HAL open science

Prédiction de déplacements de véhicules connectés en interaction pour la caractérisation de situations à risques

Tristan Klempka

To cite this version:

Tristan Klempka. Prédiction de déplacements de véhicules connectés en interaction pour la caractérisation de situations à risques. Robotique [cs.RO]. Université Paul Sabatier - Toulouse III, 2022. Français. NNT : 2022TOU30265 . tel-04103991

HAL Id: tel-04103991

https://theses.hal.science/tel-04103991

Submitted on 23 May 2023

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.

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Tristan KLEMPKA

Le 14 novembre 2022

Prédiction de déplacements de véhicules connectés en interaction pour la caractérisation de situations à risques

---

Ecole doctorale : **SYSTEMES**

Spécialité : **Informatique et Robotique**

Unité de recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Patrick DANES

Jury
M. Fawzi NASHASHIBI, Rapporteur
M. Philippe BONNIFAIT, Examinateur
M. Simon LEGLAIVE, Examinateur
Mme Caroline CHANEL, Examinatrice
M. Patrick DANES, Directeur de thèse
M. Roland CHAPUIS, Président

Abstract

Recent advances in information and communication technologies have revolutionized transportation systems. Their use has given rise to a new generation of "intelligent transportation systems" (ITS). These systems rely on perception, reasoning and action functions. They involve real-time information processing and exploit communication capabilities. ITS are used to optimize traffic, reduce pollutant emissions and improve safety.

With more than one billion cars on the road today and world's population constantly growing, road safety is a major issue. To provide an effective risk assessment and measures to prevent dangerous situations, driver assistance systems must be able to understand traffic and predict its evolution. However, driving is a complex process involving the analysis of a large amount of data from different sources, as well as the synthesis of rule-based decisions in a highly dynamic environment. That is why taking into account the behavior of drivers and their mutual interactions is required to predict traffic.

This thesis presents a comprehensive method for motion prediction of interacting vehicles. A communication infrastructure is assumed to enable a real-time exchange of information between all vehicles and a centralized processing system. The predicted road scene and the collision risk assessment can then be fed back to the vehicle drivers for local decision making. The criticality of such a system requires to set up, as much as possible, procedures of explanation, or even certification, of its correct operation. In order to avoid a monolithic closed end-to-end solution, the problem is tackled within the conventional framework of Bayesian filtering and relies on two entities which can be independently addressed before being jointly used. A probabilistic model of the prior dynamics of a fleet of vehicles is first established, which captures the uncertainties underlying the driving process as well as interactions. In view of the complexity of the entailed phenomena, this model cannot be defined in closed-form. So, it is obtained by applying machine learning algorithms to a dataset of trajectories acquired in real conditions. Based on this model and the individual locations received from the vehicles, the estimation and prediction of the fleet state are conducted. Here, specific particle filtering algorithms suited to high-dimensional state spaces are implemented in order to perform inference.

This document details the combination of supervised machine learning methods based on Probability Density Networks (PDN) or unsupervised methods using Conditional Variational Auto-Encoders (CVAE), with filtering algorihms such as the Multiple Particle Filter (MPF) or the Variational Bayesian Multiple Particle Filter (VBMPF). The obtained strategies are completed with an elementary collision risk assessment method. The whole system is evaluated and compared with end-to-end aproaches relying on Long Short Term Memory networks (LSTM) on a large set of

naturalistic highway scenarios. Special focus is given to the capture of medium-term tendencies.

Keywords: Intelligent Transportation Systems, Connected Vehicles, Probabilistic Models, Bayesian Filtering, Machine Learning, Prediction, Collision Risk Assessment

Résumé

Les avancées récentes dans les technologies de l'information et de la communication ont bouleversé les systèmes de transport. Leur exploitation a donné naissance à une nouvelle génération de "systèmes de transport intelligents" (Intelligent Transportation Systems – ITS). Ceux-ci reposent sur des fonctions de perception, de raisonnement et d'action, impliquent un traitement de l'information en temps réel et exploitent des capacités de communication. Les ITS sont utilisés à des fins d'optimisation du trafic, de réduction des émissions polluantes, de sécurité, etc.

Avec plus d'un milliard de voitures en circulation dans le monde, la sécurité routière est un enjeu majeur. Pour permettre une évaluation du risque satisfaisante et la mise en œuvre de mesures de prévention de situations dangereuses, les systèmes d'assistance à la conduite doivent comprendre la circulation et en prévoir l'évolution. Or, la conduite est un processus complexe, impliquant l'analyse d'un nombre important de données issues de différentes sources ainsi que la synthèse de décisions soumises à des règles dans un environnement hautement dynamique. C'est pourquoi la prise en compte du comportement des conducteurs et de leurs interactions mutuelles est une condition nécessaire à toute prédiction du trafic.

Cette thèse propose un schéma complet pour la prédiction des déplacements de véhicules en interaction. Une infrastructure de communication est supposée permettre un échange d'informations en temps réel entre tous les véhicules et une unité de traitement centralisée. La prédiction de la scène routière et du risque de collision par ce calculateur peut alors être retournée aux conducteurs des véhicules pour une prise de décision locale. La criticité d'un tel système nécessite de mettre en place, autant que possible, des procédures d'explication, voire de certification, de son bon fonctionnement. Afin d'éviter une solution monolithique fermée de bout en bout, le problème est abordé dans le cadre du filtrage Bayésien et s'appuie sur deux entités pouvant être appréhendées individuellement préalablement à leur exploitation conjointe. Une modélisation probabiliste de la dynamique a priori d'une flotte de véhicules est d'abord établie, qui capture les incertitudes sous-jacentes à la conduite ainsi que les interactions. Sur la base de ce modèle et des localisations individuelles reçues des véhicules, l'estimation et la prédiction de l'état de la flotte sont ensuite effectuées. La complexité des phénomènes mis en jeu dans les déplacements de véhicules en interaction ne permettant pas la définition d'un modèle de dynamique sous forme analytique, celui-ci est obtenu par application d'algorithmes d'apprentissage automatique sur une base de trajectoires acquises en conditions réelles. Des algorithmes de filtrage particulaire spécifiques sont mis en place afin de permettre l'inférence dans des espaces d'état de hautes dimensions.

Le manuscrit détaille la combinaison de méthodes d'apprentissage supervisé par réseaux de neurones à densité de probabilité (Probability Density Networks – PDN) ou non supervisé par auto-

encodeurs variationnels conditionnels (Conditional Variational Auto-Encoders – CVAE), avec des algorithmes de filtrage particulaire multiple (Multiple Particle Filter – MPF) ou variationnel Bayésien (Variational Bayes Particle Filter – VBPF). Les stratégies obtenues sont ensuite complétées par une méthode élémentaire de prédiction et d'évaluation du risque de collision. L'ensemble est évalué et comparé à des approches de bout en bout à base de réseaux à mémoire court et long terme (Long Short Term Memory networks – LSTM) sur un ensemble significatif de scénarios extraits d'une base de données publiques de véhicules autoroutiers. La capture des tendances à moyen terme fait l'objet d'une attention particulière.

Mots-clés : systèmes de transport intelligents ; véhicules connectés ; modèles probabilistes ; filtrage Bayésien ; apprentissage automatique ; prédiction ; évaluation de risque de collision.

Remerciements

J'aimerais tout d'abord remercier mon directeur de thèse, Patrick Danès, pour m'avoir appris à persévérer et à ne rien lâcher jusqu'à la toute fin de ce périple scientifique. Son savoir, ses idées, et le temps qu'il m'a accordé pendant plus de trois ans m'ont permis d'apprendre énormément et de mener à bien mon projet de thèse.

Je remercie également tous les membres de mon jury de thèse pour leur présence et les précieux retours qu'ils ont pu me faire sur mes travaux pendant, et avant, cette soutenance. Ce fut un véritable plaisir pour moi d'échanger avec eux et de défendre mes travaux lors de cette journée. Je remercie, pour leur suivi et leurs avis toujours pertinents, Roland Chapuis, rapporteur et président de mon jury, Fawzi Nashashibi, rapporteur, ainsi que Caroline Chanel, examinatrice. Je remercie également Philippe Bonnifait, examinateur. Enfin, je remercie Simon Leglaive, examinateur, pour ses commentaires supplémentaires sur mon manuscrit.

Mes remerciements vont bien sûr aussi à ma famille. Mes parents qui m'ont toujours soutenu dans mes études et qui étaient toujours présents même loin de moi, mon super jumeau Émile, Mélanie et Alice, mes petits frères, Armand et Baptiste, jamais loin de l'ordinateur pour moi, ma Mamie Jo, Anne-Laure et Laurent, Tatie Dominique et Gégé. J'ai également une pensée pour celles et ceux qui ne sont plus là.

Je remercie mes amis, Victor D., Marine, Pierre, Wafaa, Damien, Zineb, Hadrien, Victor P., Paul, Céline, Jessica, Camille, Valentin. Les week-ends et soirées passées avec vous m'ont permis de relâcher la pression et de rendre cette épreuve plus agréable. Je remercie ma petite Misha pour ses ronronnements de support pendant la rédaction de mon manuscrit.

Je tiens à remercier particulièrement ma compagne, Ghita, qui m'a soutenu sans faillir pendant toutes ces éprouvantes années. Merci d'avoir supporté les horaires et les humeurs, merci de m'avoir fait voir le positif quand je ne pouvais plus l'apercevoir, merci d'avoir relu ma prose à un moment où je ne la supportais plus. Merci d'avoir été là pour moi.

Une pensée aussi pour les collègues rencontrés pendant ma fin de rédaction... Dinesh, Christophe, Alexandre, Paul, Maëlys, Marie, Phillipe-Antoine, Bruno, Gabriel, Mathieu, Bastien, Axel, Fadi, Alex.

Je remercie le LAAS-CNRS, l'École Doctoral EDSYS, l'Université Paul Sabatier et Continental Digital Services France pour leur confiance.

Enfin, je remercie toutes celles et ceux avec qui j'ai pu échanger au cours de cette aventure scientifique au LAAS-CNRS ou ailleurs. En particulier, Luis, Jorge, Mathis, Émile, Viviane, Frédéric, Michel, Jonathan, Anaïs, Fabien.

SOMMAIRE

Remerciements 6

Chapitre 1: Introduction 13

1.1 Contexte technologique 13

1.2 Description de la problématique 17

1.3 L'approche et les solutions proposées 22

1.4 Description des contributions 23

1.5 Plan du manuscrit 24

Chapitre 2: Pré-requis 25

2.1 Introduction 25

2.2 Modélisation probabiliste 25

2.3 Estimation stochastique 30

2.4 Apprentissage automatique 40

Chapitre 3: État de l'art 51

3.1 Prédiction du mouvement pour les véhicules intelligents 51

3.2 Évaluation du risque 61

3.3 Les jeux de données 61

Chapitre 4: Une méthode de prédiction hybride mêlant estimation stochastique et apprentissage automatique supervisé 67

4.1 Introduction 67

4.2 Schéma global 69

4.3 Modélisation 70

4.4 Apprentissage hors ligne supervisé de la loi de dynamique a priori égocentrique . . 73

4.5 Estimation stochastique en ligne du vecteur d'état de la flotte de véhicules 76

4.6 Prédiction et évaluation du risque 85

4.7 Implémentation et étude de cas 86

Chapitre 5: Apprentissage non supervisé de la loi de dynamique a priori égocentrique par une architecture neuronale interprétable 101

5.1 Introduction 101

5.2 Une dynamique a priori interprétable apprise de manière non supervisée 102

5.3 Améliorations et modifications du CVAE *Vanilla* pour limiter l'effondrement du postérieur 122

5.4 Application aux algorithmes de filtrage stochastique 153

Chapitre 6: Conclusion 159

6.1 Résumé 159

6.2 Conclusions 161

6.3 Perspectives 162

References 179

Annexe A: Estimation de l'état d'un véhicule avec un IMM-EKF d'ordre non-homogène 181

A.1 Introduction 181

A.2 Modélisation 181

A.3 Estimation 183

Annexe B: Intégration d'information sémantique dans les modèles de transition basés sur le CVAE 185

Annexe C: Modèle unicycle étendu pour la modélisation cinématique du véhicule . . 187

Annexe D: Visualisations supplémentaires des résultats en estimation et prédiction . 189

D.1 MPF w/ GMM avec 100 particules 190

D.2 VBPF w/ GMM avec 250 particules 194

D.3 MPF w/ CAB-CVAE avec 100 particules 198

D.4 MPF w/ CVRAE avec 100 particules 201

CHAPITRE 1
INTRODUCTION

Avec plus d'un milliard de voitures en circulation dans le monde aujourd'hui et une population mondiale en constante augmentation, le transport et la mobilité sont des sujets primordiaux. Les moyens de transport se doivent d'être plus propres, plus rapides et plus sûrs. En 2018, un rapport de l'Organisation Mondiale de la Santé a estimé à 1,35 million le nombre de décès dus aux accidents de la route [1]. Ce rapport et ses chiffres indiquent l'importance capitale de l'amélioration de la sécurité des transports privés et publics. Détecter et prévoir les situations dangereuses sur la route est l'un des principaux défis de l'industrie automobile et des divers organismes assurant la sécurité routière. Relever ce défi consiste à prendre des mesures préventives pour éviter des situations potentiellement mortelles. Pour ce faire, il faut être capable de modéliser l'état courant du trafic et son évolution. Les prédictions fournies par le modèle (ou les modèles) peuvent constituer la base d'un système d'évaluation des risques. Cette thèse étudie l'estimation et la prédiction du déplacement de véhicules connectés en interaction en environnement autoroutier. Une méthode pour l'évaluation du risque est également proposée.

1.1 Contexte technologique

1.1.1 La voiture connectée dans les systèmes de transport intelligents

Le développement des technologies de l'information et de la communication (Information and Communication Technologies -– ICT) dans le domaine du transport a donné naissance à une transformation de la conduite automobile et à l'émergence des Systèmes de Transport Intelligents (Intelligent Transportation Systems -– ITS) [2, 3, 4]. Via le partage d'information et la collaboration entre les différents acteurs du réseau routier, les ITS permettent le développement de nouvelles applications visant à améliorer certains aspects de la conduite. Le terme de systèmes de transport intelligents et collaboratifs (Cooperative-Intelligent Transportation Systems – C-ITS) [2] est employé pour désigner les ITS qui ont recours à un partage d'information inter-systèmes pour leur fonctionnement. Les C-ITS sont vus comme une évolution par rapport aux ITS standard.

Les applications des ITS peuvent être regroupées en trois catégories [2, 4, 5].

Amélioration de la sécurité L'analyse continue et/ou collaborative des données véhiculaires (trajectoire, itinéraire, données capteurs. . .) peut pallier les défauts du conducteur, prévenir certains accidents ou anticiper des situations dangereuses. C'est dans cette catégorie que se

situent les contributions de cette thèse.

Efficacité du transport Les temps de trajet peuvent être réduits, la congestion du trafic peut être diminuée, en proposant aux conducteurs des itinéraires adaptés. La réduction de l'impact écologique peut aussi être considérée ici.

Services aux utilisateurs En se basant sur les technologies des deux catégories précédentes, il peut être proposé aux utilisateurs des services supplémentaires basés sur les données récoltées : contenus multimédias, suggestion de visites touristiques, assistance au parking en milieu urbain, etc.

On peut également classer les applications selon certains critères de fonctionnement comme les contraintes en latence ou leur criticité [6].

C'est l'intégration des ICT dans les véhicules, par le biais de systèmes embarqués, qui a permis l'apparition de véhicules « intelligents » et autres ITS. On distingue communément plusieurs classes non exclusives de véhicules.

— Les véhicules « équipés » embarquent des systèmes destinés à assister la conduite, appelés ADAS (pour Advanced Driver Assistance Systems).

— Les véhicules « autonomes » embarquent des systèmes permettant le déplacement autonome, à des degrés divers, dans l'environnement. Il existe cinq niveaux d'autonomie qui correspondent chacun à un niveau croissant d'intégration de fonctionnalités embarquées [7].

— Les véhicules « connectés » sont des véhicules, qui en plus d'être équipés ou autonomes, échangent des informations avec les autres véhicules et/ou avec l'infrastructure routière, à l'aide de modules de communication sans fil. C'est cette dernière catégorie de véhicules qui est considérée tout au long de cette thèse.

1.1.2 Les fonctionnalités nécessaires

La mise en œuvre des applications citées ci-dessus sur un ITS nécessite que celui-ci soit doté de certaines fonctionnalités. Leur définition précise dépend de la finalité de l'application, cependant des catégories génériques peuvent être exhibées [8].

