se rapproche des limites de \mathcal{P} , on se retrouve en quelque sorte dans la situation où un couloir étroit doit permettre de connecter deux larges régions de l'espace d'état (voir [Hsu *et al.*, 1999] et figure 3.11). Dans le cas présent, l'étroitesse du couloir résulte principalement du comportement dynamique du système. Or c'est précisément ce cas qui pose problème aux algorithmes exploratoires. La question de connecter ces deux régions a fait l'objet de nombreux travaux, mais aucune solution réellement satisfaisante n'a pu être apportée dans le cas des systèmes dynamiques complexes qui sont étudiés ici.

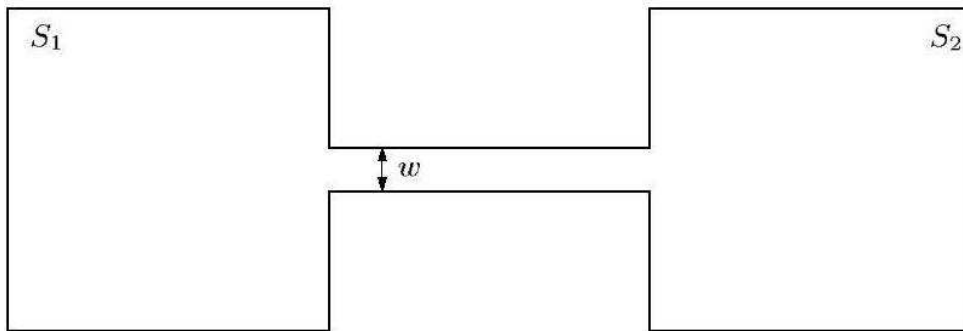


FIG. 3.11 – La connexion de deux grandes régions d'espace par un couloir étroit est l'exemple type du problème difficile à résoudre avec des méthodes exploratoires

La discrétisation de l'entrée est d'autant plus problématique que les sollicitations d'entrée doivent couvrir une plage importante d'angle et de vitesse volant. Ceci afin de n'exclure aucune solution possible. Comme cela a été vu précédemment (voir figure 1.2), ces systèmes requièrent une grande précision sur l'angle volant afin de découvrir une solution à vitesse élevée. Le nombre d'entrées testées en chaque nœud étant fixé à l'avance (20 à 30 en général), cette précision est fortement limitée par la nécessité de couvrir un large domaine de vitesses volant. Si la couverture en vitesse volant est trop faible, le véhicule ne pourra franchir les obstacles efficacement, car ces passages requièrent souvent des coups de volant violents. Par contre, si les angles volant sont discrétisés trop lâchement au profit de la vitesse volant, le véhicule ne pourra pas se stabiliser dans les étroits couloirs et entrera en collision alors qu'il existe bel et bien une solution. Ces deux besoins sont fortement contradictoires et sont une des explications de l'échec de ces techniques. Tel est le prix à payer pour profiter de l'extrême généralité de ces techniques et de leur capacité à explorer complètement un espace d'état sans se laisser piéger dans des minima locaux.

Un autre problème de ces algorithmes aléatoires reste l'importante dispersion des résultats pour une même simulation ainsi que la robustesse des résultats qui n'est pas garantie à cause de l'échantillonnage aléatoire (dans un temps de calcul fini). Ce genre de comportement est pertur-

bant en pratique et rend difficiles les études de sensibilité. Il est possible de recréer informatiquement une même suite de tirages aléatoires (en figeant la graine d'un générateur pseudo-aléatoire). Mais dans ce cas, les qualités d'exploration de ces algorithmes ne peuvent plus être garanties (minima locaux).

Les limitations informatiques liées au nombre de points calculables et à l'espace mémoire requis pour stocker des arbres aussi grands sont un autre argument à l'encontre de ces techniques. De plus, les modèles d'ESP boîte noire des fournisseurs ne sont pas utilisables avec ces algorithmes. Il est en effet impossible d'imposer leur état interne et donc de pouvoir reprendre une simulation à partir d'un nœud donné. Ce type de calcul s'est montré assez peu robuste face à des variations de paramétrage (pneu, masse, etc.) censées améliorer la vitesse de passage.

Le réglage des différents paramètres des algorithmes (discretisation des entrées, de la grille, etc.) revêt une importance capitale pour la qualité et la rapidité des solutions obtenues. Or ce paramétrage est peu intuitif et l'obtention de réglages corrects se fait grâce au retour d'expérience. Même s'il est possible d'automatiser partiellement le processus pour le cas particulier des tests consommateurs, ce point va à l'encontre de la généralité de l'approche et de son utilisation industrielle avec des modèles et obstacles quelconques.

Loin des limites, il existe de nombreuses manières différentes de passer chaque test. Parmi celles-ci, on trouve des solutions exactes du point de vue du modèle et des collisions, mais qui ne correspondent en rien aux types de solutions qu'un essayeur est susceptible de générer. La figure 3.12 présente le résultat d'une simulation où un test VDA est franchi sans toucher les plots. Le calcul a été fait sur un modèle du véhicule A à 57.6 km/h à l'aide de l'algorithme exploratoire RC-RRT. La trajectoire dessinée correspond à la simulation de l'angle volant très haché en bleu. A titre indicatif, une interpolation de l'angle volant est fournie (en pointillés rouge). Celle-ci montre clairement que la tendance générale de l'angle en bleu est bonne et proche des courbes obtenues par les essayeurs à condition de filtrer les oscillations inutiles.

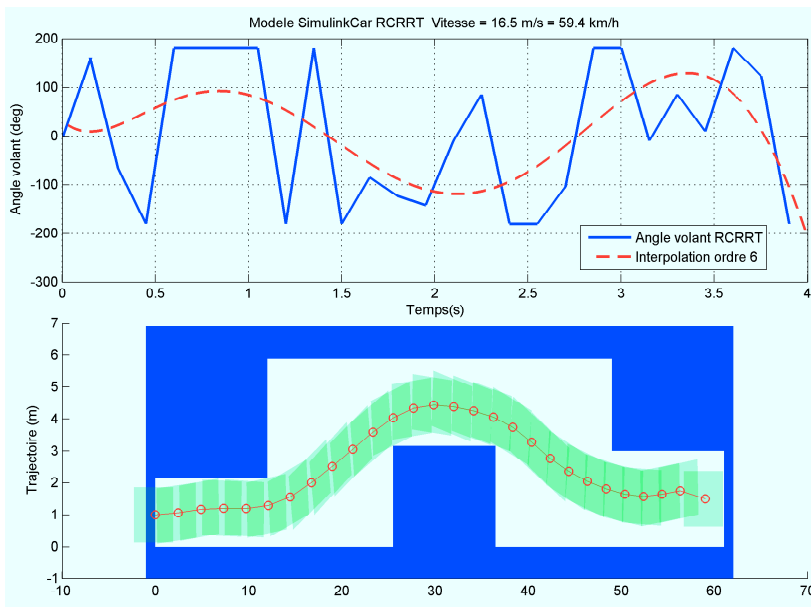


FIG. 3.12 – Un exemple de solution exacte mais aberrante du point de vue d'un essayeur humain

Au-delà de cet exemple volontairement extrême, la pratique a montré une tendance générale de ces algorithmes à fournir des solutions peu naturelles. Comme il a été vu précédemment,

cette limitation est inhérente au fonctionnement même de ces techniques et ne peut se faire sans réduction de l'espace des solutions (compromis amplitude/vitesse volant à discrétisation fixe). Ces algorithmes permettent de réaliser un certain nombre de choses intéressantes à faible vitesse, mais souffrent du problème de "pollution" des résultats par ces solutions exactes mais aberrantes. Ces solutions pourront néanmoins être utilisées dans le cadre des travaux effectués dans le chapitre 6.

En conclusion, ces techniques resteront cantonnées au domaine éloigné des limites des tests à cause de la discrétisation relativement grossière de l'état et des entrées qu'elles requièrent pour être exécutées en un temps raisonnable. Elles peuvent cependant trouver un intérêt pour la détermination d'ensembles de trajectoires solutions (voir chapitre 6), mais également pour fournir des conditions initiales à des algorithmes de contrôle optimal dont l'utilisation peut être adaptée afin de résoudre les problèmes qui nous intéressent.

L'idée fondatrice des chapitres 4 et 5 est d'essayer de pallier à ces problèmes de robustesse et de précision à l'aide d'algorithmes se servant d'informations locales et non plus globales. La généralité et la complétude de ces méthodes globales y sont sacrifiées pour bénéficier d'une bien meilleure précision.

Chapitre 4

Commande Optimale

Le potentiel des méthodes d'optimisation et de commande optimale pour la résolution numérique des problèmes 1 et 2 est grand. Leur prise en considération est donc nécessaire face aux méthodes de planification de mouvements des chapitres 3 et 5. Nous nous placerons ici uniquement d'un point de vue "utilisateur" de ces outils en ne cherchant pas à améliorer leur fonctionnement mais seulement à exploiter au mieux l'existant. La planification de mouvements a déjà été entrevue sous l'angle de l'optimisation dans les articles [Divelbiss et Wen, 1997, Divelbiss et Wen, 1994].

Un article de référence pour la comparaison et l'évaluation des différentes méthodes d'optimisation de trajectoire est disponible dans [Betts, 1998]. Ces techniques sont principalement issues du monde de l'aérospatial. L'exemple classique est de minimiser la consommation en carburant d'un satellite pour passer sur une orbite donnée tout en respectant certaines contraintes sur sa trajectoire (couloir). Le véhicule spatial est le plus souvent supposé ponctuel, la prise en compte de ces contraintes est donc relativement aisée. Dans le cas présent, la consommation ou le temps de parcours sont sans grand intérêt. La partie "résolution de contraintes" de l'optimisation suffit à elle seule à résoudre le problème pour peu que les contraintes géométriques exactes soient correctement prises en compte.

4.1 Objectifs et Définitions

L'optimisation et la commande optimale sont deux approches différentes mais liées par une parenté étroite, surtout lorsque l'on s'intéresse à leur résolution numérique. Le but de l'optimisation est de déterminer la valeur d'un jeu de paramètres $\mathbf{x} = (x_1, x_2, \dots, x_p)$ qui minimise un critère $J(\mathbf{x})$ soumis aux contraintes égalités $\mathbf{C}_{eq}(\mathbf{x}) = 0$ et inégalités $\mathbf{C}(\mathbf{x}) \leq 0$.

La commande optimale vise à trouver une commande $\mathbf{u}(s)$ d'un système dynamique qui permet de minimiser un certain critère d'optimalité J tout en respectant des contraintes sur les entrées et la trajectoire dans l'espace d'état.

4.2 Techniques existantes

4.2.1 Résolution analytique

Les livres [Trelat, 2005], [Bryson, 1975] et [Locatelli, 2001] fournissent une bonne introduction aux problématiques de la commande optimale notamment du point de vue de la théorie et de la résolution de problèmes analytiques. Il est possible de calculer analytiquement les solutions de certains problèmes de commande optimale via l'application du principe du maximum de Pontryagin (condition nécessaire) ou en résolvant l'équation d'Hamilton-Jacobi-Bellman (condition suffisante). Le principe du maximum de Pontryagin s'exprime comme suit :

Soit une grandeur H appelée Hamiltonien du système, défini par :

$$H(\mathbf{q}, \mathbf{u}, \boldsymbol{\lambda}, s) = -J(\mathbf{q}, \mathbf{u}, s) + \boldsymbol{\lambda}^T \cdot f(\mathbf{q}, \mathbf{u}, s) \quad (4.1)$$

Le vecteur $\boldsymbol{\lambda}$ est appelé vecteur adjoint, J est le critère à minimiser et s est toujours le temps. Les équations canoniques de Hamilton s'écrivent :

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} \quad (4.2)$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{q}} \quad (4.3)$$

Ces relations définissent le vecteur adjoint $\boldsymbol{\lambda}$. Le principe du maximum énonce que la commande optimale \mathbf{u}^* est celle qui maximise le hamiltonien, les contraintes et les conditions terminales étant satisfaites. L'article [Laumond, 1995] présente des techniques de commande optimale appliquées à des modèles cinématiques de robots non holonomes. Dans le cas qui nous concerne la résolution devra se faire à l'aide de techniques numériques telles que [Barclay *et al.*, 1997] les présente.

4.2.2 Méthodes numériques directes et indirectes

Il existe deux grands types de méthodes numériques en commande optimale : les méthodes directes et indirectes. Les *méthodes indirectes* consistent à résoudre numériquement un problème aux valeurs limites obtenu par application du principe du maximum de Pontryagin. L'application de ce principe amène en effet à devoir résoudre une équation différentielle avec des conditions initiales et finales (Boundary Value Problem ou BVP). Ce problème de BVP est résolu numériquement par des méthodes de "tir" simples ou multiples (voir [Shampine *et al.*, 2000]). Ces techniques sont bien adaptées aux problèmes analytiques, permettent une précision extrême mais ont un domaine de convergence assez étroit.

Une des *méthodes directes* consiste à discrétiser totalement l'état et l'entrée. Le problème de commande optimale est alors réduit à un problème d'optimisation non linéaire plus simple (Non Linear Programming : NLP). Ces techniques permettent de prendre facilement en compte des contraintes sur l'état, s'adaptent facilement à tout type de modèle mais peuvent tomber dans des minima locaux. Une autre méthode directe consiste à résoudre numériquement l'équation d'Hamilton-Jacobi. Une comparaison des avantages et inconvénients de chacune de ces deux techniques est présentée dans [von Stryk et Bulirsch, 1992] et résumée dans le tableau 4.1.

Méthodes Directes	Méthodes Indirectes
Mise en œuvre simple, sans connaissance a priori	Connaissance a priori de la structure de la trajectoire optimale
Peu sensibles au choix de la condition initiale	Très sensible au choix de la condition initiale
Facilité de prise en compte de contraintes sur l'état	Difficulté théorique de la prise en compte de contraintes sur l'état
Précision numérique basse ou moyenne	Très grande précision numérique
Efficaces en basse dimension	Efficaces en toute dimension
Gourmandise en mémoire	Calculs parallélisables
Problème de minima locaux	Petit domaine de convergence

TAB. 4.1 – Comparaison entre méthodes numériques directes et indirectes de commande optimale

4.2.3 BVP et Méthodes de tir

Les méthodes de tir utilisées dans les méthodes indirectes servent à résoudre des problèmes aux conditions limites. Un problème de BVP consiste à trouver les entrées permettant à un système dynamique régi par des équations différentielles ordinaires (ODE) de respecter à la fois des conditions initiales et finales. Ces méthodes peuvent donc être adaptées pour servir de méthode locale générique à un des algorithmes décrits au chapitre 1.6.3. Par définition, ces méthodes permettent de connecter deux configurations quelconques (initiale et finale) pour un modèle quelconque. Cependant, les problèmes numériques inhérents à ces techniques rendent cette solution difficilement exploitable, peu robuste et très lente. On pourra se référer avec utilité à [Shampine *et al.*, 2000] ou [Roberts et Shipman, 1972] pour en apprendre plus sur ces algorithmes.

4.3 Principe de fonctionnement

Du fait de nos contraintes sur le modèle et les obstacles, les techniques basées sur des équations analytiques sont inutilisables. Nous utiliserons donc ici des méthodes purement numériques aptes à gérer les critères et les contraintes non linéaires. D'après le tableau 4.1, la méthode directe de discrétisation totale est manifestement la plus adaptée. Celle-ci est suffisamment précise pour notre problème, s'adapte facilement à tout type de sollicitation, converge mieux, et permet de prendre aisément en compte les contraintes sur l'état. C'est donc elle qui sera utilisée dans la suite du chapitre. Cette technique ramène un problème de commande optimale à un problème d'optimisation non linéaire (NLP) en discrétisant les états et l'entrée. De bonnes références du domaine sont [Gill *et al.*, 1981, Nocedal et Wright, 2000]. La trajectoire du véhicule est discrétisée en un nombre n de points. En chacune de ces configurations, il faut définir une contrainte continue. Les gradients et les états adjoints sont calculés numériquement par différences finies. Le synoptique du déroulement de la procédure d'optimisation est le suivant :

1. Initialisation de l'optimisation avec un jeu de paramètres initiaux \mathbf{x}_{init}
2. Reconstitution de l'entrée correspondante \mathbf{u} par interpolation linéaire de \mathbf{x} (voir chapitre 4.5)

3. Simulation de cette entrée avec le modèle de véhicule pour obtenir une trajectoire Φ initiale
4. Sous-échantillonnage de la trajectoire en n points où seront appliquées les contraintes.
5. Calcul des contraintes géométriques appliquées en chacune de ces n configurations ("enfouissement" du véhicule dans les obstacles)
6. Calcul des contraintes supplémentaires (limitations en angle/vitesse volant, etc.)
7. A partir de ces informations l'algorithme choisi un nouveau jeu de paramètres \mathbf{x} censé rapprocher la trajectoire d'une solution et diminuant la valeur des contraintes.
8. Retour en 2 et itération jusqu'à résolution de l'ensemble des contraintes (à la tolérance définie)

On commence par résoudre le problème loin des limites en \mathcal{P} du test, puis on augmente légèrement ce paramètre en se servant du résultat précédent comme condition initiale.

4.4 Algorithme utilisé

Les méthodes d'optimisation sont étudiées depuis fort longtemps. Par conséquent, il existe sur le marché de très nombreux outils logiciels permettant la résolution de problèmes de programmation non linéaire (PNL). La méthode d'optimisation utilisée qui a été choisie est basée sur l'algorithme NLSQP (Non Linear Sequential Quadratic Programming). Dans cette méthode, un sous-problème d'optimisation quadratique est résolu à chaque itération. Tous les détails sur le fonctionnement de ces algorithmes sont disponibles dans [Han, 1977, Powell, 1978a, Powell, 1978b]. L'article [Barclay *et al.*, 1997] détaille l'utilisation de méthodes SQP dans les outils numériques de commande optimale.

Pour des raisons pratiques, l'implémentation de cette technique s'est faite via l'utilisation de la Toolbox d'optimisation de Matlab. On rappelle que l'on ne s'intéresse pas ici à la manière d'améliorer ces algorithmes mais plutôt à la manière d'en tirer les meilleurs résultats.

Comme il s'agit d'un problème de commande optimale résolu de manière directe via de la programmation non linéaire, les paramètres \mathbf{x} de l'algorithme servent à décrire l'allure de l'entrée \mathbf{u} qui sera simulée par le modèle. Les contraintes sur la trajectoire résultante sont reportées dans le critère par l'algorithme à l'aide de multiplicateurs de Lagrange ou de la méthode KKT (Karush Kuhn Tucker). L'objectif est de se ramener à un problème non contraint plus simple à résoudre.

4.5 Paramétrage de l'entrée

L'algorithme d'optimisation NLSQP requiert un jeu de paramètres \mathbf{x} en entrée sur lesquels jouer pour minimiser le critère et respecter les contraintes. Ces paramètres contenus dans \mathbf{x} servent à décrire l'entrée \mathbf{u} . Il existe de nombreuses manières de paramétrer une entrée volant quelconque. Par exemple, on peut utiliser des points de passage qui seront interpolés pour générer l'angle volant fourni au modèle. Une autre manière de faire est d'utiliser les coefficients d'une décomposition en série de Fourier de l'entrée ou bien les paramètres de Splines. Dans la pratique, c'est une interpolation linéaire entre p points de passage qui sera utilisée. Il y a donc p paramètres sur lesquels l'algorithme pourra jouer pour résoudre le problème ($\dim \mathbf{x} = p$).

4.6 Choix du critère

Dans le cadre de la résolution des problèmes numéros 1 et 2, on cherche simplement à résoudre des contraintes. Il n'est donc pas nécessaire d'utiliser un critère. On rappelle que ces deux problèmes consistent tout d'abord à trouver une solution quelconque à \mathcal{P} donné, puis à trouver la limite de ce paramètre au delà de laquelle il n'existe plus de solution. Pour calculer une solution avec \mathcal{P} fixé, il suffit de spécifier le critère comme étant une fonction nulle. On se ramène donc à un problème de résolution de contraintes.

Pour certains problèmes dont les équations sont peu différentiables, il est parfois souhaitable de rajouter un critère facilitant la convergence en lissant l'influence des contraintes. Dans la plupart des cas, cet artifice ne s'est pas avéré nécessaire grâce à la grande régularité des trajectoires issues de simulations de dynamique route. Un critère nul ne pose pas de problème dans la mesure où les contraintes y sont ramenées par l'algorithme. Le problème contraint est ainsi ramené à un autre problème non contraint plus simple à résoudre.

4.7 Choix des contraintes

Comme cela a déjà été précisé, il existe plusieurs types de contraintes :

1. Les contraintes égalité liées au mouvement dynamique du véhicule. Celles-ci sont implicitement respectées grâce à la resimulation complète du système pour l'obtention de la trajectoire.
2. Les contraintes inégalité sur la dynamique de la sollicitation volant (amplitude, vitesse, couple, etc.)
3. Les contraintes inégalité liées à la présence des obstacles. Ces contraintes sont moins régulières et plus difficiles à prendre en compte.

La trajectoire et l'entrée sont discrétisées avec n points pour le calcul, il y a donc au minimum n contraintes pour les obstacles et éventuellement d'autres si l'on souhaite brider la dynamique de l'entrée. Le calcul des contraintes liées à la limitation des angles ou vitesses volant se fait très simplement par l'application d'une contrainte du type $|u| - u_{max}$ en chaque point de l'entrée discrétisée.

Les contraintes les plus difficiles à gérer sont celles issues des obstacles car l'interaction de deux géométries est souvent très complexe. Les contraintes géométriques sont généralement des contraintes inégalités non linéaires. Dans le cas particulier du VDA, il est possible de définir des contraintes géométriques uniquement suivant y à l'aide de formules simplifiées du type :

$$\begin{aligned} q_1(x) - (y + L/2) &\geq 0 \\ (y - L/2) - q_2(x) &\geq 0 \end{aligned}$$

Où L désigne la largeur du véhicule et $q_1(x)$ et $q_2(x)$ désignent respectivement la position en y des bords gauche et droit du test en cours. Ce type de contrainte a été utilisé par [Butz et Stryk, 2005] pour optimiser le passage d'un test VDA avec un modèle simplifié de véhicule. L'algorithme utilisé est un algorithme de "collocation directe" (DIRCOL). Les résultats obtenus sont corrects mais ne garantissent en aucun cas l'absence de collisions (notamment liées à la rotation du véhicule autour de son axe de lacet), les aspects robustesse et sensibilité ne sont également pas abordés dans cet article. D'autres travaux assimilent le véhicule à un point. Ce sont des contraintes

simplifiées de ce type qui sont le plus souvent utilisées lors des travaux liés à l'optimisation de trajectoire.

La définition de ces contraintes géométriques et leur précision ont pourtant une grande influence sur la rapidité et la qualité de la convergence de l'algorithme d'optimisation. Des contraintes binaires basées sur la simple présence ou non de collisions sur la configuration en cours mènent inévitablement à l'échec de l'optimisation car une condition initiale en ligne droite constitue déjà un minimum local. Une solution plus élégante pour définir "l'enfoncement" des contraintes en chaque configuration est de définir la distance de pénétration du véhicule dans les obstacles. La définition de cette grandeur permettra de définir une contrainte variant continûment avec l'éloignement de l'espace libre. Cette prise en compte très précise de la géométrie et des collisions est une des originalités de cette thèse. Peu de travaux s'occupent d'utiliser simultanément un modèle de véhicule complexe et de prendre en compte très exactement les collisions.

4.7.1 Calcul de la distance de pénétration

Problématique

Le calcul de la distance de pénétration entre deux corps solides est connu comme étant un problème particulièrement difficile. Elle peut se définir comme étant la plus courte distance (dans l'espace des configurations) permettant à un solide de sortir d'une collision avec un autre solide. Différents paramètres sont à prendre en compte lorsque l'on souhaite calculer explicitement la distance de pénétration et notamment la convexité du véhicule et des obstacles. Il n'existe pas de résultat général lorsque les deux sont concaves. Dans notre cas, le véhicule est de forme rectangulaire et est donc convexe. Cela simplifie considérablement le problème et c'est cela qui permettra d'obtenir des résultats utilisables. Les obstacles peuvent être quelconques, on les supposera donc concaves.

Calculer la translation minimum qui permet à un véhicule de sortir d'une collision est relativement aisé et est décrit dans de nombreux articles dont [Dobkin *et al.*, 1993]. Le problème se complique singulièrement si l'on veut rajouter à ce problème la prise en compte de la rotation pour sortir de la collision, mais cela permet de trouver des solutions beaucoup plus naturelles (voir figure 4.1). Sur cette figure, la configuration initiale en collision A représente une distance de pénétration très élevée si l'on ne prend en compte que des translations pour permettre au véhicule de retourner dans l'espace libre en A' . La prise en compte de la rotation telle que cela a été effectué dans [Zhang *et al.*, 2006] permet de trouver une solution bien plus proche et naturelle représentée par A'' .

Calculer la distance de pénétration avec prise en compte des rotations revient à essayer de calculer explicitement la représentation de l'espace libre du sous-espace des configurations (x, y, ψ) . La distance de pénétration est alors la plus courte distance entre la configuration courante (en collision) et l'hypersurface délimitant cet espace libre des configurations. Dans ce cas là, la distance de pénétration devient la distance de pénétration généralisée (PD^g) conformément à la convention adoptée dans [Zhang *et al.*, 2006]. Le problème général de la détermination du PD^g entre deux obstacles concaves n'est pas encore résolu à ce jour (seul des encadrements sont disponibles).

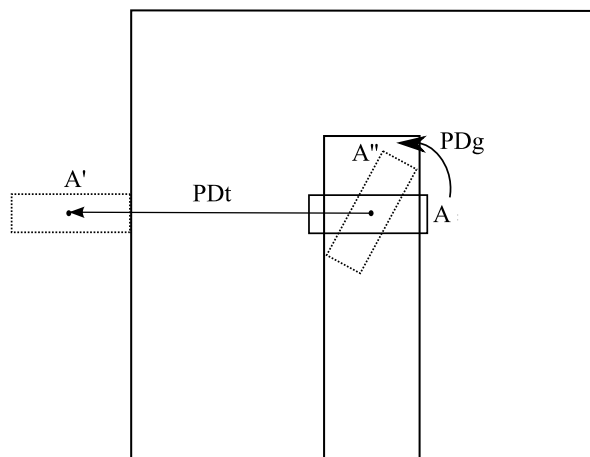


FIG. 4.1 – La prise en compte de la rotation du véhicule permet de trouver des distances de pénétration beaucoup plus réalistes

Translation seule

Dans notre cas, le véhicule et les obstacles sont bidimensionnels dans le plan (x,y) , ces deux solides sont donc représentés par des polygones fermés. Pour le moment, intéressons nous seulement à la translation du véhicule et figeons son angle de lacet ψ . Le calcul de la distance de pénétration en translation utilise la différence de Minkowski pour obtenir le polygone représentant l'espace libre des configurations. La somme de Minkowski se définit comme étant :

$$A \oplus B = \{a + b \mid a \in A \text{ and } b \in B\} \quad (4.4)$$

La différence de Minkowski est $A \ominus B = A \oplus (-B)$. Pour calculer $-B$, on choisit un point quelconque de B comme référence, et on prend l'opposé des coordonnées de chaque point de B par rapport à cette référence (voir la gauche de la figure 4.2). La différence de Minkowski consiste à soustraire à chaque point des obstacles, tous les points de $-B$. Cette opération purement géométrique prend un ensemble de points en entrée et fournit un autre ensemble de points en sortie. La différence de Minkowski de deux polygones convexes est un polygone convexe. Si les obstacles sont convexes, il est donc possible de passer de l'ensemble de points résultant au polygone associé grâce à un calcul de polygone convexe englobant (convex hull) décrit dans [Barber *et al.*, 1996].

La figure 4.2 présente une illustration de la différence de Minkowski entre un véhicule et un obstacle triangulaire convexe. Le changement de point de référence est sans importance : cela va changer l'allure du polygone de Minkowski, mais pas la valeur de la distance de pénétration. Celle-ci se calcule via la détermination de la plus courte distance entre le point de référence de la configuration à tester et ce polygone de Minkowski.

Lorsque les obstacles sont concaves, il faut décomposer ces obstacles en éléments convexes plus simples et réitérer ce calcul sur chacun de ces éléments. L'union géométrique de tous les polygones convexes résultant est calculée de manière à obtenir la différence de Minkowski finale (non nécessairement convexe). Cette union utilise l'algorithme SweepLine implémenté dans la librairie géométrique GPC. La procédure peut se résumer comme suit :

- Décomposition des obstacles concaves en un ensemble de triangles convexes par triangulation (Algorithme de Seidel, voir [Seidel, 1991])
- Différence de Minkowski entre le véhicule (convexe) et chacun de ces triangles (convexe)

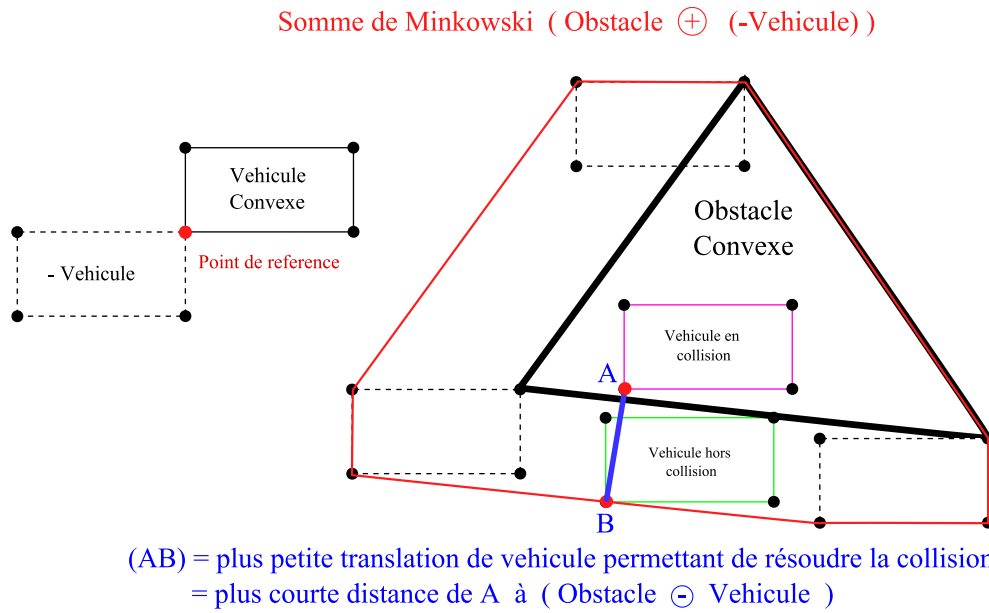


FIG. 4.2 – Calcul de la différence de Minkowski entre un véhicule convexe et un obstacle convexe

- Le polygone résultant est obtenu par calcul du polygone convexe englobant les points de chaque différence de Minkowski
- Union géométriques des polygones résultants (algorithme Sweep Line)

La figure 4.3 illustre la réalisation de cette procédure avec un obstacle concave.

Translation + Rotation

La figure 4.4 présente le polygone de Minkowski correspondant aux obstacles d'un test VDA pour un véhicule ayant un angle de lacet $\psi = 0$. La figure 4.5 présente une superposition de ces polygones pour de nombreux ψ différents. Cette superposition est une représentation approchée mais crédible de l'espace libre. L'axe vertical des z correspond à l'angle de lacet ψ du véhicule autour de son centre de gravité. Une trajectoire sans collision est une trajectoire qui reste à l'intérieur de la structure tridimensionnelle ainsi définie. Un exemple de trajectoire solution est tracé en bleu sur la figure 4.5. Pour obtenir la distance de pénétration en termes de rotation+translation, il suffit de calculer la plus courte distance entre cette hypersurface et une configuration en collision située à "l'extérieur" de cette surface. C'est la librairie PQP qui se charge de ce travail.

La figure 4.6 présente l'utilisation de ce calcul dans le cas concret d'un véhicule en collision dans le test VDA. Il y a bien utilisation conjointe de la translation et de la rotation pour trouver la plus courte distance de pénétration.