Capacités de perception [9] L'ITS, vu comme un système de systèmes, est constitué d'entités : véhicules, mobilier urbain, objets connectés, etc. Chacune d'elles constitue une sonde qui fournit des observations relatives à son environnement *e.g.*, une caméra comptant les véhicules passant par un péage. Les observations provenant des véhicules sont perçues tout au long de la conduite via des capteurs présents sur ceux-ci. Les capteurs embarqués sur ces véhicules peuvent être extéroceptifs (caméras, LIDAR...) ou proprioceptifs (vitesse, GPS, alertes de freinage d'urgence...).

Les données capteurs (positions et vitesses des véhicules, flux vidéo, nuages de points 3D, etc.) alimentent les traitements nécessaires aux applications. Dans le contexte C-ITS, ces données peuvent être mutualisées de diverses manières (agrégation, fusion précoce ou tardive, etc.), pour alimenter des traitements de plus haut niveau.

Capacités de traitement local [10, 4, 5] Certains traitements peuvent être effectués localement aux véhicules. On retrouve souvent, à cette échelle, les systèmes d'aide à la conduite (Advanced Driver-Assistance Systems – ADAS), qui intègrent plusieurs fonctions : détection d'obstacles, détection de piétons, suivi de lignes, détection des sorties de voies, gestion des distances entre les véhicules, etc. Les ADAS génèrent des alertes au conducteur en cas de danger ; dans certaines situations critiques, ils peuvent agir directement sur le véhicule (*e.g.*, freinage d'urgence). Les traitements sont effectués localement car ils sont partie intégrante d'applications de sécurité active soumises à des contraintes de temps réel fortes qui ne tolèrent pas de latence induite par un réseau.

Capacités de traitement global [8, 3, 11] Ici, on retrouve les traitements liés aux applications C-ITS qui nécessitent de collecter des données en provenance d'entités différentes pour produire des résultats tels que la réduction de la congestion du trafic, le calcul du temps de trajet ou encore la détection de situations routières dangereuses. Afin de produire ces résultats, l'ITS doit être en mesure de collecter, stocker et traiter en ligne des données provenant d'entités qui le constituent. L'ITS doit également permettre de capitaliser des données hors ligne : cartes, modèles a priori, consolidation de données antérieures, etc. Ces capacités peuvent être centralisées ou distribuées.

Capacités de communication Les véhicules connectés partagent des données avec les différentes entités d'un C-ITS à l'aide de modules sans fil. Les modes de communication entre les entités constituent un critère de classification [2, 3, 7], ainsi :

— On parle de V2V (*Vehicle-to-Vehicle*) lorsque les véhicules s'échangent mutuellement des informations.

— On parle de V2I (*Vehicle-to-Infrastructure*) lorsque les véhicules connectés communiquent avec le mobilier urbain.

— Les autres modes de communication sont résumés sous le vocable V2X (*Vehicle-to- Everything*) qui désigne le partage, entre l'ensemble des entités, des données potentiellement utiles à l'application ITS. Ce terme englobe V2V et V2I, mais aussi V2P (*Vehicle-to-Pedestrian*) pour la liaison piéton-véhicule, V2D (*Vehicle-to-Device*) pour la communication avec les objets connectés et V2G (*Vehicle-to-Grid*) pour la gestion du parc de voitures électriques. Ces différents modes de communication ne sont pas exclusifs.

Les différents modes de communication décrits précédemment peuvent être supportés par une technologie d'échange direct ou via un routeur. Voici un aperçu des différentes technologies de communication utilisées dans les ITS.

— La norme WAVE IEEE 802.11p spécifie une technologie Wifi spécialement développée pour les ITS. Il s'agit d'un mode direct permettant les communications sur de courtes distances (350 mètres maximum). Cette portée peut être étendue en constituant un réseau mobile appelé VANET (Vehicular Ad-hoc NETwork) [2, 3].

— Les technologies GSM, 3G, 4G, 5G ou encore WiMax permettent une grande couverture du territoire dans un mode routé avec des débits importants. Les normes 4G LTE et 5G proposent des latences de communication suffisamment faibles pour permettre leur utilisation dans le cadre d'applications de sécurité critiques [3].

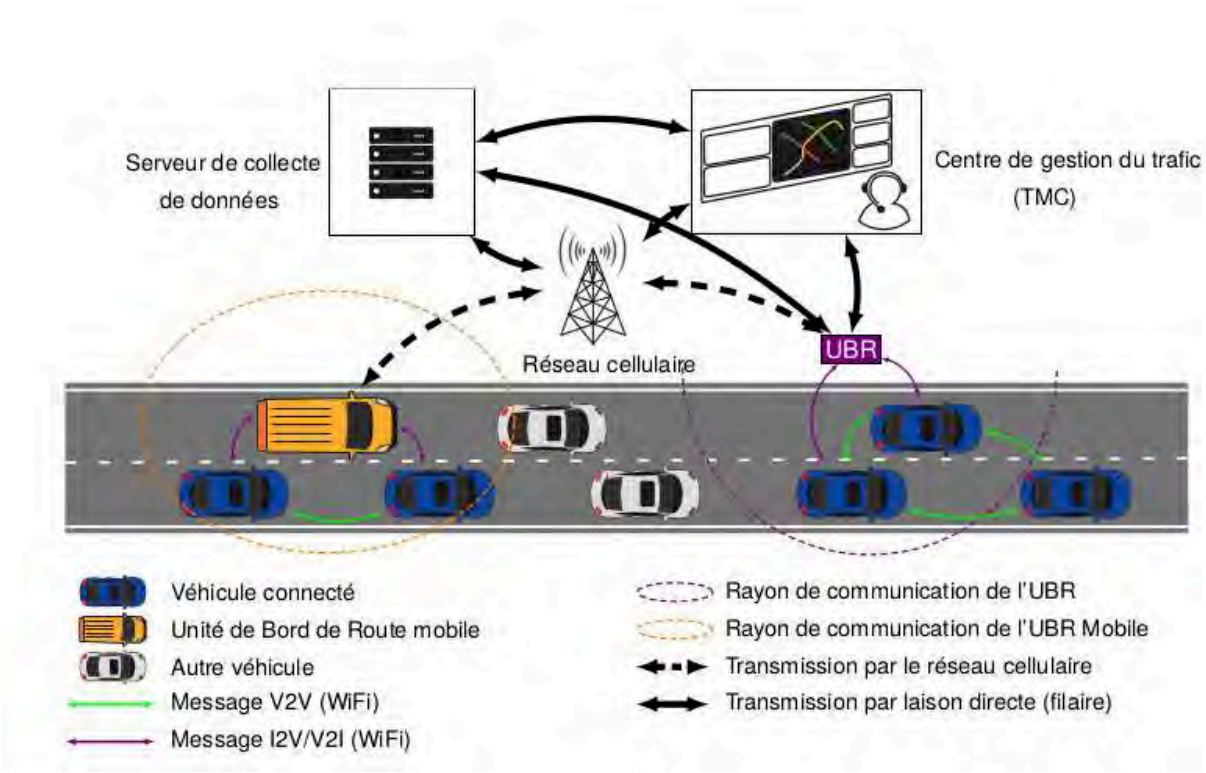


FIGURE 1.1 – Schématisation possible issue de [2] des communications entre les entités composant un système de transport intelligent collaboratif, impliquant des UBR (Unités de Bord de Route).

Capacités d'action Un résultat produit par l'ITS, quel que soit sa nature, peut provoquer une action sur un sous-ensemble de l'ITS. Que ce soit au niveau du véhicule, comme dans le cas de cette thèse, ou à un plus haut niveau (*e.g.*, station de contrôle du réseau routier), les entités de l'ITS doivent posséder des interfaces qui les ouvrent à des flux de contrôle et de données entrants. Ces interfaces peuvent être de natures très différentes : tableau de bord, assistant audio, contrôle des systèmes embarqués de la voiture, écrans de visualisation du trafic, actions sur le mobilier urbain, etc.

1.1.3 Un assistant routier dans le « cloud »

Cette thèse s'est déroulée dans le cadre du dispositif de Convention Industrielle de Formation par la REcherche (CIFRE) au sein de l'entreprise Continental. Par le biais de sa filiale Continental Digital Services France (CDSF), Continental développe des solutions « cloud » afin de proposer des services d'assistance pour le conducteur à bord de son véhicule. L'un des objectifs de cet assistant, nommé « eHorizon », est d'étendre l'horizon de perception du conducteur.

En effet, un conducteur perçoit l'environnement sur une zone réduite, typiquement une centaine de mètres dans des conditions optimales de visibilité. Le système assistant de CDSF fournira à chaque conducteur des informations pertinentes vis-à-vis de son itinéraire en amont de sa position pour pallier ses capacités de perception limitées. Ces informations sur le caractère statique et dynamique du réseau routier renforceront la sécurité du conducteur, et rendront les déplacements plus confortables et efficaces. Une telle technologie est transférable aux systèmes de conduite entièrement automatisés comme les voitures autonomes. Les sorties de ce système peuvent donner lieu à une prise de décision par un composant logiciel installé sur la voiture.

1.2 Description de la problématique

La problématique de cette thèse est l'estimation et la prédiction du mouvement de véhicules en interaction pour la détection de situations routières dangereuses. Les trois objectifs suivants sont extraits de cette problématique :

— Estimer avec suffisamment de précision l'état d'une flotte de véhicules en interaction depuis des observations bruitées.

— Sur la base de cette estimation, prédire à moyen terme, dans un horizon de 3 à 5 secondes, le mouvement de cette flotte de véhicules.

— Enfin, exploiter la prédiction pour évaluer le risque de la scène routière analysée.

Dans cette étude, l'évolution de plusieurs véhicules sur un même segment autoroutier est étudiée. La flotte de véhicules est supposée connectée à une architecture de communication commune centralisée permettant de recueillir et d'échanger les données des véhicules en temps-réel. Afin de prévoir et éviter les situations à risque impliquant ces véhicules, il faut être en mesure de prédire leurs mouvements à moyen terme sur la base de leurs trajectoires observées. Il existe une multitude de risques et de dangers pouvant être évalués lors de la conduite [8]. Dans cette thèse, seule la détection des collisions entre les véhicules est considérée. Un conducteur humain prend des décisions en fonction du comportement des agents qui l'entourent et de son environnement. Ce processus de décision complexe est également dépendant du conducteur. L'objectif est de fournir des prédictions des trajectoires de tous les véhicules en capturant les intentions et les interactions mutuelles

qui sous-tendent le raisonnement implicite mené par les conducteurs. Depuis la détection d'une situation dangereuse, certains systèmes peuvent entreprendre une action pour mitiger le risque du conducteur. Dans les ADAS, il peut être par exemple question de commander le véhicule pour effectuer une trajectoire d'évitement ou d'afficher une alerte pour le conducteur. Ces sous-systèmes liés à la prise de décision et au contrôle du véhicule ne sont pas abordés dans cette étude.

Les humains construisent leurs décisions sur la base des données qu'ils observent. Les capteurs utilisés peuvent être divers et peuvent donc avoir des caractéristiques de bruit très distinctes. Dans un souci de généricité, les approches proposées doivent être suffisamment agnostiques au type de capteur utilisé afin d'éviter tout traitement spécifique à celui-ci.

La criticité des ADAS pour la conduite impose aux composants qui permettent l'évaluation du risque pour les conducteurs de fournir des éléments pour attester de leur bon fonctionnement. Les approches modulaires composées de sous-ensembles aux rôles clairement définis sont privilégiées. Les approches monolithiques de bout en bout ne sont considérées qu'en dernier recours afin de favoriser l'interprétabilité et l'explicabilité des résultats produits.

Les algorithmes utilisés doivent mettre à disposition d'un opérateur humain des méthodes pour la compréhension des processus d'estimation, de prédiction et d'analyse des scènes routières. Les algorithmes utilisés peuvent être interprétables en exhibant des relations de cause à effet dans leur fonctionnement. Certains algorithmes sont explicables et il est possible pour un être humain d'expliquer le fonctionnement interne de l'algorithme depuis une sortie produite spécifique. La compréhension du fonctionnement des algorithmes composant notre approche est un des pré-requis pour s'assurer de son bon fonctionnement. De même, la rigueur mathématique des méthodes employées est recherchée afin d'être en mesure de vérifier l'absence de comportements erronés lors de leur utilisation. Afin de permettre le fonctionnement en situation routière réelle, les approches présentées doivent être en mesure de proposer des performances (quasi) temps réel pour une quantité de véhicules raisonnablement élevée. Les possibilités d'implémentation parallèle et le passage à l'échelle doivent donc être également considérés. La Figure 1.2 illustre et situe dans le cas d'un ADAS les sous-systèmes nécessaires pour répondre à la problématique.

1.2.1 Exemple de scénario étudié

Pour valider les objectifs présentés précédemment, une série de scénarios autoroutiers complexes sont sélectionnés. Les différents résultats en estimation, prédiction et évaluation de risque sont comparés avec des méthodes de l'état de l'art. La Figure 1.3 présente un exemple impliquant trois véhicules roulant sur un même segment autoroutier. Dans les premiers instants (a-b), il est important d'estimer correctement l'état de la flotte. En effet, de la qualité de cette estimation découlent les performances de la prédiction et de l'évaluation de risque. Dans ce scénario, le véhicule

jaune semble entamer une manœuvre de dépassement. Une estimation d'état mauvaise ou absente sur la base de mesures bruitées pourrait, par exemple, amener le système à localiser le véhicule au milieu de sa voie alors que celui-ci se prépare à dépasser. Lors de la phase de prédiction, les véhicules voisins et l'environnement routier doivent être pris en compte pour prédire un futur plausible à moyen terme de la scène routière. Les approches qui ne prennent pas en compte le voisinage et les interactions entre les véhicules ne permettent pas la représentation d'une manœuvre complexe de dépassement comme celle entreprise par ces trois véhicules et prédirait un futur difficilement exploitable par les couches systèmes de décision et de contrôle du véhicule. Dans cet exemple, la prédiction de l'évolution de la scène routière (c) indique que le véhicule jaune pourrait couper la route du véhicule rouge ou se retrouver à distance dangereusement proche de ce dernier s'il continuait sa manœuvre de dépassement. L'état prédit pour le véhicule jaune dans le cas où il ne dépasserait pas n'est pas source de dangerosité : le véhicule concerné ralentit et laisse passer le véhicule rouge en suivant le véhicule orange. Ce scénario d'exemple illustre succinctement quelques motivations de l'approche proposée dans cette thèse: la combinaison d'une méthode d'estimation et de prédiction multimodales pour la compréhension de la scène routière et l'évaluation du risque.

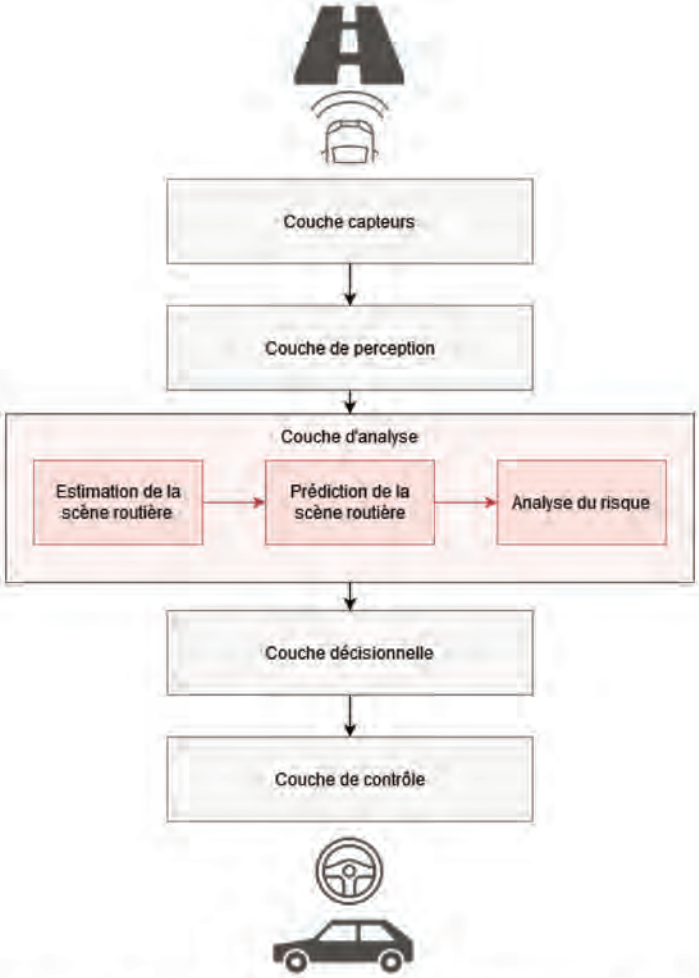
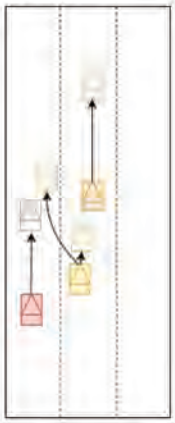


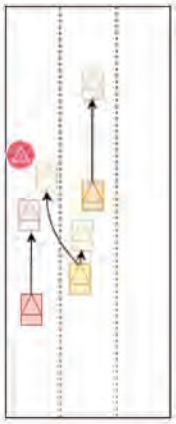
FIGURE 1.2 – Schématisation simplifiée d'un système ADAS où les blocs rouges représentent les sous-systèmes étudiés dans cette thèse. Après le traitement des données capteurs brutes par la couche de perception se trouve la couche d'analyse. Ce sous-système a pour but l'analyse de la scène routière dans son ensemble pour l'application considérée. Dans notre cas, ce sous-système a pour objectif de comprendre et prédire la scène routière afin d'en évaluer le risque pour le conducteur. Cette évaluation peut ensuite être utilisée par le sous-système de décision pour choisir une action à réaliser pour mitiger le risque détecté.

(a) L'état initial des véhicules nécessite d'être estimé à partir d'observations bruitées. Les rectangles et les flèches colorés illustrent l'état réel des véhicules en position et en vitesse respectivement. L'étoile correspond à la mesure de position qui leur est associée.

(b) Au fur et à mesure que les véhicules évoluent sur le segment routier, leur état est estimé via les mesures.



(c) A tout moment, il peut être nécessaire de connaître le futur de la scène routière étudiée. À ce moment là, l'estimation peut s'arrêter et la prédiction de l'état des véhicules, en transparent sur le schéma, peut être réalisée. La prédiction multimodale pour le véhicule jaune illustre la possibilité d'avoir des futurs multiples pour chaque véhicule.



(d) Le risque associé à la scène routière peut ensuite être établi depuis les états prédits. Ici le véhicule jaune pourrait entrer en collision avec le véhicule rouge.

FIGURE 1.3 – Exemple de scénario autoroutier motivant les objectifs présentés. L'estimation (a,b) des véhicules précède la prédiction (c) qui permet de réaliser l'évaluation du risque (d) associé à cette scène.

1.3 L'approche et les solutions proposées

De nombreuses publications attestent de l'efficacité des solutions de bout en bout qui font correspondre des trajectoires d'entrée à des trajectoires futures en sortie par le biais d'architectures neuronales. Ces deux trajectoires s'étendent sur plusieurs instants. De telles méthodes sont indéniablement puissantes, mais leurs résultats peuvent être difficiles à qualifier et leur interprétabilité et explicabilité peuvent être remises en question [12].

L'approche proposée repose sur trois piliers fondamentaux. Le premier consiste à se placer dans le cadre des systèmes Markoviens pour modéliser l'état complet (caché) de la flotte de véhicules. Le deuxième propose l'exploitation de techniques éprouvées de filtrage stochastique Bayésien pour estimer et prédire l'état de la flotte sur la base d'observations individuelles des véhicules collectées par une interface de communication commune centralisée. Ce cadre de travail répond aux objectifs évoqués précédemment au moyen d'outils basés sur des principes mathématiques solides : gestion des différentes sources d'incertitudes ; certificats de consistance ; évaluation de l'effet du bruit de mesure, etc. Cependant, l'estimation stochastique s'appuie généralement sur des modèles de transition (lois de dynamique a priori) exprimés analytiquement. Il existe dans la littérature quelques modèles de dynamique sous forme analytique permettant de décrire le déplacement des véhicules. Cependant, leur complexité intrinsèque limitée ne leur permet pas de capturer ni les interactions entre les conducteurs ni les interactions avec l'environnement routier. C'est pourquoi le troisième et dernier pilier consiste en l'apprentissage de la dynamique des véhicules en interaction par le biais de méthodes d'apprentissage automatique. L'augmentation de la puissance de calcul des ordinateurs ainsi que la disponibilité croissante de grands jeux de données véhiculaires permettent de mettre en place des techniques d'apprentissage profond et de profiter de la puissance de représentation de celles-ci pour l'obtention de tels modèles probabilistes.

1.4 Description des contributions

Les contributions principales de cette thèse sont les suivantes.

— **Une approche unifiée d'estimation et de prédiction du déplacement des véhicules en interaction testée sur de multiples scénarios autoroutiers**. L'algorithme "hybride" proposé dans cette thèse combine dans une partie hors-ligne l'apprentissage de la dynamique a priori des véhicules en interaction par apprentissage profond. Avec l'estimation et la prédiction récursive en ligne de l'état, l'approche est validée sur une sélection de scénarios autoroutiers complexes mettant en jeu de multiples interactions. Différents niveaux de bruit sont simulés afin d'évaluer la robustesse des algorithmes aux incertitudes.

— **Une représentation d'état stochastique de la flotte de véhicules en interaction** caractérisée par un vecteur d'état Markovien dont les composantes ont un sens intuitif, et son modèle d'observation linéaire avec bruits de mesure additifs.

— **Deux méthodes d'apprentissage automatique pour la modélisation de la dynamique a priori**. Le modèle de transition des états cachés de la flotte est appris sur la base d'un jeu de données autoroutières. Deux approches sont étudiées: une approche « supervisée » reposant sur l'utilisation de multiples réseaux neuronaux de densité de probabilité et une segmentation en manœuvres du jeu de données ; une approche « non-supervisée » à base d'un autocodeur variationnel conditionnel (Conditional Variational AutoEncoder – CVAE) cherchant à apprendre depuis le jeu de données des codes latents sur le processus de génération des trajectoires observées.

— **Deux schémas de filtrage stochastique « approchés » pour l'estimation de l'état de la flotte de véhicules en interaction**. La dynamique a priori préalablement apprise est utilisée au sein d'algorithmes de filtrage récursif approchés pour l'estimation de l'état de la flotte de véhicules sur la base de séquence d'observations. Le filtre particulaire multiple (Multiple Particle Filter – MPF) ainsi que le filtre particulaire multiple variationnel Bayésien (Variational Bayes Multiple Particle Filter – VBMPF) sont envisagés.

— **Une méthode élémentaire d'évaluation du risque pour les scènes autoroutières**. Une méthode simple d'évaluation du risque d'une scène routière depuis l'état prédit de la flotte est proposée.

1.5 Plan du manuscrit

Les plan du manuscrit se compose comme suit :

Chapitre 2: Pré-requis. Ce chapitre aborde les notions essentielles théoriques sur la théorie des probabilités, le filtrage stochastique et l'apprentissage automatique.

Chapitre 3: État de l'art. Ce chapitre met en perspective la littérature existante sur la prédiction du mouvement et l'évaluation de risque dans le cadre des systèmes de transport intelligents. Il présente également les principales bases de données publiques de la littérature et jeux de données utilisés.

Chapitre 4: Une méthode de prédiction hybride mêlant estimation stochastique et apprentissage automatique supervisé. La dynamique a priori d'une flotte de véhicules en interaction apprise depuis un jeu de données segmenté peut être utilisée au sein d'un algorithme de filtrage pour estimer et par la suite prédire le déplacement de la flotte. Une première méthode d'apprentissage automatique « supervisée » ainsi que deux algorithmes de filtrage stochastique approximatifs sont présentés ici. L'approche globale est évaluée sur une multitude de scénarios autoroutiers.

Chapitre 5: Apprentissage non supervisé de la dynamique égocentrique avec un modèle neuronal interprétable. Ce chapitre s'appuie sur le précédent pour se concentrer sur une amélioration majeure de l'apprentissage de la dynamique a priori. Ce nouvel apprentissage s'affranchit du besoin de segmentation des données en exploitant des techniques d'apprentissage non supervisé pour extraire automatiquement de l'information à partie du jeu de trajectoires. Plusieurs algorithmes sont étudiés et comparés sur des données synthétiques et réelles.

Chapitre 6: Conclusion. Ce dernier chapitre résume les contributions et présente un ensemble de perspectives.

CHAPITRE 2
PRÉ-REQUIS

2.1 Introduction

Ce chapitre présente les concepts et outils théoriques utilisés tout au long de cette étude pour l'application ITS envisagée. Le premier défi majeur quant à la réalisation de cette application provient de l'incertitude omniprésente dans l'estimation et la prédiction de l'état des véhicules roulant sur un segment routier. En effet, la conduite en interaction résulte d'un processus de raisonnement et de décision humain complexe difficile à modéliser directement via un modèle analytique et/ou de manière certaine. De plus, les capteurs utilisés dans les ITS sont variés et produisent des observations entachées de bruit. Enfin, l'évaluation de risque découlant de l'estimation et de la prédiction de l'état réel de la flotte de véhicules, le risque associé à une scène routière est donc également incertain.

Une première partie présente succinctement les bases du cadre probabiliste utiles pour représenter et raisonner sur les multiples sources d'incertitude du problème. Dans un second temps, les notions de filtrage stochastique nécessaires à l'estimation récursive de l'état des véhicules depuis des observations bruitées sont abordées. Enfin la dernière section de ce chapitre traite des notions d'apprentissage automatique nécessaires pour cette étude.

Les pré-requis théoriques présentés ici sont en grande partie extraits de [13, 14, 15, 16, 17]. Le lecteur est invité à consulter ces références pour approfondir les sujets traités dans ce chapitre.

2.2 Modélisation probabiliste

La théorie des probabilités nous permet de raisonner sur des événements aléatoires. Un événement noté ω est lié à une expérience aléatoire et représente un sous-ensemble des résultats possibles pour cette expérience. Dans la théorie des probabilités, une expérience aléatoire définit une expérience dont le résultat ne peut être prévu à l'avance et dont celui-ci peut changer même avec des conditions identiques. Selon l'interprétation des probabilités choisie, fréquentiste ou Bayésienne, la théorie des probabilités permet de modéliser et analyser la fréquence d'occurrence des événements ou alors de quantifier l'incertitude sur ceux-ci. Dans cette thèse, c'est cette dernière interprétation qui est adoptée.

2.2.1 Notations des variables et processus aléatoires

Une variable aléatoire (VA) univariée est notée x. Une valeur pouvant être prise par cette VA est notée x. Pour un événement ω, $\mathsf{x}(\omega) = x$ où x est appelée la « réalisation » de x. Dans le cas multidimensionnel, cette notation devient $\mathbf{x}$ et $\boldsymbol{x}$ pour la variable aléatoire vectorielle et pour la valeur prise par cette dernière respectivement. Usuellement, en considérant une notation simplifiée, $P(\mathsf{x})$ désigne la distribution sur la variable aléatoire x. Lorsque l'on note $P(x)$, c'est la distribution de probabilité évaluée pour la valeur x qui est désignée [1]. Pour lever l'ambiguïté sur la variable aléatoire discrète à utiliser, on peut noter explicitement $P(\mathsf{x} = x)$ ou même $P_{\mathsf{x}}(x)$. Il est possible de définir en premier une variable aléatoire et d'indiquer sa distribution associée avec la notation $\mathsf{x} \sim P(\mathsf{x})$.

Les même remarques de notation peuvent être faites pour le cas continu. Pour une variable aléatoire continue x, sa densité de probabilité s'écrit p. Une variable vectorielle continue est notée $\mathbf{x}$.

Un processus aléatoire ou processus stochastique peut être désigné par une suite de variables aléatoires telles que $\mathbf{x}_{0:n} = \mathbf{x}_0, \ldots, \mathbf{x}_n$ où chaque variable aléatoire est indexée par le temps $n \in \mathbb{N}$. La séquence $\boldsymbol{x}_{0:n}$ obtenue pour une expérience donnée constitue une réalisation du processus $\mathbf{x}_{0:n}$.

2.2.2 Distributions de probabilité utiles

Dans cette partie, plusieurs distributions de probabilité sont présentées. Ces distributions communes, sont toutes paramétriques : un ensemble fini de valeurs permet de décrire entièrement leur forme. Lorsque que l'on écrit une distribution paramétrique, « ; » veut dire « est paramétrée par ».

2.2.2.1 Distribution de Bernoulli et distribution catégorielle

La distribution de Bernoulli est une distribution de probabilité sur une variable aléatoire binaire $\mathsf{x} \in \{0, 1\}$. Elle possède un unique paramètre $\lambda \in [0, 1]$ et est définie par

$$P(x; \lambda) = \lambda^x (1 - \lambda)^{1-x} \tag{2.1}$$

La distribution catégorielle est une loi discrète qui généralise la distribution de Bernoulli dans le cas où une variable aléatoire peut prendre plus que deux valeurs différentes. Cette distribution de probabilité est paramétrée par le vecteur $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)^T$ contenant les probabilités de chaque valeur possible, avec $\lambda_k \in [0, 1]$ et $\sum_{k=1}^{K} \lambda_k = 1$. Si $\boldsymbol{x}^{\text{encodage}}$ est un vecteur d'encodage 1 parmi n,

1. En toute rigueur, des notations plus rigoureuses devraient être utilisées : $P_{\mathsf{x}}(x)$ désignerait la densité de probabilité de la variable aléatoire x, fonction de la variable muette x réalisation de x ; l'espérance mathématique devrait s'écrire $\mathbb{E}_{\mathsf{x}}[g(\mathsf{x})]$. Dans ce manuscrit, la différence entre variables aléatoires et réalisations n'est pas toujours explicitée.

avec ici $n = K$, permettant de représenter la valeur d'une variable discrète x alors sa distribution peut être définie comme suit :

$$P(x; \boldsymbol{\lambda}) = \prod_{k=1}^{K} \lambda_k^{x^{\text{encodage},k}} \tag{2.2}$$

2.2.2.2 *Distribution normale*

La distribution normale ou Gaussienne univariée est une distribution continue. Elle est paramétrée par sa moyenne μ qui est également son mode et sa variance σ^2 qui contrôle l'étalement de la distribution autour de sa moyenne. Elle est définie comme suit :

$$\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{2.3}$$

La distribution normale ou Gaussienne multivariée est une généralisation du cas univarié à une variable aléatoire vectorielle continue. Pour un vecteur aléatoire $\mathbf{x}$ vivant dans $\mathbb{R}^k$, cette distribution est paramétrée par un vecteur de moyennes $\boldsymbol{\mu}$ de taille k qui est aussi le mode ainsi que par une matrice $k \times k$ de covariance $\boldsymbol{\Sigma}$ définie positive. Sa densité de probabilité est donnée par :

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^k \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{2.4}$$

2.2.2.3 *Mélange de Gaussiennes*

La densité de probabilité d'un mélange de Gaussiennes (Gaussian Mixture Model – GMM) est définie comme une combinaison linéaire convexe de K lois Gaussiennes. Cette distribution est paramétrée par une distribution catégorielle $P(C)$ qui représente la probabilité de chacune de ses K hypothèses. Chacune des K Gaussiennes est paramétrée par un vecteur moyenne $\boldsymbol{\mu_k}$ et une matrice de covariance $\boldsymbol{\Sigma_k}$ de sorte que que la distribution de probabilité du mélange de Gaussiennes s'écrit

$$P(\mathbf{x} = \boldsymbol{x}) = \sum_{k=1}^{K} P(C = k) \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.5}$$

2.2.3 Divergence de Kullback-Leibler

La divergence de Kullback-Leibler (Kullback-Leibler Divergence – KLD) ou divergence KL mesure la dissimilarité entre deux distributions de probabilité $P(\mathsf{x})$ et $Q(\mathsf{x})$ sur la même variable

aléatoire x. La divergence de Kullback-Leibler de Q vers P est définie comme suit :

$$D_{KL}(P||Q) = \mathbb{E}_{\mathsf{x}\sim P(\mathsf{x})}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{\mathsf{x}\sim P(\mathsf{x})}[\log P(x) - \log Q(x)]. \tag{2.6}$$

Dans le cas continu, elle s'écrit

$$D_{KL}(p||q) = \mathbb{E}_{\mathsf{x}\sim p(\mathsf{x})}\left[\log \frac{p(x)}{q(x)}\right] = \mathbb{E}_{\mathsf{x}\sim p(\mathsf{x})}[\log p(x) - \log p(x)]. \tag{2.7}$$

La divergence KL est non négative et est égale à 0 si et seulement si $P(\mathsf{x})$ et $Q(\mathsf{x})$ sont identiques. Comme l'illustre la Figure 2.1, la divergence de KL n'est pas symétrique, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

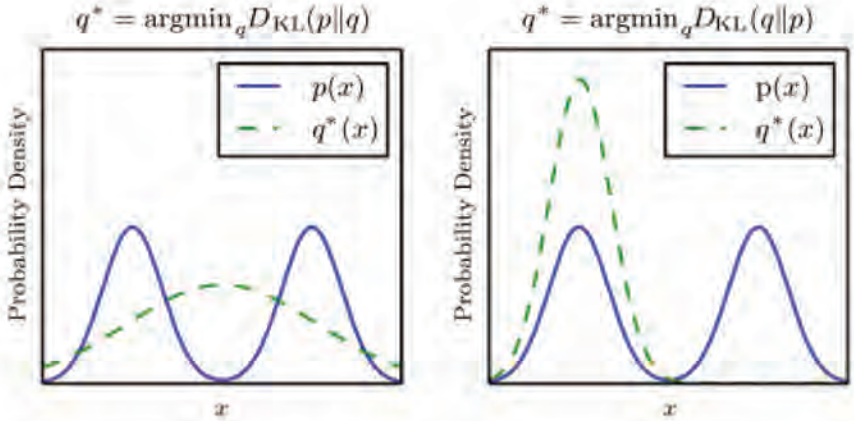


FIGURE 2.1 – Figure extrait de [13]. Pour illustrer l'asymétrie de la divergence KL, [14] propose d'approcher un mélange de Gaussiennes $p(x)$ par une Gaussienne $q^*(x)$ minimisant la divergence de KL $D_{KL}(p||q)$ ou $D_{KL}(q||p)$. À gauche, minimiser $D_{KL}(p||q)$, appelée aussi « la divergence KL directe » revient à choisir une distribution de probabilité q qui est élevée là où p l'est également. Sur le graphe, on remarque que q ne peut pas être nul là ou p est important sous peine que la divergence KL soit élevée. q recouvre tous les modes de telle manière que ceux ci n'appartiennent pas aux queues de q. À l'inverse, à droite, lorsque $D_{KL}(q||p)$ est minimisée, q est sélectionné de manière à être faible dans les zones où p l'est aussi *e.g.* p ne peut pas être faible alors que q est élevée.