Cette distance de pénétration permet de définir de manière continue une fonction de contrainte pour chaque configuration en collision de la trajectoire discrétisée. Ce type de calcul est intéressant d'un point de vue théorique mais son utilisation pratique requiert beaucoup de mémoire et ralentit sensiblement la procédure d'optimisation. C'est pourquoi il est parfois plus opportun de s'en remettre à des méthodes fournissant une approximation de ces résultats (tout en détectant exactement les collisions).

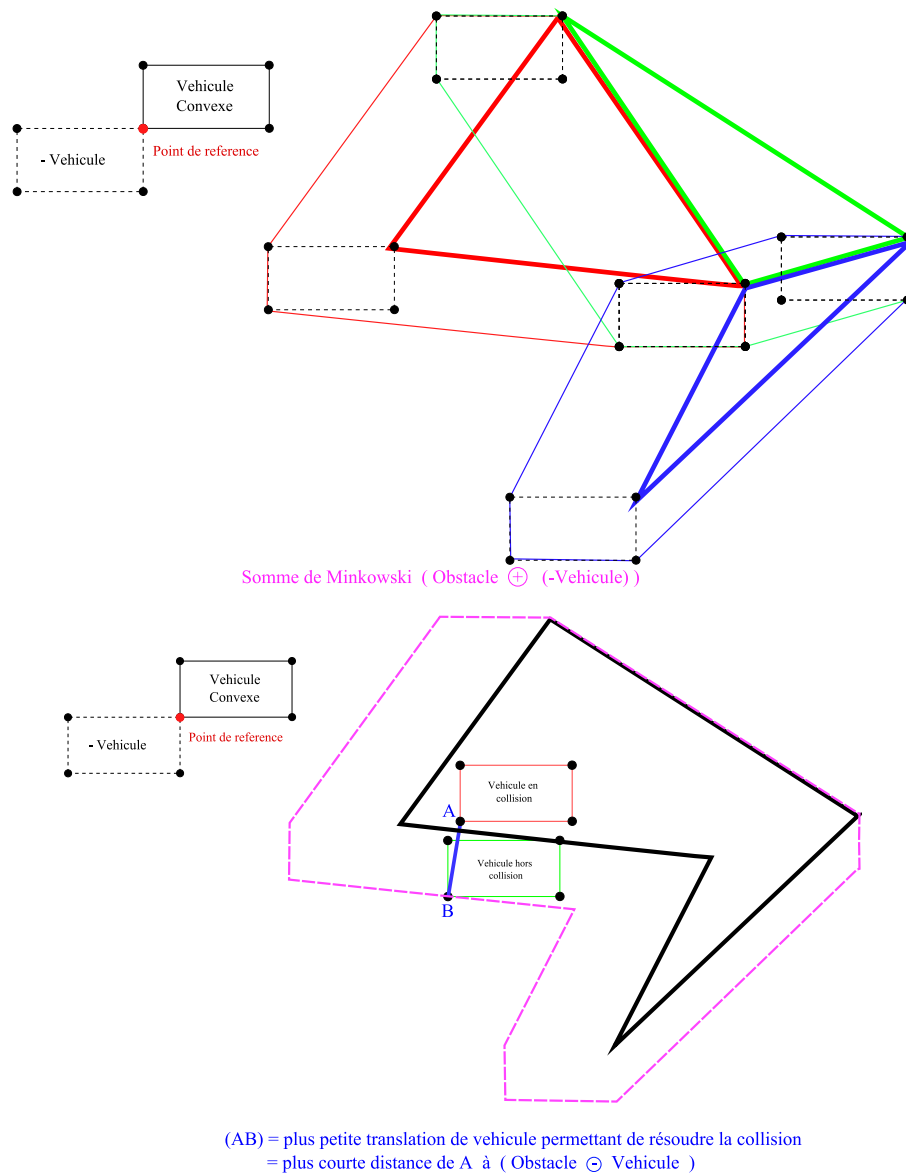


FIG. 4.3 – Calcul de la différence de Minkowski entre un véhicule convexe et un obstacle concave décomposé en triangles

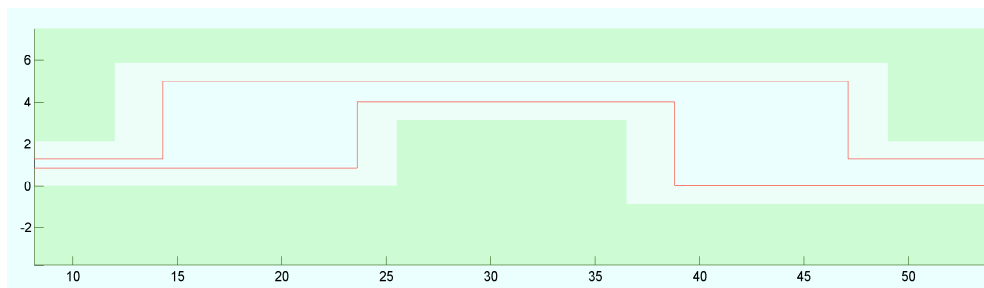


FIG. 4.4 – Polygone de Minkowski correspondant à des obstacles concaves de type VDA pour un véhicule ayant l'orientation $\psi = 0$

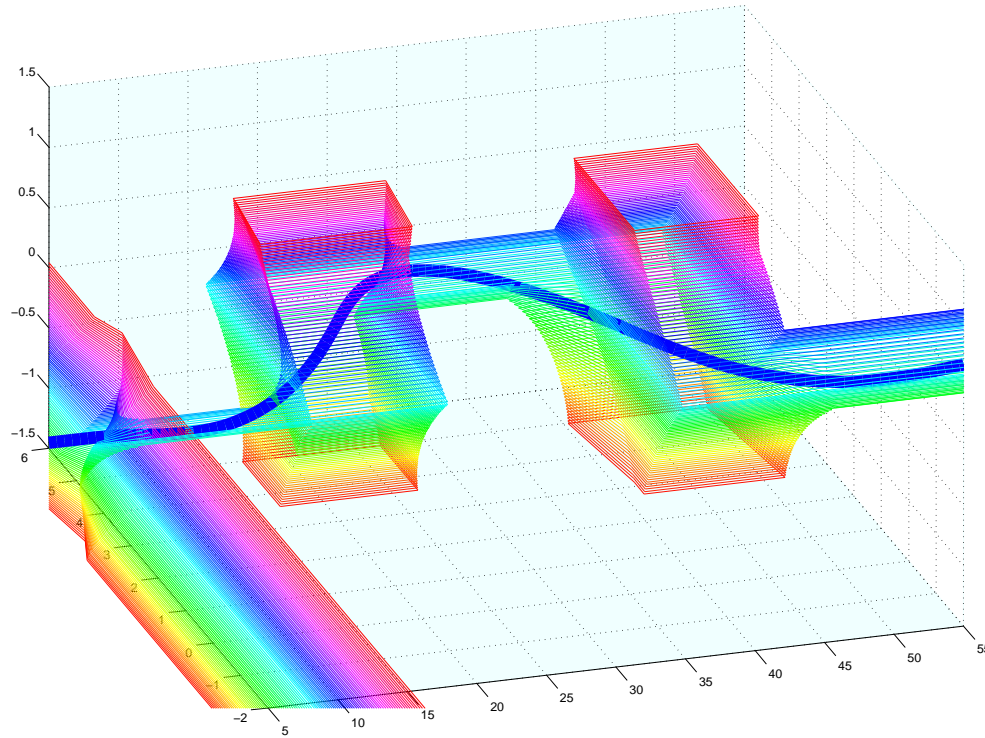


FIG. 4.5 – Superposition de polygones de Minkowski calculés pour de nombreux ψ , un exemple de trajectoire solution du VDA est donné



FIG. 4.6 – Utilisation de la distance de pénétration généralisée pour trouver la configuration hors collision la plus proche en termes de translation+rotation

4.7.2 Contraintes heuristiques

Les limitations du calcul précédent ont mené à la définition d'une approximation de la distance de pénétration qui profite des particularités communes à toutes les manœuvres existantes à ce jour (navigation en couloir notamment). L'approximation utilise la librairie GPC (General Polygon Clipper) écrite en C pour déterminer les différents polygones d'intersection entre le véhicule et les obstacles. Le barycentre $[x_c, y_c]$ de chaque polygone intersection est ensuite calculé à l'aide de la formule suivante :

$$x_c = \iint x.dxdy = \sum_{i=1}^n (6.x_i.y_i.\Delta x - 3.x_i^2.\Delta y + 3.y_i.\Delta x^2 + \Delta x^2.\Delta y)/12$$

$$y_c = \iint y.dxdy = \sum_{i=1}^n (3.y_i^2.\Delta x - 6.x_i.y_i.\Delta y - 3.x_i.\Delta y^2 - \Delta x.\Delta y^2)/12$$

Cette formule est valable pour un polygone fermé de n points de coordonnées $[x_i, y_i]$ et avec $\Delta x = x_{i+1} - x_i$ et $\Delta y = y_{i+1} - y_i$. La distance de ce barycentre au coeur libre du test est calculée à l'aide de la librairie PQP et de ses routines de calcul de distance entre deux objets quelconques (voir figure 4.7). Si plusieurs zones de collision existent c'est la somme des différentes

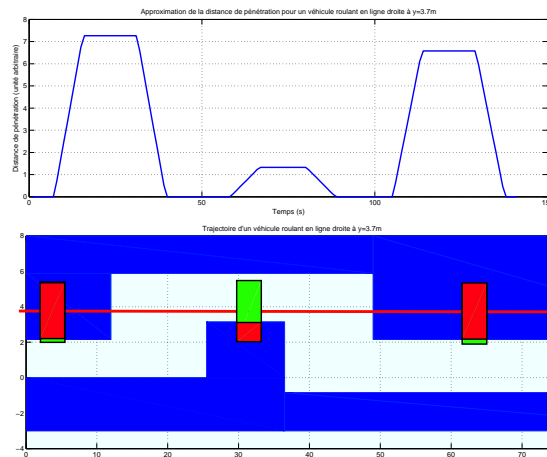


FIG. 4.7 – Il est possible d'approximer rapidement et efficacement la distance de pénétration en utilisant un algorithme de détection de collisions

distances de pénétration approchées qui est renvoyée. La valeur renvoyée est toujours continue et vaut 0 s'il n'y a pas de collision. Ces calculs ne fournissent qu'une approximation de la distance de pénétration mais se basent sur une détection exacte des collisions et offrent donc une bien meilleure prise en compte de la rotation que d'autres critères plus simples.

4.8 Paramétrage de l'algorithme

Le réglage de ces algorithmes est la principale difficulté liée à l'utilisation de l'optimisation dans l'industrie. Un bon choix de l'algorithme, un bon réglage des différents paramètres et le choix de bonnes conditions initiales sont les conditions nécessaires mais non suffisantes pour arriver au succès. Il convient de bien contraindre le système et de le placer suffisamment près

d'une solution pour obtenir la convergence. La vitesse du calcul est dépendante de la taille du vecteur \mathbf{x} et du nombre n de contraintes prises en compte.

L'algorithme dispose d'un système permettant de limiter la variation de chaque élément du vecteur \mathbf{x} entre deux itérations de l'algorithme SQP. Ce dispositif permet de limiter les trop fortes variations de l'entrée qui pourraient amener le système dans des zones aberrantes. Ce bridage ne porte que sur le paramétrage de l'entrée et non pas sur la trajectoire elle-même. Celle-ci peut donc se déplacer grandement car une faible perturbation d'entrée peut avoir de grandes conséquences (voir figures 4.12 et 1.2). Les paramètres les plus importants à surveiller pour l'algorithme sélectionné sont :

1. Les conditions initiales (angle volant).
2. Le nombre de points n de la trajectoire et donc le nombre de contraintes
3. La dimension p du vecteur de paramètres \mathbf{x} qui décrit l'entrée.
4. L'amplitude des perturbations élémentaires pour le calcul des gradients par différences finies.
5. Le pas de variation maximum de chaque élément de \mathbf{x} .
6. La tolérance acceptée sur les contraintes (pénétration maximale). Il faut en effet autoriser une certaine tolérance pour éviter à l'algorithme de trop osciller. La valeur généralement retenue est de $1e^{-4}$.
7. La tolérance sur le critère
8. Le critère d'arrêt (nombre de calcul maximum, valeur des gradients, etc.)

4.9 Résultats

4.9.1 Recherche du \mathcal{P} limite

La figure 4.8 présente le résultat de la procédure de recherche de vitesse maximale menée à l'aide de l'algorithme d'optimisation décrit ci-dessus dans le cas d'un évitement VDA. Les angles volant obtenus entre chaque incrémentation de vitesse sont superposés. L'incrément est généralement de 1 km/h , et la solution de l'optimisation précédente est utilisée comme condition initiale pour la nouvelle optimisation. La trajectoire a été discrétisée avec $n = 100$ points. $p = 20$ points de passage ont été interpolés pour décrire l'entrée \mathbf{u} .

La figure 4.9 présente une superposition similaire pour un évitement ville. On constate que les allures des différentes courbes sont proches de l'expérience de même que l'ordre de grandeur des vitesses de passage. Le temps de calcul pour une seule vitesse est de l'ordre d'une demi-heure à une heure. Lorsque l'on se rapproche des limites de \mathcal{P} le processus oscille légèrement et prend un peu plus de temps à se stabiliser. L'arrêt du calcul se fait lorsque le gradient est trop faible car l'on se trouve dans un minimum local. Le temps de calcul global (avec incrémentation de \mathcal{P}) est de plusieurs heures.

Les valeurs de V_{max} sont proches de celles obtenues dans le chapitre 5 bien que souvent inférieures de 2 ou 3 km/h . Aucune explication objective de cet écart n'a pu être avancée. La robustesse de cet algorithme vis-à-vis de variations paramétriques est identique à ce qui sera vu au chapitre 5.10.4 pour la déformation de trajectoires.

La figure 4.10 présente le résultat d'un calcul visant à déterminer la pression de freinage maximum lors d'un freinage en courbe. On constate qu'il faut débraquer le volant pour réussir à conserver le véhicule sur sa trajectoire courbe.

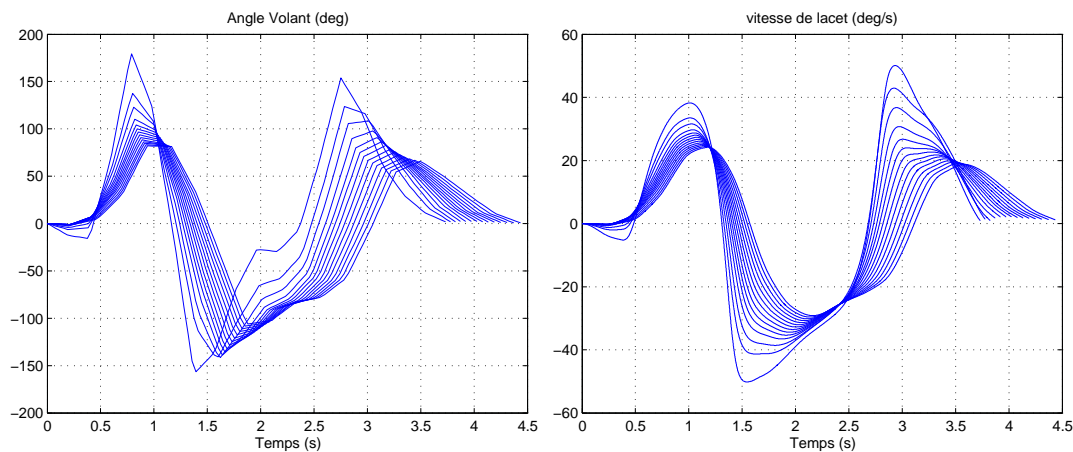


FIG. 4.8 – Résultats de la procédure d'optimisation appliquée au passage d'un test VDA avec le véhicule A pour V variant de 57.6 à 68.4 km/h

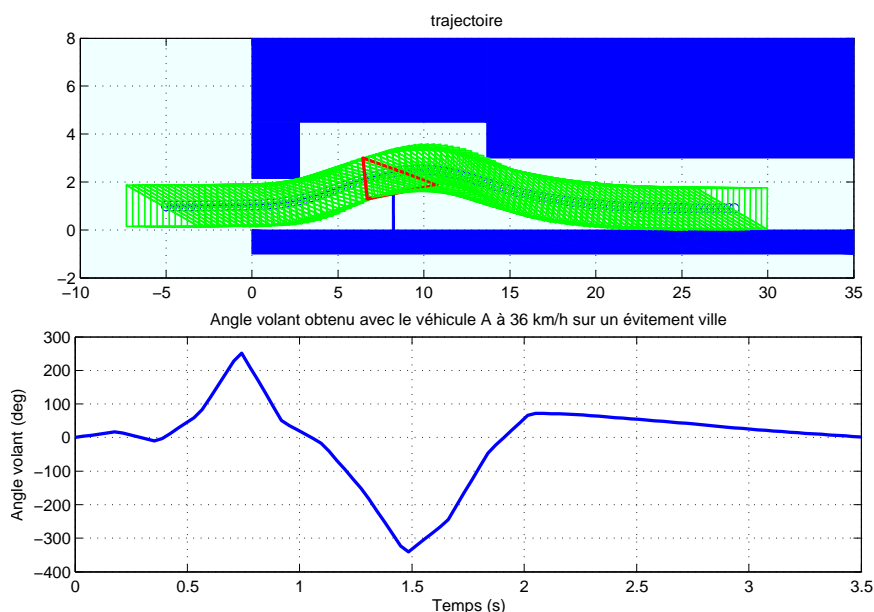


FIG. 4.9 – Exemple de trajectoire solution obtenue sur un évitement ville franchi par le véhicule A à 36 km/h

La procédure s'arrête lorsqu'une des roues est bloquée par le freinage ou que l'ABS s'enclenche. La figure 4.11 illustre ce cas limite.

4.9.2 Convergence

La figure 4.12 présente plusieurs exemples de trajectoires suivies par le véhicule au cours du déroulement de la procédure d'optimisation appliquée au cas particulier du test VDA. La trajectoire subit de fortes variations entre deux itérations, et est sollicitée de manières très différentes

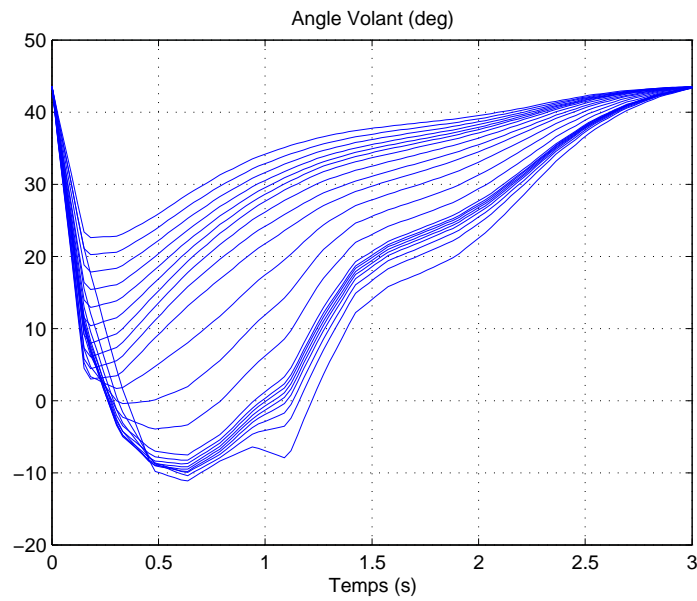


FIG. 4.10 – Résultats de la procédure d'optimisation appliquée au passage d'un freinage en courbe avec le véhicule A pour $P_{freinage}$ variant de 0 à 34 bar

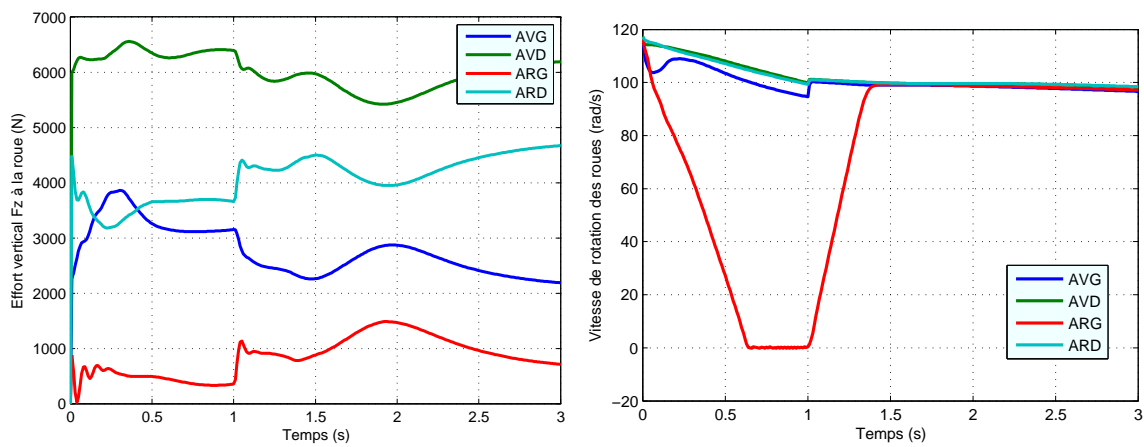


FIG. 4.11 – Illustration d'un blocage de roue lors d'un freinage en courbe ($P_{freinage} = 34$ bar)

jusqu'à la convergence vers une solution. Une faible variation sur l'entrée en début de simulation peut entraîner de grandes conséquences sur la trajectoire (et peut-être même un tête-à-queue).

Il est possible de brider les variations de l'entrée dans le paramétrage de l'algorithme mais cela ne limite aucunement ces variations sur la trajectoire. La technique de déformation de trajectoire qui sera vue au chapitre 5 ne présente pas ce genre de problèmes. L'algorithme bien paramétré converge vers une solution acceptable pour un certain nombre de problèmes. Certaines modifications requièrent cependant de fournir au modèle une condition initiale suffisamment proche du résultat pour obtenir la convergence.

4.9.3 Robustesse

Les résultats sont proches de ceux obtenus au chapitre 5.10.4 tant en termes de limites de \mathcal{P} , que de robustesse aux mêmes variations paramétriques appliquées au système. Les tests de robustesse et les valeurs correspondantes seront détaillés dans cette section afin d'éviter les répétitions.

4.10 Limitations

La principale limitation est liée à la difficulté de convergence du système qui peut facilement partir dans des domaines où le tête-à-queue est inévitable ou rester bloqué dans des minima locaux absurdes. De manière purement informatique, certaines sollicitations trop violentes où la dérive au centre de gravité dépasse 90 deg peuvent d'ailleurs faire échouer le modèle et bloquer Matlab. Ces phénomènes ont été constatés sur toutes les manœuvres essayées. Le modèle de véhicule C équipé de son ESP n'a d'ailleurs pas pu être testé faute d'avoir jamais réussi à obtenir une convergence correcte. Le comportement assez différent induit par la présence de cet équipement met donc en défaut la procédure d'optimisation.

De par leur rapidité et leur précision ces techniques ont un fort potentiel, mais la difficulté d'appréhender leur fonctionnement, les problèmes de convergence et la difficulté de paramétrage posent souvent problème.

Les méthodes heuristiques d'optimisation telles que les algorithmes génétiques, méthode tabou ou recuit simulé (voir [Bernard *et al.*,] ou [Ahuactzin *et al.*, 1992]) permettent de s'affranchir partiellement des problèmes de minima locaux dans les méthodes d'optimisation. Cependant, leur caractère aléatoire pose problème pour les études de robustesse et de sensibilité, car les résultats renvoyés peuvent être très différents entre deux exécutions successives d'un même calcul (run). Cela soulève le même type de limitations que pour les méthodes exploratoires aléatoires type RCRRT ou KDP. Les problèmes de stabilité informatique du modèle face à des sollicitations incongrues ne sont pas non plus résolus. Il a donc été décidé de ne pas développer et tester ces méthodes par ailleurs déjà fort bien documentées. De même, il existe une multitude d'autres méthodes d'optimisation qui auraient pu convenir, mais leur apprentissage et leur étude exhaustive auraient demandé bien trop de temps.

Ces limitations ont conduit à essayer une méthode hybride entre la planification de mouvements et la commande optimale quitte à dégrader légèrement les temps de calcul. C'est l'objet du chapitre 5 qui présente une technique de déformation de trajectoire, au fonctionnement aisément appréhendable, aux performances similaires, au réglage simple et à la convergence plus robuste pour ce type de problème.

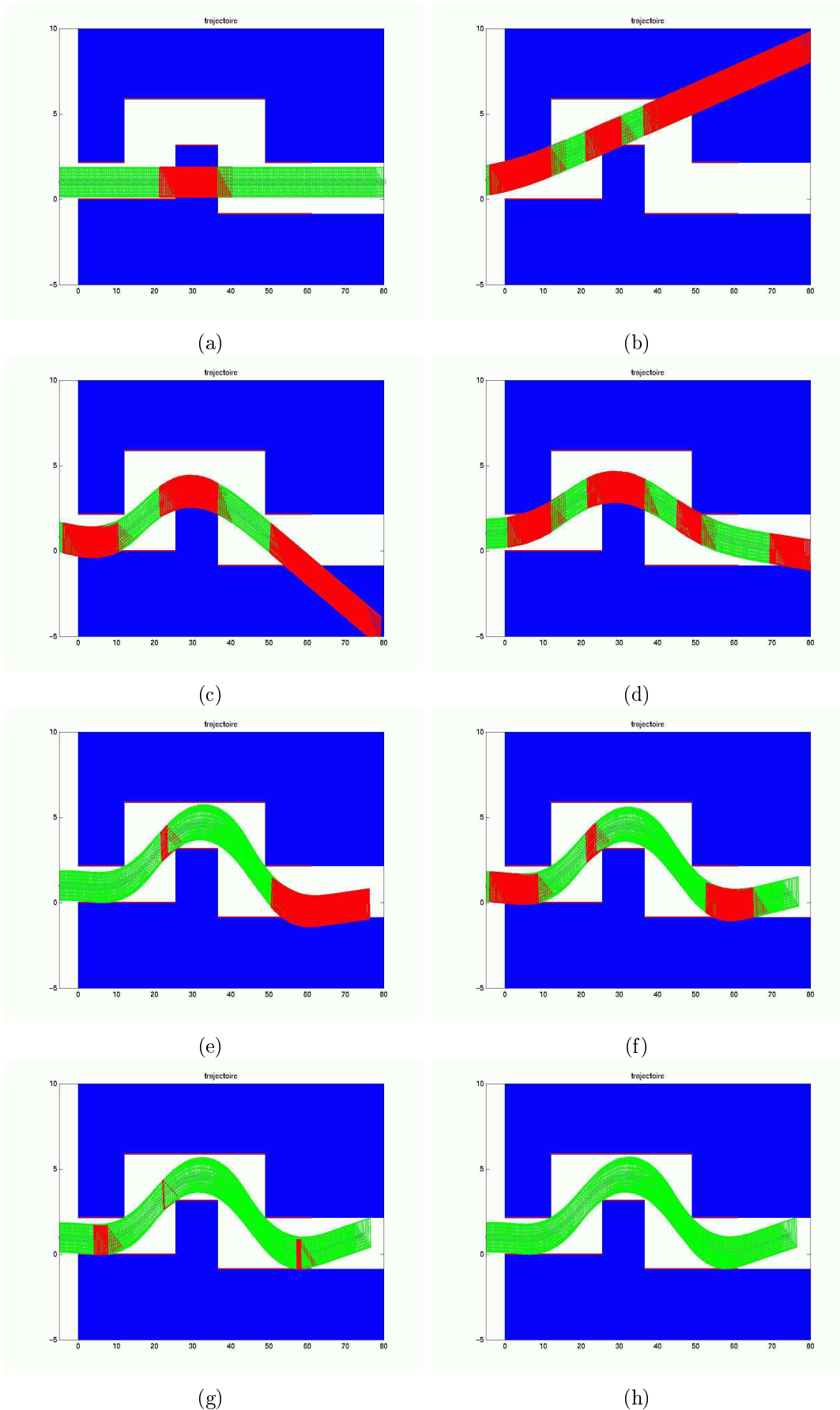


FIG. 4.12 – Évolution de la trajectoire du véhicule lors des itérations successives d'une procédure d'optimisation appliquée au test VDA

Chapitre 5

La déformation de trajectoire

5.1 Introduction

La déformation de trajectoire telle que présentée ici est inspirée des travaux de Florent Lamiroux et David Bonnafous dans [Bonnafous, 2003] et [Lamiroux et Bonnafous, 2002]. Au cours de cette thèse, cet outil a été étendu et généralisé afin de pouvoir prendre en compte efficacement les modèles de véhicules dynamiques ainsi que les collisions particulièrement nombreuses lors d'une navigation en couloir. Le problème de la limitation de la dynamique volant a également pu être correctement pris en compte. C'est cette version généralisée qui est présentée ci-dessous et dans l'article [Boyer et Lamiroux, 2006]. Ce travail constitue la principale contribution scientifique de cette thèse.

Cet algorithme a déjà été utilisé industriellement pour le compte de la société Airbus afin de valider la trajectoire d'un convoi spécial transportant une charge volumineuse (aile d'A380) au travers des routes étroites des villages de Gimont et Levignac (voir figure 5.1). Le camion utilisé possède une cinématique complexe (déplacement horizontal de la remorque), mais se déplace trop lentement pour que des phénomènes dynamiques interviennent. L'application de cet algorithme a permis de prouver l'existence d'une trajectoire solution qui a été réellement empruntée par le véhicule lors du transport des prototypes (voir [Lamiroux *et al.*, 2004b]). Ce problème de détermination de la trajectoire d'un véhicule au comportement complexe au travers d'un couloir d'obstacles étroit est suffisamment proche du problème abordé dans cette thèse pour confirmer l'intérêt de l'étude de cette méthode.

5.2 Principe de fonctionnement

Le synoptique de l'algorithme est le suivant :

- Une trajectoire initiale est obtenue à partir d'une sollicitation arbitraire \mathbf{u}_{init} (potentiellement en collision).
- Les informations sur le comportement du véhicule autour de la trajectoire en cours sont récupérées. La trajectoire est vue dans sa globalité. Ce comportement est supposé linéaire dans un petit domaine autour de la trajectoire de référence
- En chaque configuration, des forces qui permettront au véhicule de sortir de collision sont calculées. Ce calcul purement géométrique est fonction des obstacles alentour et de la configuration courante du véhicule.

- Une petite perturbation de l'angle volant est calculée. Celle-ci conjugue au mieux les forces et le comportement du véhicule de manière à ce que ce dernier s'écarte des obstacles tout en restant dans le domaine de validité des informations.
- Le modèle est resimulé en boucle ouverte avec le nouvel angle volant constitué de l'ancien angle plus la petite perturbation.
- La procédure est réitérée jusqu'à résolution de l'ensemble des collisions.

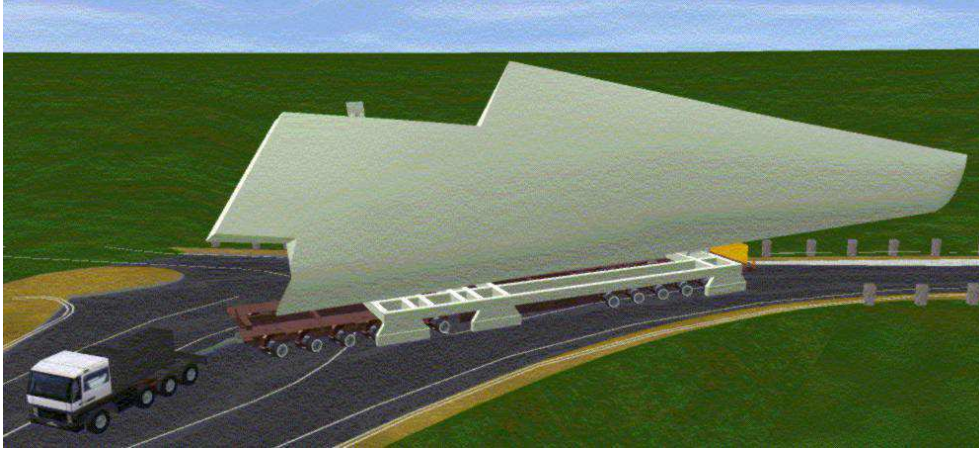


FIG. 5.1 – Utilisation de la technique de déformation de trajectoire chez Airbus pour valider des trajectoires d'un camion volumineux au travers de routes étroites

Pour trouver les limites du véhicule vis-à-vis du paramètre \mathcal{P} , on part d'une entrée quelconque (nulle) et loin des limites de \mathcal{P} , puis on déforme afin de résoudre les collisions. Une fois ce travail accompli, \mathcal{P} est incrémenté. L'entrée volant n'est alors plus vraiment adaptée à la configuration en cours, ce qui crée de nouvelles collisions avec les obstacles. Il ne reste alors plus qu'à déformer à nouveau pour résoudre ces collisions. Ce principe est illustré sur la figure 5.2 dans le cas de l'incrémentation de vitesse d'un test VDA.