2.2.4 Modèles probabilistes et propriété de Markov

Lors de la modélisation probabiliste d'un problème réel, il n'est pas rare d'avoir recours à un nombre important de variables aléatoires. Décrire la probabilité jointe de l'ensemble de ces variables est une tâche fastidieuse. Il est alors avantageux d'exprimer cette loi jointe en un produit de lois plus facilement paramétrables une à une. Les relations de dépendance et d'indépendance sont exprimées via des probabilités conditionnelles. Un exemple d'une telle factorisation pour trois

variables aléatoires a,b et c pourrait être :

$$p(\mathsf{a}, \mathsf{b}, \mathsf{c}) = p(\mathsf{a})p(\mathsf{b}|\mathsf{a})p(\mathsf{c}|\mathsf{b}). \tag{2.8}$$

Dans cet exemple, a influence la valeur pouvant être prise par b et, conditionnellement à b, c est indépendant de a. Il est possible de visualiser cette distribution de probabilité sous forme d'un graphe orienté acyclique, ou réseau Bayésien, tel que celui présenté sur la Figure 2.2.

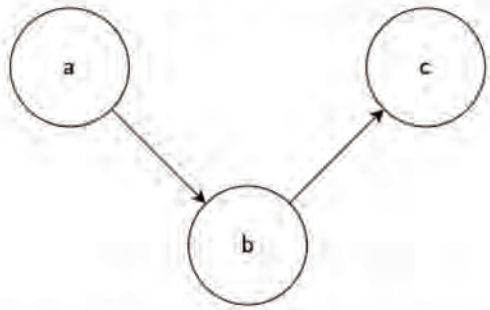


FIGURE 2.2 – Représentation en graphe orienté acyclique composé de trois variables aléatoires de probabilité jointe $p(\mathsf{a}, \mathsf{b}, \mathsf{c}) = p(\mathsf{a})p(\mathsf{b}|\mathsf{a})p(\mathsf{c}|\mathsf{b})$. Ce graphe correspond à la factorisation définie dans 2.8.

Un réseau bayésien est un modèle graphique probabiliste permettant de décrire la loi jointe d'un ensemble de variables aléatoires. Le graphe d'un réseau Bayésien est composé de noeuds représentant les variables aléatoires en jeu et d'arêtes exprimant les dépendances conditionnelles entre ces variables. Pour un vecteur de K variables aléatoires $\mathbf{x} = (\mathsf{x}^1, \ldots, \mathsf{x}^K)^T$, cette factorisation satisfait

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(\mathsf{x}^k|\mathsf{pa}(\mathsf{x}^k)) \tag{2.9}$$

où $\mathsf{pa}(\mathsf{x}^k)$ désigne l'ensemble des parents du noeud x^k.

L'étude de processus aléatoires est un thème central de cette thèse. La notion de réseau Bayésien peut être étendue pour modéliser des séquences de variables aléatoires indexées dans le temps. Dans ce cas, cette nouvelle représentation est appelée **réseau Bayésien dynamique**. La Figure 2.3 illustre un exemple de réseau Bayésien dynamique pour un ensemble réduit de variables aléatoires. Pour une séquence de K variables aléatoires indexées dans le temps par n, la factorisation de $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, où $\forall l,\ \mathbf{x}_l = (\mathbf{x}_l^1, \ldots, \mathbf{x}_l^K)^T$, s'écrit :

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \prod_{k=1}^{K} p(\mathsf{x}_n^k|\mathsf{pa}(\mathsf{x}_n^k)), \tag{2.10}$$

où $\mathrm{pa}(\mathsf{x}_n^k)$ est l'ensemble des parents du noeud x_n^k à l'instant n. Dans le cas spécifique d'un réseau dynamique Bayésien respectant la propriété de Markov, les parents de ce noeud peuvent être soit un parent direct dans le réseau à l'instant n, soit un parent direct à l'instant $n-1$. Cette factorisation est également appelée la loi de transition du réseau.

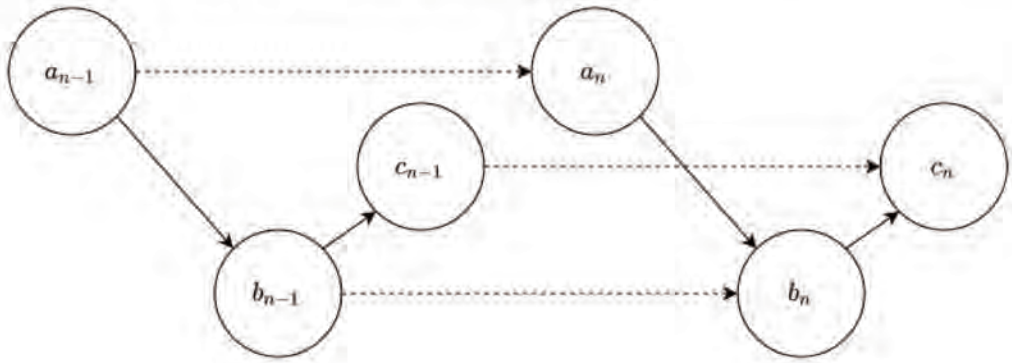


FIGURE 2.3 – Exemple de réseau Bayésien Dynamique respectant la propriété de Markov pour une factorisation de trois variables aléatoires a, b et c indexées dans le temps. Les flèches pointillées représentent les relations temporelles. Via la structure du graphe, la loi de transition peut s'écrire $p(\mathsf{a}_n, \mathsf{b}_n, \mathsf{c}_n | \mathsf{a}_{n-1}, \mathsf{b}_{n-1}, \mathsf{c}_{n-1}) = p(\mathsf{c}_n | \mathsf{c}_{n-1}, \mathsf{b}_n) p(\mathsf{b}_n | \mathsf{a}_n, \mathsf{b}_{n-1}) p(\mathsf{a}_n | \mathsf{a}_{n-1})$.

La loi jointe sur N instants peut s'écrire comme suit :

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_N) = \prod_{k=1}^{K} p(\mathsf{x}_0^k | \mathrm{pa}(\mathsf{x}_0^k)) \times \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathsf{x}_n^k | \mathrm{pa}(\mathsf{x}_n^k)). \tag{2.11}$$

La propriété de Markov : un processus aléatoire est dit Markovien s'il possède la propriété de Markov. Un tel processus vérifie :

$$\forall n_j < n_{j+1} < \ldots < n_{i-1} < n_i, \qquad p(\mathbf{x}_{n_i} | \mathbf{x}_{n_{i-1}}, \ldots, \mathbf{x}_{n_{j+1}}, \mathbf{x}_{n_j}) = p(\mathbf{x}_{n_i} | \mathbf{x}_{n_{i-1}}) \tag{2.12}$$

2.3 Estimation stochastique

Un des piliers centraux de cette thèse repose sur l'estimation de l'état de la flotte des véhicules en interaction. Dans cette section, les notions fondamentales d'estimation stochastique sont présentées. La problématique de l'estimation stochastique est introduite dans le cas d'un modèle stochastique statique puis les notions présentées sont ensuite étendues pour le cas dynamique d'estimation d'un processus aléatoire.

2.3.1 Problématique de l'estimation stochastique

L'estimation stochastique repose sur l'utilisation d'un modèle stochastique pour représenter mathématiquement un phénomène ou un système réel. La forme de ce dernier est la suivante :

$$\mathbf{z} = \mathcal{M}(\boldsymbol{x}, \mathbf{v}) \tag{2.13}$$

où $\boldsymbol{x} \in \mathbb{R}^M$ désigne le vecteur des paramètres « caché » à estimer. $\mathbf{z} \in \mathbb{R}^N$ désigne le vecteur aléatoire d'observation (ou de mesure). Pour l'expérience en cours, celui-ci se réalise en le vecteur $\boldsymbol{z}$. Enfin, $\mathbf{v}$ modélise les bruit et le perturbations. Le modèle intègre une description statistique de ce vecteur.

À travers cette modélisation, la problématique de l'estimation stochastique s'énonce alors comme suit : *Depuis la réalisation $\boldsymbol{z}$ de $\mathbf{z}$, quelle information peut-on obtenir sur $\boldsymbol{x}$?*

Il convient de distinguer deux cas concernant $\boldsymbol{x}$. Si aucune information a priori n'est considérée pour $\boldsymbol{x}$, alors ce dernier est considéré comme déterministe inconnu. Si on dispose d'une information a priori sur les valeurs que peut prendre $\boldsymbol{x}$, alors celle-ci est exprimée à travers la distribution de probabilité de la variable aléatoire vectorielle $\mathbf{x}$ dont $\boldsymbol{x}$ est supposé être une réalisation inconnue.

Dans cette thèse, seule la dernière catégorie est jugée d'intérêt, celle des **estimateurs Bayésiens**. Pour des V.A. continues, l'estimation Bayésienne repose sur le modèle

$$p(\mathbf{x}, \mathbf{z}), \tag{2.14}$$

dans lequel

$$p(\mathbf{x}) \tag{2.15}$$

est la **loi a priori** et

$$p(\mathbf{z}|\mathbf{x}) \tag{2.16}$$

le **modèle d'observation**. Avec bien sûr, $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{z}|\mathbf{x})$.

En utilisant le théorème de Bayes, il est possible d'écrire

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}, \tag{2.17}$$

où $p(\mathbf{x}|\mathbf{z})$ est la **loi a posteriori** (ou loi postérieure) qui capture toute l'information sur $\boldsymbol{x}$ contenue dans l'observation $\boldsymbol{z}$. $p(\mathbf{z})$ est une constante de normalisation.

Pour une réalisation $\boldsymbol{x}$, la **fonction de vraisemblance** est donnée par

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x}). \tag{2.18}$$

La fonction de vraisemblance permet de caractériser la capacité d'une valeur candidate du vecteur de paramètres a expliquer les observations.

2.3.2 Estimation stochastique récursive

Dans cette thèse, nous sommes intéressés par l'estimation de l'état de véhicules en interaction sur un segment autoroutier. Pour cela, il faut être en mesure d'intégrer séquentiellement de l'information sur l'état des véhicules depuis des capteurs délivrant des mesures entachées de bruit. Pour notre problématique ITS, il s'agit d'estimer récursivement, pour chaque instant n, l'état des véhicules $\mathbf{x}_n$ *e.g.* leur position et vitesse. Cette estimation est effectuée sur la base d'une série de mesures bruitées $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ liées à $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Plus formellement, ce schéma consiste en l'estimation du processus d'état caché $\mathbf{x}_{0:n}$ à partir du processus de mesure $\mathbf{z}_{1:n}$ et d'une distribution initiale notée $p(\mathbf{x}_0)$. Cette nouvelle problématique se place dans le cadre d'un schéma récursif d'estimation d'un processus aléatoire.

La Figure 2.4 propose une représentation, via un réseau Bayésien dynamique, des dépendances statistiques sur les processus aléatoires $\mathbf{x}_{0:n}$ et $\mathbf{z}_{1:n}$ considérés conjointement pour notre problématique d'estimation récursive. La **loi jointe a posteriori** $p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n} = \boldsymbol{z}_{1:n})$ caractérise toute l'information sur le processus d'état caché $\mathbf{x}_{0:n}$ pouvant être extraite de la trajectoire de mesure $\boldsymbol{z}_{1:n}$.

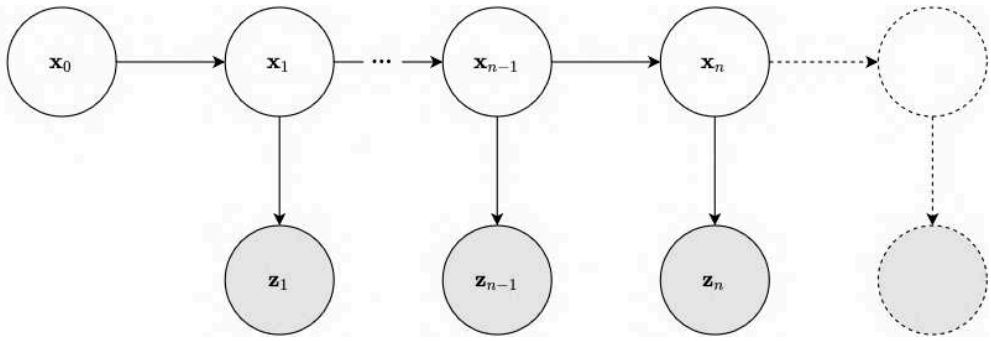


FIGURE 2.4 – Réseau Bayésien décrivant les processus aléatoires et leur dépendances mises en jeu dans le problème d'estimation récursive considéré. Le processus d'état caché $\mathbf{x}_{0:n}$ est représenté par la suite de noeuds blancs sur le graphe. Les variables du processus de mesure $\mathbf{z}_{1:n}$ sont en gris.

Dans une telle représentation plusieurs hypothèses sont effectuées et certains modèles sont supposés connus a priori :

— Le processus aléatoire caché $\mathbf{x}_{0:n}$ est supposé Markovien dont la loi de dynamique a priori $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ est connue. Cette dernière est donc également Markovienne et modélise la transition de l'état entre deux instants $n-1$ et n. Elle peut être équivalente à une équation d'état à temps discret avec bruits additifs de statistiques données.

— Le modèle d'observation $p(\mathbf{z}_n|\mathbf{x}_n)$ qui relie la mesure courante et l'état courant.

— La distribution initiale $p(\mathbf{x}_0)$ représentant la connaissance de l'état à l'instant $n = 0$.

En marginalisant la loi jointe a posteriori, on obtient :

— La loi de filtrage $p(\mathbf{x}_n|\mathbf{z}_{1:n} = \boldsymbol{z}_{1:n})$ qui représente la distribution de probabilité de l'état à l'instant courant n après l'assimilation des mesures jusqu'à ce même instant.

— La loi de prédiction $p(\mathbf{x}_n|\mathbf{z}_{1:n'} = \boldsymbol{z}_{1:n'})$ avec $n' < n$ qui représente la distribution de probabilité de l'état à l'instant futur n après l'assimilation de mesures jusqu'à un instant passé n'.

— La loi de lissage $p(\mathbf{x}_n|\mathbf{z}_{1:n'} = \boldsymbol{z}_{1:n'})$ avec $n' > n$.

Ces deux premières lois sont utilisées dans les chapitres suivants.

2.3.3 Solution récursive exacte

La solution récursive exacte à ce problème d'estimation Bayésienne appelée aussi filtrage de Bayes est composée de deux étapes qui peuvent être explicitées à travers la loi jointe postérieure. D'après la formule de Bayes, dans le cas général, on peut écrire la loi jointe postérieure comme suit :

$$p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}) = \frac{p(\mathbf{z}_n|\mathbf{x}_{0:n}, \mathbf{z}_{1:n-1})p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})} \tag{2.19}$$

$$\text{avec } p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{1:n-1})p(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1}). \tag{2.20}$$

Cette formulation peut être simplifiée en utilisant la modélisation présentée dans la Figure 2.4 qui prend en compte l'hypothèse Markovienne du processus aléatoire d'état : $p(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$. Et similairement, elle exhibe le fait que la mesure à l'instant n est conditionnellement indépendante aux autres états conditionnellement à l'étant courant : $p(\mathbf{z}_n|\mathbf{x}_{0:n}, \mathbf{z}_{1:n-1}) = p(\mathbf{z}_n|\mathbf{x}_n)$. On a alors

$$p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}) = \frac{p(\mathbf{z}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})} \tag{2.21}$$

$$\text{avec } p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1}). \tag{2.22}$$

En marginalisant 2.22 et 2.20 selon $\mathbf{x}_{0:n-1}$ on obtient respectivement :

— L'équation de Chapman-Kolgomorov, qui donne la loi de prédiction $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ à l'instant

n à partir de la loi de filtrage $p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ à l'instant précédent $n-1$,

$$p(\mathbf{x}_n|\mathbf{z}_{1:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1} \tag{2.23}$$

— La règle de Bayes qui permet d'obtenir la loi de filtrage $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ à l'instant n à partir de la loi de prédiction $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ par assimilation de la mesure $\mathbf{z}_n$:

$$p(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{p(\mathbf{z}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{p(\mathbf{z}_n|\mathbf{z}_{1:n-1})} \tag{2.24}$$

On peut alors réécrire la loi jointe a posteriori sous la forme

$$p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}) \propto p(\mathbf{z}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1}) \tag{2.25}$$

en omettant la constante de normalisation $p(\mathbf{z}_n|\mathbf{z}_{1:n-1})$. Comme pour le cas statique, la connaissance de la loi de filtrage et de prédiction, permet d'obtenir des estimés $\hat{\boldsymbol{x}}_n$.

2.3.4 Filtre de Kalman

Les équations (2.23) et (2.24) donnent la solution optimale de l'estimation Bayésienne récursive. Elles sont importantes sur le plan théorique mais elles peuvent difficilement être appliquées directement à une problématique de filtrage réelle. En effet, les intégrales de ces équations sont généralement « intractables » *i.e.*, il n'existe pas de moyen d'obtenir dans un temps raisonnable une solution à ce calcul. Cependant, avec des simplifications sur la modélisation du problème d'estimation, il est possible de retrouver des solution exploitables pour un problème réel. Dans le cas où l'état est continu ; le modèle de dynamique du processus d'état caché $\mathbf{x}_{0:n}$ est décrit par une équation d'état stochastique linéaire à temps discret, avec bruits blancs, centrés, Gaussiens ; le lien entre $\mathbf{x}_n$ et $\mathbf{z}_n$ est décrit par un modèle d'observation stochastique linéaire avec bruits de mesure blancs mutuellement indépendants, centrés, Gaussiens alors le filtre de Kalman [18] peut être utilisé pour obtenir l'expression exacte des moments de la loi de filtrage Gaussienne sous forme récursive. Dans sa forme canonique, les bruits sont supposés mutuellement indépendants et indépendants de l'état initial. Ces deux modèles s'écrivent :

$$\mathbf{x}_n = \boldsymbol{F}_{n-1}\mathbf{x}_{n-1} + \mathbf{W}_{n-1} \qquad \text{avec } \mathbf{W}_{n-1} \sim \mathcal{N}(0, \boldsymbol{Q}_{n-1}) \tag{2.26}$$

$$\mathbf{z}_n = \boldsymbol{H}_n\mathbf{x}_n + \mathbf{V}_n \qquad \text{avec } \mathbf{V}_n \sim \mathcal{N}(0, \boldsymbol{R}_n) \tag{2.27}$$

où $\boldsymbol{F}_n$ et $\boldsymbol{H}_n$ sont les matrices de transition et d'observation à l'instant n respectivement. $\mathbf{W}_n$ et $\mathbf{V}_n$ sont les bruits de dynamique et de mesure à l'instant n respectivement. Il vient alors que $p(\mathbf{x}_n|\mathbf{z}_{1:n})$ est une distribution Gaussienne de moyenne $\hat{\boldsymbol{x}}_{n|n}$ et de covariance $\boldsymbol{P}_{n|n}$. Similairement, la loi Gaussienne $p(\mathbf{x}_n|\mathbf{z}_{1:n-1})$, de moyenne $\hat{\boldsymbol{x}}_{n|n-1}$ et de covariance $\boldsymbol{P}_{n|n-1}$ est la loi de prédiction sur l'état à l'instant n. Les moments de la loi de filtrage et de prédiction sont obtenus récursivement et analytiquement via les équations du filtre de Kalman :

Initialisation	$\hat{\boldsymbol{x}}_{0\|0} = \boldsymbol{m}_{\mathbf{x}_0}$	$\boldsymbol{P}_{0\|0} = \boldsymbol{P}_0$
Prédiction	$\hat{\boldsymbol{x}}_{n\|n-1} = \boldsymbol{F}_{n-1}\hat{\boldsymbol{x}}_{n-1\|n-1}$	$\boldsymbol{P}_{n\|n-1} = \boldsymbol{F}_{n-1}\boldsymbol{P}_{n-1\|n-1}\boldsymbol{F}_{n-1}^T + \boldsymbol{Q}_n$
Mise à jour		$\hat{\boldsymbol{x}}_{n\|n} = \hat{\boldsymbol{x}}_{n\|n-1} + \boldsymbol{K}_n(\boldsymbol{z}_n - \boldsymbol{H}_n\hat{\boldsymbol{x}}_{n\|n-1})$ $\boldsymbol{P}_{n\|n} = \boldsymbol{P}_{n\|n-1} - \boldsymbol{K}_n\boldsymbol{H}_n\boldsymbol{P}_{n\|n-1}$ où $\boldsymbol{K}_n = \boldsymbol{P}_{n\|n-1}\boldsymbol{H}_n^T(\boldsymbol{R}_n + \boldsymbol{H}_n\boldsymbol{P}_{n\|n-1}\boldsymbol{H}_n^T)^{-1}$

(2.28)

2.3.5 Méthodes approximatives

2.3.5.1 Filtres de Kalman étendu et « unscented »

Il existe des extensions du filtre de Kalman reposant sur des approximations permettant de considérer des modèles non linéaires : le **filtre de Kalman étendu** (Extended Kalman Filter – EKF) [19] suppose l'utilisation de fonctions différentiables f et h pour les modèles de transition et/ou d'observation. Afin de pouvoir appliquer ces fonctions sur les matrices de covariance des équations du filtre, il est nécessaire de calculer, pour chaque instant n, les matrices Jacobiennes $\boldsymbol{F}_{n-1} = \frac{\partial f}{\partial \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_{n-1|n-1}}$ et $\boldsymbol{H}_n = \frac{\partial h}{\partial \boldsymbol{x}}|_{\hat{\boldsymbol{x}}_{n|n-1}}$ pour le modèle de dynamique et le modèle d'observation respectivement. Ce processus permet donc de linéariser les fonctions non linéaires autour de l'estimation actuelle.

Le **filtre de Kalman « unscented »** (Unscented Kalman Filter – UKF) [20] est une autre extension du filtre de Kalman standard. Quand les fonctions f et h sont non linéaires, la mise a jour des covariances après linéarisation réduisent drastiquement les performances en estimation de l'EKF. Dans le cas de l'UKF, un ensemble fini de points autour de la moyenne est choisi de manière déterministe pour représenter la distribution de probabilité de l'état estimé. Ces points sont ensuite propagés via les fonctions non linéaire du modèle avec lesquels une nouvelle moyenne et une nouvelle covariance peuvent être calculées à l'aide de la transformation « unscented » [21].

Tout comme les versions alternatives du filtre de Kalman standard, il existe d'autres méthodes d'estimation stochastique permettant de produire des estimés de l'état à partir de modèles de tran-

sition et/ou d'observation non linéaires et/ou non-Gaussiens. Deux familles de méthodes sont présentées par la suite, le filtre particulaire et le filtre variationnel Bayésien.

2.3.5.2 Méthode de Monte Carlo et filtre particulaire

L'idée principale du filtre particulaire repose sur la représentation de la loi jointe postérieure par une approximation de celle-ci obtenue à partir d'une méthode connue sous le nom de *méthode de Monte Carlo*. Cette représentation permet l'utilisation de lois de probabilité quelconques pour le modèle de dynamique et le modèle d'observation. Dans le filtre particulaire, la loi jointe postérieure est estimée au moyen de lois discrètes. Dans le cas général, une densité de probabilité quelconque continue est approchée par une distribution discrète

$$\hat{p}(\mathbf{x}) \triangleq \sum_{i=1}^{N} w^{(i)} \delta(\boldsymbol{x} - \boldsymbol{x}^{(i)}), \text{ avec } \sum_{i=1}^{N} w^{(i)} = 1. \tag{2.29}$$

$\boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(N)}$ et $w^{(i)}, \ldots, w^{(N)}$ constituent l'ensemble des particules pondérées servant à représenter une approximation de $p(\boldsymbol{x})$ où les $\boldsymbol{x}^{(i)}$ sont échantillonnées aléatoirement et les $w^{(i)}$ sont définis de sorte que $\hat{p}(\mathbf{x})$ constitue une bonne approximation de $p(\mathbf{x})$. D'après le théorème de transfert

$$p(\varphi) = \mathbb{E}_{p(\mathbf{x})}[\varphi(\boldsymbol{x})] = \int \varphi(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}, \tag{2.30}$$

l'image de toute fonction $\varphi(\cdot)$ d'une variable aléatoire se distribuant selon $p(\mathbf{x})$ est approchée par

$$\hat{I}(\varphi) = \mathbb{E}_{\hat{p}(\mathbf{x})}[\varphi(\boldsymbol{x})] = \sum_{i=1}^{N} w^{(i)} \varphi(\boldsymbol{x}^{(i)}). \tag{2.31}$$

On a donc un estimateur $\hat{I}(\varphi)$ associé à toute espérance selon $p(\boldsymbol{x})$. Par exemple, il peut être intéressant d'approcher la moyenne et la covariance d'une variable aléatoire $\mathbf{x}$ se distribuant selon $p(\mathbf{x})$.

Si possible, les échantillons $\boldsymbol{x}^{(i)}$ sont tirés depuis la loi $p(\mathbf{x})$. Dans le cas de l'**échantillonnage idéal**, on a :

$$\boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(N)} \overset{i.i.d.}{\sim} p(\mathbf{x}) \text{ et } \forall i \in \{1, \ldots, N\}, w^{(i)} = \frac{1}{N}. \tag{2.32}$$

Dans le cas contraire où il n'est pas possible ou pas souhaitable d'échantillonner $p(\mathbf{x})$, une distribution $q(\mathbf{x})$, appelée fonction d'importance, est utilisée pour échantillonner les particules. Le support de cette distribution erronée doit recouvrir celui de $p(\mathbf{x})$. L'ensemble des poids $w^{(i)}$ compense l'erreur induite par la distribution $q(\mathbf{x})$ choisie. C'est le cas de l'échantillonnage préférentiel,

ou « échantillonnage d'importance » (importance sampling). Dans ce cas on a :

$$\boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(N)} \overset{i.i.d.}{\sim} q(\mathbf{x}) \text{ et } w^{(i)} \propto \frac{p(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})}. \tag{2.33}$$

Dans un **filtre particulaire séquentiel avec échantillonage pondéré** [22], l'objectif est d'estimer la loi jointe a posteriori de l'équation (2.25) de manière récursive. Pour ce faire une fonction d'importance facile à échantillonner $q(\mathbf{x}_{0:n}|\mathbf{z}_{1:n})$ est utilisée. Cette fonction doit être « causale », c'est à dire

$$\forall n' \geq n, q(\mathbf{x}_{0:n}|\mathbf{z}_{1:n'}) = q(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}). \tag{2.34}$$

Il vient que

$$q(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}) = q(\mathbf{x}_n|\mathbf{x}_{0:n-1}, \mathbf{z}_{1:n})q(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1}) \tag{2.35}$$

L'approximation $\hat{p}(\mathbf{x}_{0:n}|\mathbf{z}_{1:n})$ de $p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n})$ est donc calculée comme 2.29 et définie par :

$$\boldsymbol{x}_{0:n}^{(i)} \overset{i.i.d.}{\sim} q(\mathbf{x}_{0:n}|\mathbf{z}_{1:n}) \text{ et } w^{(i)} \propto \frac{p(\boldsymbol{x}_{0:n}^{(i)}|\boldsymbol{z}_{1:n})}{q(\boldsymbol{x}_{0:n}^{(i)}|\boldsymbol{z}_{1:n})}. \tag{2.36}$$

Dans un schéma récursif, l'objectif est d'établir l'approximation particulaire de $p(\mathbf{x}_{0:n}|\mathbf{z}_{1:n})$ à partir de $p(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1})$ et de la nouvelle observation $\boldsymbol{z}_n$. Généralement, les particules (trajectorielles) sont obtenues par :

$$\boldsymbol{x}_{0:n-1}^{(i)} \sim q(\mathbf{x}_{0:n-1}|\mathbf{z}_{1:n-1}), \text{ et } \boldsymbol{x}_n^{(i)} \sim q(\mathbf{x}_n|\mathbf{x}_{n-1}^{(i)}, \mathbf{z}_n). \tag{2.37}$$

Le poids associé à chaque particule se calcule comme suit :

$$w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p(\boldsymbol{z}_n|\boldsymbol{x}_n^{(i)})p(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)})}{q(\boldsymbol{x}_n^{(i)}|\boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{z}_n)} \tag{2.38}$$

Après avoir normalisé les poids, on obtient l'approximation particulaire suivante de la densité de filtrage :

$$p(\mathbf{x}_n|\mathbf{z}_{1:n}) \approx \hat{p}(\mathbf{x}_n|\mathbf{z}_{1:n}) = \sum_{i=1}^{N} w_t^{(i)} \delta(\boldsymbol{x}_n - \boldsymbol{x}_n^{(i)}). \tag{2.39}$$

Cet algorithme souffre d'un phénomène appelé « dégénérescence » : Après plusieurs itérations, la majorité des particules sont affectées d'un poids négligeable. La dégénérescence du nuage de

particules peut être évaluée en calculant la taille efficace du N-échantillon comme suit [22] :

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w_n^{(i)})^2}. \tag{2.40}$$

Afin de limiter ce phénomène, une étape de rééchantillonnage est ajouté à la fin d'une étape de récursion. Pendant cette étape, chaque particule $\boldsymbol{x}_n^{(i)}$ est dupliquée un nombre N_j de fois proportionnel à son poids $w_n^{(i)}$. En moyenne, on peut obtenir $\mathbb{E}[N_j] = N w_n^{(i)}$ mais tout rééchantillonnage ajoute de la variance appelée « variance de Monte Carlo » sur N_j. L'algorithme de « rééchantillonnage systématique » permet de rééchantillonner les particules avec une variance de Monte Carlo minimale en garantissant que le nombre de duplications de chaque particule ne diffère pas de $N w_n^{(i)}$ de plus de 1. L'ajout d'une étape de rééchantillonnage donne naissance au **filtre particulaire séquentiel avec rééchantillonage pondéré** (Sequential Importance Resampling – SIR). Une version générique de cet algorithme est présenté dans l'algorithme 1.

Algorithme 1 Filtre particulaire séquentiel avec rééchantillonnage pondéré

1: **fonction** FILTRE_PARTICULAIRE($[\boldsymbol{x}_{n-1}^{(i)}, w_{n-1}^{(i)}]_{i=1}^N, z_n$)
2: Échantillonner $\boldsymbol{x}_0^{(1)}, \ldots, \boldsymbol{x}_0^{(i)}, \ldots, \boldsymbol{x}_0^{(N)}$ i.i.d selon $p(\boldsymbol{x}_0)$, et poser $w_0 = \frac{1}{N}$
3: **pour** $i = 1$ **à** N **faire**
4: Échantillonner $\boldsymbol{x}_n^{(i)} \sim q(\boldsymbol{x}_n | \boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{z}_n)$
5: Mettre à jour le poids selon

$$w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p(\boldsymbol{z}_n | \boldsymbol{x}_n^{(i)}) p(\boldsymbol{x}_n^{(i)} | \boldsymbol{x}_{n-1}^{(i)})}{q(\boldsymbol{x}_n^{(i)} | \boldsymbol{x}_{n-1}^{(i)}, \boldsymbol{z}_n)} \tag{2.41}$$

6: Normaliser l'ensemble des poids.
7: $[\boldsymbol{x}_n^{(i)}, w_n^{(i)}]_{i=1}^N$ permet d'approcher la loi de filtrage par $\hat{p}(\mathbf{x}_n | \mathbf{z}_{1:n}) = \sum_{i=1}^N w_t^{(i)} \delta(\boldsymbol{x}_n - \boldsymbol{x}_n^{(i)})$
8: Si N_{eff} est inférieur à un seuil préalablement fixé, rééchantillonner le nuage de particules avec l'algorithme de rééchantillonnage systématique.
9: **retourne** $[\boldsymbol{x}_n^{(i)}, w_n^{(i)}]_{i=1}^N$

2.3.5.3 Approximation et filtre de Kalman variationnels

Cette section présente le filtrage Bayésien variationnel. Certains algorithmes comme le filtre particulaire se sont avérés particulièrement populaires pour approcher les lois de probabilité quelconques dont l'obtention repose sur des calculs intractables. Le filtre particulaire souffre de deux principaux défauts : le phénomène de dégénérescence du nuage de particules et son coût calculatoire important, particulièrement quand la dimension de l'état filtré est grande. Les algorithmes de filtrage basés sur des approximations variationnelles cherchent à régler ces problèmes tout en

produisant des estimés de bonne qualité [23].