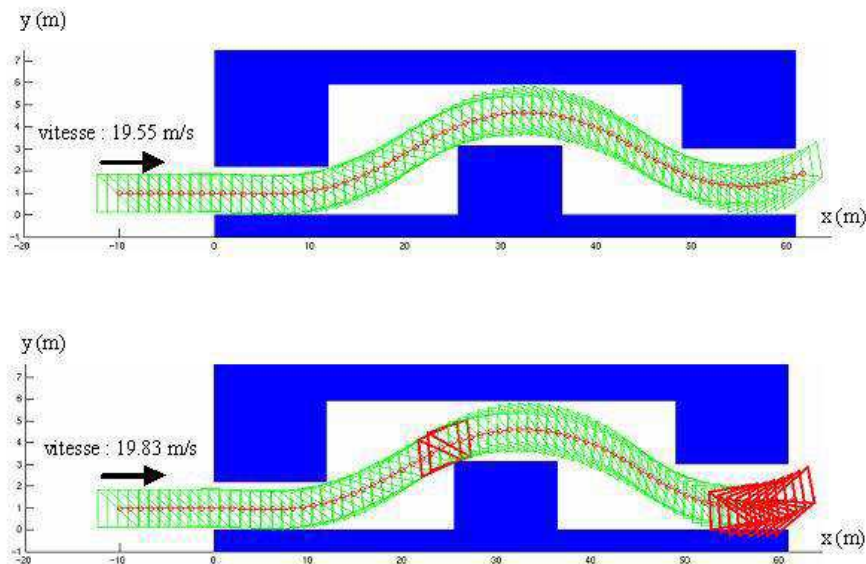


FIG. 5.2 – Incrémenter légèrement la vitesse avec un même angle volant crée de nouvelles collisions

5.3 Théorie de la déformation de trajectoire

Les entrées de l'algorithme sont :

- La définition géométrique des obstacles et du véhicule
- Le modèle dynamique du véhicule modélisé par l'équation (5.1)
- Une configuration initiale \mathbf{q}_{init}
- Une entrée initiale pour l'entrée $\mathbf{u}(s)$

5.3.1 Idée initiale

L'idée de départ de cette méthode est de déformer la trajectoire $\mathbf{q}(s)$ en perturbant l'entrée $\mathbf{u}(s)$. Par conséquent, on travaille sur un ensemble de trajectoires $\mathbf{q}(s, \tau)$ similaire à celui présenté sur la figure 5.3. Cet ensemble est indexé par le paramètre $\tau \in [0, +\infty[$ qui est incrémenté entre deux itérations (tel que cela est décrit dans [Lamiriaux *et al.*, 2004a]). $\mathbf{u}(s, \tau)$ est l'entrée correspondant à la trajectoire déformée $\mathbf{q}(s, \tau)$.

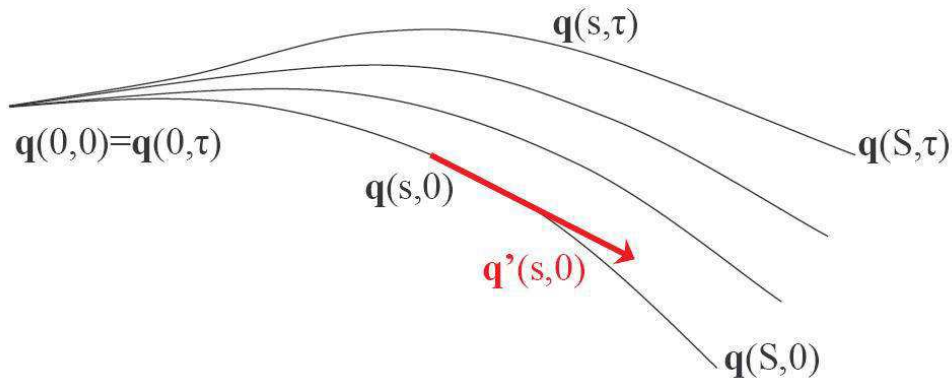


FIG. 5.3 – Chaque itération donne lieu à une trajectoire différente paramétrée par τ

L'équation explicite de notre système dynamique peut maintenant être réécrite ainsi :

$$\frac{\partial \mathbf{q}}{\partial s}(s, \tau) = \dot{\mathbf{q}}(s, \tau) = f(\mathbf{q}(s, \tau), \mathbf{u}(s, \tau)) \quad (5.1)$$

Si l'on dérive cette expression par rapport à τ , on obtient :

$$\frac{\partial \dot{\mathbf{q}}}{\partial \tau} = \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}, \mathbf{u}) \cdot \frac{\partial \mathbf{q}}{\partial \tau}(s, \tau) + \frac{\partial f}{\partial \mathbf{u}}(\mathbf{q}, \mathbf{u}) \cdot \frac{\partial \mathbf{u}}{\partial \tau}(s, \tau)$$

Définissons la matrice $\mathbf{A}(s, \tau)$ de dimensions $n \times n$ en tant que :

$$\mathbf{A}(s, \tau) = \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}(s, \tau), \mathbf{u}(s, \tau)) \quad (5.2)$$

Puis définissons la matrice $\mathbf{B}(s, \tau)$ de dimensions $n \times k$ en tant que :

$$\mathbf{B}(s, \tau) = \frac{\partial f}{\partial \mathbf{u}}(\mathbf{q}(s, \tau), \mathbf{u}(s, \tau)) \quad (5.3)$$

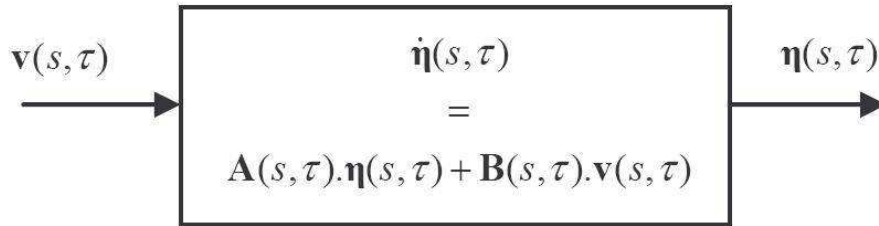


FIG. 5.4 – Ce système dynamique permet de calculer la direction de déformation induite par une perturbation quelconque

Nous appellerons respectivement *perturbation d'entrée* et *direction de déformation* les fonctions vectorielles suivantes :

$$[0, S] \times [0, +\infty[\rightarrow \mathbf{R}^k$$

$$\mathbf{v}(s, \tau) = \frac{\partial \mathbf{u}}{\partial \tau}(s, \tau)$$

et

$$[0, S] \times [0, +\infty[\rightarrow \mathbf{R}^n$$

$$\boldsymbol{\eta}(s, \tau) = \frac{\partial \mathbf{q}}{\partial \tau}(s, \tau)$$

Avec cette notation, nous obtenons l'équation différentielle (5.4) qui est le *linéarisé tangent* du système :

$$\dot{\boldsymbol{\eta}}(s, \tau) = \mathbf{A}(s, \tau) \cdot \boldsymbol{\eta}(s, \tau) + \mathbf{B}(s, \tau) \cdot \mathbf{v}(s, \tau) \quad (5.4)$$

Afin de pouvoir intégrer cette équation, la condition initiale $\boldsymbol{\eta}(s, \tau) = 0$ est rajoutée (i.e. l'état initial ne doit pas bouger). Cette équation permet le calcul de la direction de déformation $\boldsymbol{\eta}(s)$ selon laquelle la trajectoire bouge lorsqu'une perturbation $\mathbf{v}(s)$ lui est appliquée (voir figure 5.4).

5.3.2 Décomposition de la perturbation

L'ensemble des perturbations possibles est un espace de dimension C^∞ . Afin de pouvoir représenter cette perturbation, on décompose $\mathbf{v}(s)$ dans un sous-espace vectoriel de dimension finie généré par la base de $\mathbf{e} = \{\mathbf{e}_i(s)\}_{i \in [1, p]}$ de p fonctions linéairement indépendantes ($p > n$)

$$\mathbf{v}(s, \tau) = \sum_{i=1}^p \lambda_i(\tau) \mathbf{e}_i(s) \quad (5.5)$$

Comme on a supposé que le système était linéaire dans un petit domaine autour de la trajectoire de référence $\mathbf{q}(s, \tau)$, la direction de déformation correspondante $\boldsymbol{\eta}(s, \tau)$ peut être décomposée en directions de déformation élémentaires $\mathbf{E}_i(s, \tau)$:

$$\boldsymbol{\eta}(s, \tau) = \sum_{i=1}^p \lambda_i(\tau) \mathbf{E}_i(s, \tau) \quad (5.6)$$

Ces directions de déformation élémentaires peuvent être calculées pour chaque perturbation grâce à l'intégration de (5.4). On a alors :

$$\dot{\mathbf{E}}_i(s, \tau) = \mathbf{A}(s, \tau) \cdot \mathbf{e}_i(s, \tau) + \mathbf{B}(s, \tau) \cdot \mathbf{e}_i(s) \quad (5.7)$$

Ces $\mathbf{E}_i(s, \tau)$ représentent les déformations élémentaires correspondants à l'application des perturbations élémentaires $\mathbf{e}_i(s, \tau)$ au linéarisé tangent.

L'amplitude des fonctions $\mathbf{e}_i(s, \tau)$ n'a pas vraiment d'importance ici car on travaille sur un système linéarisé et non pas par différences finies sur un modèle numérique. Ce sont ces fonctions $\mathbf{E}_i(s, \tau)$ qui représentent la connaissance locale du système autour de la trajectoire courante. Cette information n'est valable que dans un petit domaine autour de cette trajectoire et il faudra bien prendre garde à ne pas déplacer la trajectoire hors de ce domaine entre deux itérations.

5.3.3 La fonction de potentiel

Un potentiel $U(\mathbf{q}(s, \tau))$ est introduit pour chaque configuration d'état. Ce potentiel est conçu pour être fort lorsque la configuration est proche des obstacles et nul autrement (voir figure 5.5). Pour caractériser la proximité de l'ensemble de la trajectoire envers l'ensemble des obstacles, nous introduisons une fonction de potentiel global $V(\tau)$ qui intègre l'ensemble des potentiels élémentaires $U(\mathbf{q}(s, \tau))$ le long de la trajectoire :

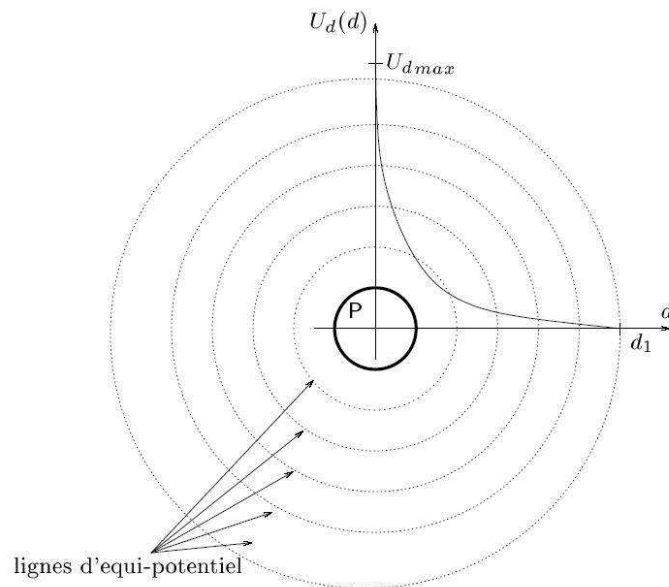


FIG. 5.5 – Exemple de définition possible pour la fonction de potentiel U . Le potentiel est fort lorsque l'on est près des obstacles et nul lorsque la distance d devient trop grande.

$$V(\tau) = \int_0^S U(\mathbf{q}(s, \tau)) ds$$

Le but de la déformation est de s'écarter des obstacles et donc de voir ce potentiel diminuer, par conséquent il faut :

$$\frac{dV}{d\tau}(\tau) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \cdot \frac{\partial \mathbf{q}}{\partial \tau}(s, \tau) \cdot ds < 0$$

i.e.

$$\frac{dV}{d\tau}(\tau) = \sum_{i=1}^p \lambda_i(\tau) \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \cdot \mathbf{E}_i(s, \tau) \cdot ds < 0$$

Cela est assuré à condition de choisir

$$\lambda_i(\tau) = - \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \cdot \mathbf{E}_i(s, \tau) \cdot ds \quad (5.8)$$

Dans ce cas là, on a bien $\frac{dV}{d\tau}(\tau) = \sum_{i=1}^p -(\int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \cdot \mathbf{E}_i(s, \tau) \cdot ds)^2$ qui est une grandeur toujours strictement négative. $V(\tau)$ est donc toujours décroissant entre deux itérations. Le véhicule s'écarte bien des obstacles locaux.

Le gradient $\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau))$ de dimension $n \times 1$ peut être assimilé à une force dans l'espace des configurations (voir figure 5.6), cette force pousse la trajectoire loin des obstacles. La direction de déformation $\boldsymbol{\eta}(s, \tau)$ est calculée grâce aux équations (5.6) et (5.8). Ces directions sont différentes de celles des forces, car elles prennent également en compte le comportement dynamique du système. C'est cela qui permet à cet algorithme d'anticiper les effets de l'inertie et de la dérive pour résoudre les collisions représentées par les forces.

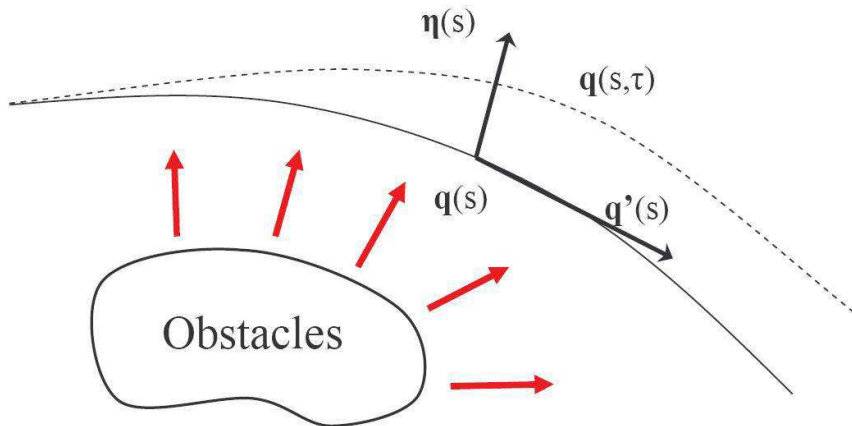


FIG. 5.6 – Les forces générées par les obstacles poussent la trajectoire loin de ceux-ci

5.3.4 Perturbation des entrées

L'intensité maximum des forces peut être quelconque dans la mesure où seule leur direction $\boldsymbol{\eta}$ importe pour le calcul. Par conséquent, les paramètres $\Delta\tau$ et η_{max} sont introduits afin de limiter l'intensité de la déformation de trajectoires résultantes. Le paramètre η_{max} est choisi expérimentalement de manière à s'assurer que la trajectoire ne se déplace pas trop entre deux itérations et respectera l'hypothèse de linéarité locale représentée par l'équation (5.4). Pour obtenir ce résultat, on impose :

$$\begin{cases} \Delta\tau = \frac{\eta_{max}}{\|\boldsymbol{\eta}\|} \text{ si } \|\boldsymbol{\eta}\| \geq \eta_{max} \\ \Delta\tau = 1 \text{ sinon} \end{cases}$$

Pour déformer l'entrée, il suffit d'ajouter à l'entrée courante la perturbation calculée avec (5.5) et (5.8) et de limiter l'intensité de la déformation avec le coefficient $\Delta\tau$:

$$\mathbf{u}(s, \tau) = \mathbf{u}(s, \tau) + \Delta\tau \cdot \mathbf{v}(s, \tau) \quad (5.9)$$

L'itération est maintenant terminée, il suffit d'incrémenter l'index τ de l'ensemble de trajectoires et de réitérer la procédure jusqu'à résolution de l'ensemble des collisions.

$$\tau \leftarrow \tau + \Delta\tau$$

5.3.5 Résumé de l'algorithme

L'algorithme consiste à itérer sur la boucle suivante jusqu'à ce que toutes les collisions aient été éliminées :

1. Calculer $\mathbf{q}(s)$ par intégration de (5.1) avec $\mathbf{u}(s)$
2. Calculer $\mathbf{A}(s)$ et $\mathbf{B}(s)$ par différences finies de (5.2) et (5.3) ou analytiquement
3. Calculer chaque $\mathbf{E}_i(s, \tau)$ par intégration de (5.4)
4. Calculer le gradient de $U(s)$
5. Calculer chaque λ_i par intégration de (5.8)
6. Calculer $\mathbf{v}(s)$ en utilisant (5.5)
7. Déformer les entrées en utilisant (5.9)

5.4 Calcul des forces

Le calcul pratique des différentes forces appliquées peut se faire de différentes manières suivant le type de calcul que l'on souhaite effectuer. Avant tout il faut savoir que ces forces existent dans l'espace d'état et peuvent donc s'appliquer autant sur la position x, y que sur l'angle de lacet ψ ou sur toute autre grandeur qui serait disponible en sortie du système. Les méthodes exactes de détection de collisions qui sont utilisées ici garantissent toujours une parfaite prise en compte des collisions liées aux rotations ψ du véhicule autour de l'axe vertical passant par son cdg.

5.4.1 Distance aux obstacles

Lorsque l'on se trouve dans un cas où les collisions sont peu nombreuses et la distance à parcourir relativement importante, il est possible d'utiliser directement la distance aux obstacles comme gradient du potentiel U (voir chapitre 5.3.3). Ce résultat a été obtenu par Lefebvre et Lamiriaux dans [Lefebvre *et al.*, 2005]. Une fonction spécifique est utilisée afin de pondérer les forces. Lorsque la distance aux obstacles est faible, les forces sont importantes. Au-delà d'une certaine distance d'influence les forces deviennent même nulles (voir figure 5.5). En cas de collision, les forces sont mises à 0. Ce sont donc les configurations aux alentours des obstacles qui bénéficient de forces importantes et qui vont tirer les configurations en collision vers l'espace libre.

Lorsque les collisions sont trop nombreuses ou l'espace libre trop réduit, ce mécanisme est mis en échec. La résolution des tests en couloir souffre typiquement de ce problème. Cette manière de faire n'a pas pu être utilisée avec succès lors de cette thèse. Face à la prépondérance des collisions, il faut alors envisager des forces basées sur la prise en compte directe de la distance de pénétration à l'intérieur des obstacles.

5.4.2 Distance de pénétration

De même qu'il a été fait au chapitre 4.7.1, il est possible d'utiliser le gradient de la distance de pénétration pour déterminer le gradient de U par rapport à \mathbf{q} et générer des forces. Une autre possibilité est d'utiliser la librairie GPC qui peut fournir une approximation de la distance de pénétration comme cela a été présenté au chapitre 4.7.2. On obtient alors une valeur de U qu'il faut dériver par rapport à \mathbf{q} pour obtenir les forces. Ces approches bien que parfaitement utilisables théoriquement, ralentissent fortement le calcul, sont moins précises et posent certains problèmes de dérivabilité de la distance de pénétration par rapport à l'angle de lacet. Elles n'ont donc pas été retenues lors des calculs présentés au chapitre 5.10.

5.4.3 Forces heuristiques

Pour le problème étudié ici, il est possible de profiter d'un certain nombre de simplifications. En effet, toutes les manœuvres étudiées se placent dans un cas particulier où le véhicule navigue dans un couloir. On peut donc facilement calculer les normales à ces couloirs et s'en servir comme axe directeur pour les forces. Ces forces vont toujours ramener le véhicule vers le centre du couloir.

Lors d'une collision, le gradient de potentiel peut être imposé directement sans passer par le calcul explicite d'un champ de potentiel U tel que présenté sur la figure 5.7. La force appliquée est toujours la même que l'on soit en bord de collision ou loin à l'intérieur des obstacles. Une simple détection de collision binaire suffit à ce calcul. L'intensité K des forces est sans importance car la déformation maximum de la trajectoire est limitée par l'algorithme. La figure 5.8 donne un exemple de déformation $\boldsymbol{\eta}$ obtenue lors de l'application de forces purement normales aux obstacles (suivant y). Dans ce cas-là, on a :

$$\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) = \begin{pmatrix} \vdots \\ \frac{\partial U}{\partial y} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \pm K \\ \vdots \end{pmatrix} \quad (5.10)$$

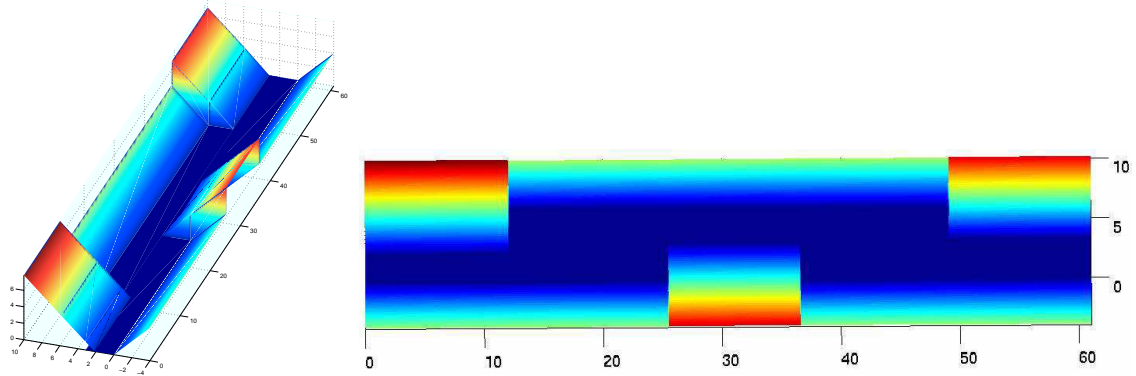


FIG. 5.7 – Exemple de champ de potentiel utilisable sur un VDA

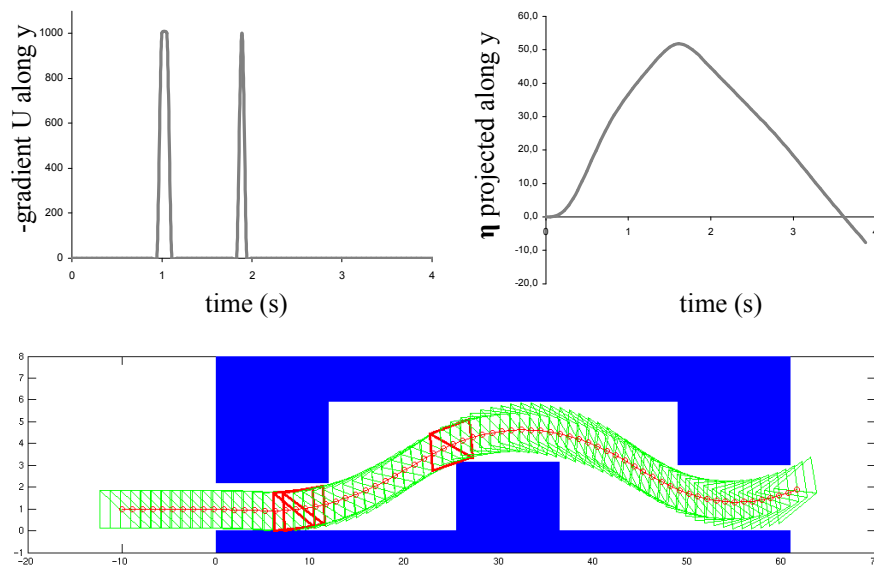


FIG. 5.8 – Exemple de forces nées des collisions avec les obstacles d'un VDA pour le véhicule A. Le η associé est projeté suivant y

Avec les algorithmes d'optimisation vus au chapitre 4, il est possible d'imposer manuellement le gradient des contraintes par rapport au vecteur de paramètre \mathbf{x} . Mais ce gradient est impossible à fournir manuellement, l'algorithme passe donc par un calcul de différences finies. Imposer manuellement le gradient des contraintes (ou de U) par rapport à la configuration \mathbf{q} est beaucoup plus intéressant. Ce gradient a en effet une signification physique et une valeur aisément appréhendable. Cette approche qui se base sur des forces dans l'espace des configurations physiques et non dans l'espace abstrait des paramètres \mathbf{x} est un des grands avantages de la DT.

5.5 Choix de la base de décomposition $\{\mathbf{e}_i\}_{i=[1,p]}$

Ce choix n'est pas anodin car il conditionne le sous-ensemble des entrées potentiellement utilisables par l'algorithme. La base peut-être de tout type : base de Fourier, de triangles ou même d'exponentielles. Il est important que la base soit orthogonale afin de récupérer des informations non redondantes sur le système. La figure 5.9 présente un exemple de pseudo base de Fourier et de triangles. La base de triangles n'est pas exactement orthogonale mais en constitue une approximation acceptable. Cette dernière est la plus utilisée en pratique car elle permet d'avoir une entrée continue constituée de segments. Les bases de Fourier génèrent parfois des effets de bord ou des oscillations inutiles qui sont préjudiciables à la qualité de la déformation.

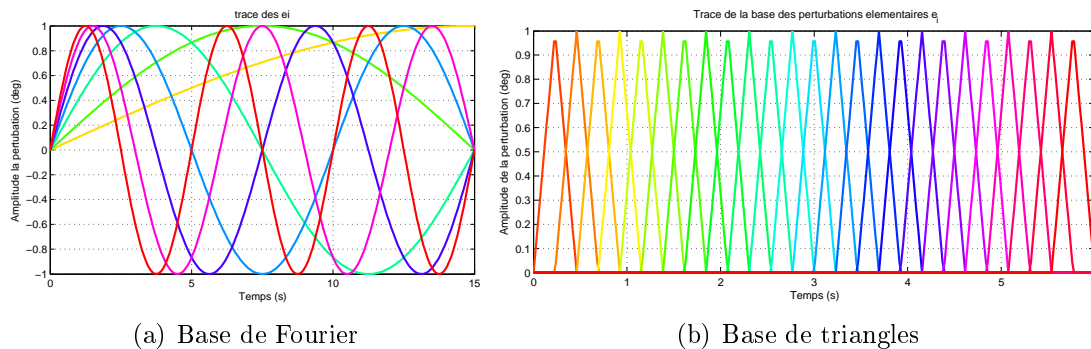


FIG. 5.9 – Différentes bases de décomposition sont possibles pour l'entrée

5.6 Calcul numérique des \mathbf{E}_i

L'algorithme initial prévoit de calculer explicitement les matrices $\mathbf{A}(\mathbf{s})$ et $\mathbf{B}(\mathbf{s})$ en chaque point de la trajectoire discrétisée, et ce pour chaque itération afin de pouvoir obtenir les \mathbf{E}_i . Cette approche est avantageuse dans le cas où l'équation analytique du système est connue et que ces matrices peuvent être différenciées de manière formelle. Leur calcul est alors très rapide. Dans le cas des systèmes dynamiques complexes étudiés ici, le calcul des gradients présents dans les matrices \mathbf{A} et \mathbf{B} est uniquement possible numériquement par différences finies. Ces calculs approchés sont peu précis et créent des problèmes de stabilité numérique de l'algorithme.

Face à ces problèmes, la détermination des \mathbf{E}_i a été améliorée en les calculant directement à partir du modèle. Il suffit pour cela de faire la différence entre la trajectoire initiale \mathbf{q} et une trajectoire issue de la simulation de l'entrée initiale plus une perturbation élémentaire \mathbf{e}_i . L'amplitude de cette perturbation doit être très faible de manière à avoir un très léger déplacement

de la trajectoire. Le \mathbf{E}_i résultant est donc issu d'un calcul par différences finies portant sur la trajectoire entière. La figure 5.10 illustre bien le fonctionnement de cette méthode. Cette manière

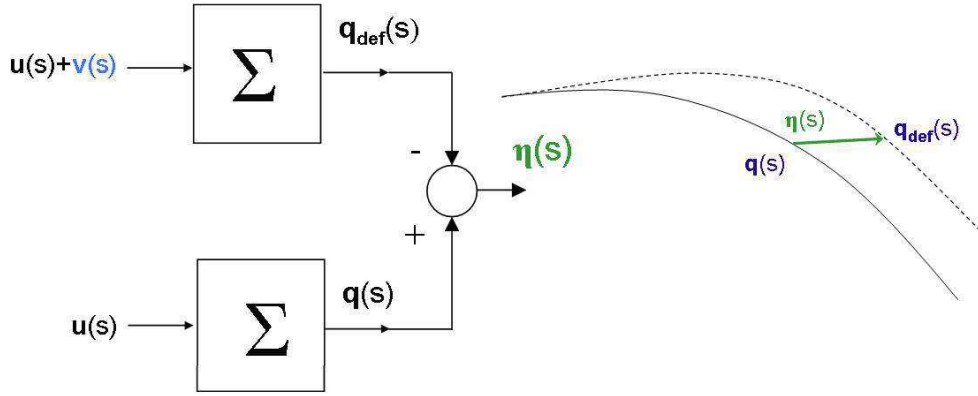


FIG. 5.10 – Le calcul de $\boldsymbol{\eta}$ se fait numériquement par soustraction d'une trajectoire initiale et d'une trajectoire perturbée

de faire s'est révélée plus précise et robuste que la méthode utilisant les matrices \mathbf{A} et \mathbf{B} , et ce dans la plupart des cas étudiés. Elle a cependant le défaut d'augmenter linéairement le temps de calcul avec la dimension p de la base de fonctions. La méthode utilisant le calcul explicite des matrices \mathbf{A} et \mathbf{B} augmente le temps de calcul linéairement avec le nombre de points n de la trajectoire discrétisée. Dans ce dernier cas, le temps de calcul des \mathbf{E}_i est anecdotique une fois ces informations acquises.

La figure 5.11 présente l'évolution de ces \mathbf{E}_i sur l'exemple d'un test VDA. Chaque figure correspond à un test réussi. La vitesse à l'entrée du couloir est croissante. La base $\{\mathbf{e}_i\}_{i=1,p}$ est constituée de triangles, et les \mathbf{E}_i représentés sont projetés uniquement suivant l'angle de lacet ψ . On voit sur cette succession de figures que le véhicule perd de plus en plus de sa capacité à tourner et à se décaler dans les couloirs. Or c'est cette capacité qui est essentielle pour les changements de files d'un VDA. Ce sont bien ces phénomènes dynamiques qui expliquent l'existence d'une vitesse de passage maximum. Initialement, le véhicule va lentement et se comporte donc quasiment cinématiquement. Ajouter une perturbation à n'importe quel instant donne donc toujours la même variation de l'angle de lacet. Par contre, une fois que l'accélération latérale devient plus élevée, on voit que le comportement local du véhicule se complexifie.

La figure 5.12 présente une comparaison des \mathbf{E}_i entre une trajectoire initiale en ligne droite et une autre autour d'une trajectoire solution d'un VDA. Seules les projections suivant y et ψ sont représentées. Le véhicule A entre dans le VDA à la même vitesse dans les deux cas. Comme précédemment, l'allure des \mathbf{E}_i autour de la trajectoire solution est nettement plus tourmentée, témoignant de l'utilisation du potentiel du véhicule. En ligne droite, une perturbation appliquée à un moment ou à un autre donne toujours le même résultat, car la capacité du véhicule à se décaler latéralement n'est pas entamée.