Soit $\mathbf{x}$ une variable aléatoire et $\mathbf{x}^1, \ldots, \mathbf{x}^K$ ses partitions en K blocs. Soit $\mathcal{Q}$ l'ensemble des densités de probabilité séparables en la variable $\mathbf{x}$, *i.e.*, $\mathcal{Q} = \{\tilde{q}(\mathbf{x}) = \prod_{k=1}^{K} \tilde{q}(\mathbf{x}^k)\}$. Une densité de probabilité (séparable) $q(\mathbf{x})$ est définie comme l'**approximation variationnelle Bayésienne** (Variational Bayesian Approximation – VB-Approximation) d'une densité $p(\mathbf{x})$ si elle minimise, sur $\mathcal{Q}$, la divergence de Kullback-Leibler de p vers q, *i.e.*,

$$q(\mathbf{x}) := \prod_{k=1}^{K} q(\mathbf{x}^k) = \arg \min_{\tilde{q}(\mathbf{x}) := \prod_{k=1}^{K} \tilde{q}(\mathbf{x}^k)} \mathrm{KL}\big(\tilde{q}(\mathbf{x}) \| p(\mathbf{x})\big). \tag{2.42}$$

On montre que la solution de (2.42) s'écrit comme suit [23] :

$$q(\mathbf{x}) := \prod_{k=1}^{K} q(\mathbf{x}^k), \quad q(\mathbf{x}^k) \propto \exp\Big(\mathbb{E}_{q(\mathbf{x}^{k-})}\big[\ln p(\mathbf{x})\big]\Big), \tag{2.43}$$

où $\mathbf{x}^{k-}$ indique le complément de $\mathbf{x}^k$ dans $\mathbf{x}$. La deuxième partie de l'équation (2.43) ne constitue pas une solution explicite pour calculer les K facteurs optimaux $q(\mathbf{x}^k)$ de l'approximation variationnelle recherchée. En effet, ils dépendent chacun de moments calculés sur leur facteurs complémentaires. Il est donc nécessaire d'estimer cycliquement les moments de l'ensemble des facteurs [23].

Dans [23] le potentiel des méthodes variationnelles est étudié pour le cas du filtrage bayésien avec comme objectif d'approcher de manière récursive la densité de filtrage $p(\mathbf{x}_n|\mathbf{z}_{0:n})$. Un **filtre de Kalman Bayésien variationnel** (Variational Bayesian Kalman Filter – VBKF) est développé comme possible solution à ce problème. Les deux partitions dans $\mathbf{x}_{0:n}$ sont respectivement, pour chaque n instant, $\mathbf{x}^1 = \mathbf{x}_n$ et $\mathbf{x}^2 = \mathbf{x}_{n-1}$. Dans ce cadre, les approximations VB suivantes sont impliquées :

$$q(\mathbf{x}_n|\mathbf{z}_{0:n-1}) \propto \exp\Big(\mathbb{E}_{q(\mathbf{x}_{n-1}|\mathbf{z}_{0:n})}\big[\ln p(\mathbf{x}_n|\mathbf{x}_{n-1})\big]\Big) \tag{2.44}$$

$$q(\mathbf{z}_n|\mathbf{x}_{n-1}, \mathbf{z}_{0:n-1}) \propto \exp\Big(\mathbb{E}_{q(\mathbf{x}_n|\mathbf{z}_{0:n})}\big[\ln p(\mathbf{x}_n|\mathbf{x}_{n-1})\big]\Big). \tag{2.45}$$

Elles correspondent respectivement à une approximation VB du modèle de transition et d'observation du filtre. De ces deux approximations VB sont exprimées les marginales VB suivantes :

$$q(\mathbf{x}_n|\mathbf{z}_{0:n}) \propto p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{z}_{0:n-1}) q(\mathbf{x}_n|\mathbf{z}_{0:n-1}) \tag{2.46}$$

$$q(\mathbf{x}_{n-1}|\mathbf{z}_{0:n}) \propto q(\mathbf{z}_n|\mathbf{x}_{n-1}, \mathbf{z}_{0:n}) q(\mathbf{x}_{n-1}|\mathbf{z}_{0:n-1}). \tag{2.47}$$

Ces deux expressions correspondent respectivement à la distribution de filtrage VB et la distribution de lissage VB à l'instant n. La distribution de filtrage VB est propagée au fil des instants. La distribution de lissage VB n'est pas propagée au fil des instants mais ses moments sont utilisés pour calculer la marginale VB de filtrage. Il est intéressant de remarquer que les étapes standards de prédiction et de mise à jour Bayésienne sont remplacées par deux étapes de « mise à jour VB » reposant sur l'utilisation des deux marginales VB définies dans 2.47. Pour le VBKF, il n'est pas nécessaire de recourir à un algorithme itératif pour estimer les moments sur les facteurs complémentaires des approximations, l'ensemble des calculs pouvant être fait analytiquement.

Dans [23], les auteurs proposent une série de schémas de filtrage Bayésien pour l'application des approximations VB.

2.4 Apprentissage automatique

Dans ce document, l'apprentissage du modèle de dynamique a priori des véhicules en interaction est étudié. Les algorithmes d'apprentissage automatique reposent sur des techniques statistiques et d'optimisation avec une attention particulière sur leur mise en place pratique sur des calculateurs informatiques. Une tâche résolue par de tels algorithmes décrit usuellement quelles sorties, notées $\boldsymbol{y} \in \mathbb{R}^M$, sont souhaitées du système d'apprentissage automatique après avoir été alimenté avec des données d'entrée, notées $\boldsymbol{x} \in \mathbb{R}^N$. Une tâche peut être vue comme l'apprentissage d'une fonction f reliant $\boldsymbol{x}$ et $\boldsymbol{y}$. Dans cette thèse, trois tâches sont particulièrement intéressantes :

— **La classification** : l'algorithme d'apprentissage doit apprendre à produire une fonction $f : \mathbb{R}^N \to \{1, \ldots, k\}$. Avec $y = f(\boldsymbol{x})$, le modèle assigne une des k catégories de sortie à un vecteur d'entrée.

— **La régression** : Cette tâche est similaire à la précédente, seul le domaine de sortie de l'algorithme change. Ici, le système apprend une fonction telle que $f : \mathbb{R}^N \to \mathbb{R}^M$.

— **L'estimation de densité de probabilité** : Dans ce problème, la fonction f devient p_{model} avec $p_{\text{model}} : \mathbb{R}^N \to \mathbb{R}^M$. $p_{\text{model}}(\boldsymbol{y}|\boldsymbol{x})$ est interprétée comme une densité de probabilité sur les données de sortie conditionnellement aux données d'entrée.

Après la définition de la tâche à accomplir par l'algorithme d'apprentissage, il faut être en mesure d'évaluer les performances du système pour l'accomplissement de cette tâche. Le critère utilisé pour l'évaluation dépend de la tâche à accomplir, des données et de l'algorithme d'apprentissage utilisé. Par exemple dans le cas de la classification, il peut être intéressant de mesurer le taux d'erreur sur les catégories prédites à partir des entrées. Dépendant de la façon dont sont entraînés les algorithmes, trois catégories d'apprentissage peuvent être dégagées.

— **L'apprentissage supervisé** rassemble les algorithmes entraînés sur un jeu de données contenant des données d'entrées $\boldsymbol{x}$ associées à des données de sorties $\boldsymbol{y}$. Ces données de sortie sont souvent appelées cibles ou étiquettes selon la tâche considérée.

— **L'apprentissage non supervisé** regroupe les cas où le système est entraîné avec un jeu de données sans association explicite entre des données $\boldsymbol{x}$ et des cibles $\boldsymbol{y}$. Généralement, l'objectif est d'apprendre à extraire automatiquement des propriétés utiles sur le jeu de données observé.

— **L'apprentissage par renforcement** traite des algorithmes qui sont appris sur un jeu de données évoluant au cours de l'apprentissage. Ces algorithmes interagissent avec un environnement de façon à créer une boucle rétroactive entre eux-mêmes et des données d'apprentissage obtenues depuis leur évolution dans cet environnement. Cette catégorie de système ne sera pas détaillée dans ce document.

L'apprentissage profond est une sous-catégorie de l'apprentissage automatique qui traite des algorithmes basés sur des réseaux de neurones. Ces réseaux approchent une fonction f dépendante de la tâche voulue par le biais de couches successives composées d'unités de calcul. Un nombre important d'architectures de réseaux de neurones existe [13]. Dans les parties suivantes nous présentons l'architecture des trois types de réseaux élémentaires utilisés dans la suite du manuscrit.

2.4.1 Réseaux de neurones à propagation avant

Les réseaux de neurones à propagation avant (Feed-Forward Neural Networks – FFNNs) constituent le premier type de réseaux présentés dans ce document. Leur présentation permet également de détailler les principales étapes du processus d'apprentissage des réseaux de neurones. Dans ces réseaux, les données ne transitent que vers l'avant en partant de $\boldsymbol{x}$ jusqu'à $\boldsymbol{y}$ en passant par des calculs intermédiaires définissant la fonction du réseau. Aucune boucle n'est présente dans les FFNNs et ils peuvent être représentés par un graphe acyclique orienté. La fonction apprise peut donc s'écrire sous la forme $\boldsymbol{y} = f(\boldsymbol{x}) = f_i(f_{i-1}((\ldots)f_1(\boldsymbol{x})))$ où les sous-fonctions $f_1, \ldots, f_i$ sont désignées comme les couches composant le réseau.

2.4.1.1 Structure d'un FFNN

Entre les noeuds d'entrées et de sortie d'un réseau de neurones à propagation avant se trouvent les couches de **noeuds cachées** notés $\boldsymbol{h} = \boldsymbol{h}_1, \ldots, \boldsymbol{h}_i$. À la fin de la structure du réseau, les **noeuds de sortie** ont pour objectif de transformer la sortie de la séquence de couches de noeuds cachés en un résultat permettant de remplir l'objectif applicatif. Une forme élémentaire pour les noeuds de sortie est une transformation affine des sorties des noeuds cachés. Soit $\boldsymbol{h} = f(\boldsymbol{x}; \boldsymbol{\theta})$ un ensemble

de noeuds cachés, une couche de sortie élémentaire est définie par :

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{h} + \boldsymbol{b} \tag{2.48}$$

avec $\boldsymbol{W}$ et $\boldsymbol{b}$ paramètres du réseau représentant les poids et les biais de la couche de sortie respectivement. Il convient d'adapter la couche de sortie du réseau afin d'approcher la fonction f voulue.

Pour la $i^{\text{ième}}$ couche du réseau, son vecteur de noeuds cachés $\boldsymbol{h}_i$ est défini par :

$$\boldsymbol{h}_i = g(\boldsymbol{W}_i^T \boldsymbol{h}_{i-1} + \boldsymbol{b}_i) \tag{2.49}$$

où $g(.)$ est une fonction non linéaire appelée fonction d'activation, $\boldsymbol{W}_i$ et $\boldsymbol{b}_i$ paramètres du réseau représentant les poids et biais de la couche de noeuds cachée respectivement. Dans un FFNN, les **noeuds d'entrée** correspondent simplement à la première couche cachée qui prend en entrée directement les données d'apprentissage. Dans ce cas, dans l'équation (2.49), on a $\boldsymbol{h}_{i-1} = \boldsymbol{x}$. Les applications successives d'une fonction non linéaire sur une transformation affine des données d'entrée permettent théoriquement d'apprendre, via le réseau de neurones, une fonction quelconque [24].

Le choix du nombre de couches cachées, de noeuds par couche, des fonctions d'activations utilisées et des relations entre les différentes entrées et sorties des couches du réseau constitue une problématique majeure dans la littérature de l'apprentissage profond. En plus des choix d'architecture, la plupart des algorithmes d'apprentissage automatique ont plusieurs paramètres utilisés pour contrôler leur comportement. Ces paramètres sont appelés hyperparamètres. Les valeurs de ces hyperparamètres ne sont pas apprises par l'algorithme d'apprentissage.

2.4.1.2 Fonction de coût et optimisation des paramètres du réseau

Formellement, un réseau de neurones cherche à apprendre une fonction f telle que $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ avec $\boldsymbol{x}$ et $\boldsymbol{y}$ les données d'entrée et de sortie, respectivement. Au cours de l'apprentissage, la valeur des paramètres $\boldsymbol{\theta}$ du réseau - ses biais et ses poids – est optimisée de manière à produire la meilleure approximation f^* recherchée de la fonction f.

Un choix important dans la construction d'un algorithme d'apprentissage automatique est le choix de la fonction de coût $J(\boldsymbol{\theta})$ qui formalise l'objectif d'optimisation des paramètres à partir des entrées et sorties du réseau. Un choix populaire lors de l'apprentissage d'une probabilité conditionnelle entre les entrées $\boldsymbol{x}$ et les sorties $\boldsymbol{y}$ est l'anti-log-vraisemblance. Cette dernière est calculée entre les données d'apprentissage distribuées selon p_{data} et des sorties obtenues depuis l'évaluation de la distribution de probabilité p_{model} représentée par le réseau. Cette fonction de coût est donnée

par :

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{y}|\boldsymbol{x}). \tag{2.50}$$

La forme spécifique de cette fonction de coût $J(\boldsymbol{\theta})$ dépend de la forme de p_{model} (et dans ce cas de la façon d'approcher l'espérance). Cette forme spécifique est liée aux choix des noeuds de sortie du réseau.

2.4.1.3 Rétropropagation et descente de gradient

Dans les sections précédentes, les considérations élémentaires d'architecture d'un réseau de neurones à propagation avant ont été abordées. Ici, l'apprentissage proprement dit des paramètres du réseau est expliqué. La non-linéarité du réseau de neurones a pour effet de rendre non convexes la plupart des fonctions de coût utilisées pour apprécier sa performance dans la réalisation d'une tâche. Ajouté à cela leur nombre important de paramètres et de données traitées, il est nécessaire d'avoir recours à des méthodes d'optimisation numérique itératives pour pouvoir réaliser l'apprentissage du réseau. Ces méthodes proposent seulement des garanties vers un optimum local. Lorsqu'un réseau neuronal à propagation avant accepte une entrée $\boldsymbol{x}$ et produit une sortie $\hat{\boldsymbol{y}}$, l'information circule dans le réseau « vers l'avant ». Les entrées $\boldsymbol{x}$ sont propagées à travers les couches cachées pour finalement produire $\hat{\boldsymbol{y}}$. Ce processus est appelé propagation avant. Pendant la phase d'apprentissage, les sorties produites pendant la propagation avant permettent de produire un coût scalaire $J(\boldsymbol{\theta})$. L'algorithme de rétro-propagation [25] permet de faire circuler en sens inverse dans le réseau le gradient du coût par rapport aux paramètres $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ en fonction de l'erreur calculée sur la sortie $\hat{\boldsymbol{y}}$ par rapport à la sortie $\boldsymbol{y}$ provenant du jeu de données. Le gradient de la fonction de coût est calculé en chaîne pour les paramètres de chaque couche de neurones à partir d'algorithmes de dérivation automatiques [26, 27] reposant sur le théorème de dérivation des fonctions composées.

Appliqués itérativement sur un ensemble de données d'apprentissage $\{\boldsymbol{x}, \boldsymbol{y}\}$, la propagation avant, le calcul du coût scalaire depuis l'erreur entre $\hat{\boldsymbol{y}}$ et $\boldsymbol{y}$ avec la fonction de coût, puis la rétro-propagation du gradient du coût sur les paramètres du réseau permettent au réseau de représenter une fonction $\hat{f}$ s'approchant graduellement de la fonction f^*. La définition de la fonction de coût par rapport à la tâche à effectuer par le réseau est primordiale pour se rapprocher au mieux de f^*. Généralement, la condition d'arrêt du processus d'optimisation du réseau est obtenue en observant les valeurs de la fonction de coût sur un ensemble de données de validation que le réseau n'a pas observé pour son apprentissage. Lorsque que le coût scalaire, calculé au fur et à mesure sur cet ensemble, ne diminue plus (ou même augmente), l'apprentissage peut être arrêté. Après cela, le réseau est utilisé lors de l'inférence sur un ensemble de test pour produire les $\hat{\boldsymbol{y}}$ correspondant aux sorties souhaitées pour l'application considérée. Ce sous-ensemble n'a pas été observé par le réseau

ni pendant l'apprentissage, ni pendant la phase de validation.

2.4.2 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (Convolutional Neural Networks – CNNs) sont spécialisés dans le traitement des données possédant une topologie en grille. Les images et les séries temporelles sont deux exemples communs. Les premières peuvent être vues comme des tableaux 2D de pixels ; les secondes comme un tableau 1D de données prises à un intervalle de temps constant. L'unique différence entre les réseaux convolutifs et les réseaux à propagation avant est l'utilisation, sur au moins une des couches du réseau, d'une opération de convolution entre les poids du réseau et les données. Cette opération remplace la multiplication matricielle standard sur les réseaux à propagation avant.

Dans le cas spécifique de l'apprentissage automatique, les fonctions convoluées sont souvent discrètes et sous forme de tableaux multidimensionnels appelé tenseurs. Les noyaux utilisés pour traiter ces données sont également le plus souvent des tableaux multidimensionnels. Dans le domaine de l'apprentissage automatique et dans ce document, par abus de langage, l'opération de corrélation croisée n'est pas différenciée de celle de la convolution. Le produit de convolution d'une image 2D I et d'un noyau 2D K s'écrit :

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n). \tag{2.51}$$

Une des motivations principales de l'emploi des CNNs dans une tâche d'apprentissage est l'utilisation d'un tenseur de poids unique pour chaque couche réalisant une opération de convolution. Ce partage de paramètres permet de réduire l'empreinte mémoire, d'améliorer les performances du réseau et de permettre une invariance par translation du traitement des données. En effet, un réseau de neurones standard relie entièrement l'entrée de chaque couche à la sortie de la couche suivante augmentant ainsi énormément le nombre de paramètres nécessaires pour apprendre un modèle traitant des données de haute dimension.

2.4.3 Réseaux de neurones récurrents

Les réseaux de neurones récurrents (Recurrent Neural Networks – RNNs) sont spécialisés dans le traitement des données séquentielles. Comme pour les CNNs, les RNNs ont recours à un partage de paramètres leur permettant d'obtenir des propriétés spéciales quant au traitement de leur données en entrée. Dans le cas des RNNs, ce partage permet aux réseaux de traiter efficacement des séquences de données de longueurs variables dont la position des données dans ces dernières varie. Elle repose sur l'hypothèse qu'un même jeu de paramètres et de fonctions d'activation peut

être utilisé pour traiter des instants différents d'une séquence de données. Ce partage se manifeste par la présence de boucles dans l'architecture du réseau reliant les données et/ou les paramètres sur des instants n différents. Le prix à payer pour un nombre réduit de paramètres est un processus d'apprentissage plus difficile.

2.4.3.1 Propagation avant dans les réseaux de neurones récurrents

La littérature définit l'état d'un réseau récurrent comme les sorties de ses noeuds cachés $\boldsymbol{h}$. En considérant une séquence de données d'entrée $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, l'état à l'instant n est défini par :

$$\boldsymbol{h}_n = g(\boldsymbol{W}\boldsymbol{h}_{n-1} + \boldsymbol{b} + \boldsymbol{U}\boldsymbol{x}_n), \tag{2.52}$$

où $\boldsymbol{b}$ sont les vecteurs de biais avec $\boldsymbol{W}$ et $\boldsymbol{U}$ les vecteurs de poids des connections [états cachés]→[états cachés] et [données d'entrée]→[états cachés] respectivement. g désigne la fonction d'activation. On remarque dans (2.52) que les poids, les biais et la fonction d'activation sont partagés pour tous les instants n de la séquence traitée. Afin de produire la séquence de sortie désirée $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, une couche de sortie similaire à celle d'un réseau à propagation avant est utilisée. Une forme élémentaire de cette couche reliant l'état caché et la sortie à un instant n pourrait être définie comme suit :

$$\boldsymbol{y}_n = \boldsymbol{V}\boldsymbol{h}_n + \boldsymbol{c}, \tag{2.53}$$

avec $\boldsymbol{V}$ et $\boldsymbol{c}$ les poids et les biais, respectivement. Selon les architectures employées, un réseau de neurones récurrent peut produire une sortie à tous les instants ou bien agréger la valeur de ses états sur plusieurs instants pour produire une unique sortie. Les connexions entre les états précédents et les entrées du réseau peuvent différer selon le modèle considéré. Les équations (2.52) et (2.53) constituent le processus de propagation avant générique pour un réseau récurrent standard. Le graphe d'un tel réseau est illustré dans la Figure 2.5.

2.4.3.2 Fonction de coût et rétro-propagation du gradient

Dans le cas courant où le réseau apprend à relier une séquence d'entrée $\boldsymbol{X}$ à une séquence de sortie $\boldsymbol{Y}$ de même taille, le coût total est la somme des coûts sur tous les instants

$$J(\boldsymbol{\theta}) = -\sum_{n}^{N} \log p_{\text{model}}(y_n | \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}), \tag{2.54}$$

où $p_{\text{model}}(y_n | \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\})$ est représenté par le réseau récurrent.

Pour calculer le gradient du coût relativement aux paramètres d'un réseau de neurones récur-

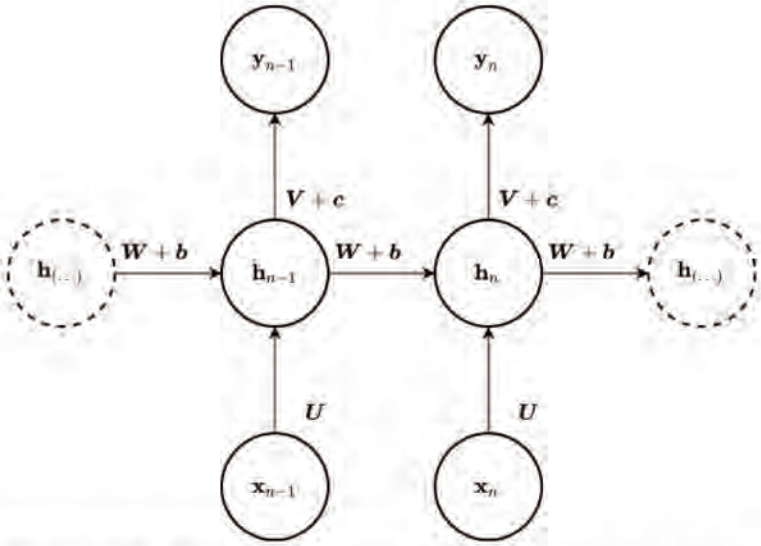


FIGURE 2.5 – Le graphe d'un réseau récurrent qui met en correspondance une séquence d'entrées $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ avec une séquence de sorties $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$. Le réseau possède des connexions [entrées]→[états cachés] paramétrées par une matrice de poids $\boldsymbol{U}$, des connexions récurrentes [états cachés]→[états cachés] paramétrées par une matrice de poids $\boldsymbol{W}$ et un vecteur de biais $\boldsymbol{b}$, et des connexions états cachés-sorties paramétrées par une matrice de poids $\boldsymbol{V}$ et les biais $\boldsymbol{c}$.

rent, il suffit d'appliquer récursivement l'algorithme de rétro-propagation introduit dans la section 2.4.1.3 à travers le temps pour chaque couche du réseau. L'utilisation de la rétro-propagation sur le graphe ainsi « déroulé » est appelée algorithme de rétro-propagation dans le temps. Les gradients ainsi obtenus par rétro-propagation sont ensuite utilisés pour optimiser les paramètres du réseau et apprendre la fonction désirée.

2.4.3.3 Le « problème de disparition du gradient » et les réseaux récurrents à mémoire court et long terme

L'une des difficultés majeures lors de l'apprentissage des réseaux récurrents est la capture des dépendances long terme dans les données. Ce problème provient de la diminution (ou plus rarement de l'explosion) progressive du gradient du coût au fil de sa propagation dans le réseau en raison du nombre important de multiplications des dérivées partielles de la fonction de coût par rapport aux paramètres du réseau. Il devient alors compliqué d'apprendre des relations long terme entre les entrées et sorties du jeu de données.

Une solution pour traiter les dépendances à long terme est de concevoir un modèle qui fonctionne sur plusieurs échelles de temps, de sorte que certaines parties du réseau fonctionnent à des échelles de temps fines, tandis que d'autres parties fonctionnent à des échelles de temps plus gros-

sières et transfèrent plus efficacement les informations du passé vers l'instant courant.

Les approches les plus populaires dans la littérature reposent sur des architectures basées sur des réseaux récurrents à portes (Gated Recurrent Unit network – GRU network) [28]. Ce type de réseau possède des connections et des poids qui leur permettent d'accumuler ou d'oublier de l'information à travers le temps. Lors de l'apprentissage, le réseau apprend au fur et mesure la quantité d'information à garder ou à oublier pour capturer correctement les dépendances temporelles dans les données.

Les réseaux récurrents à mémoire court et long terme (Long Short Term Memory networks – LSTM networks ou simplement LSTM) [29] sont un type particulier de réseau à portes qui comprend des boucles internes associées à des poids qui dépendent du contexte d'apprentissage. Cette propriété très importante permet aux LSTMs d'apprendre l'échelle de temps sur laquelle les données exhibent des dépendances temporelles ce qui leur permet de capturer des relations passées proches mais aussi lointaines.

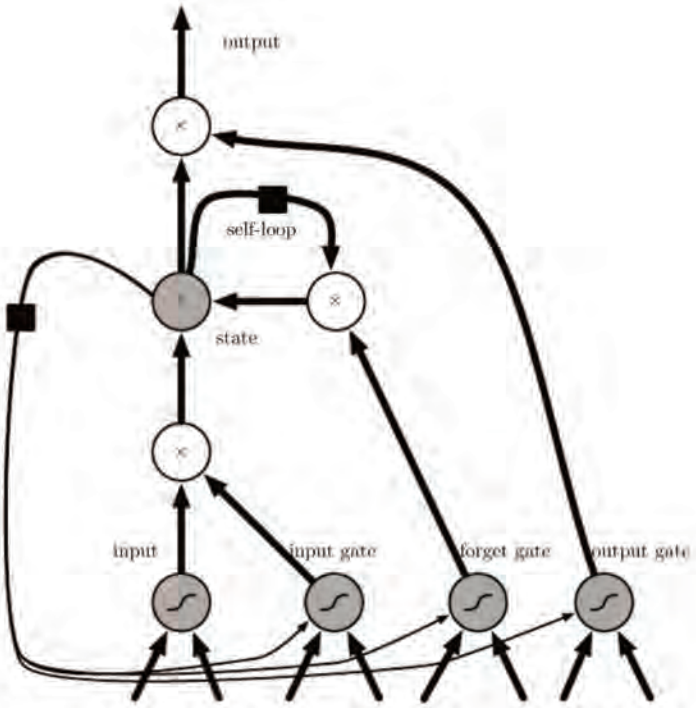


FIGURE 2.6 – Schéma fonctionnel d'une cellule LSTM dans un réseau récurrent [13]. Les nœuds sont connectées de manière récurrente les uns aux autres. Les entrées sont traitées avec un nœud standard de FFNN. Sa valeur en sortie peut être accumulée dans l'état si la porte d'entrée le permet. Le nœud d'état possède une boucle linéaire sur lui-même dont le poids est contrôlé par la porte d'oubli. La sortie de la cellule LSTM peut être modulée par la porte de sortie. La fonction d'activation des portes est la fonction sigmoïde. Le nœud d'entrée peut avoir une fonction différente. Le carré noir indique un retard d'un pas de temps.

Un LTSM profond comporte plusieurs noeuds ou « cellules » possédant une boucle interne, en plus de la récurrence externe du réseau de neurones récurrents. Chaque cellule admet les mêmes entrées et sorties qu'un réseau récurrent classique, mais possède davantage de paramètres et un système de portes de contrôle qui modulent le flux d'informations comme illustré dans la Figure 2.6. Par la suite, chaque porte possède un ensemble de poids récurrents $\boldsymbol{W}$, de poids d'entrée $\boldsymbol{U}$ et de biais $\boldsymbol{b}$. Le composant le plus important d'un réseau LSTM est son vecteur mémoire (ou vecteur état) $\boldsymbol{s}_n$ qui possède une boucle interne commandée par une porte « d'oubli » comme suit :

$$\boldsymbol{f}_n = \sigma\big(\boldsymbol{U}^f \boldsymbol{x}_n + \boldsymbol{W}^f \boldsymbol{h}_{n-1} + \boldsymbol{b}^f\big), \tag{2.55}$$

où $\boldsymbol{x}_n$ est l'entrée vectorielle courante, $\boldsymbol{h}_{n-1}$ la couche cachée de l'instant précédent et $\sigma(.)$ la fonction sigmoïde. La porte d'entrée externe qui contrôle la quantité d'information à intégrer dans l'état des cellules depuis les données d'entrées est définie de la même façon que pour la porte d'oubli :

$$\boldsymbol{g}_n = \sigma\big(\boldsymbol{U}^g \boldsymbol{x}_n + \boldsymbol{W}^g \boldsymbol{h}_{n-1} + \boldsymbol{b}^g\big). \tag{2.56}$$

La mise à jour du vecteur d'état interne au réseau est réalisée avec les équations suivantes :

$$\tilde{\boldsymbol{s}}_n = \tanh\big(\boldsymbol{U}^s \boldsymbol{x}_n + \boldsymbol{W}^s \boldsymbol{h}_{n-1} + \boldsymbol{b}^s\big) \tag{2.57}$$

$$\boldsymbol{s}_n = \boldsymbol{f}_n \circ \boldsymbol{s}_{n-1} + \boldsymbol{g}_n \circ \tilde{\boldsymbol{s}}_n, \tag{2.58}$$

où $\circ$ indique le produit matriciel d'Hadamard. La sortie $\boldsymbol{o}_n$ des cellules est modulée par sa porte de sortie $\boldsymbol{q}_n$:

$$\boldsymbol{q}_n = \sigma\big(\boldsymbol{U}^o \boldsymbol{x}_n + \boldsymbol{W}^o \boldsymbol{h}_{(n-1)} + \boldsymbol{b}^o\big) \tag{2.59}$$

$$\boldsymbol{o}_n = \tanh\left(\boldsymbol{s}_n\right) \circ \boldsymbol{q}_n. \tag{2.60}$$

Plusieurs travaux démontrent l'efficacité des LSTM à apprendre des dépendances long terme par rapport aux architectures récurrentes classiques sur des tâches difficiles de traitement de séquences où des performances de pointe sont atteintes [30, 31, 32].

2.4.4 Modèles génératifs et modèles discriminatifs profonds

Les modèles génératifs profonds (Deep Generative Models – DGM) et les modèles discriminatifs profonds (Deep Discriminative Models – DDM) [13] représentent deux catégories fondamentalement différentes de réseaux de neurones. Ils diffèrent tous les deux de par la modélisation statistique employée pour répondre à une tâche d'apprentissage. Les modèles génératifs cherchent à

représenter la distribution de probabilité des données observées et ainsi permettre, par exemple, de générer des instances similaires. Certains de ces modèles permettent également d'exhiber des liens de dépendance statistique dans les données traitées et de faire des hypothèses sur les relations qui sous-tendent celles-ci. A l'opposé, les modèles discriminatifs apprennent directement à modéliser la distribution de probabilité postérieure conditionnelle des données de sortie conditionnellement aux données d'entrée.

En considérant une tâche de classification standard avec comme jeu de données, des entrées $\boldsymbol{x}$ distribuées selon $p_{\text{data}}(\mathbf{x})$ et des catégories de sortie souhaitées $\boldsymbol{y}$ distribuées selon $p_{\text{data}}(\mathbf{y})$, les modèles discriminatifs se contentent d'apprendre à produire directement la probabilité conditionnelle $p(\boldsymbol{y}|\mathbf{x} = \boldsymbol{x})$ qui permet de classifier l'entrée observée. Les modèles génératifs visent à résoudre le problème plus général d'apprentissage de la distribution jointe $p(\mathbf{y}, \mathbf{x})$ sur les deux variables du problème. En utilisant le théorème de Bayes, $p(\mathbf{y}|\mathbf{x})$ peut être calculée pour ensuite obtenir le résultat de la classification [33].

Modéliser la distribution de probabilité des données impliquent souvent des calculs intractables sur les distributions de probabilité. Une des approches pour apprendre de tels modèles repose sur l'utilisation de variables latentes intermédiaires de faible dimension pour approcher la véritable distribution de probabilité des données. Ce type d'apprentissage est récemment devenue l'une des approches dominantes des modèles génératifs profonds, sous la forme de l'auto-encodeur variationnel [34]. Son extension, l'auto-encodeur variationnel conditionnel [35] est décrite dans le Chapitre 5.

CHAPITRE 3
ÉTAT DE L'ART

Dans ce chapitre, un état de l'art en trois parties est présenté. Premièrement, un état de l'art sur la prédiction de mouvement pour les véhicules intelligents est proposé. Cette partie traite des nombreuses techniques permettant la prédiction de l'évolution des scènes de trafic. Certaines modélisations sont également présentées. Dans une seconde partie, une présentation élémentaire de la littérature sur l'évaluation de risque pour les véhicules est effectuée. Enfin, le chapitre se termine par un descriptif des principaux jeux de données véhiculaires.