Si l'on souhaite utiliser une base de $\{\mathbf{e}_i\}_{i=1,p}$ avec beaucoup de fonctions il peut être plus intéressant de passer par un calcul explicite des matrices \mathbf{A} et \mathbf{B} (par différences finies) car celui-ci n'est pas fonction de la dimension p de cette base, mais uniquement du nombre de points n de la trajectoire.

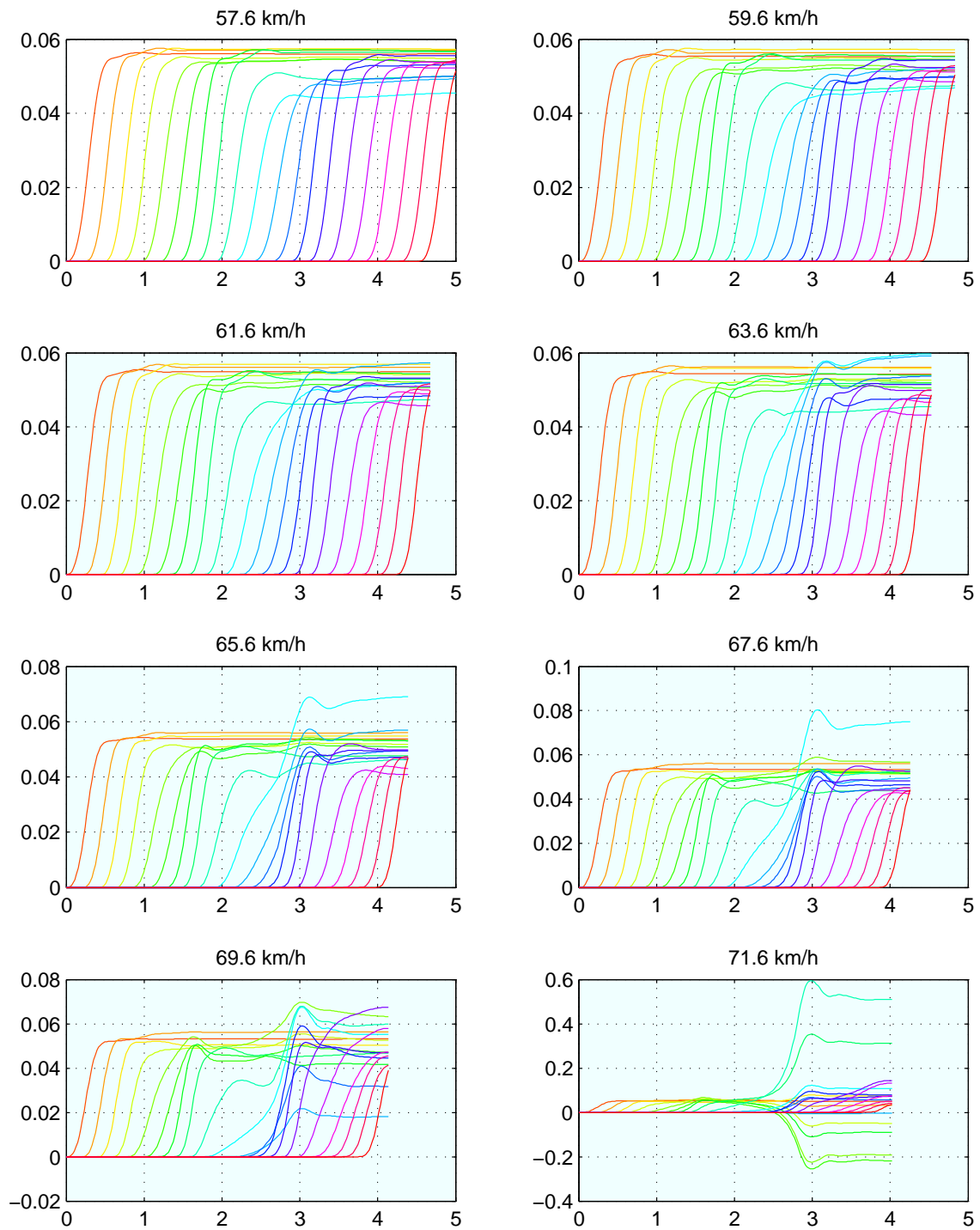


FIG. 5.11 – Comparaison de l'évolution des \mathbf{E}_b sur un test VDA en fonction de la vitesse de passage pour une base $\{\mathbf{e}_i\}_{i=[1,p]}$ de triangles

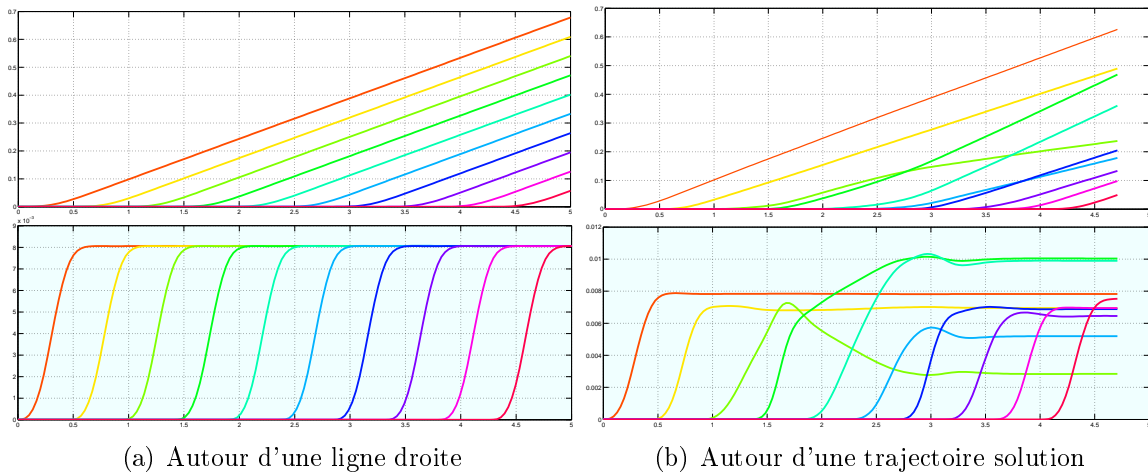


FIG. 5.12 – Comparaison des \mathbf{E}_y et \mathbf{E}_ψ entre une trajectoire initiale en ligne droite et autour d’une trajectoire solution. Le véhicule A entre dans le VDA à la même vitesse dans les deux cas.

5.7 Conditions aux limites et orthonormalisation

Conditions aux limites Si l’on applique brutalement la méthode vue précédemment, la procédure de déformation oscille beaucoup et met du temps à converger. Afin d’accélérer le processus de déformation, il est possible d’utiliser certaines astuces mathématiques portant sur les valeurs des λ_i calculées par l’équation (5.8). Comme l’on travaille directement avec des trajectoires et non pas dans l’espace des entrées comme avec l’optimisation, il est possible d’imposer à la trajectoire perturbée de ne pas se déplacer selon y ou ψ en un point quelconque de la trajectoire. Dans le cas du VDA, on choisit généralement la configuration finale. Il suffit pour cela de projeter le $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ sur le sous-espace des perturbations qui respectent :

$$\boldsymbol{\eta}(S, \tau) = \sum_{i=1}^p \lambda_i(\tau) \cdot \mathbf{E}_i(S, \tau) = 0_{\mathbb{R}^m} \quad (5.11)$$

Le symbole S désigne le temps à la fin de la trajectoire déformée et m est la dimension de l’espace d’état. On définit la matrice \mathbf{L} de dimensions $m \times p$ comme étant égale à $\mathbf{L} = (\mathbf{E}_1(S, \tau), \dots, \mathbf{E}_p(S, \tau))$. Le vecteur $\boldsymbol{\lambda}^{proj}$ doit donc respecter $\mathbf{L} \cdot \boldsymbol{\lambda}^{proj} = 0$, ce qui est assuré si :

$$\boldsymbol{\lambda}^{proj} = (\mathbf{I}_p - \mathbf{L}^+ \cdot \mathbf{L}) \cdot \boldsymbol{\lambda} \quad (5.12)$$

Le symbole \mathbf{L}^+ désigne la pseudo-inverse de la matrice non carrée \mathbf{L} . La pseudo-inverse d’une matrice quelconque se calcule par la formule suivante : $\mathbf{L}^+ = \mathbf{L}^T \cdot (\mathbf{L} \cdot \mathbf{L}^T)^{-1}$. La notation \mathbf{L}^T désigne la transposée de la matrice \mathbf{L} tandis que \mathbf{I}_p désigne la matrice identité carrée de dimensions $p \times p$. Dans le cas du VDA, bloquer la déformation selon x n’aurait pas de sens et empêcherait le processus de se dérouler correctement. Il faut donc forcer à 0 les lignes de la matrice \mathbf{L} que l’on souhaite continuer à voir évoluer. La figure 5.13 illustre bien l’impact de cette manipulation mathématique. Pour le freinage en courbe cette astuce n’est pas utilisée, car la position finale peut beaucoup varier du fait du ralentissement du véhicule.

Il est possible d’observer une légère dérive du procédé due aux approximations précédentes. Les trajectoires réellement obtenues après resimulation ne sont jamais rigoureusement conformes

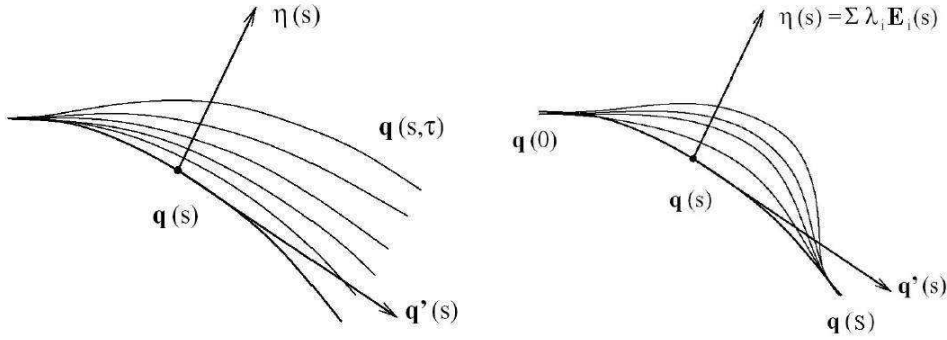


FIG. 5.13 – Impact de la projection des λ_i sur le sous-espace des perturbations imposant $\eta(S, \tau) = 0$

à celles prédites par l'algorithme qui ne se base que sur les informations locales des \mathbf{E}_i . Cela ne pose cependant aucun problème, car en cas de collision les forces viennent toujours rétablir l'équilibre.

Orthonormalisation des \mathbf{E}_i Il a été montré dans la thèse [Bonnafous, 2003] qu'il est toujours plus intéressant d'utiliser une base orthonormée de \mathbf{E}_i pour le produit scalaire L^2 . En effet, travailler dans une base orthonormée permet d'assurer une décroissance beaucoup plus forte au potentiel global V , et donc une convergence plus rapide. Les déformations issues de l'utilisation d'une base de \mathbf{E} orthonormée utilisent bien mieux le potentiel du véhicule et convergent beaucoup plus vite. Pour orthonormaliser la base de \mathbf{E}_i , il faut commencer par calculer la matrice de Gram \mathbf{G} associée à cet ensemble de fonctions. On a ainsi :

$$\mathbf{G} = \begin{pmatrix} \langle \mathbf{E}_1 | \mathbf{E}_1 \rangle & \cdots & \langle \mathbf{E}_1 | \mathbf{E}_p \rangle \\ \vdots & & \vdots \\ \langle \mathbf{E}_p | \mathbf{E}_1 \rangle & \cdots & \langle \mathbf{E}_p | \mathbf{E}_p \rangle \end{pmatrix} \quad (5.13)$$

où $\langle \mathbf{E}_m | \mathbf{E}_n \rangle = \int_0^S \mathbf{E}_m^T(s) \cdot \mathbf{E}_n(s) \cdot ds$ définit un produit scalaire dans l'espace des fonctions L^2 .

L'orthonormalisation de la matrice de Gram se fait via la très classique procédure d'orthonormalisation de Gram-Schmidt. Celle-ci fournit en sortie la matrice de passage \mathbf{P} qui permet le passage de la base des \mathbf{E}_i initiale à la même base orthonormée. Le nouveau λ^{proj} à utiliser se calcule alors avec :

$$\lambda^{proj} = \mathbf{P} \cdot \mathbf{P}^T \cdot \lambda \quad (5.14)$$

Les deux méthodes peuvent être utilisées conjointement sans aucune limitation. Dans ce cas, le nouveau λ se calcule à l'aide de la formule :

$$\lambda^{proj} = (\mathbf{I}_p - \mathbf{P} \cdot (\mathbf{L} \cdot \mathbf{P})^+ \cdot \mathbf{L}) \cdot \mathbf{P} \cdot \mathbf{P}^T \cdot \lambda \quad (5.15)$$

Les vecteurs \mathbf{E}_i de dimension m sont homogènes à des vecteurs d'état \mathbf{q} . Ils contiennent donc à la fois des longueurs, des angles ou des vitesses. L'orthonormalisation ne peut se faire sans une homogénéisation des unités de ces grandeurs. Cette harmonisation se fait via l'application de coefficients adéquats.

La figure 5.14 illustre bien l'impact des différentes modifications possibles. On voit que ces modifications permettent au potentiel global V de descendre entre deux itérations. Afin d'évaluer l'importance de la décroissance de V , on calcule la grandeur $\frac{dV}{d\tau}(\tau) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \cdot \boldsymbol{\eta}(s, \tau)$, qui représente bien la variation de ce potentiel d'une itération à l'autre. Sans aucun traitement $\frac{dV}{d\tau}(\tau) = 0.44$, avec orthonormalisation $\frac{dV}{d\tau}(\tau) = 1.06$. L'allure de la perturbation \mathbf{v} résultante est également assez fortement impactée.

Une fois toutes ces manipulations effectuées il suffit de recalculer la perturbation \mathbf{v} à l'aide de l'équation (5.5) et du nouveau $\boldsymbol{\lambda}$.

5.8 Prise en compte de contraintes sur la commande

Le conducteur du véhicule étant un être humain, il est nécessaire de borner les vitesses et les accélérations angulaires à des valeurs physiologiquement réalistes (voir chapitre 2.3.2). Cette prise en compte est très facile à implémenter en rajoutant simplement une sortie supplémentaire au système. Celle-ci sera considérée comme étant une variable d'état supplémentaire sur laquelle on peut appliquer une force de manière à la contraindre à rester dans un domaine défini. Le vecteur \mathbf{q} contient alors des sorties du modèle au sens des automaticiens et non le vrai état interne. Cette sortie peut même être l'entrée, la dérivée de l'entrée ou le couple volant reconstitué.

Une illustration de cette technique est présentée en figure 5.15 avec le modèle Simulink d'*Hilare 2*. La procédure de déformation étant très progressive avec un départ en ligne droite, elle ne fournit que très rarement des sollicitations volant importantes. Généralement, c'est plutôt la solution présentant le moins de perturbations de l'angle volant qui ressort en premier. Cette prise en compte n'est donc nécessaire que pour les évitements ville où les dynamiques volant sont importantes même à faible vitesse.

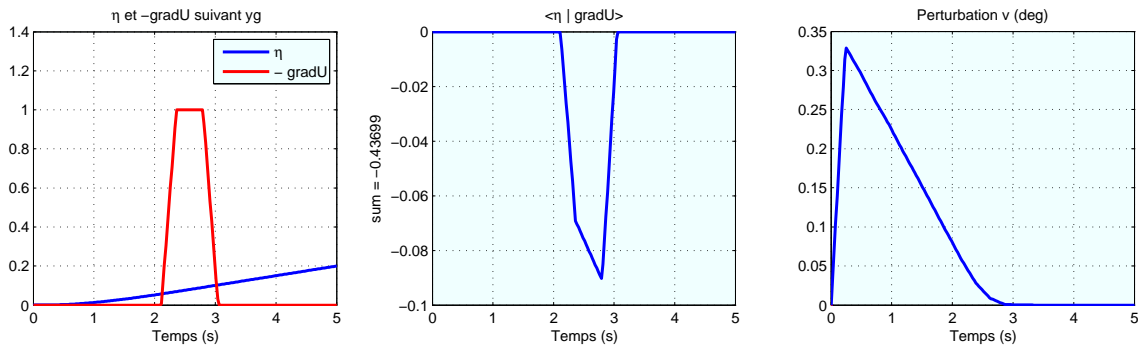
Pour démontrer l'efficacité de ce contrôle de la dynamique, le modèle de véhicule B est utilisé lors d'un VDA avec des limites sur la dynamique de l'entrée volontairement très faibles. La figure 5.16 présente un exemple de limitation volontaire de la vitesse volant à 170 deg/s et autre un exemple de limitation de l'angle volant à 57 deg .

La figure 5.28 présentera un autre exemple de déformation où le couple volant renvoyé par la crémaillère du véhicule B est bridé à 20 N.m . Le couple n'est pas une variable d'état mais une sortie que l'on a reconstituée et intégrée au vecteur \mathbf{q} de manière à la contrôler en y imposant des forces.

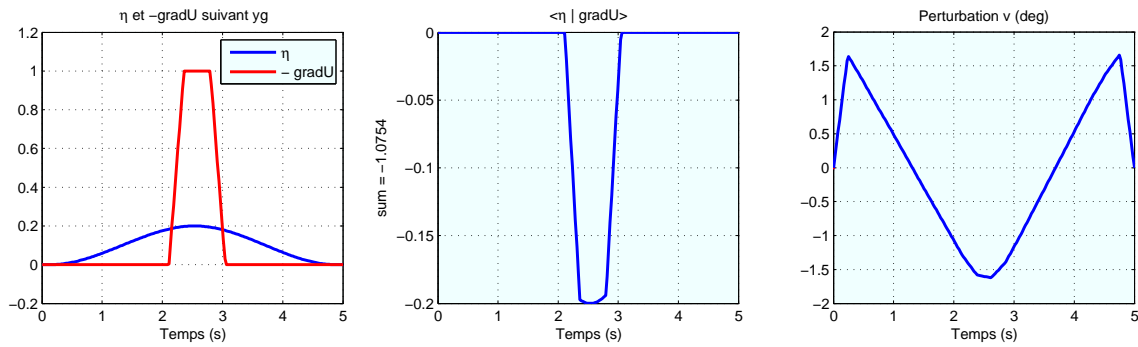
Il est également possible de travailler avec des modèles possédant plusieurs entrées. Il suffit pour cela de modifier le vecteur \mathbf{u} pour qu'il contienne les variables désirées. L'algorithme s'applique ensuite de la même manière. Historiquement, cet algorithme a d'ailleurs été utilisé avec le robot *Hilare 2* qui nécessite deux entrées pour sa commande (vitesse linéaire et vitesse de lacet).

5.9 Implémentation et paramétrage

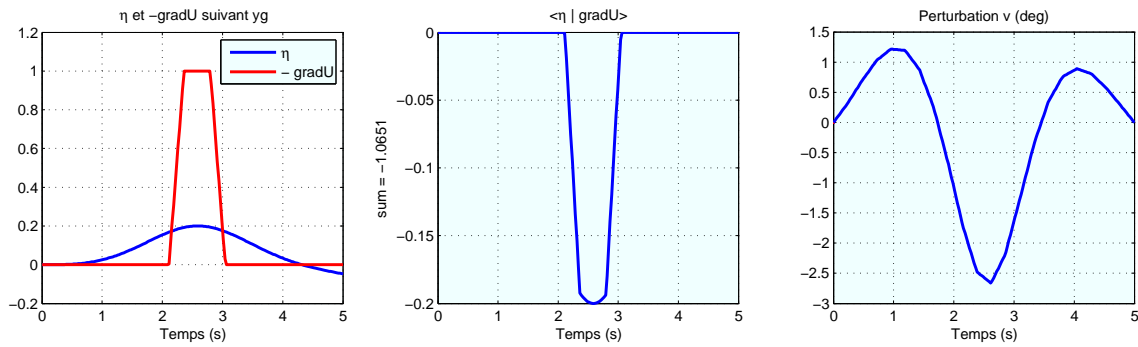
Il existe de nombreuses astuces capables d'accélérer le calcul de la déformation de trajectoire. Par exemple, il est possible de ne pas recalculer les \mathbf{E}_i à chaque fois, mais seulement lorsque la trajectoire a suffisamment bougé. Le nombre p de perturbations élémentaires qui représente la



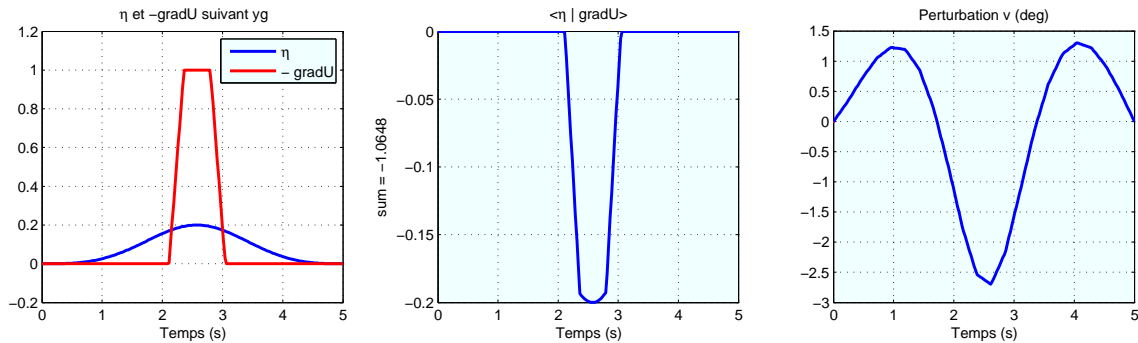
(a) Rien



(b) Projection



(c) Orthonormalisation



(d) Proj + Ortho

FIG. 5.14 – Impact des différentes modifications possibles sur η , v et $\frac{dV}{d\tau}(\tau)$. Le véhicule va en ligne droite au travers d'un VDA (première itération). $\eta_{max} = 0.2$

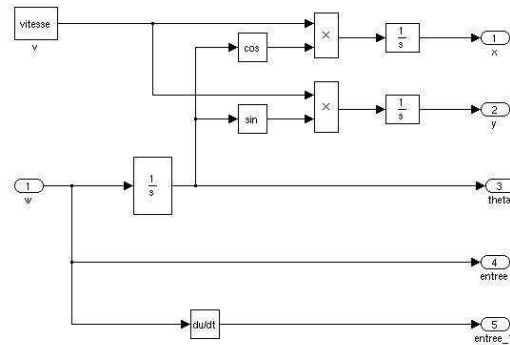


FIG. 5.15 – Modélisation simplifiée d'un modèle de *Hilare 2* instrumenté pour contrôler la dynamique de son entrée

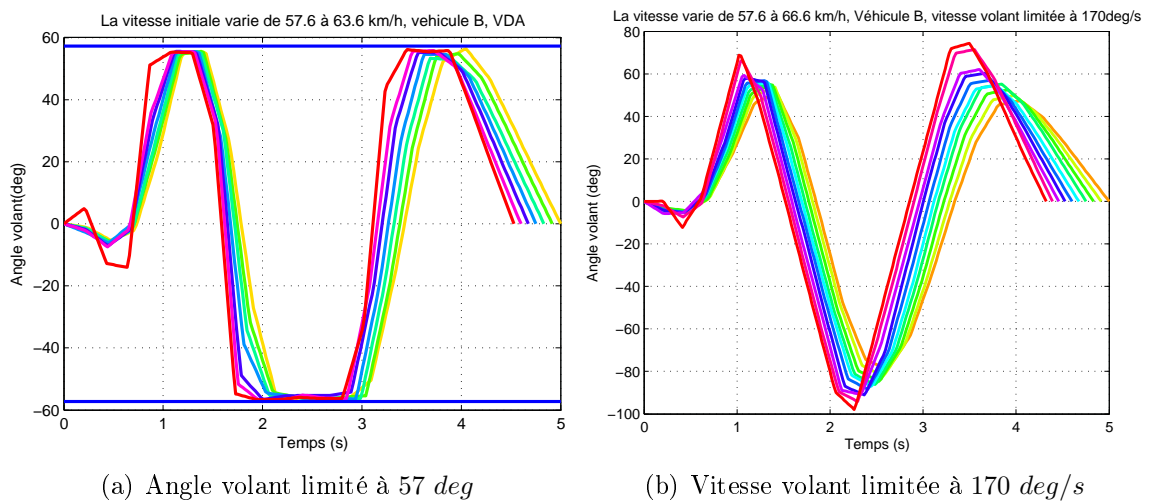


FIG. 5.16 – Impact de la limitation volontaire de la vitesse volant et de l'amplitude volant pour le modèle B à des valeurs très faibles lors un VDA

majorité des calculs doit être ajusté au mieux pour tenir dans les délais impartis (généralement $20 \leq p \leq 50$). Le nombre de perturbations définit la qualité des informations dynamiques récupérées sur le système. Le nombre de points n de la trajectoire discrétisée est de l'ordre de 200 points. Cette dernière valeur n'est pas cruciale, car elle ne conditionne que le rapide calcul des forces.

La valeur de η_{max} définit la rapidité et la qualité de la convergence et doit être ajustée expérimentalement (généralement η_{max} est de l'ordre de quelques centimètres). Le η_{max} doit également être suffisamment petit pour rester dans le domaine de validité des \mathbf{E}_i . La discrétisation de la trajectoire peut être relativement fine car elle n'influence pas ou peu le temps de calcul et permet une détection de collisions pertinente. Généralement n représente plusieurs centaines de points pour 5 ou 6 secondes de simulation. Le processus de déformation s'arrête après un nombre prédéfini d'itérations infructueuses (généralement 1000). La dernière valeur de \mathcal{P} qui a fourni une solution sans collision est alors considérée comme étant la limite du test en cours.

5.10 Résultats

5.10.1 Recherche des limites de \mathcal{P}

Cas du test VDA

La figure 5.17 présente les résultats obtenus lors d'une recherche de vitesse maximum avec cet algorithme sur un test VDA. Les différents angles volant correspondent aux différentes vitesses superposées. La vitesse du véhicule C qui est utilisé ici varie entre 57.6 km/h et 68.4 km/h . Le $\|\eta_{max}\|$ vaut 0.1, la base utilisée est triangulaire et comprend $p = 25$ fonctions. L'incrément de vitesse lorsque toutes les collisions ont été résolues vaut 1 km/h . Le temps de calcul est de l'ordre d'une nuit.

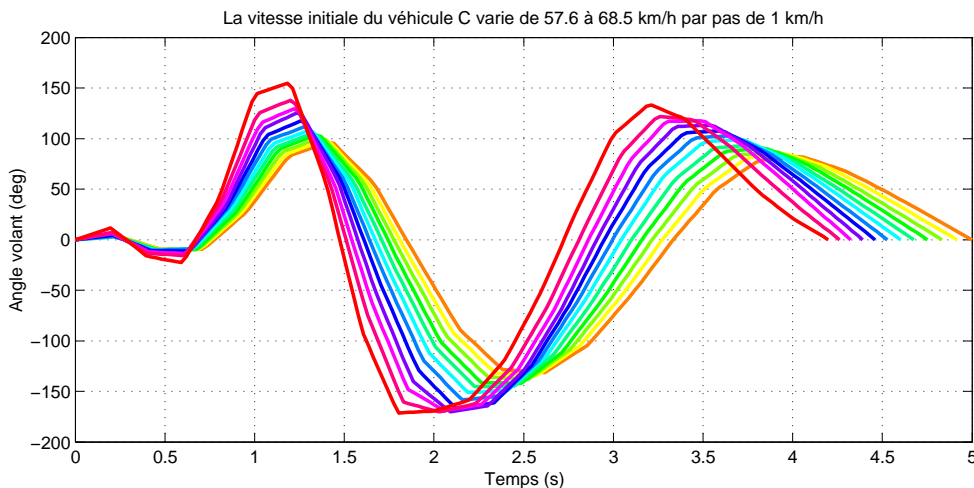


FIG. 5.17 – Angles volant et trajectoires solutions pour une vitesse croissante lors d'un test VDA avec le véhicule C

On remarque sur cette figure un petit décroché au début de l'angle volant (vers 0.5 s) qui correspond à une manœuvre bien connue des essayeurs expérimentés. Cette manœuvre permet de pré-positionner le véhicule dans le premier couloir et d'arrondir la trajectoire. On constate avec

satisfaction que cet algorithme est suffisamment précis pour découvrir de lui-même l'intérêt de ce genre de manœuvre. L'évolution de l'amplitude et du positionnement temporel des différents maxima/minima d'angle volant fournissent également une information intéressante à exploiter.

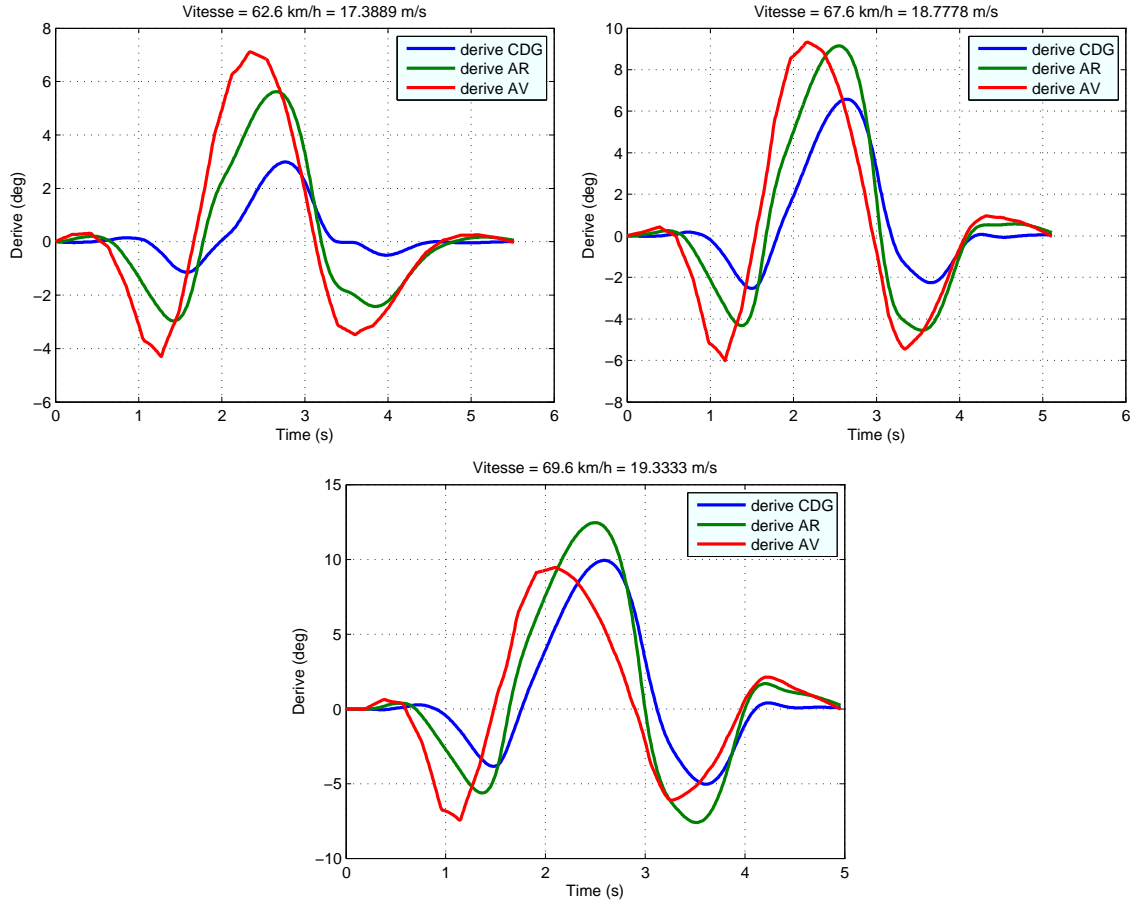


FIG. 5.18 – Dérives AV, AR et CDG calculées avec la DT sur un test d'évitement VDA avec le véhicule A à différentes vitesses. On retrouve les mêmes types de comportements que sur la mesure

Les courbes de dérive de la figure 5.18 présentent les 3 pics caractéristiques des 3 couloirs du test VDA et pour δ_{AV} , δ_{AR} et δ_{cdg} . On retrouve bien les mêmes allures et les mêmes types de comportement (sur-vireur/sous-vireur/neutre) que lors des mesures essayeurs des figures 2.23, 2.21, 2.22.

La figure 5.19 présente le résultat de calculs de sur-braquage dynamique avec la DT. On constate bien l'existence d'un "mur" de la dérive qui empêche le franchissement du test VDA à des vitesses supérieures. On notera que suivant les caractéristiques du véhicule, ce "mur" peut être orienté vers le haut (sous-virage dynamique $\delta_{AV} > \delta_{AR}$ sur le deuxième pic) ou vers le bas (sur-virage dynamique $\delta_{AR} > \delta_{AV}$ sur le deuxième pic).

Ces résultats sont très similaires à ceux présentés au chapitre 2.1.1 et confirment que les V_{max} obtenus sont proches des limites du véhicule. Dans le cas particulier du modèle du véhicule B, la courbe de sur-braquage dynamique est inversée par rapport aux mesures de la figure 2.24. Le modèle est dynamiquement sous-vireur tandis que le véhicule réel est dynamiquement sur-vireur. Pour autant, les vitesses de passage observées sont similaires en simulation et en réalité (de l'ordre de 70 km/h). La simulation et la mesure échouent à la même vitesse, mais pas pour

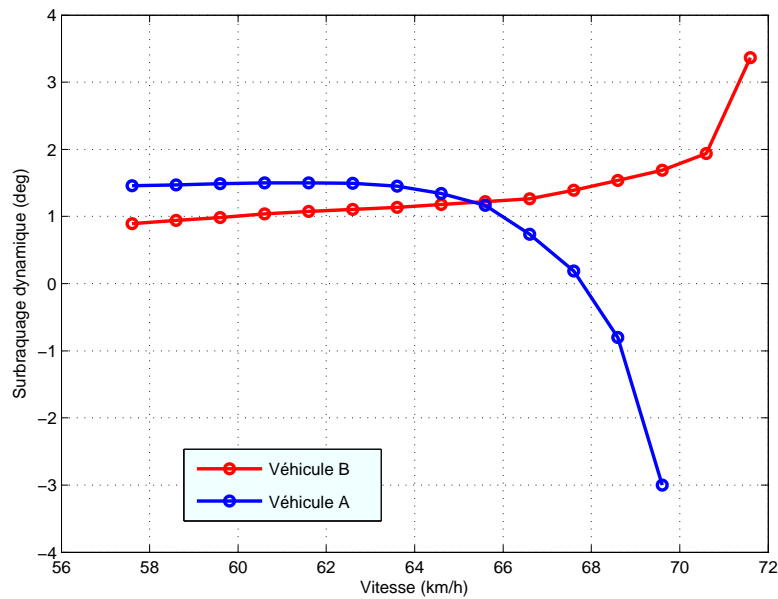


FIG. 5.19 – Courbes de sur-braquage dynamique calculées avec la DT sur des tests VDA réalisés avec les véhicules A et B

la même raison. Ce problème est manifestement imputable à un écart de modélisation. Cela confirme bien qu'il faut s'intéresser à des variations de paramétrage et non à des valeurs brutes de V_{max} lorsque l'on compare la simulation aux mesures sur piste. C'est d'ailleurs bien ce qui est fait lors de l'étude de robustesse présentée au chapitre 5.10.4.

Cas de l'évitement ville

La figure 5.20 présente les mêmes calculs effectués avec un évitement ville. Les amplitudes et les vitesses au volant ont très nettement augmenté ce qui est cohérent par rapport aux mesures des essayeurs. L'évolution des positions et amplitudes des 3 pics caractéristiques d'un évitement est riche d'enseignements sur la manière dont il faut compenser la perte progressive de la capacité du véhicule à tourner.

Freinage en courbe

La figure 5.21 présente le résultat d'une déformation de trajectoire appliquée à un freinage en courbe avec pression de freinage croissante. La durée du freinage est de 1 s. Le détail de la procédure peut être trouvé au chapitre 2.3.4. On constate qu'il faut débraquer le volant de plus en plus fortement afin de permettre au véhicule de suivre sa voie. La procédure s'arrête lorsque des roues sont bloquées. La déformation a permis de pousser le véhicule au maximum de freinage possible pour ce test tout en évitant la sortie de route. La figure 5.22 présente l'évolution de la dérive au cdg et la vitesse de lacet associée lorsque le freinage augmente. On voit que ces grandeurs restent sous contrôle grâce à l'action de correction entreprise au volant. La figure 5.23 présente les mêmes courbes lorsque le volant conserve l'angle d'avant le freinage. Le véhicule part rapidement en tête-à-queue si rien n'est fait pour contrecarrer la dérive croissante.

Les calculs effectués avec différents modèles (AMESim, ESP ON/OFF, etc.) présentent des différences assez significatives sur ce test particulier.

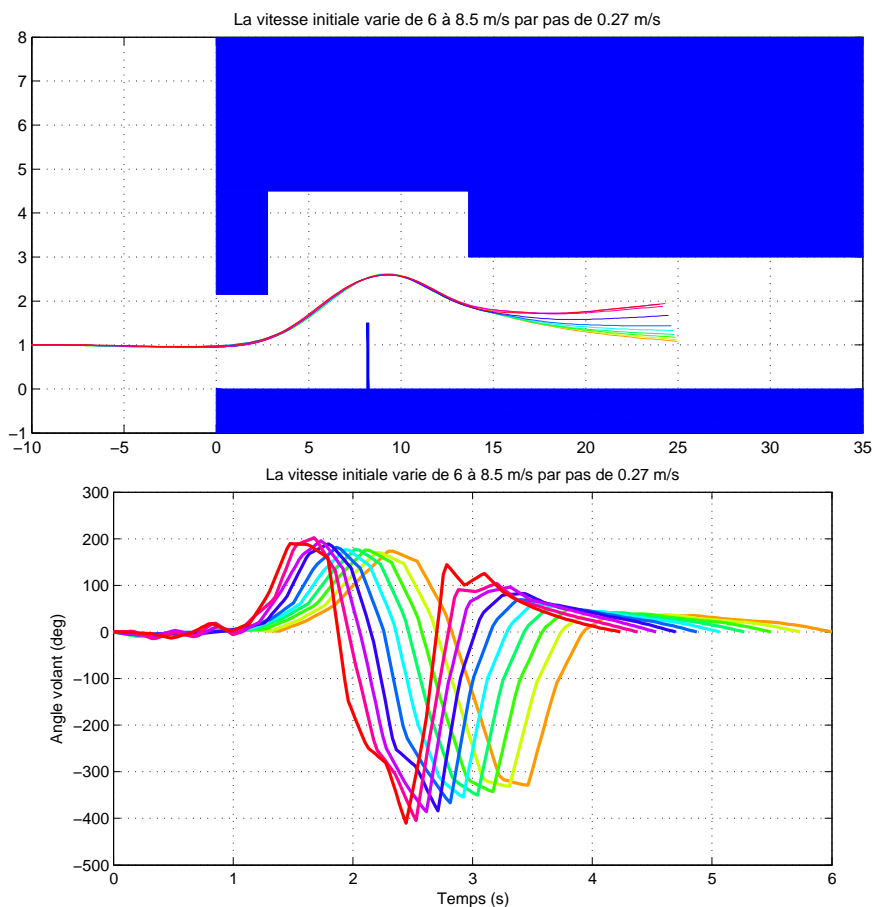


FIG. 5.20 – Angles volant et trajectoires solutions pour une vitesse croissante lors d'un évitement ville avec le véhicule C

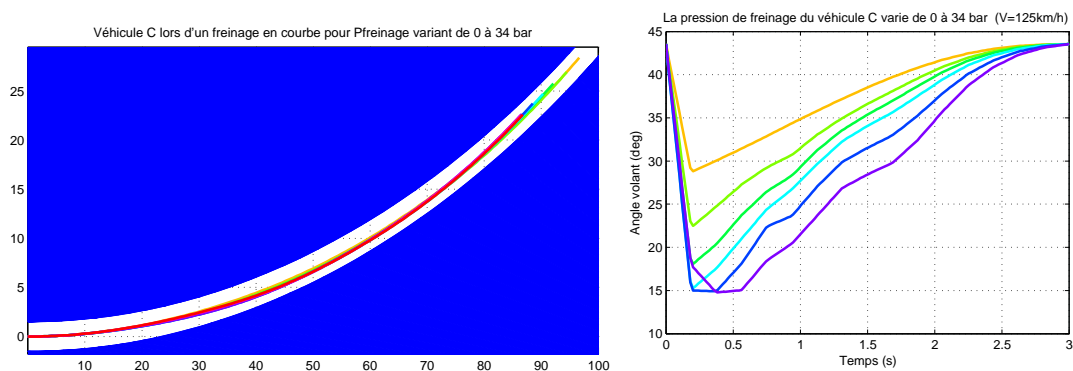


FIG. 5.21 – Angles volant et trajectoires solutions lors d'un freinage en courbe avec le véhicule C ($P_{freinage}$ varie de 0 à 34 bars)

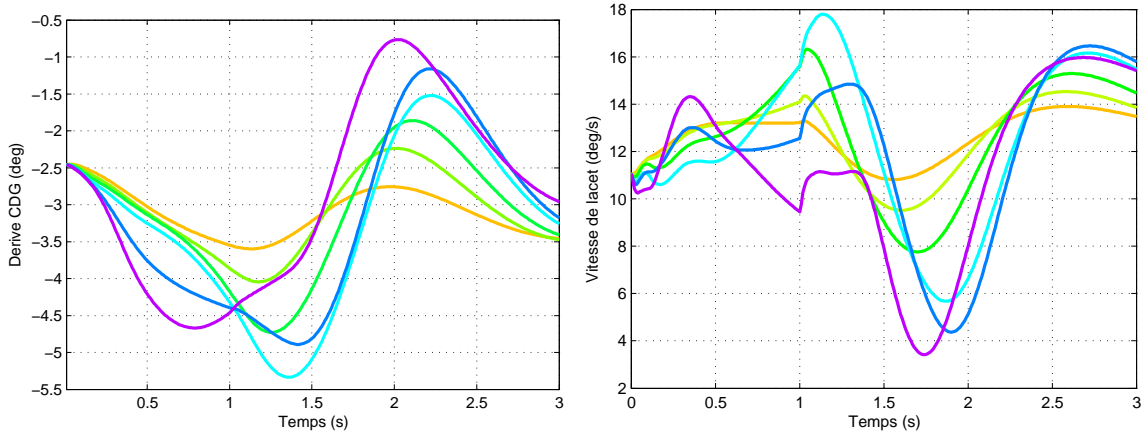


FIG. 5.22 – Dérives au centre de gravité et vitesses de lacet sans collisions lors d'un freinage en courbe avec le véhicule A pour $P_{freinage}$ variant entre 0 et 34 bars

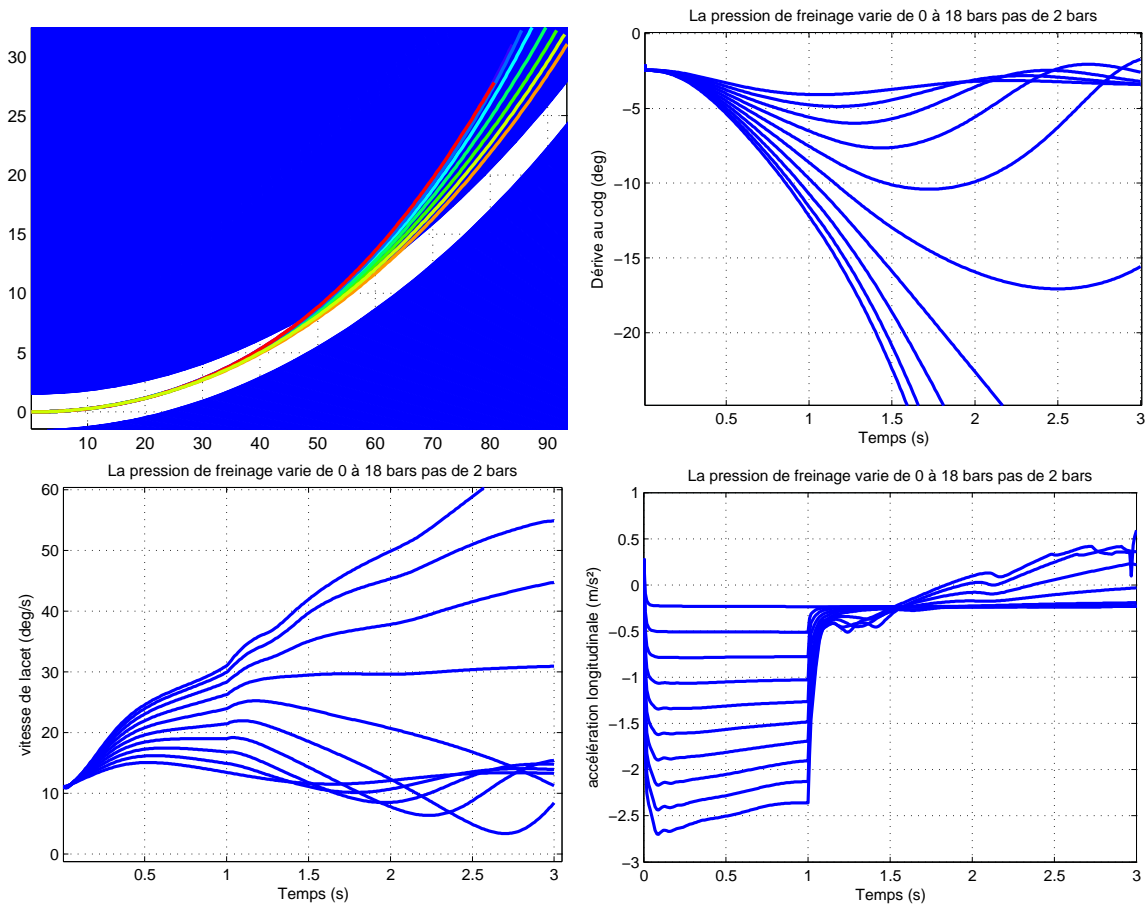


FIG. 5.23 – Trajectoires, dérives au centre de gravité, vitesses de lacet et accélérations longitudinales lors d'un freinage en courbe avec volant tenu. $P_{freinage}$ varie entre 0 et 18 bars

5.10.2 Convergence

La figure 5.24 présente un exemple d'évolution de la trajectoire du véhicule A lors des itérations d'une procédure de déformation appliquée au VDA. Il est intéressant de comparer cette manière de converger à celle de l'optimisation en figure 4.12. La trajectoire initiale est rectiligne. On observe clairement le caractère très progressif et local de cet algorithme. Les obstacles sont pris en compte au fur et à mesure de leur découverte. La figure 5.25 présente les angles volant obtenus lors de la résolution des collisions. La trajectoire initiale cherche juste à éviter l'obstacle représenté par la zone interdite du deuxième couloir, d'où l'aspect inhabituel de l'angle volant des premières itérations. Mais lorsque les autres contraintes sont prises en compte (obstacles supérieurs), l'angle volant converge rapidement vers la bonne solution. Cette manière de converger est bien plus adaptée que celle de la commande optimale du chapitre 4 car elle évite de solliciter le modèle dans des domaines où il est instable. C'est bien la trajectoire $\mathbf{q}(s)$ qui est déformée progressivement, et non le vecteur \mathbf{x} qui décrit l'entrée. On évite ainsi tous les problèmes informatiques et de non convergence liés à la manière de procéder de l'optimisation.

Il est possible de s'assurer de la qualité de la convergence en fournissant à l'algorithme des conditions initiales tirées aléatoirement, puis en vérifiant que l'on converge vers des résultats proches. Ces tests ont été effectués et ont donné de bons résultats dans la plupart des cas. Les trajectoires résultantes sont souvent assez semblables de par la nature des forces appliquées : ce sont celles qui passent au plus près des obstacles qui ressortent en premier. Il est également possible d'appliquer des forces dans l'espace libre de manière à avoir des trajectoires plus distantes des obstacles.

5.10.3 Respect de l'hypothèse de linéarité locale

Il est possible de comparer les $\boldsymbol{\eta}$ théoriquement prédits par l'équation (5.6) à ceux réellement obtenus après simulation de l'entrée perturbée afin de valider notre hypothèse de superposition et de linéarité des \mathbf{E}_i . Le $\boldsymbol{\eta}$ de corrélation est obtenu a posteriori par soustraction entre le \mathbf{q} courant et le \mathbf{q} de l'itération précédente. Un système intégré à l'algorithme permet de valider cette corrélation à chaque itération afin de garantir la robustesse du système. Cette procédure avertit l'utilisateur lorsque le véhicule ne se comporte plus comme l'algorithme le prédit. La figure 5.26 présente une illustration de cette comparaison où seules les projections suivant x , y et ψ sont représentées. Sur cet exemple, la corrélation entre les deux $\boldsymbol{\eta}$ est quasiment parfaite, le véhicule va exactement là où l'algorithme le pousse. Cette figure illustre le bien-fondé de notre hypothèse initiale de comportement linéaire autour d'une trajectoire de référence.

5.10.4 Robustesse

Comme on l'a vu précédemment, la confrontation directe des calculs aux essais physiques est limitée par un biais de modélisation. L'évaluation de robustesse doit donc être plus qualitative que quantitative. La robustesse de l'algorithme est ici évaluée uniquement dans le cas du test VDA en confrontant les variations de vitesse de passage maximale à des variations de paramétrage du modèle connues pour améliorer celle-ci.

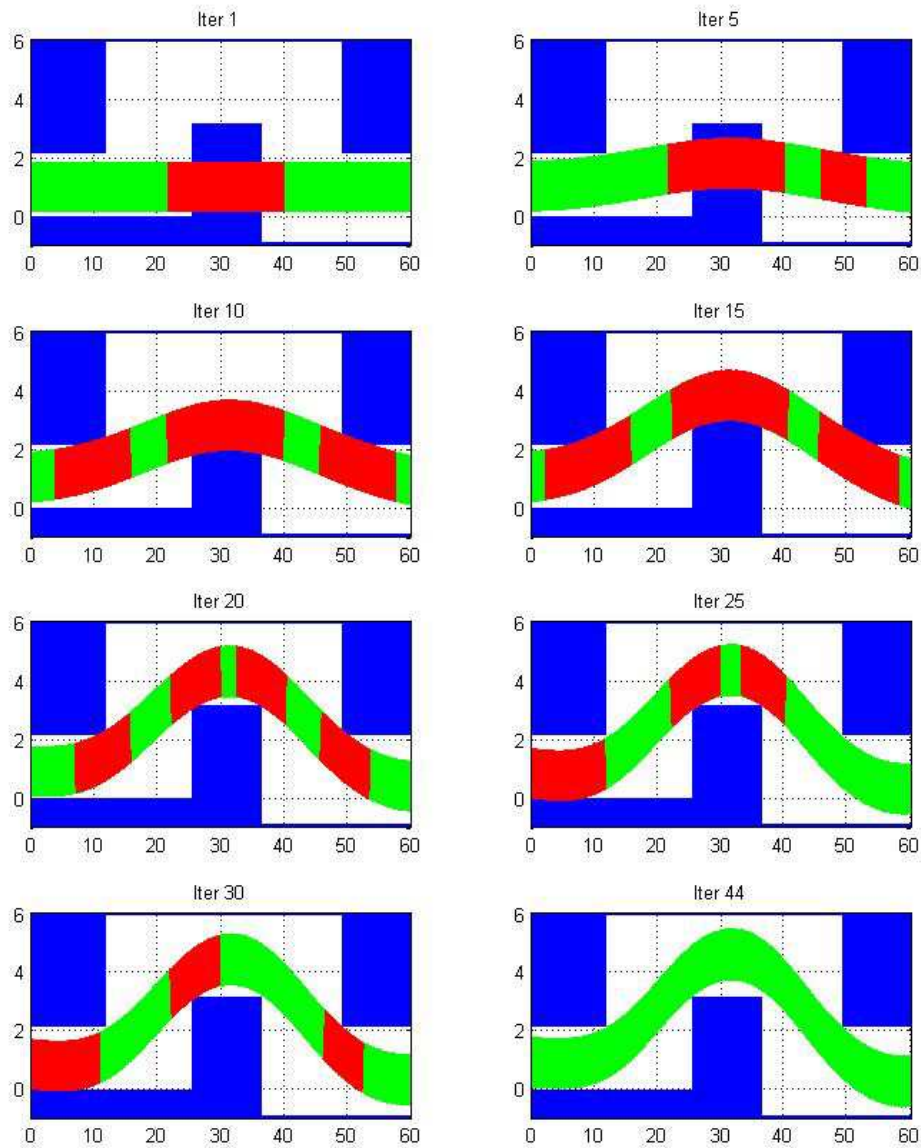


FIG. 5.24 – Exemple de trajectoires issues des itérations d’une procédure de déformation de trajectoire avec le véhicule B à 57.6 km/h

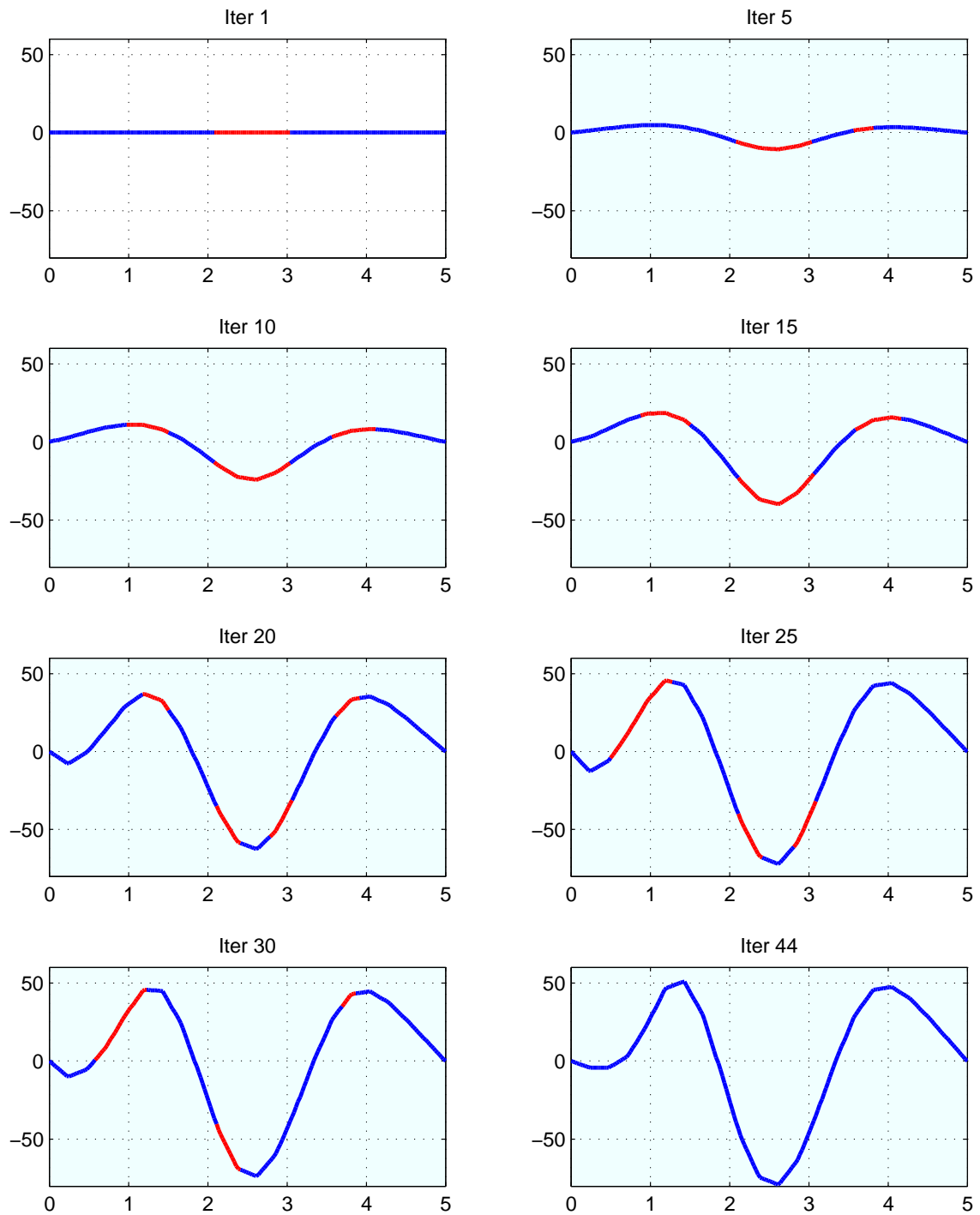


FIG. 5.25 – Exemple d'angles volant issus des itérations d'une procédure de déformation de trajectoire avec le véhicule B à 57.6 km/h

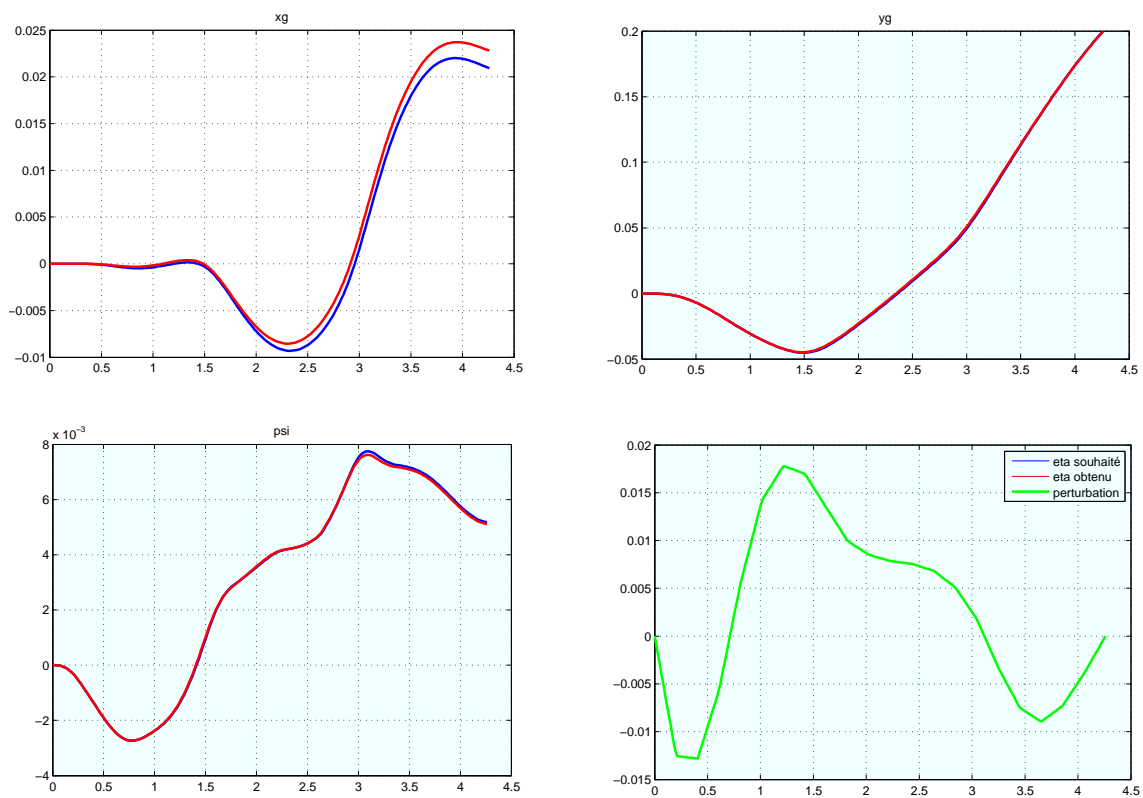


FIG. 5.26 – Comparaison entre un η prédit par l'algorithme et un η réellement obtenu après simulation de l'entrée perturbée (projection suivant x , y et ψ)

Pneumatiques

Une modification susceptible d'améliorer fortement la V_{max} est d'améliorer le comportement des pneumatiques en augmentant leur rigidité de dérive ou leur adhérence maximum (voir chapitre 2.1.4). Dans les deux cas les forces transversales pouvant être générées par les pneus sont augmentées et la capacité du véhicule à éviter un obstacle est améliorée. La figure 5.27 illustre graphiquement les diverses améliorations qui ont été apportées sur le comportement transversal du pneu. Le tableau 5.1 résume les résultats obtenus en termes de V_{max} . Ces résultats montrent bien une amélioration très sensible de la capacité d'évitement du véhicule en relation directe avec les modifications des pneus. Ce résultat est conforme à l'expérience. L'incrément de vitesse entre chaque déformation réussie est de 1 km/h , tous ces résultats doivent donc être compris comme possédant une tolérance de $\pm 1\text{ km/h}$. A cause des phénomènes de saturation, la force générée par le pneu a une limite maximum qui empêche le véhicule de tourner plus vite et donc d'améliorer sa vitesse de passage. Dans le cas des systèmes cinématiques type *Hilare 2*, la vitesse peut augmenter indéfiniment car il n'y a pas de saturations.

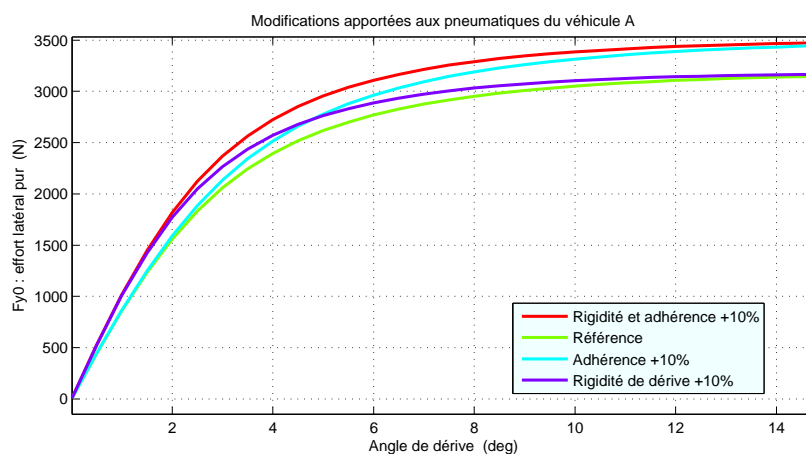


FIG. 5.27 – Exemple de modifications possibles des caractéristiques transversales des pneumatiques du véhicule A

Modifications apportées	Vitesse de passage maximum
Pneu de référence	70.6 km/h
Adhérence maximum augmentée de 10%	72.4 km/h
Rigidité de dérive augmentée de 10%	72.4 km/h
Adhérence et rigidité de dérive augmentés de 10%	74.9 km/h

TAB. 5.1 – Impact de différentes variations du paramétrage du pneu sur la vitesse de passage maximum du véhicule A au travers du test VDA

Afin de valider au mieux cet algorithme et d'évaluer la sensibilité de V_{max} à la variation de certains paramètres, on effectue quelques calculs ne respectant pas les normes du test VDA.

Géométrie

Si la géométrie du véhicule C est remplacée par celle du véhicule A tout en conservant le comportement du véhicule C (sans adapter les dimensions des cônes), la vitesse de passage maximum passe de 69.6 km/h à 74.6 km/h . Cette modification correspond à un écart très important uniquement imputable à l'espace supplémentaire alloué qui a bien été exploité par l'algorithme.

Couple volant

Si un modèle de direction sans aucune assistance est rajouté sur le véhicule A, et que l'on impose une limitation du couple renvoyé par la crémaillère à 20 N.m à l'aide de la même technique que celle décrite au chapitre 5.8. La vitesse de passage maximum décroît de 69.6 km/h à 60.8 km/h et l'allure des angles volant résultants change radicalement (voir figure 5.28).

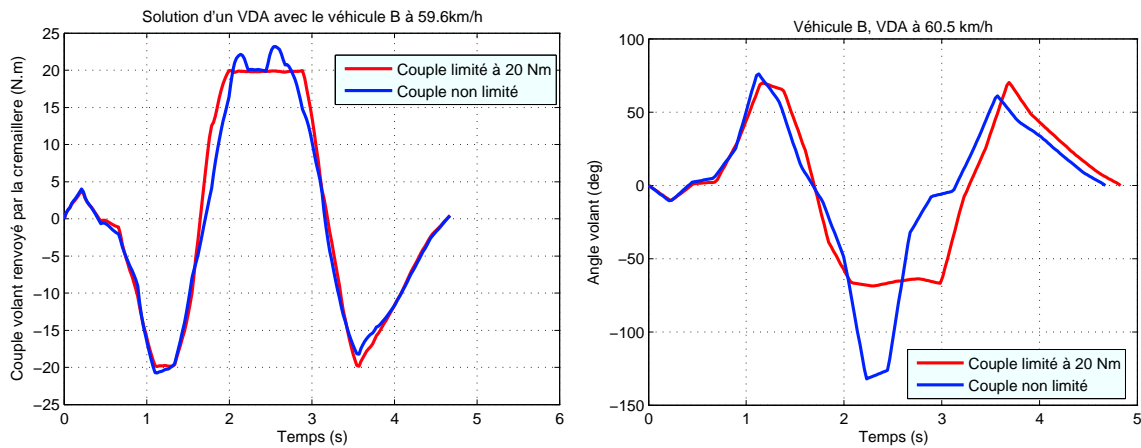


FIG. 5.28 – Évolution de la solution fournie par la DT lorsque l'on applique une contrainte supplémentaire sur le couple volant du véhicule B lors d'un VDA

Angle et vitesse volant

Comme on l'a vu au chapitre 5.8, il est possible d'imposer des contraintes arbitraires à la dynamique de l'entrée et d'en voir l'impact sur la V_{max} d'un VDA. Dans le cas de la limitation de l'angle volant du véhicule B à 57 deg/s , la vitesse de passage maximum sur le VDA descend à 63 km/h . Lors de la limitation de la vitesse volant à 170 deg/s , la V_{max} descend à 66 km/h . La figure 5.16 présente l'allure des angles volant correspondants. Ces calculs sont très intéressants car ils permettent d'évaluer en simulation l'impact de conducteurs plus limités physiquement. La prise en compte de contraintes arbitraires est une des grandes forces de la DT.

Adhérence

Si on tente de passer le VDA avec une adhérence $\mu = 0.5$ sur la piste, la vitesse de passage maximum du véhicule A est de 53 km/h au lieu de 69.5 km/h . Les angles volant ont une amplitude bien plus faible. De manière assez surprenante, la dérive ne part pas vers des niveaux très élevés, car la trajectoire est justement bien maîtrisée par l'algorithme. Une trop grande dérive

génère nécessairement des collisions car la largeur des couloirs est très faible. Une simulation boucle ouverte aurait donné des résultats aberrants (tête-à-queue) car des angles volant mesurés n'auraient pas été adaptés à la situation réellement simulée sur le véhicule (adhérence fixe et régulière).

Masse et Inertie

L'influence de la masse peut s'évaluer facilement lorsque l'on compare un modèle du véhicule B avec un cas de charge standard (2P14) au même véhicule chargé à son maximum (Masse Totale Admissible en Charge de 1649kg). La vitesse de passage maximum lors d'un test VDA chute alors de près de 1.8 *km/h*. Il faut cependant faire très attention lors de l'interprétation de ces résultats : suivant le niveau de saturation auquel se trouve le pneu lors d'un VDA à vide, l'augmentation de masse ne pénalise pas forcément aussi fortement la V_{max} que ce à quoi l'on pourrait s'attendre. Un véhicule qui sous-utilise le potentiel de ses pneus peut ne pas se dégrader en termes de V_{max} car l'augmentation de charge sur le pneu permet à celui-ci de travailler dans une zone où il est plus efficace. Il faut donc faire très attention lors de la définition de ces tests et de l'interprétation des résultats. La sollicitation, la trajectoire et le V_{max} changent pour chaque simulation ce qui exige une expertise physique fine pour bien interpréter des résultats parfois surprenants.

Essence/Diesel

Un autre exemple plus pratique de l'intérêt de ces techniques est la comparaison entre un véhicule B équipé d'un moteur essence et le même véhicule utilisant un moteur diesel plus lourd de 75 *kg* (tous deux chargés en 2P14). Ces deux véhicules présentent une vitesse de passage proche, car la LAS du véhicule diesel a été adaptée afin de prendre en compte ce poids supplémentaire (changement des pneus, barre anti-dévers, etc.).

Elasto-cinématique

La comparaison entre différentes souplesses de cales du train donne en simulation strictement la même vitesse de passage tandis que les essayeurs constatent un écart de quelques *km/h*. La faute peut en être imputée au modèle de véhicule probablement trop grossier.

ESP

L'ajout d'un modèle d'ESP sur le véhicule C augmente légèrement la valeur de V_{max} . Celle-ci passe en effet de 68.4 *km/h* à 71.2 *km/h*. Il sera vu au chapitre 6.9 que la présence de cet équipement augmente significativement la taille de l'ensemble des trajectoires possibles.

Dans toutes ces comparaisons, les sollicitations solutions sont différentes car l'algorithme s'adapte au comportement du véhicule. Néanmoins, force est de reconnaître que trajectoires et angles volant présentent toujours des allures similaires d'une simulation à une autre. La figure 5.29 illustre les trajectoires obtenues lors de la procédure de déformation suivant que l'ESP soit activé ou non. La vitesse varie par pas de 1 *km/h* de 61.2 *km/h* à V_{max} .

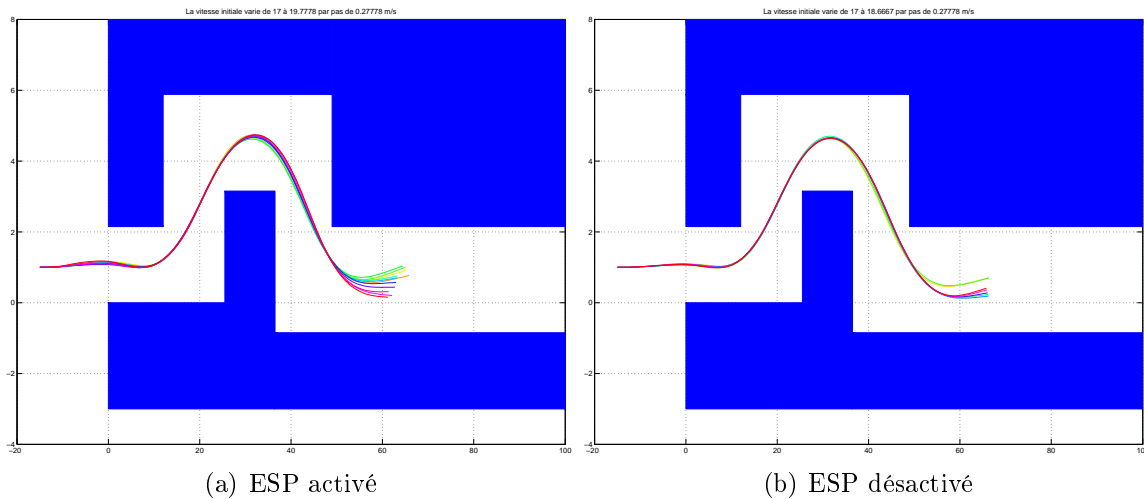


FIG. 5.29 – Trajectoires VDA obtenues lors de la procédure de déformation pour le véhicule C équipé ou non d'un ESP. La vitesse varie entre 61.2 et 68.4 km/h sans ESP et entre 61.2 et 71.2 km/h avec ESP.

Typage

Comme on l'a vu au chapitre 2.1.5, il est possible de rendre un véhicule plus ou moins sous-vireur statiquement en jouant sur les raideurs des barres anti-dévers. La figure 5.30 illustre le changement d'allure de la courbe de sur-braquage dynamique vue au calcul, lors d'une modification de la répartition des raideurs AV/AR du véhicule A lors d'un VDA. La somme totale des raideurs des deux BAD reste constante. Le véhicule de référence a une répartition de 55 % à l'AV et de 45 % à l'AR. Le véhicule modifié a une répartition de 68%/32%. Le véhicule modifié est plus raide de l'AV et est donc sous-vireur statiquement. Mais comme on le voit sur la figure 5.30, il est également plus sous-vireur dynamiquement.

5.11 Limitations

La principale limitation de cet algorithme est de nature théorique et est la même que pour les autres : l'absence de preuve d'exactitude du résultat. A notre connaissance il n'existe pas de système simplifié sur lequel on pourrait prouver analytiquement l'existence d'une limite à \mathcal{P} . Il n'est donc pas possible de valider que les limites atteintes soient bien les bonnes. L'absence d'un critère d'arrêt réellement objectif est l'origine principale du problème. Si la valeur de η_{max} est choisie trop grande, le processus peut osciller en rebondissant entre les couloirs de droite et de gauche et ne jamais converger. A l'opposé, si elle est choisie trop petite, le processus sera trop lent et s'arrêtera prématurément lors de l'activation du critère d'arrêt. Les valeurs de η_{max} et du nombre d'itérations infructueuses avant arrêt sont les deux principaux critères à surveiller lors de la définition d'un calcul de déformation. L'incrément du paramètre du test \mathcal{P} lorsque toutes les collisions sont résolues définit la précision avec laquelle cette valeur peut être déterminée. Ces calculs se révèlent souvent fiables et robustes, mais ils ne sont pas forcément discriminants dans le cas de modifications subtiles du paramétrage du véhicule. Cela signifie qu'une petite variation de paramétrage se traduira en sortie de l'algorithme par une même valeur de \mathcal{P} , mais sans donner de résultat aberrant.

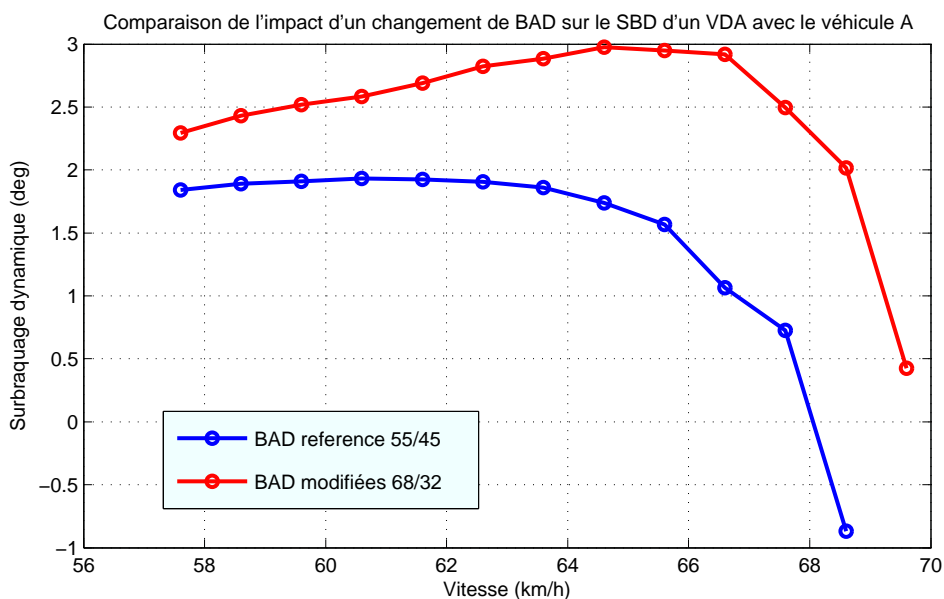


FIG. 5.30 – Impact d'un changement de répartition de raideur AV/AR sur l'allure de la courbe de sur-braquage dynamique du véhicule A

5.12 Synthèse

La précision et la robustesse dont fait montre la DT sont du même niveau, voire supérieures à l'optimisation. Son fonctionnement est plus naturel et compréhensible et nécessite moins de paramétrage. Les résultats intermédiaires sont toujours intéressants (pas de trajectoire aberrante en cas d'échec). Les angles volant sont souvent très réguliers car issus de la trajectoire nécessitant le moins de perturbations. Il est donc rare d'obtenir des solutions erratiques comme cela est possible en optimisation. En résumé, la déformation de trajectoire peut se concevoir comme une solution hybride, plus opérationnelle, située entre la planification et l'optimisation. La prise en compte de contraintes originales, l'approche physique plus concrète et proche de l'expérience des essayeurs caractérisent cet outil.

Ces trois techniques (exploration, optimisation, déformation) répondent chacune à leur manière à la question du calcul de la limite de \mathcal{P} . Ces résultats permettent d'obtenir de précieuses informations sur le comportement du véhicule. Cependant, ces techniques ont encore un fort potentiel inexploité pour évaluer l'ensemble des trajectoires solutions disponibles à une vitesse donnée. Ce genre de calcul devrait permettre d'extraire encore plus d'informations sur le comportement du système. Cela est l'objet du chapitre suivant.

Chapitre 6

Recherche d'un ensemble de solutions

La caractérisation de l'ensemble des solutions possibles lors d'un test quelconque est une tâche autrement plus ardue que la détermination robuste des limites en terme de \mathcal{P} . En effet, le nombre de solutions possibles au problème de planification de mouvements est un ensemble de trajectoires de dimension infinie. Il n'est évidemment pas possible de caractériser cet ensemble de manière analytique au vu des contraintes posées. L'objectif se restreindra alors à obtenir un sous-ensemble de trajectoires (ou d'angles volant, dérives, etc.) aussi représentatif que possible. La forme et la taille des faisceaux de trajectoires ainsi obtenus pourront être utilisées et interprétées par les ingénieurs. La suite de ce chapitre présente la manière dont les méthodes vues précédemment ont pu être utilisées afin de caractériser ces faisceaux. Cette possibilité était également entrée en ligne de compte lors de leur sélection parmi l'ensemble des techniques disponibles. Les contraintes en termes d'accessibilité du système boîte noire et de détection de collisions sont les mêmes que précédemment.

6.1 Approche par les méthodes exploratoires

La méthode FDP développée au chapitre 3.2.1 peut facilement être étendue afin de fournir des faisceaux de trajectoires intéressants. Il suffit pour cela de laisser l'algorithme s'exécuter après la découverte d'une trajectoire solution. De nouveaux points sont alors rajoutés à l'arbre, ce qui étend la couverture de l'ensemble des solutions. Le modèle utilisé sur la figure 6.1 est celui du véhicule A passant un VDA à 40 *km/h*. L'algorithme FDP discrétise ici avec un pas de temps de 0.3 *s*, la grille utilisée a une résolution de 300 cases suivant x , 300 cases suivant y et 100 cases suivant ψ .

Les méthodes aléatoires type RCRRT ou KDP (voir chapitres 3.2.3 et 3.2.4) fournissent naturellement des solutions différentes lors de chaque exécution de l'algorithme. Cette particularité peut être exploitée pour fournir les ensembles de solutions recherchés. Il suffit de résoudre de multiples fois le même problème avec ces algorithmes, puis de superposer les résultats. C'est cette approche qui est utilisée avec l'algorithme RCRRT et le véhicule A à la vitesse de 57.6 *km/h* dans la figure 6.2. Les faisceaux de dérive ou d'angle volant associés sont évidemment disponibles une fois ces calculs réalisés.

Cependant, les problèmes de temps de calcul prohibitif et de robustesse des solutions viennent réduire l'intérêt de cette approche. Les techniques aléatoires ne permettent pas d'obtenir les mêmes faisceaux entre deux "run" successifs à moins d'avoir un temps très grand à disposition.

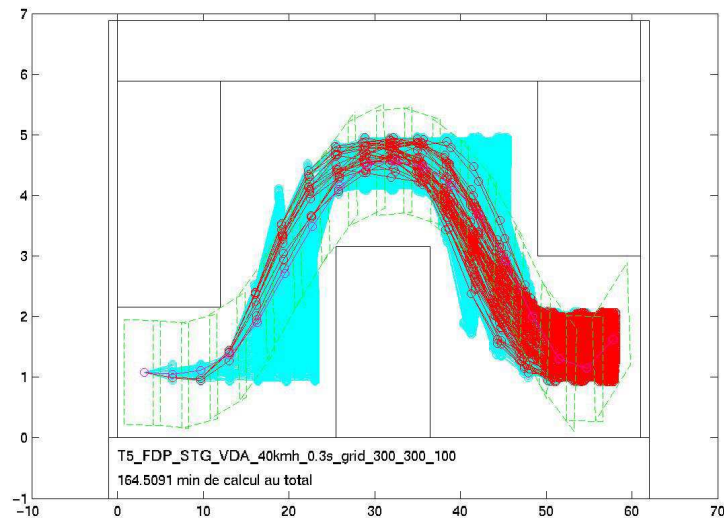


FIG. 6.1 – Faisceau de trajectoires issu d'un algorithme FDP à 40 km/h avec le véhicule A

Les limitations en termes d'éloignement des limites de \mathcal{P} sont toujours les mêmes : on ne peut travailler qu'à faible vitesse sur un VDA ou faible pression de freinage sur un freinage en courbe. La discrétisation des entrées est en effet trop grossière pour permettre d'approcher les limites.

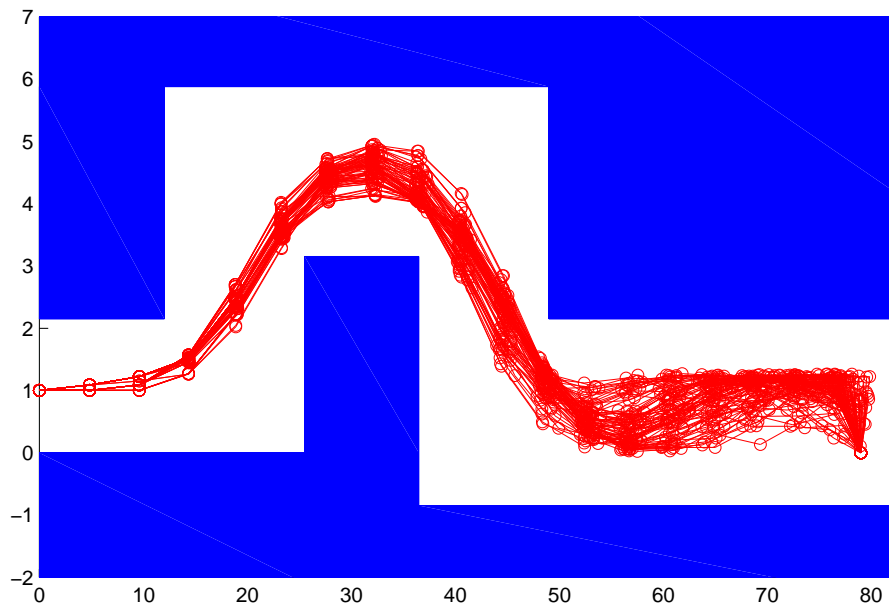


FIG. 6.2 – Faisceau de trajectoires issu de la superposition de multiples exécutions de l'algorithme RCRRT

6.2 Approche par l'optimisation

En faisant varier le critère comme cela a été fait dans [Butz et Stryk, 2005], il est possible d'obtenir plusieurs trajectoires solutions qui peuvent correspondre à différents types de conducteurs. Dans cet article, le modèle utilisé est simplifié et les contraintes sont simplement des contraintes en y basées sur la largeur du véhicule (voir chapitre 4.7.1). Les contraintes liées à la rotation du véhicule ne sont pas prises en compte.

Le faisceau résultant présenté sur la figure 6.3 est par conséquent faux, mais permet de se faire une idée de la technique à employer avec les algorithmes vus au chapitre 4.

Du fait des contraintes de cette thèse, on utilisera le modèle de véhicule complet SimulinkCar et non un modèle analytique simplifié pour représenter le comportement du véhicule. De même, des contraintes en collision rigoureuses (détection de collision) et non un simple critère en y (qui ne prend pas le lacet en compte et sous-contraint le problème) sont utilisées ici.

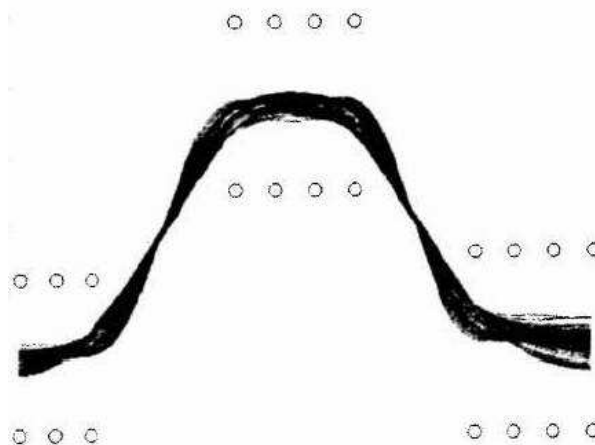


FIG. 6.3 – Faisceau obtenu par Butz et Stryk en faisant varier les paramètres de leur critère

Afin d'obtenir des solutions différentes, il est possible de définir différents critères à minimiser lors de la réalisation de la procédure d'optimisation du chapitre 4. On peut par exemple chercher à minimiser :

1. L'intégrale de la valeur absolue (ou du carré) de l'angle volant
2. L'intégrale de la valeur absolue (ou du carré) de la vitesse volant
3. L'intégrale de la valeur absolue (ou du carré) du couple volant
4. Maximiser la distance parcourue (i.e. minimiser le temps de parcours)

La figure 6.4 présente les résultats obtenus sur un évitement ville lors de l'utilisation de 3 différents critères. La figure 6.5 présente le faisceau d'angles volant résultant de ce calcul. Il comporte peu de solutions, mais est intrinsèquement large du fait de la diversité des critères utilisés. Les faisceaux de dérive ou de trajectoires associés sont disponibles une fois ces calculs effectués.

Il est possible de faire varier le critère continûment afin d'obtenir d'autres solutions. Il suffit pour cela de sommer les critères précédents en les pondérant avec des coefficients que l'on fait varier. Le temps de calcul nécessaire pour obtenir une unique solution est de l'ordre de plusieurs heures. Cette solution pourtant prometteuse est donc limitée par des questions d'ordre pratique. Les collisions présentes sur la figure 6.4 sont dans l'intervalle de tolérance défini sur les contraintes au chapitre 4.8.

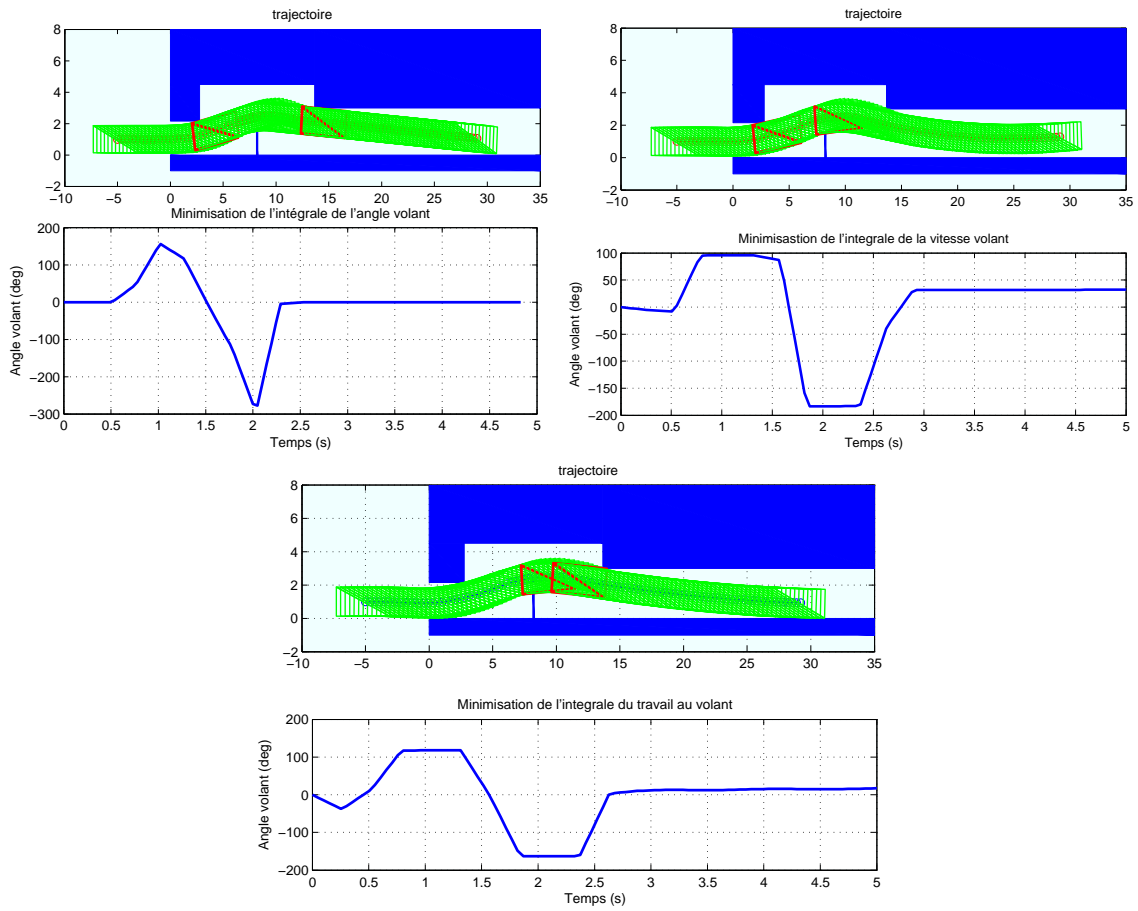


FIG. 6.4 – Résultats obtenus via l'optimisation de différents critères lors du franchissement d'un évitement ville avec le véhicule A à 25 km/h

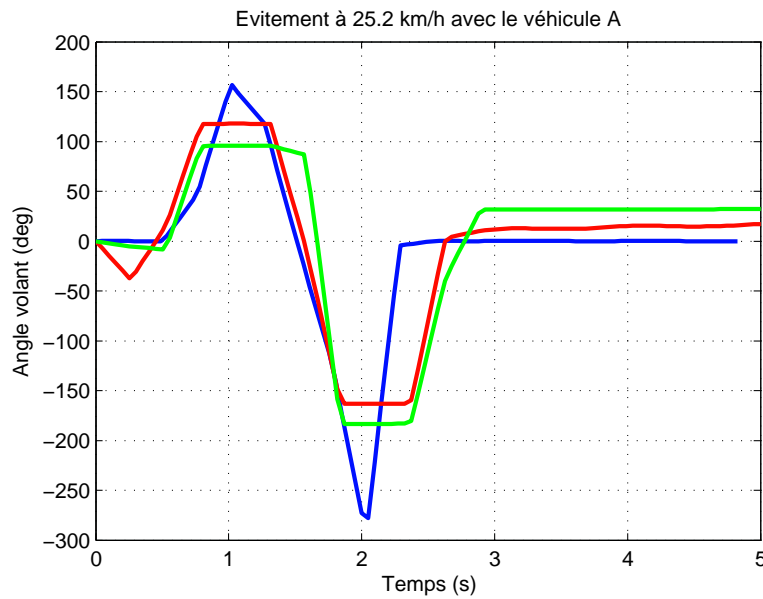


FIG. 6.5 – Faisceau d'angle volant issu de l'utilisation de différents critères lors de l'optimisation

6.3 Approche par la déformation de trajectoire

Le calcul d'un faisceau de trajectoires utilisant la DT se fait de la manière suivante. On utilise une trajectoire de référence \mathbf{q}_{ref} comme point de départ. Puis, on applique un grand nombre de perturbations aléatoires au système. On combine les informations des \mathbf{E}_i avec ces perturbations comme cela a été vu au chapitre 5.3. L'amplitude $\Delta\tau$ des perturbations est ensuite ajustée afin de s'assurer que toutes les trajectoires sont sans collisions. La figure 6.6 illustre l'ensemble des étapes à suivre pour obtenir ces faisceaux.

Cette méthode utilise uniquement des informations locales et ne peut donc découvrir un faisceau de solutions qu'à l'intérieur d'un petit tube autour de la trajectoire de référence $\mathbf{q}_{ref}(s)$. Cependant, lorsque l'on s'approche des limites de \mathcal{P} , le faisceau de trajectoires devient de plus en plus fin et cette technique est alors utilisable. Pour les vitesses plus faibles où le nombre de solutions est important, mieux vaut s'en remettre aux méthodes exploratoires. Cette méthode est aléatoire et présente donc a priori les mêmes problèmes de robustesse que les autres. Néanmoins, l'extrême rapidité du calcul (la trajectoire n'est pas resimulée pour chaque perturbation) autorise le calcul de suffisamment de perturbations pour être pertinent.

6.3.1 Perturbation des entrées

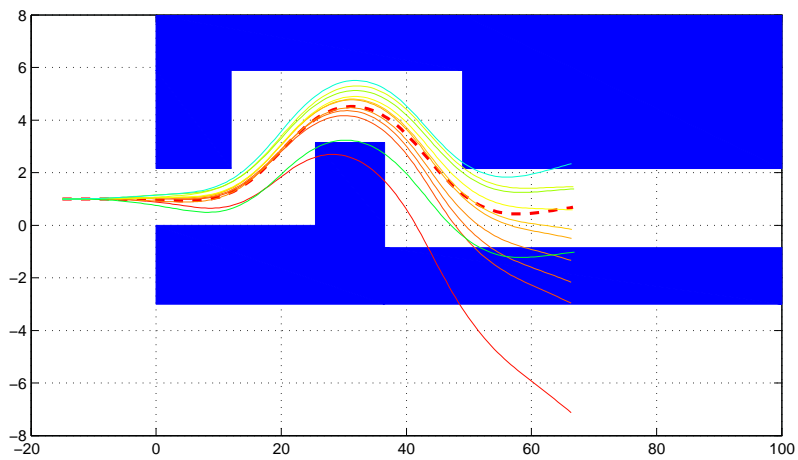
La génération de nombres aléatoires classique souffre d'un certain nombre de défauts tels que la mauvaise répartition des perturbations liée à un échantillonnage uniforme de l'ensemble de perturbations possibles. La génération des perturbations pseudo-aléatoires peut s'effectuer grâce à des séquences de Halton, Sobol ou Faure telles que décrites en [Morokoff et Cafisch, 1994], [Halton, 1960], [Niederreiter, 1992]. Ces séquences présentent des propriétés remarquables et couvrent bien mieux et bien plus vite l'ensemble des fonctions de perturbation possibles [Chazelle, 2000]. La figure 6.7 illustre la différence entre un tirage aléatoire uniforme de points dans un plan et le même tirage effectué avec des séquences de Halton. On voit bien qu'il est plus intéressant d'utiliser ces séquences pour explorer plus efficacement tous les cas possibles. Pour les simulations présentées par la suite, seules des séquences de Halton seront utilisées. Dans le cas étudié ici, ce sont les λ_i qui sont choisis aléatoirement par cette méthode afin de créer des perturbations \mathbf{e} quelconques.

6.3.2 Projection et orthonormalisation

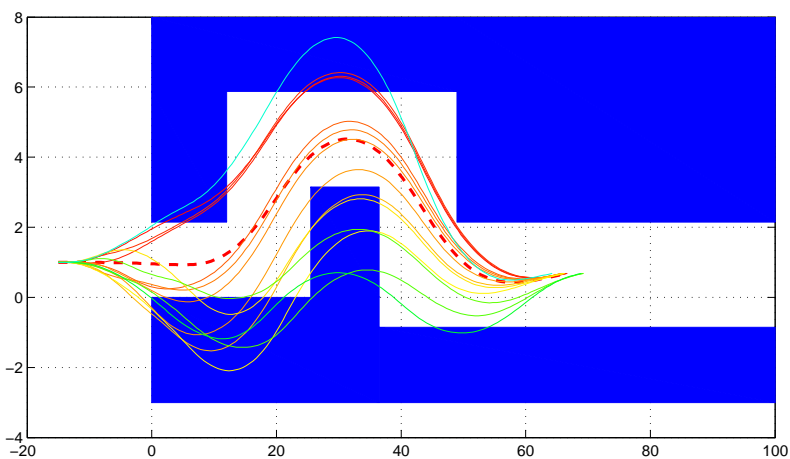
Les \mathbf{E}_i calculés lors du processus de déformation sont projetés et ortho-normalisés suivant la méthode vue au chapitre 5.7 si besoin est. Il ne reste alors plus qu'à reconstituer la perturbation résultante à l'aide de l'équation $\boldsymbol{\eta}(s, \tau) = \sum_{i=1}^p \lambda_i(\tau) \mathbf{E}_i(s, \tau)$. La trajectoire perturbée se déduit ensuite par $\mathbf{q}(s, \tau) = \mathbf{q}_{ref}(s) + \tau \boldsymbol{\eta}(s)$

6.3.3 Ajustement de l'amplitude des perturbations

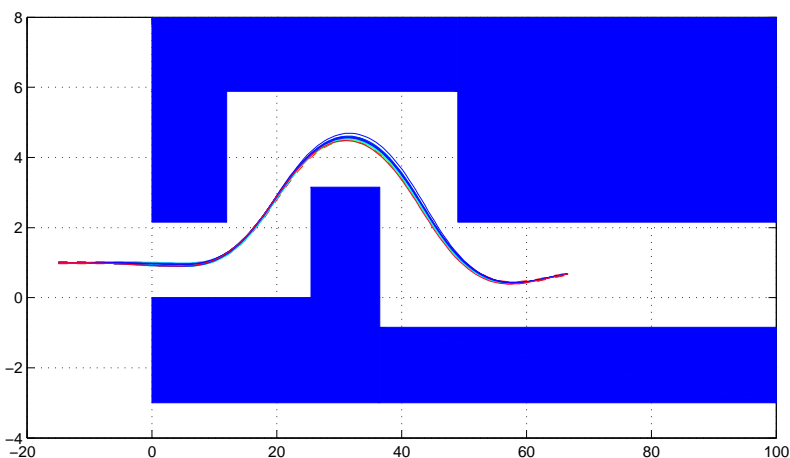
Afin d'obtenir un faisceau le plus épais possible, il faut positionner la trajectoire résultante de manière à ce qu'au moins une de ses configurations soit à la limite de la collision (à une tolérance près). Ce positionnement s'effectue grâce à un algorithme qui détermine le $\Delta\tau$ qu'il faut appliquer à la trajectoire afin d'obtenir ce résultat. Ce calcul est trivial si le véhicule est assimilé à un point (simple calcul de distance à l'espace libre x, y). Dans notre cas, l'ensemble de la géométrie est



(a) perturbations aléatoire



(b) projection et orthonormalisation



(c) ajustement d'amplitude

FIG. 6.6 – Les grandes étapes de la création d'un faisceau de trajectoires via l'utilisation de la déformation de trajectoires

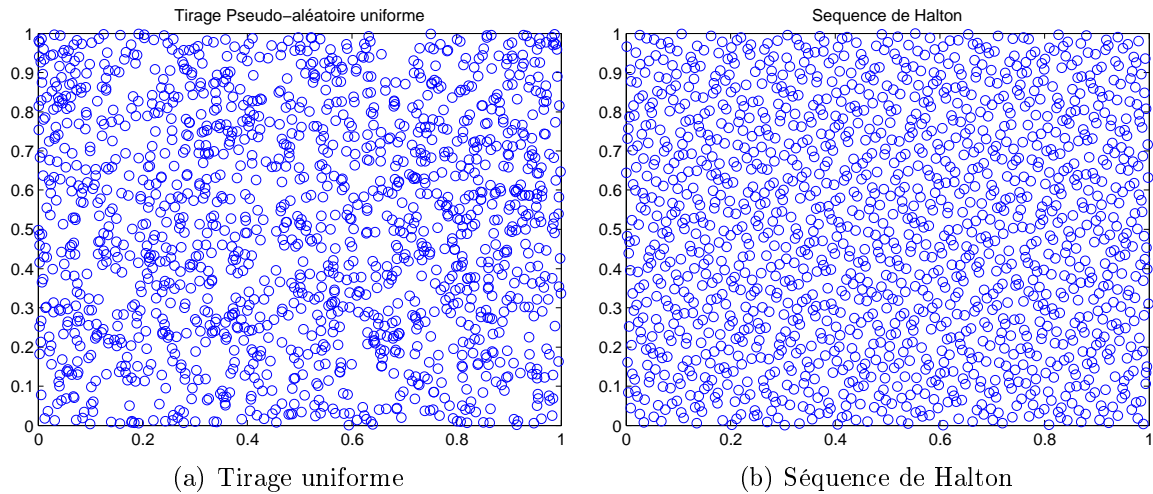


FIG. 6.7 – Comparaison entre un tirage pseudo-aléatoire uniforme de points dans le plan et un tirage issu d’une séquence de Halton

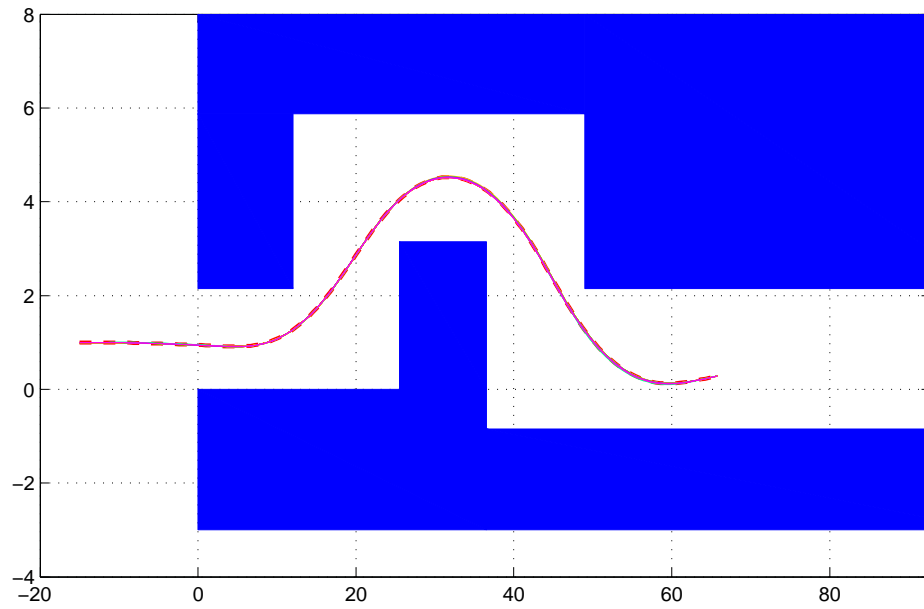
prise en compte et il faut plusieurs itérations afin de placer correctement la trajectoire. Une autre méthode plus directe pour placer un véhicule à la limite de la collision est d’utiliser le calcul de l’espace libre vu au chapitre 4.7.1. Il suffit alors de calculer la distance à cette surface suivant la direction $\boldsymbol{\eta}(s)$. Ce calcul est répété en chaque configuration de la trajectoire discrétisée et c’est le plus petit τ qui est retenu pour le calcul de la trajectoire entière $\mathbf{q}_{ajust}(s) = \mathbf{q}_{ref}(s) + \Delta\tau \cdot \boldsymbol{\eta}(s)$.

6.3.4 Résultats et robustesse

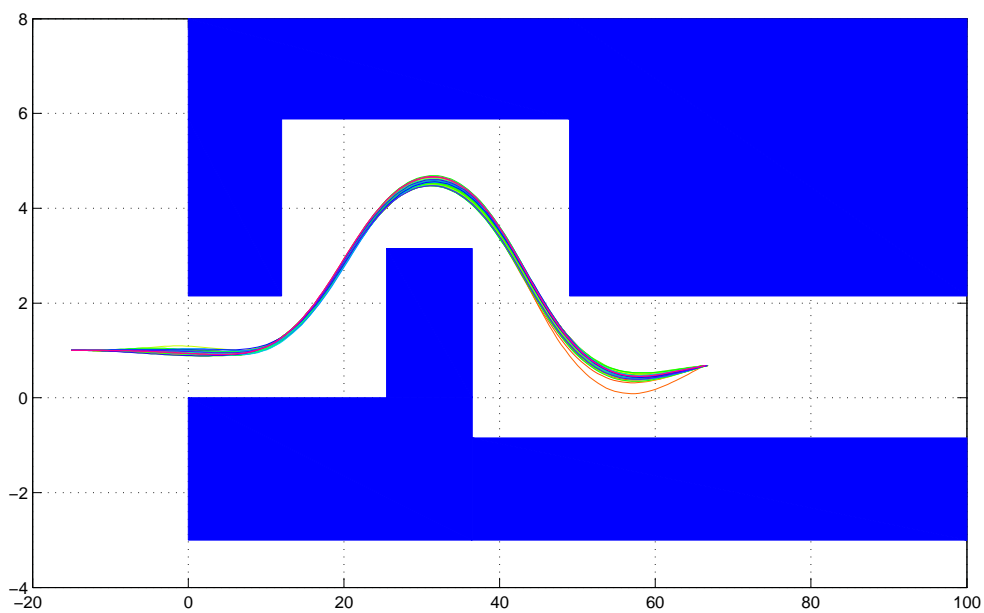
La figure 6.8 présente la comparaison entre deux faisceaux de trajectoires pour le véhicule A mais à deux vitesses différentes. On voit que le faisceau se rétrécit nettement lorsque l’on approche de la vitesse de passage maximum. Cela prouve bien que l’ensemble du potentiel de la voiture a déjà été correctement utilisé pour franchir le test. Plusieurs milliers de perturbations différentes ont été utilisées, mais seules celles faisant augmenter l’aire du faisceau ont été retenues sur la figure. Le nombre important de calculs permet d’assurer une robustesse relative. Les faisceaux de dérive ou d’angle volant associés sont évidemment disponibles une fois ces calculs réalisés.

La figure 6.9 évalue l’impact de la présence d’un ESP sur un véhicule pour une seule et même vitesse de passage à l’intérieur d’un VDA (inférieure à V_{max}). Le faisceau est nettement plus épais lorsque l’ESP est présent. Cela illustre bien que celui-ci permet de faciliter le passage d’un VDA en offrant des solutions supplémentaires au conducteur.

Ces calculs ont surtout pour objectif d’évaluer la qualité du potentiel caché à l’intérieur des \mathbf{E}_i . Sur les figures 5.12 et 5.11 il est relativement aisé de visualiser les véhicules qui ont atteint leurs limites et qui n’ont plus de potentiel pour tourner. Les petites sollicitations ne permettent plus de se décaler dans les couloirs, voire même ont l’effet inverse. C’est ce comportement que les faisceaux cherchent à caractériser, ils permettent d’objectiver la qualité des \mathbf{E}_i . Néanmoins, il faut toujours garder à l’esprit que cette caractérisation se fait autour d’une trajectoire de référence et que celle-ci possède aussi son influence sur les résultats. Ce type de calcul illustre également la tolérance du véhicule aux petites erreurs de pilotage que pourrait faire un essayeur.

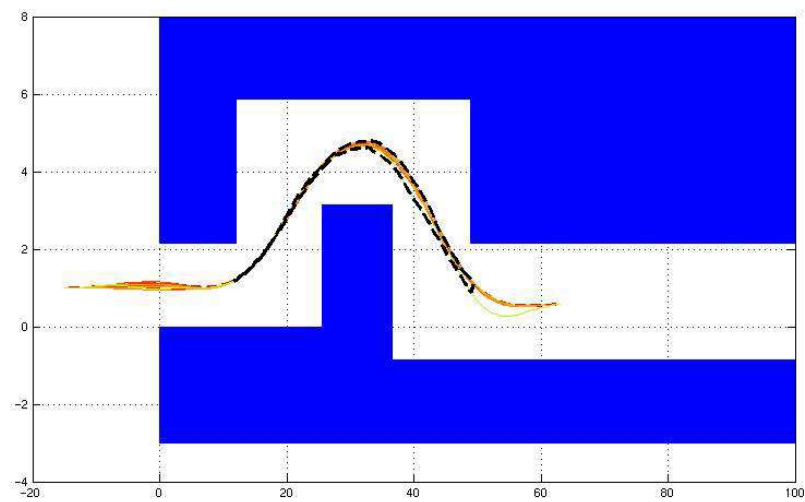


(a) 72 km/h

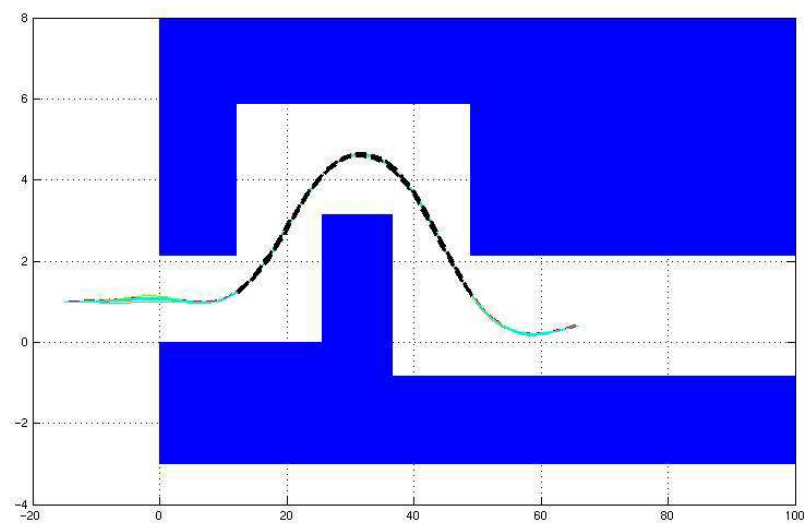


(b) 65 km/h

FIG. 6.8 – Comparaison de deux faisceaux de trajectoires issus du véhicule A mais calculés pour deux vitesses de passage différentes avec la méthode de DT



(a) ESP activé



(b) ESP désactivé

FIG. 6.9 – Comparaison de deux faisceaux de trajectoires issus du véhicule C franchissant le test à la vitesse de 67 km/h mais avec activation ou non de l'ESP

6.4 Approche avec des essayeurs humains

Les faisceaux de trajectoires peuvent aussi être interprétés comme étant la capacité d'un véhicule à permettre à des conducteurs aux styles de conduite différents de franchir un seul et même test sans encombre (par exemple le franchissement d'un test VDA à une vitesse inférieure à V_{max}). En effet de nombreuses études telles que [Pascali *et al.*, 2002], [Rizzi, 2005] et [Whittaker *et al.*, 2001] montrent l'existence de styles de conduite très différents sur ces tests. D'un côté, certains essayeurs suivent des trajectoires très dynamiques en mettant trop d'angle et de vitesse volant puis en corrigeant. Ces sollicitations sont caractérisées par des oscillations importantes et inutiles, tandis que les trajectoires présentent souvent des dérives excessives. D'autre part, certains pilotes ont tendance à minimiser les actions au volant et à ne mettre que le minimum d'angle et de vitesse volant nécessaire pour franchir le test. Les trajectoires résultantes sont souvent assez tendues et passent au plus près des obstacles. La figure 6.10 illustre bien ces différences.

Une autre constatation expérimentale est qu'un essayeur ayant passé un test avec succès plusieurs fois, a tendance à reproduire le même type d'angle volant. La variance entre deux conducteurs est bien plus importante qu'entre différents passages d'un seul et même essayeur. La recherche d'un ensemble de trajectoires solutions n'est pas le but des essais réels. Cependant, des personnes ont pu piloter le modèle SimulinkCar et déterminer des faisceaux de trajectoires grâce à l'utilisation d'un simulateur de conduite.

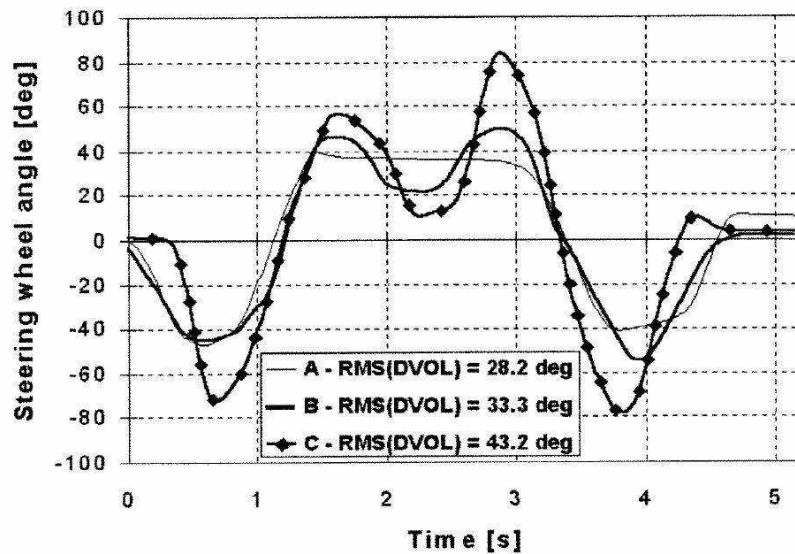


FIG. 6.10 – Différents styles de conduites sur un test d'évitement ISO

6.4.1 Description du simulateur de conduite

Le simulateur de conduite SHERPA est un système temps-réel où un conducteur humain est présent dans la boucle de simulation pour fournir ses sollicitations au modèle. Le modèle de véhicule SimulinkCar est utilisé pour intégrer les équations du mouvement et il fournit la position du véhicule résultant du calcul. L'état du véhicule est ensuite retranscrit sous forme d'une vue

tridimensionnelle qui est affichée sur 3 écrans géants (voir figure 6.12). Un retour d'effort au volant est fourni par un moteur électrique afin de simuler le couple renvoyé par la LAS. Le conducteur peut alors adapter son comportement en fonction des informations disponibles. L'immersion est complète grâce au réalisme de la cabine de pilotage (Citroën C3 de série) et du système de contrôle (volant et colonne de direction de série). La figure 6.11 présente un exemple du rendu 3D de ces simulations.



FIG. 6.11 – Rendu 3D d'un passage de VDA avec un essai humain pilotant un modèle SimulinkCar



FIG. 6.12 – Simulateur de conduite permettant à un essai humain de piloter le modèle SimulinkCar

Ce genre d'essais se situe à la frontière entre l'étude objective des capacités mécaniques du véhicule à éviter un obstacle et l'étude de la capacité d'un être humain à exploiter ce potentiel. Par conséquent, elle sort quelque peu du cadre officiel de cette thèse, mais permet d'obtenir des résultats très pertinents notamment pour les faisceaux.

L'utilisation d'un simulateur permet de réaliser de nombreux tests qui seraient bien plus contraignants et coûteux à réaliser avec des essais sur piste. Autre avantage : les différences liées à la non-représentativité du modèle par rapport au vrai véhicule sont supprimées, car essayeurs humains et techniques de planification de mouvements exploitent strictement le même modèle. A défaut de pouvoir trouver de très nombreux conducteurs, il a été demandé à quelques personnes suffisamment expérimentées de se forcer à faire varier leur style de conduite, de manière

à trouver un maximum de manières de franchir le test (i.e. de "solutions possibles" au problème de planification de mouvements dynamiques).

6.4.2 Protocole de mesure

Le protocole utilisé pour la détermination des faisceaux de trajectoires avec un modèle piloté par essayeur humain est le suivant :

- Utilisation du test d'évitement VDA comme test de référence
- Deux séries de tests sur deux véhicules différents (A et C)
- Chaque essayeur sur chaque véhicule effectue une recherche de la vitesse de passage maximale V_{max} du test VDA
- Chaque série est faite à au moins 2 vitesses différentes ($V_{max} \times 0.8$ et $V_{max} \times 0.9$) en partant toujours du même point (10 m avant les premiers cônes du VDA au centre du premier couloir). Le respect de ces conditions initiales est assuré par le travail sur simulateur.
- Chaque essayeur passe le VDA en s'efforçant de varier son style de conduite et en essayant de découvrir un maximum de solutions différentes (en termes de trajectoire, dérive, angle volant, vitesse volant, etc.). On retient au minimum une quinzaine de solutions différentes (trajectoires sans collisions) pour chaque trinôme essayeur/véhicule/vitesse
- Seules les trajectoires sans collisions sont conservées et l'on enregistre au passage les grands classiques de la dynamique route
- Post-traitement et exploitation des faisceaux

Le modèle d'ESP du véhicule C n'a pas pu être testé, car cet ajout rend le modèle plus lent que le temps réel. De plus, des problèmes informatiques d'intégration de cette boîte noire dans le simulateur ont condamné cette tentative.

6.4.3 Résultats

La figure 6.13 présente les angles volant obtenus lors de la détermination de V_{max} sur ce modèle. Il est intéressant de les comparer à ceux obtenus par des essayeurs réels sur la figure 2.19 et à ceux obtenus par les différentes méthodes détaillées précédemment. On constate que la corrélation entre mesures véhicule/mesures SHERPA/planification de mouvements est toujours correcte.

A contrario, les valeurs des vitesses de passage maximum se sont révélées assez inférieures à celles obtenues par les algorithmes détaillés précédemment. La difficulté de passage s'est avérée particulièrement élevée même à faible vitesse ($\leq 50 \text{ km/h}$). L'extrême rigueur de l'algorithme de détection de collisions qui sanctionne des trajectoires qui auraient vraisemblablement été validées en réalité explique en partie ces résultats. Les V_{max} ainsi obtenues avec le véhicule A ont rarement dépassé les 60 km/h contre 70 km/h pour l'algorithme de déformation de trajectoire. Une évaluation de la "quantité" de collisions générées pendant un test raté peut-être donnée par le calcul de l'aire (en m^2) de l'intersection entre la surface balayée par la géométrie du véhicule et les obstacles du test (voir figure 6.14). La surface balayée est l'union des géométries du véhicule prises en chaque configuration de la trajectoire discrétisée. Ce calcul est implémenté à l'aide de la librairie GPC et de l'algorithme SweepLine.

Le taux d'échec s'est révélé particulièrement élevé avec des valeurs de l'ordre de 1 passage réussi pour 50 essais. Afin de rendre le test plus facile et plus réaliste, une tolérance de 0.2 m^2 a été arbitrairement définie. Celle-ci permet de valider les trajectoires légèrement en collision. Le

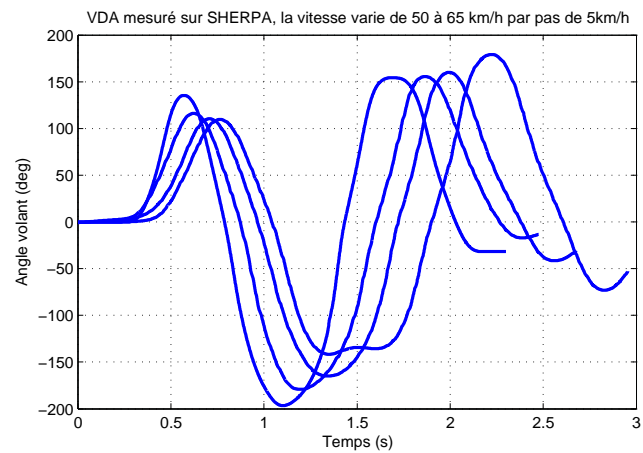


FIG. 6.13 – Recherche de la vitesse de passage maximum d'un test VDA par un essayeur humain sur le simulateur SHERPA avec le véhicule A

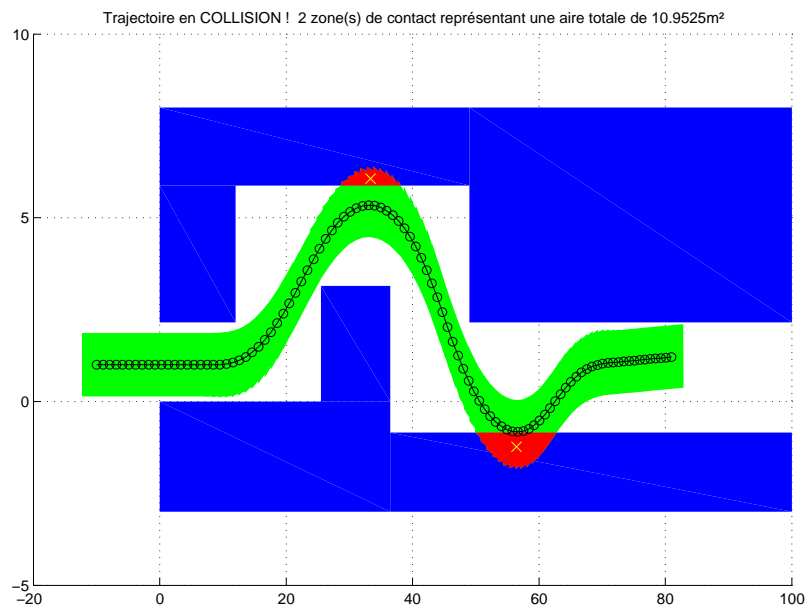


FIG. 6.14 – Evaluation de la "quantité" de collisions d'un test raté grâce à l'aire des zones d'intersection (en rouge) entre la surface balayée par le véhicule et les obstacles du test

taux d'échec remonte ainsi à un peu moins de 1 sur 10. Les figures 6.15 et 6.16 présentent les faisceaux de trajectoires, d'angle volant et de dérive obtenus par la superposition de l'ensemble des résultats des essayeurs humains pilotant le véhicule A à 50 km/h et à 60 km/h .

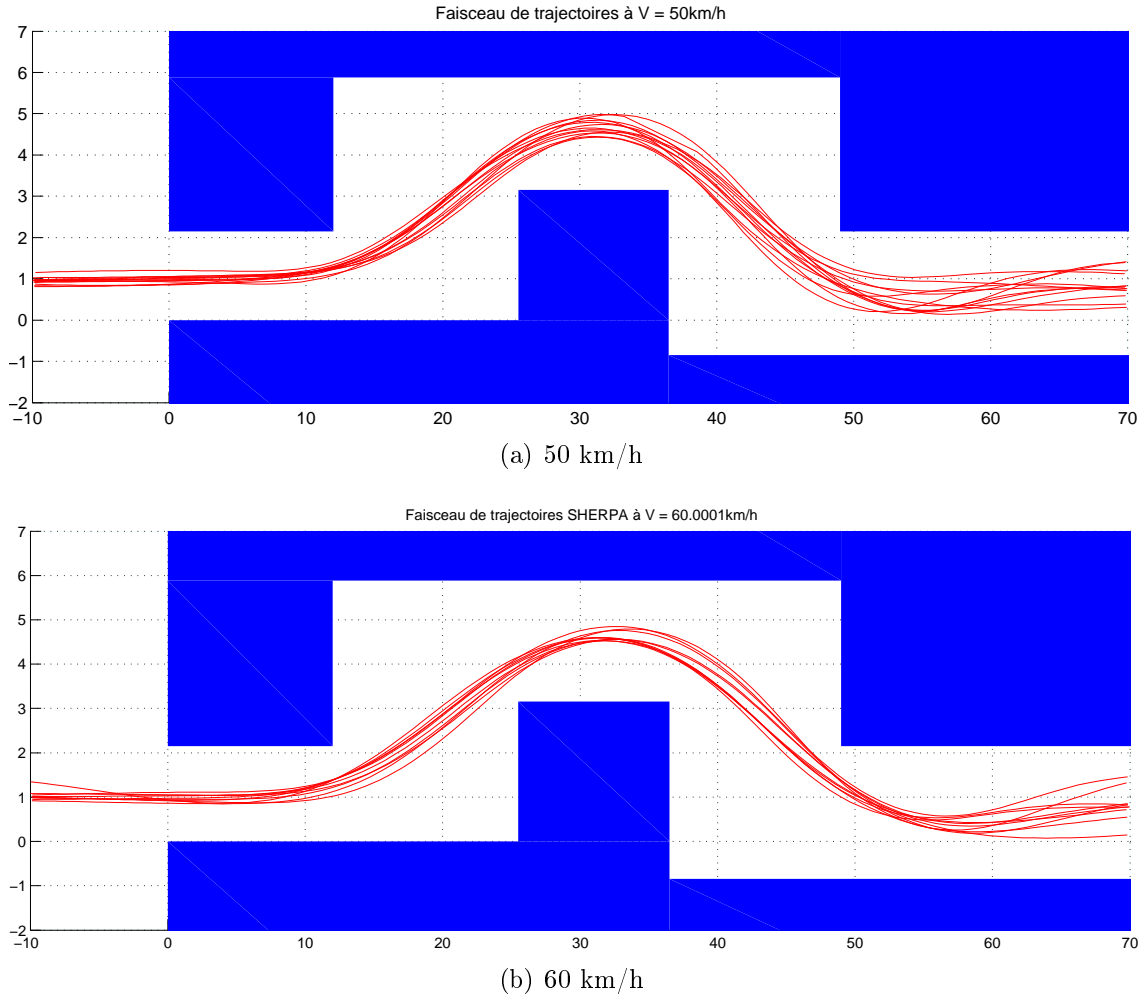
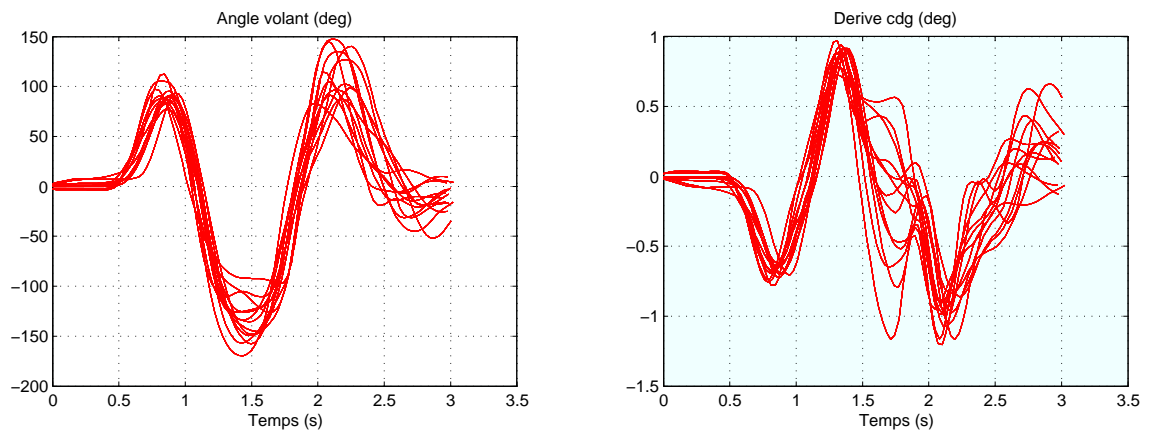


FIG. 6.15 – Faisceau de trajectoires obtenu par l'ensemble des essayeurs humains sur le modèle du véhicule A

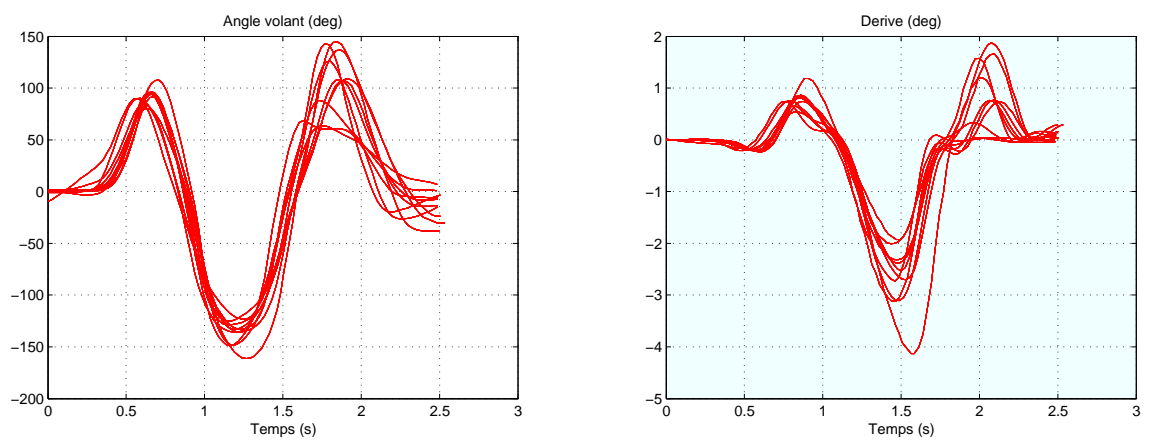
A 50 km/h , on observe une grande diversité des angles volant solutions. Cela confirme l'incapacité des méthodes locales à identifier l'ensemble des solutions possibles à des vitesses aussi faibles et la nécessité d'utiliser des méthodes exploratoires. La corrélation la plus intéressante, à proximité de V_{max} , n'a pu être effectuée faute d'avoir des mesures essayeurs à ces vitesses là.

6.5 Synthèse

Tous ces résultats encourageants sont à considérer avec précaution. Aucune garantie ne peut être donnée tant en termes de robustesse qu'en fiabilité intrinsèque des résultats fournis. Un des défauts de ces techniques est leur manque de précision, i.e. leur incapacité à discriminer des véhicules proches. Leur robustesse à des variations paramétrique paraît acceptable. Ces calculs



(a) 50 km/h



(b) 60 km/h

FIG. 6.16 – Faisceau d'angles volant et de dérives cdg obtenu par l'ensemble des essayeurs humains sur le modèle du véhicule A

permettent d'obtenir des informations jusqu'alors inaccessibles. La taille et la forme des différents faisceaux pourront être utilisées par les ingénieurs afin de concevoir des véhicules plus sûrs.

Chapitre 7

Conclusion

7.1 Synthèse technique

Suivant le problème considéré, chacune des techniques précédentes a son champ d'application propre qu'il convient d'exploiter au mieux.

Problèmes 1 et 2 La question est ici de trouver une solution quelconque au test considéré, ainsi que la limite de \mathcal{P} après laquelle il n'existe plus de solution.

- Exploration : Calcul de solutions uniquement loin des limites. Robustesse et précision faible.
- Optimisation : Résolution rapide sur tout le domaine de \mathcal{P} . Convergence non garantie, résultats parfois aberrants. Robustesse un peu en retrait par rapport à la DT.
- Déformation : Calcul des limites de \mathcal{P} plus lent, mais précis et robuste. Résultats intermédiaires cohérents et crédibles.

Problème 3 La question est de déterminer un ensemble de trajectoires solutions aussi représentatif que possible pour une valeur de \mathcal{P} donnée.

- Exploration : Larges faisceaux de trajectoires uniquement loin des limites de \mathcal{P} . Robustesse faible.
- Optimisation : Faisceaux lents, solutions peu nombreuses mais représentatives. Robustesse moyenne.
- Déformation : Faisceau local rapide, mais limité au domaine proche des limites de \mathcal{P} . Robustesse moyenne.

Résumé Il convient de bien choisir son algorithme suivant ce que l'on souhaite obtenir comme résultat. Des combinaisons de ces différentes techniques ont d'ailleurs été réalisées afin de bien exploiter le potentiel de chacune. Par exemple, il est possible de commencer par une exploration loin des limites de \mathcal{P} afin d'obtenir une condition initiale cohérente pour la méthode de contrôle optimal. Si la méthode ne converge pas, on utilise la méthode de déformation afin d'obtenir un résultat cohérent. Les combinaisons possibles et cas d'utilisation sont nombreux.

La robustesse des calculs effectués est généralement acceptable, mais leur précision et leur capacité de discrimination ne sont pas aussi élevées que ce que l'on aurait pu souhaiter. La

corrélation avec les mesures réelles est généralement bonne, mais les modèles manquent souvent de représentativité à ces niveaux de sollicitations. La part de responsabilité incombant à la modélisation ou à la planification est difficile à évaluer. Aucune garantie de convergence ou preuve de fiabilité ne peut d'ailleurs être donnée de manière formelle, notamment à cause de la diversité des systèmes ou situations de vies potentiellement étudiables. Néanmoins, les résultats des simulations ont été jugés suffisamment encourageants et fiables pour permettre leur éventuelle exploitation chez PSA Peugeot Citroën.

Une des grandes forces de l'approche développée dans cette thèse est la complète indépendance vis-à-vis des paramètres du problème (modèles, obstacles). En effet, il est tout à fait possible de rajouter facilement des éléments quelconques au modèle pour réaliser des tests inédits et plus réalistes vis-à-vis des situations rencontrées par les clients. Les innovations futures du monde automobile pourront facilement être intégrées en modifiant simplement le modèle informatique. Nul besoin de recalculer les gradients d'un modèle analytique ou de coder les modifications dans un langage éloigné des réalités de la conception automobile.

7.2 Utilisation industrielle

Afin de faciliter la prise en main de ces techniques, une interface graphique a été réalisée sous forme d'un assistant permettant le paramétrage automatique et le choix des algorithmes adaptés au problème étudié. L'idée est de permettre l'utilisation de ces outils par des ingénieurs novices qui devront obtenir les résultats souhaités de la manière la plus facile possible (i.e. avec le moins de formation possible). L'assistant pose un ensemble de questions permettant de déterminer le paramétrage des algorithmes le plus adapté (voir figure 7.1). A chaque étape, des outils sont proposés afin de faciliter l'introduction des données nécessaires (géométrie, fichier FDV, etc.). Un mode avancé permet de modifier les paramètres pré-réglés par l'assistant. On présente ci-dessous un exemple d'étapes suivies par un utilisateur :

- Introduction/Aide
- Choix du modèle de véhicule SimulinkCar à utiliser
- Choix du fichier FDV de paramétrage (importateur disponible)
- Choix de la géométrie du véhicule (éditeur disponible)
- Choix de la géométrie des obstacles (éditeur disponible)
- Choix du type de calcul à effectuer (recherche de limites de \mathcal{P} , solution à \mathcal{P} donné, faisceaux, détection de collisions, etc.) et de l'algorithme à employer.
- Choix des conditions initiales si besoin est (\mathcal{P})
- Lancement du calcul
- Lancement de l'interface de post-traitement des résultats adaptée (faisceaux, évolution de \mathcal{P} et des grandeurs physiques associées).

7.3 Perspectives

Grâce aux choix qui ont été faits au chapitre 1.3 et à la grande généralité des techniques étudiées, une infinité de tests nouveaux pourront facilement être définis et étudiés. Les prestations en sécurité active d'un modèle de direction à démultiplication variable ou l'adjonction d'un train arrière directif pourront aisément être évaluées sans nécessiter la réalisation d'un prototype. Tous les tests consommateurs qui viendraient à être utilisés dans l'avenir (voir chapitre 2.3)

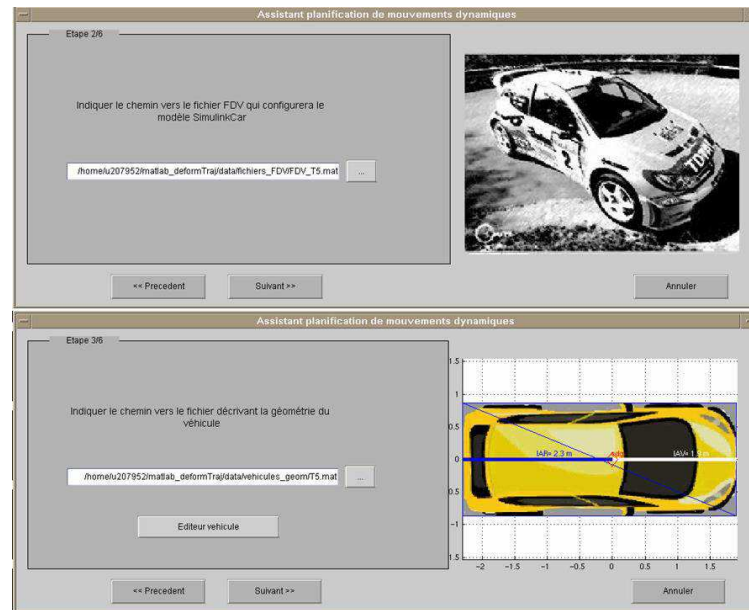


FIG. 7.1 – Assistant permettant aux ingénieurs d'utiliser simplement les outils de planification de mouvements dynamiques

pourront facilement être intégrés à l'outil et étudiés dans les limites vues précédemment. De plus, n'importe quelle contrainte autre que la simple "non-collision" avec les obstacles peut facilement être prise en compte et potentiellement donner naissance à bien plus que les tests standardisés d'aujourd'hui. De même, l'étude de l'ensemble des trajectoires solutions pourrait permettre aux ingénieurs de concevoir des véhicules à la fois plus performants et plus sécurisants pour tous les conducteurs.

Annexe A

Interface Matlab pour la Motion Strategy Library

La librairie Motion Strategy Library (MSL) est une ensemble de classes C++ initialement développées par Lavalle et Kuffner afin de pouvoir utiliser facilement des algorithmes de planification de mouvements dynamiques (FDP, RRT, PRM, etc.). L'ensemble de classes définit un "Framework" complet qui permet d'utiliser facilement toutes sortes de planificateurs, de modèles ou de solveurs. La librairie de détection de collisions PQP est intégrée. La MSL se configure par l'utilisation de nombreux fichiers textes contenant les définitions nécessaires à la description du problème (obstacles, véhicule, modèle, CI, etc.). Une interface spécifique de post-traitement/visualisation des résultats en OpenGL est aussi présente. Cette librairie est proposée gratuitement en OpenSource tant pour les applications académiques que commerciales à l'adresse <http://msl.cs.uiuc.edu/msl/>. L'objectif est de faciliter la diffusion de ces techniques encore trop confidentielles.

Cette librairie, bien que très puissante, s'utilise très difficilement avec Matlab de cette manière. Il faut générer le code C correspondant au modèle Simulink à l'aide d'une Toolbox spécifique de Matlab (Real Time Workshop), créer une classe dédiée pour utiliser le code généré, puis recompiler l'ensemble du programme. Cette procédure très lourde empêche la modification rapide du modèle de véhicule par les ingénieurs. Pour pallier à ces problèmes, le coeur du code de la MSL a été interfacé avec Matlab sous forme de mex-fonctions pré-compilées. On peut ainsi directement paramétrer la simulation sous Matlab à l'aide d'une structure `options` dans le workspace. Les fichiers textes de la MSL sont ainsi avantageusement remplacés. La mex-fonction `motion_planner` est appelée de manière transparente depuis Matlab comme s'il s'agissait d'une vraie fonction Matlab. Le code C++ de la MSL est appelé nativement avec les paramètres adéquats permettant ainsi un calcul bien plus rapide que si tous ces algorithmes avaient du être retranscrits sous Matlab. Les structures du C++ sont en effet bien plus optimisées et adaptées que leurs équivalents en langage Matlab interprété.

Le code dispose à présent d'une classe de modèle supplémentaire `ModelSimulink` qui correspond à un modèle Simulink pouvant être appelé nativement depuis la mex-fonction sans génération de code. Le résultat du calcul est retourné sous Matlab dans une structure `results`. Cette approche permet d'allier la simplicité et la souplesse Matlab pour le pré/post-traitement et la modélisation, à la puissance du C++ pour l'implémentation des algorithmes de planification de mouvements.

La modification des algorithmes de planification eux-même requiert cependant une recompilation des mex-fonctions (automatisée). La compilation des mex-fonctions a été testée sous Windows 2000, HPUNIX et Sun avec Matlab 7.1 et ne pose aucun problème grâce à la réduction du code originel qui a été opérée (suppression de l'interface graphique et des appels non portables). Le post-traitement de cette structure `results` est assuré grâce à des scripts Matlab classiques et aux fonctions de tracé propres à Matlab. Pour l'instant seuls les algorithmes FDP, RRT, RCRTT, KDP et leurs variantes sont disponibles. Les algorithmes PRM sont beaucoup moins bien adaptés à cette approche, car les méthodes locales sont difficiles à implémenter sous Simulink.

Un exemple d'utilisation de cette interface est présenté ci-dessous. A l'exception des paramètres liés à Simulink et aux fichiers FDV, les paramètres de la structure `options` correspondent exactement aux noms des fichiers textes requis pour faire fonctionner la MSL en temps normal. Leur signification et les valeurs possibles sont détaillées dans la documentation associée à la librairie MSL.

`%Exemple d'utilisation de l'interface Matlab de la MSL`

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Réglage des paramètres de la simulation
options.Model='ModelSimulink';%utilisation d'un modèle Simulink
options.SimulateFcn='simulate_skcar';%nom du modèle simulink appelé
options.StateDim=24;%nombre de variables d'état
options.InputDim=1;%nombre d'entrées
options.Planner='FDP';%'FDP';'RRT';'RCRRT';'KDP'; %Choix du planificateur
options.Geom='GeomPQP2DRigid';%choix de PQP pour la détection de collision
options.GeomDim=2;%monde en 2D
options.NumNodes=100000;%nombre de points essayés avant arrêt
options.PlannerDeltaT=0.2;%pas de temps
options.Robot=[1.906900 0.865000 -2.295000 0.865000 -2.295000 -0.865000 ...
  1.906900 -0.865000];%définition géométrique du véhicule
options.GoalState=[79 0 0.58 0 0 0 13.8889 0 0 0 0 0 0 0 0 0 0 0 0 ...
  0 0 0 0 0];%état d'arrivée
options.LowerState=[0.000000 0.000000 0 0 0 -3.14 0 0 0 0 0 0 0 0 0 0 ...
  0 0 0 0 0 0 0 0];%Bornes min sur l'état
options.UpperState=[91.000000 5.883000 0 0 0 3.14 0 0 0 0 0 0 0 0 0 0 ...
  0 0 0 0 0 0 0 0];%Bornes sup sur l'état
options.GridDimensions=[100 100 1 1 1 100 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
%Discrétisation de la grille utilisée par FDP
options.ModelDeltaT=1e-3;%inutile avec les modèles simulink
% utilisation du solveur choisi dans le modèle
options.GapError=[3 3 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 ...
  1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 1e+10 ...
  1e+10];%tolérance sur la zone d'arrivée
options.VitVolantMax=14;%limitation en vitesse volant
options.AccelVolantMax=84;
options.ThetaIndex=6;%position de l'angle de rotation dans le vecteur d'état
options.MultiPath_activated=0;%faisceaux ON/OFF
options.FDVFile='FDV_T5.mat';%choix du fichier FDV
% présent pour info, à charger manuellement

```

```

options.Comments='SkCar simulation';%commentaires
load(options.FDVFile);%chargement du fichier FDV
stabilisation;%calcul des états initiaux de SimulinkCar
% grâce à une routine spécifique (crée X0 XOS et XOR)
options.InitialState=[X0 XOS XOR];%Etat initial
options.InitialInput=[0];%entrée courante à t=0 %must be of InputDim
options.Inputs=[linspace(-pi,pi,25)'];%discrétisation de l'entrée
options.Obst=[...
12.0000    2.1500   12.0000    5.8800         0    5.8800;...
12.0000    2.1500         0    5.8800         0    2.1500;...
49.0000    5.8800   49.0000    8.0000         0    8.0000;...
49.0000    5.8800         0    8.0000         0    5.8800;...
100.0000    2.1500  100.0000    8.0000   49.0000    8.0000;...
100.0000    2.1500   49.0000    8.0000   49.0000    2.1500;...
36.5000   -3.0000   36.5000         0         0         0;...
36.5000   -3.0000         0         0         0   -3.0000;...
36.5000         0   36.5000    3.1500   25.5000    3.1500;...
36.5000         0   25.5000    3.1500   25.5000         0;...
100.0000   -3.0000  100.0000   -0.8500   36.5000   -0.8500;...
100.0000   -3.0000   36.5000   -0.8500   36.5000   -3.0000;...
]%définition géométrique des obstacles sous forme de triangles
%x1 y1 x2 y2 x3 y3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Lancement de la planification de mouvement proprement dite
results=motion_planner(options);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%lancement du post-traitement des résultats (affichage trajectoire,entrée,...)
post_motion_planner(results);

```


Bibliographie

- [Ahuactzin, 1994] AHUACTZIN, J. (1994). *Le Fil d'Ariane : une méthode de planification générale. Application à la planification automatique de trajectoires*. Thèse de doctorat, INP, Grenoble, France.
- [Ahuactzin et al., 1992] AHUACTZIN, J. M., TALBI, E.-G., BESSIÈRE, P. et MAZER, E. (1992). Using genetic algorithms for robot motion planning. In *10th Europ. Conf. Artific. Intelligence*, pages 671–675, London, England.
- [Allen et J.P.Christos, 2002] ALLEN, R. et J.P.CHRISTOS (2002). Driver/vehicle modeling and simulation. *SAE Transactions Paper*, 2002-01-1568.
- [Aurenhammer, 1991] AURENHAMMER, F. (1991). Voronoï diagrams – A survey of a fundamental geometric structure. *ACM Computing Surveys*, 23:345–405.
- [Barber et al., 1996] BARBER, C., DOBKIN, D. et HUHDANPAA, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483.
- [Barclay et al., 1997] BARCLAY, A., GILL, P. et ROSEN, J. (1997). Sqp methods and their application to numerical optimal control.
- [Barraquand et al., 1997] BARRAQUAND, J., KAVRAKI, L. E., LATOMBE, J.-C., LI, T.-Y., MOTWANI, R. et RAGHAVAN, P. (1997). A random sampling scheme for path planning. *International Journal on Robotics Research*, 16(6):759–775.
- [Barraquand et al., 1992] BARRAQUAND, J., LANGLOIS, B. et LATOMBE, J.-C. (1992). Numerical potential field techniques for robot path planning. *IEEE Tr. Syst., Man, and Cybern.*, 22(2):224–241.
- [Barraquand et Latombe, 1993] BARRAQUAND, J. et LATOMBE, J.-C. (1993). Nonholonomic multibody mobile robots : Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155.
- [Bellman, 1957] BELLMAN, R. E. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ.
- [Bernard et al.,] BERNARD, J., GRUENING, J. et HOFFMEISTER, K. Evaluation of vehicle/driver performance using genetic algorithms. SAE technical paper.
- [Bertsekas, 2001] BERTSEKAS, D. P. (2001). *Dynamic Programming and Optimal Control, Vol. I, 2nd Ed.* Athena Scientific, Belmont, MA.
- [Betts, 1998] BETTS, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207.
- [Bonnafoous, 2003] BONNAFOOUS, D. (2003). *Exécution réactive de trajectoire pour robots mobiles non-holonomes*. Thèse de doctorat, INP Toulouse.
- [Bosch, 2001] BOSCH (2001). Cahier technique : Contrôle dynamique de trajectoire esp. Rapport technique.

- [Boyer et Lamiroux, 2006] BOYER, F. et LAMIRAUX, F. (2006). Trajectory optimization applied to kinodynamic motion planning for a realistic car model. *IEEE International Conference on Robotics and Automation*, pages 487–492.
- [Bryson, 1975] BRYSON, A. E. (1975). *Applied Optimal Control*. Taylor & Francis.
- [Butz et Stryk, 2005] BUTZ, T. et STRYK, O. (2005). Optimal control based modeling of vehicle driver properties. *SAE Technical Papers*.
- [Chazelle, 1987] CHAZELLE, B. (1987). Approximation and decomposition of shapes. In SCHWARTZ, J. T. et YAP, C. K., éditeurs : *Algorithmic and Geometric Aspects of Robotics*, pages 145–185. Lawrence Erlbaum Associates, Hillsdale, NJ.
- [Chazelle, 1991] CHAZELLE, B. (1991). Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6(5):485–524.
- [Chazelle, 2000] CHAZELLE, B. (2000). *The Discrepancy Method*. Cambridge University Press, Cambridge, U.K.
- [Cheng, 2005] CHENG, P. (2005). *Sampling-Based Motion Planning with Differential Constraints*. Thèse de doctorat, University of Illinois, Urbana, IL.
- [Cheng et LaValle, 2001] CHENG, P. et LAVALLE, S. M. (2001). Reducing metric sensitivity in randomized trajectory design. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 43–48.
- [Day et Metz, 2000] DAY, T. et METZ, L. (2000). The simulation of driver inputs using a vehicle driver model. *SAE Technical Papers*, (2000-01-1313).
- [Dijkstra, 1959] DIJKSTRA, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.
- [Divelbiss et Wen, 1994] DIVELBISS, A. W. et WEN, J. T. (1994). Nonholonomic path planning with inequality constraints. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 52–57.
- [Divelbiss et Wen, 1997] DIVELBISS, A. W. et WEN, J. T. (1997). A path-space approach to nonholonomic planning in the presence of obstacles. *IEEE Transactions on Robotics & Automation*, 13(3):443–451.
- [Dobkin et al., 1993] DOBKIN, D. P., HERSHBERGER, J., KIRKPATRICK, D. G. et SURI, S. (1993). Computing the intersection-depth of polyhedra. *Algorithmica*, 9(6):518–533.
- [Donald et al., 1993] DONALD, B., XAVIER, P., CANNY, J. et REIF, J. (1993). Kinodynamic motion planning. *Journal of the ACM*, 40:1048–1066.
- [Fenaux, 2005] FENAUX, E. (2005). Problèmes particuliers pour évaluer la sécurité active présentation d’une démarche. *Articles techniques SIA*.
- [Fraichard et Scheuer, 1994] FRAICHARD, T. et SCHEUER, A. (1994). Car-like robots and moving obstacles. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 64–69.
- [Fraichard et Scheuer, 2004] FRAICHARD, T. et SCHEUER, A. (2004). From Reeds and Shepp’s to continuous-curvature paths. *IEEE Transactions on Robotics*, 20(6):1025–1035.
- [Fujioka et al., 1991] FUJIOKA, T., NOMURA, K., TAKUBO, N. et CHUNG, D. H. (1991). Driver model using neural network system. *SAE Technical Papers*, (912562).
- [Garrott et al., 1982] GARROTT, W., WILSON, D. et SCOTT, R. (1982). Closed loop automobile maneuvers using preview-predictor models. *SAE Technical Papers*, (820305).
- [Gill et al., 1981] GILL, P., MURRAY, W. et WRIGHT, M. (1981). *Practical Optimization*. Academic Press. references matlab.

- [Gillespie, 1992] GILLESPIE, T. N. (1992). *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, Warrendale, PA.
- [Hajjaji et Ouladsine, 2001] HAJJAJI, A. E. et OULADSINE, M. (2001). Modeling human vehicle driving by fuzzy logic for standardized iso double lane change maneuver. *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, pages 499 – 503.
- [Halconruy, 1995] HALCONRUY, T. (1995). *Les liaisons au sol*. ETAI.
- [Halton, 1960] HALTON, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90.
- [Han, 1977] HAN, S. (1977). A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297. reference matlab.
- [Hermann et Krener, 1977] HERMANN, R. et KRENER, A. (1977). Nonlinear controllability and observability. *IEEE Tr. on Automatic Control*, 22:728–740.
- [Hsu, 2000] HSU, D. (2000). *Randomized Single-Query Motion Planning in Expansive Spaces*. Thèse de doctorat.
- [Hsu et al., 1999] HSU, D., LATOMBE, J.-C. et MOTWANI, R. (1999). Path planning in expansive configuration spaces. *International Journal of Computational Geometry & Applications*, 4:495–512.
- [ISO, 1996] ISO (1996). Norme iso 12021-1 véhicules routiers – sensibilité au vent latéral – partie 1 : Méthode en boucle ouverte avec génération de vent.
- [ISO, 1999] ISO (1999). Norme iso 3888-1 voitures particulières – piste d’essai de déboîtement latéral brusque – partie 1 : Double déboîtement.
- [ISO, 2002] ISO (2002). Norme iso 3888-2 voitures particulières – piste d’essai de déboîtement latéral brusque – partie 2 : Évitement d’obstacle.
- [ISO, 2006] ISO (2006). Norme iso 7975 voitures particulières – freinage en virage – méthode d’essai en boucle ouverte.
- [Jacobs et al., 1991] JACOBS, P., REGE, A. et LAUMOND, J.-P. (1991). Non-holonomic motion planning for hilare-like robots. *In International Symposium on Intelligent Robotics*, pages 338–347, Bangalore.
- [Kavraki et al., 1996] KAVRAKI, L. E., SVETKA, P., LATOMBE, J.-C. et OVERMARS, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics & Automation*, 12(4):566–580.
- [Khatib et Chatila, 1995] KHATIB, M. et CHATILA, R. (1995). An extended potential field approach for mobile robot sensor-based motions. *In International Conference on Intelligent Autonomous Systems*, pages 490–496, Karlsruhe, Germany.
- [Lamiriaux et Bonnafous, 2002] LAMIRAUX, F. et BONNAFOUS, D. (2002). Reactive trajectory deformation for nonholonomic systems : Application to mobile robots. *In International Conference on Robotics and Automation*, pages 3099–3104, Washington D.C. IEEE.
- [Lamiriaux et al., 2004a] LAMIRAUX, F., BONNAFOUS, D. et LEFEBVRE, O. (2004a). Reactive path deformation for non-holonomic mobile robots. *IEEE Transactions on Robotics*, 20(6):967–977.
- [Lamiriaux et Laumond, 1997] LAMIRAUX, F. et LAUMOND, J.-P. (1997). Flatness and small-time controllability of multi-body mobile robots : applications to motion planning. *In European Control Conference*, Brussels, Belgium.

- [Lamiroux *et al.*, 2004b] LAMIRAUX, F., LAUMOND, J.-P., VANGHEEM, C., BOUTONNET, D. et RAUST, G. (2004b). Trailer-truck trajectory optimization for airbus a380 component transportation. *IEEE Robotics and Automation Magazine*, (Robotic Technologies Applied to Intelligent Transportation Systems). to appear.
- [Lamiroux *et al.*, 1999] LAMIRAUX, F., SEKHAVAT, S. et LAUMOND, J.-P. (1999). Motion planning and control for Hilare pulling a trailer. *IEEE Transactions on Robotics & Automation*, 15(4):640–652.
- [Larsen *et al.*, 1999] LARSEN, E., GOTTSCHALK, S., LIN, M. et MANOCHA, D. (1999). Fast proximity queries with swept sphere volumes. Rapport technique, Department of Computer Science, University of N. Carolina.
- [Laumond, 1995] LAUMOND, J. (1995). Nonholonomic motion planning via optimal control. In GOLDBERG, K., HALPERIN, D., LATOMBE, J.-C. et WILSON, R., éditeurs : *Algorithmic Foundations of Robotics*, pages 227–238, Wellesley, MA. A K Peters, Ltd.
- [LaValle, 2006] LAVALLE, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K. Available at <http://planning.cs.uiuc.edu/>.
- [LaValle et Kuffner, 1999] LAVALLE, S. M. et KUFFNER, J. J. (1999). Randomized kinodynamic planning. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 473–479.
- [LaValle et Kuffner, 2000] LAVALLE, S. M. et KUFFNER, J. J. (2000). Rapidly-exploring random trees : Progress and prospects. In *Proceedings Workshop on the Algorithmic Foundations of Robotics*.
- [Lefebvre *et al.*, 2005] LEFEBVRE, O., LAMIRAUX, F. et BONNAFOUS, D. (2005). Fast computation of robot-obstacle interactions in nonholonomic trajectory deformation. In *International Conference on Robotics and Automation*, Barcelona, Spain.
- [Lin *et al.*, 1997] LIN, M. C., MANOCHA, D., COHEN, J. et GOTTSCHALK, S. (1997). Collision detection : Algorithms and applications. In LAUMOND, J.-P. et OVERMARS, M. H., éditeurs : *Algorithms for Robotic Motion and Manipulation*, pages 129–142. A.K. Peters, Wellesley, MA.
- [Locatelli, 2001] LOCATELLI, A. (2001). *Optimal Control - An Introduction*. Birkhäuser.
- [Morokoff et Caflisch, 1994] MOROKOFF, W. J. et CAFLISCH, R. E. (1994). Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279.
- [Muehlmeier et Mueller, 2002] MUEHLMEIER, M. et MUELLER, N. (2002). Optimization of the driving line on a race track. *SAE Technical Papers*, (2002-01-3339).
- [Naudé et Steyn, 1993] NAUDÉ, A. et STEYN, J. (1993). Objective evaluation of the simulated handling characteristics of a vehicle in a double lane change manoeuvre. *SAE Transactions Paper*, 930826.
- [Niederreiter, 1992] NIEDERREITER, H. (1992). *Random Number Generation and Quasi-Monte-Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia.
- [Nocedal et Wright, 2000] NOCEDAL, J. et WRIGHT, S. (2000). *Numerical Optimization*. Springer.
- [O’Hara et Robert, 2005] O’HARA et ROBERT, S. (2005). Vehicle path optimization of emergency lane change maneuvers for vehicle simulation. Mémoire de D.E.A., University of Maryland.
- [Pacejka, 2002] PACEJKA, H. (2002). *Tire and Vehicle Dynamics*. Society of Automotive Engineers.

- [Pascali *et al.*, 2002] PASCALI, L., C.DATA, S., SANTI, C. et TAMBURRO, A. (2002). Handling objective evaluation using a parametric driver model for iso lane change simulation. *SAE Technical Papers*.
- [Post *et al.*, 1997] POST, J., BURKE, R., DREXEL, M. et ROBERTSON, J. (1997). An adaptive control model for lateral path following with closed-loop handling simulations. *SAE Technical Papers*.
- [Powell, 1978a] POWELL, M. (1978a). The convergence of variable metric methods for nonlinearly constrained optimization calculations. *Nonlinear Programming 3*. reference matlab.
- [Powell, 1978b] POWELL, M. (1978b). A fast algorithm for nonlinearly constrained optimization calculations. *Lecture Notes in Mathematics*, 630. reference matlab.
- [Prokop et Günther, 2001] PROKOP et GÜNTHER (2001). Modeling human vehicle driving by model predictive online optimization. *Vehicle System Dynamics*, 35:19.
- [Reeds et Shepp, 1990] REEDS, J. et SHEPP, R. (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific J. of Mathematics*, 145:367–393.
- [Reimpell *et al.*, 2001] REIMPELL, J., STOLL, H. et BETZLER, J. W. (2001). *The Automotive Chassis : Engineering Principles*. Society of Automotive Engineers, Troy, MI.
- [Renski, 1998] RENSKI, A. (1998). The driver model and identification of its parameters. *SAE Technical Papers*, (980011).
- [Rimon et Koditschek, 1992] RIMON, E. et KODITSCHKEK, D. E. (1992). Exact robot navigation using artificial potential fields. *IEEE Transactions on Robotics & Automation*, 8(5):501–518.
- [Rizzi, 2005] RIZZI, M. (2005). Steering behaviour of 44 drivers in lane change manoeuvres on a slippery surface. Rapport technique, Linköping University, Department of Science and Technology.
- [Roberts et Shipman, 1972] ROBERTS, S. et SHIPMAN, J. (1972). Two-point boundary value problems : shooting methods. *Elsevier*.
- [Scheuer et Fraichard, 1997] SCHEUER, A. et FRAICHARD, T. (1997). Collision-free and continuous-curvature path planning for car-like robots. *In Proceedings IEEE International Conference on Robotics & Automation*, pages 867–873.
- [Seidel, 1991] SEIDEL, R. (1991). A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry : Theory and Applications*.
- [Shampine *et al.*, 2000] SHAMPINE, L., REICHEL, M. et KIERZENKA, J. (2000). Solving boundary value problems for ordinary differential equations in matlab with bvp4c. Rapport technique.
- [Svestka et Overmars, 1998] SVESTKA, P. et OVERMARS, M. (1998). Probabilistic path planning. *In LAUMOND, J.-P., éditeur : Robot Motion Planning and Control*, pages 255–304. Lecture Notes in Control and Information Sciences, Springer, NY.
- [Trelat, 2005] TRELAT, E. (2005). *Contrôle optimal - théorie et application*. Ciel et Terre.
- [Uzunsoy et O.Olatunbosun, 2003] UZUNSOY, E. et O.OLATUNBOSUN (2003). A generic fuzzy pilot and path planning for vehicle handling dynamics simulation studies. *SAE Technical Papers*, (2003-01-2263).
- [Vögel *et al.*, 2003] VÖGEL, M., von STRYK, O., BULIRSCH, R., WOLTER, T.-M. et CHUCHOŁOWSKI, C. (2003). An optimal control approach to real-time vehicle guidance. *In JÄGER, W. et KREBS, H.-J., éditeurs : Mathematics - Key Technology for the Future*, pages 84–102. Springer-Verlag.

- [von Stryk et Bulirsch, 1992] von STRYK, O. et BULIRSCH, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373.
- [Whittaker et al., 2001] WHITTAKER, L., SPILLANE, A. et JONES, R. (2001). Human performance modeling : Representation of different steering styles. *SAE Technical Papers*, (2001-01-2100).
- [Zhang et al., 2006] ZHANG, L., KIM, Y. J., VARADHAN, G. et MANOCHA, D. (2006). Generalized penetration depth computation. In *SPM '06 : Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 173–184, New York, NY, USA. ACM Press.
- [Zhong et al., 2000] ZHONG, C., K.GUO et H.GUAN (2000). Research on closed-loop comprehensive evaluation method of vehicle handling and stability. *SAE Technical Papers*.