3.1 Prédiction du mouvement pour les véhicules intelligents

Pour être en mesure d'évaluer la dangerosité future d'une scène routière, un système de transport intelligent doit être capable de prédire avec une fiabilité suffisante l'évolution du trafic. Cette prédiction repose sur une analyse des données en provenance des véhicules de façon à déterminer l'intention des conducteurs et prédire l'état de la flotte. La prédiction du mouvement (*motion prediction*) constitue le domaine principal de la littérature lié à la problématique de cette thèse. Les approches de prédiction du mouvement pour les véhicules sont généralement classées en trois types [36] : les approches basées sur des modèles physiques, sur des modèles de manœuvres, ou sur des modèles permettant la prise en compte des interactions inter-véhicules. L'organisation de cette section suivra cette partition, illustrée Figure 3.1. La référence [37] recense un grand nombre d'articles traitant de la prédiction de trajectoires de piétons ou de véhicules. Les auteurs proposent une taxonomie sensiblement différente basée sur des variations du paradigme *Percevoir-Raisonner-Agir* et sur les informations contextuelles utilisées pour la prédiction.

3.1.1 Modèles basés sur la physique

3.1.1.1 Modélisation

Si les modèles de mouvement basés sur la physique représentent chaque véhicule comme des entités dynamiques régies par les lois de la physique, alors leur mouvement est représenté par des modèles cinématiques et dynamiques reliant certaines entrées de commande (*e.g.*, la direction, l'accélération), les propriétés du véhicule (*e.g.*, la masse) et les conditions externes (par exemple, le coefficient de friction de la surface de la route) à l'évolution de l'état du véhicule (*e.g.*, la position, le cap, la vitesse). La prédiction de mouvement peut être ensuite effectuée en intégrant cette modé-

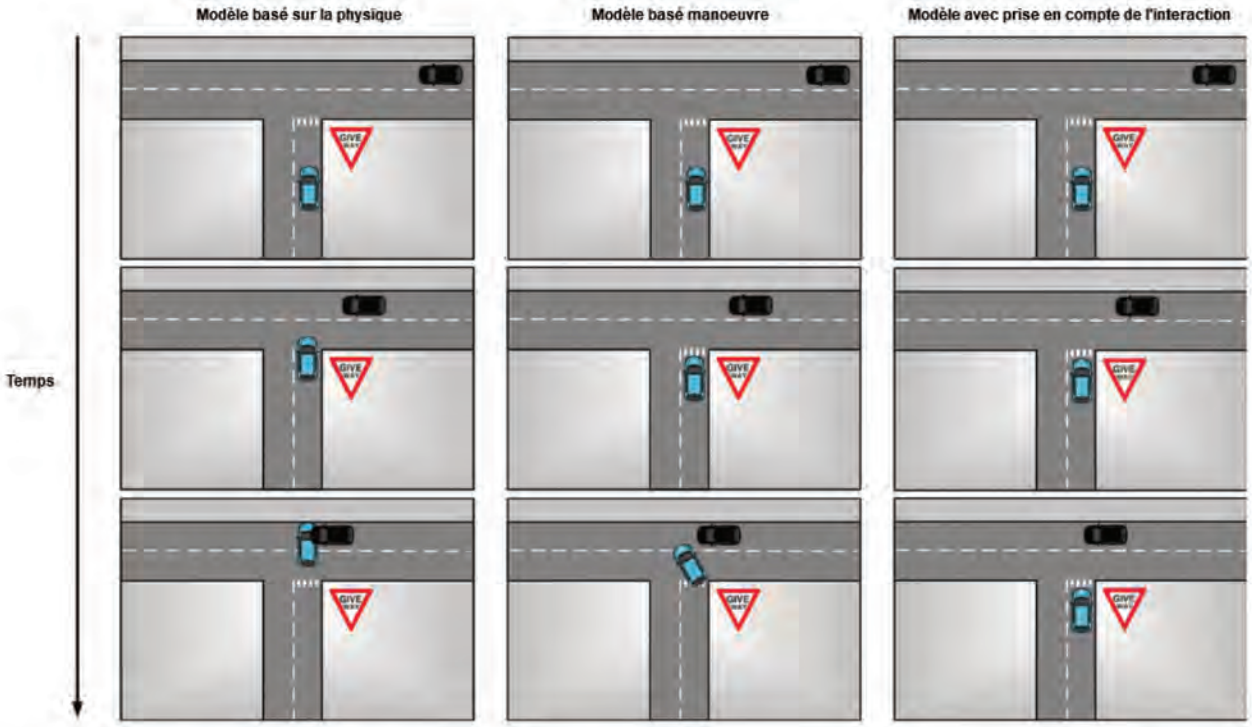


FIGURE 3.1 – Exemples extraits de [36] de prédiction de mouvement basée sur différents types de modèles. Un scénario d'intersection entre une voiture bleue et une voiture noire prioritaire permet d'illustrer les différentes catégories de modèles de mouvement considérés. (À gauche) Le modèle basé physique se contente de faire évoluer la position des deux voitures dans le temps sans considération de la topologie, de la signalétique ni des interactions entre les véhicules. (Au milieu) La topologie et la signalétique sont considérées. La voiture bleue s'arrête au panneau avant de repartir pour s'insérer dans la voie d'en face. Cependant, les interactions n'étant toujours pas considérées, les deux voitures semblent être vouées à entrer en collision. (À droite) Dans cette dernière catégorie, en plus des autres considérations sur l'environnemt routier, les interactions entre les véhicules sont prises en compte. Dans ce cas, la voiture bleue cède le passage à la voiture noire.

lisation dans un schéma d'inférence. Ces modèles souffrent d'un faible pouvoir de représentation, qui les empêche d'être utilisés dans un autre contexte que celui de la prédiction du mouvement à court terme (inférieur à une seconde).

Les modèles les plus courants de cette famille sont les modèles cinématiques. Les modèles à vitesse constante (Constant Velocity – CV) et à accélération constante (Constant acceleration – CA) sont parmi les plus simples [38, 39]. Légèrement plus sophistiqués, les modèles à vitesse de rotation constante et à vitesse constante (Constant Turn Rate and Constant Velocity – CTRV), les modèles à vitesse de rotation constante et à accélération constante (Constant Turn Rate and Acceleration – CTRA) prennent en compte les contraintes de non-holonomie des véhicules de type voiture pour la réalisation de trajectoires courbées en introduisant les variables d'angle de lacet et de vitesse de

lacet dans le vecteur d'état du véhicule [40, 41].

3.1.1.2 Inférence

Les modèles cinématiques précédents peuvent servir de socle pour l'estimation de l'état sur la base d'observations bruitées. Dans [38], le modèle *Constant Velocity* est utilisé comme fonction de transition pour le véhicule dans le cadre d'un filtre de Kalman. Les bruits de dynamique et d'observation permettent de prendre en compte respectivement l'imperfection du modèle à représenter l'évolution de l'état du véhicule et l'incertitude induite par les mesures bruitées pour estimer l'état. En plus de l'hypothèse de linéarité du le modèle de transition, l'incertitude est représentée à l'aide d'une distribution normale. En répétant plusieurs fois l'étape de prédiction, on obtient un vecteur de moyenne et une matrice de covariance de la loi de prédiction assimilée à une Gaussienne de l'état de chaque véhicule pour chaque pas dans le futur. Il est possible d'étendre l'utilisation de ces modèles en leur appliquant des entrées (*e.g.*, les vitesses ou accélérations) échantillonnées aléatoirement. Ces simulations de Monte Carlo permettent l'utilisation de modèles non linéaires pouvant être ensuite utilisées pour la construction d'un ensemble de trajectoires possibles pour les véhicules de la scène routière [42].

Certaines approches étoffent les modèles basés sur la physique avec des information issues de cartes statiques. Lorsque les véhicules sont en mesure de se déplacer librement, certains travaux raisonnent sur une grille construite en prenant en compte la cartographie connue du segment routier traité [43, 44].

De par leur simplicité, les modèles cinématiques sont beaucoup plus populaires que les modèles dynamiques pour la prédiction de trajectoire. Ces derniers relient les forces et les moments des forces s'exerçant sur le véhicule à son mouvement. Cependant, dans [45], les auteurs modélisent les forces s'exerçant sur le véhicule dans un contexte de détection des collisions.

3.1.2 Modèles basés manœuvres

Les modèles basés sur les manœuvres peuvent conduire à de meilleures prédictions sur des sections de route spécifiques où les comportements des conducteurs dépendent de paramètres statiques (*e.g.*, la signalisation routière, la configuration de la route) sans besoin de modéliser les interactions entre les véhicules évoluant dans la scène. Les véhicules sont considérés comme des entités indépendantes qui effectuent une série d'opérations pour réaliser un mouvement voulu. Les manœuvres peuvent être également définies comme un « comportement » ou une « intention » [46].

La prédiction du mouvement avec les modèles basés manœuvres repose souvent sur une détection de l'intention du véhicule considéré suivie de la prédiction de la trajectoire future correspondant à la manœuvre entreprise. D'après [47], une façon de traiter la complexité induite par la

multitude de déplacements possibles pour le véhicule est de considérer un ensemble fini d'hypothèses de manœuvres discrètes. La distribution de probabilité postérieure sur les manœuvres peut servir ensuite de base à la prédiction du mouvement continu.

Le filtre à modèles multiples interactifs (Interacting Multiple Models filter – IMM filter) est une technique d'inférence souvent utilisée dans les applications de suivi où plusieurs modèles cinématiques sont considérés pour décrire l'objet en mouvement [48]. Les transitions entre les modes de fonctionnement sont décrites par une chaîne de Markov à temps discret dont les probabilités de transitions a priori sont données. La dynamique a priori du système conditionnellement aux modes est décrite par une banque de modèles d'état stochastiques partageant un vecteur d'état commun. Le filtre est dit « à modèles interactifs » dans le cas où les transitions entre les modes sont considérées à tout instant dans l'élaboration de la loi postérieure du vecteur d'état continu.

Le modèle représentant au mieux le déplacement du véhicule est estimé tout le long du filtrage. Quelques travaux font correspondre pour un véhicule, la probabilité a posteriori d'être dans un mode particulier à la probabilité d'être dans une manœuvre spécifique [49]. Les techniques reposant sur les filtres IMM souffrent d'un besoin important de réglage de paramètres pour obtenir des résultats d'inférence satisfaisants, *e.g.*, les probabilités a priori des sauts, les probabilités initiales des modes de fonctionnement, les bruits de dynamique et d'observation.

Les estimateurs d'intention ont souvent recours à des techniques de classification supervisée. En utilisant des données véhiculaires (positions, vitesses, orientations... etc.) et des informations sur le segment routier (géométrie, topologie, signalisation... etc.) de nombreux travaux cherchent à estimer la manœuvre entreprise par le véhicule d'intérêt. Des heuristiques définies manuellement sur le contexte routier peuvent être utilisés pour déterminer les manœuvres futures probables [50]. Une multitude de méthodes d'apprentissage ont été appliquées à la problématique d'estimation de manœuvre : les réseaux de neurones profond à propagation avant [51], les modèles de régression logistique [52], les machines à vecteur de pertinence (Relevance Vector Machines – RVM) [53] ou encore les machines à séparateurs vastes marges (Support-Vector Machine – SVM) [54] sont autant de techniques utilisées dans la littérature. Plus récemment, dans [55], les auteurs démontrent les bonnes performances de classification des manœuvres via un réseau de neurones récurrent alimenté avec des données LIDAR capturées sur une intersection. Dans l'ensemble, l'application d'algorithmes d'apprentissage supervisé pour la prédiction de l'intention des conducteurs constitue une solution immédiate et relativement simple à mettre en place. Cependant, selon les architectures et la complexité des manœuvres, les méthodes d'apprentissage requièrent une quantité de données conséquente (par rapport à leur nombre de paramètres) et nécessitent le réglage (ou « tuning ») de nombreux hyper-paramètres pour obtenir de bonnes performances.

Les Modèles de Markov caché (Hidden Markov Model – HHM) sont une alternative aux al-

gorithmes d'apprentissage et représentent une chaîne d'événements consécutifs correspondant à chaque manoeuvre possible. Les probabilités de transition entre les différents événements peuvent être apprises à partir de données, ainsi que le modèle d'observation. Pour une nouvelle séquence d'observations, l'intention de manœuvre est estimée en comparant la vraisemblance de chaque HMM vis à vis des observations, *i.e*, en comparant l'évaluation de la probabilité de la séquence des observations (ou « évidence ») pour chaque HMM en compétition [46, 56].

Une fois l'estimation de l'intention effectuée, il est nécessaire de faire correspondre la manœuvre détectée à une trajectoire continue future effectuée par le véhicule. Quand la détection de manœuvre est couplée à un modèle physique comme pour l'IMM, dans ce cas les futurs états du véhicule sont propagés en utilisant le modèle physique correspondant. Dans la même veine, la référence [57] fait correspondre une manœuvre spécifique à un HMM. Chaque HMM peut ensuite générer une trajectoire par le biais d'un modèle reposant sur un réseau bayésien dynamique. La trajectoire peut être générée de manière déterministe comme dans [58] ou par le biais de méthodes génératives probabilistes via des Processus Gaussiens (Gaussian Processes – GP) [59, 60] ou des arbres aléatoires d'exploration rapide (Rapidly-exploring Random Tree – RRT) [61, 62].

Certains travaux [63, 60] basées sur les Processus Gaussiens traitent conjointement le problème d'estimation de la manœuvre et de prédiction du mouvement. Ici le modèle repose sur l'hypothèse que les trajectoires des véhicules sont des fonctions issues de l'échantillonnage d'un GP. L'apprentissage consiste donc à faire correspondre une distribution Gaussienne sur ces fonctions via l'apprentissage d'une fonction noyau. Les principaux avantages des GPs sont leur robustesse au bruit présent dans les trajectoires observées et leur capacité à capturer les variations dans l'exécution d'un modèle de mouvement d'une manière probabiliste.

Même si les modèles basés manœuvre permettent de considérer des mouvements futurs sur un horizon temporel plus grand, ils ne prennent pas en compte directement les interactions entre les véhicules. Dans des segments routiers où l'interaction est forte, les performances de tels modèles se voient amputées si bien qu'ils risquent de produire des prédictions erronées pouvant nuire à la détection de situations dangereuses.

3.1.3 Modèles avec prise en compte de l'interaction

Les modèles présentés dans les deux sections précédentes ignorent les interactions entre les véhicules. Ils s'appuient uniquement sur des hypothèses plus ou moins fortes de manœuvres ou de trajectoires indépendantes du trafic avoisinant. Pourtant, le processus complexe de conduite, bien qu'individuel, est modulé par les déplacements et les comportements observés des autres véhicules situés autour du conducteur. Seuls les modèles prenant en compte les interactions, parfois subtiles, entre les mouvements de l'ego-véhicule et des véhicules qui l'entourent, peuvent permettre une

estimation et une prédiction fiables de l'état de la flotte et une évaluation du risque. L'incorporation d'autres considérations (nature de la route, mobilier urbain...) peut s'avérer pertinente.

3.1.3.1 Modèles simples analytiques

Il existe dans la littérature quelques modèles analytiques prenant en compte les interactions entre les véhicules pour la prédiction du mouvement. Le *modèle du conducteur intelligent* (Intelligent Driver Model – IDM) et ses dérivés peuvent être utilisés pour décrire l'évolution d'un véhicule par rapport au véhicule qui le précède [64]. Hors du domaine des ITS, les modèles basés sur les forces sociales permettent de modéliser le déplacement en interaction de deux piétons [65]. Ces modèles analytiques font l'hypothèse de conducteurs rationnels qui évitent le danger à tout prix. Dans [66], un modèle prenant en compte les commandes du conducteur est adopté. La prédiction sur les commandes est obtenue après la résolution d'un problème de contrôle optimal où le critère d'optimalité est défini manuellement pour prendre en compte les autres usagers de la route. Ici, la prédiction est restreinte à des scénarios de suivi entre véhicule sur une seule voie. Dans [67] et [68], les auteurs proposent de déterminer la probabilité de conflit entre deux véhicules effectuant une manœuvre choisie depuis un ensemble fini connu. Une fonction de coût établie manuellement pénalisant les comportements dangereux ou conflictuels permet de calculer les probabilités postérieures de manœuvres conflictuelles.

Ces modèles sont limités en termes de pouvoir de représentation, fastidieux à mettre en place et restreignent fortement la complexité des interactions envisagées. Ils sont donc à proscrire pour la prédiction du mouvement dans un environnement routier complexe où les interactions sont nombreuses et complexes à décrire formellement.

3.1.3.2 Modèles obtenus par apprentissage automatique

L'une des approches les plus populaires de la prédiction de mouvement pour les véhicules intelligents repose sur l'utilisation de modèles appris sur de larges bases de données de trajectoires acquises en conditions réelles. Au lieu de modéliser le déplacement des véhicules à l'aide d'un modèle analytique, il est possible d'apprendre ce modèle à l'aide d'un jeu de données d'apprentissage. Le mouvement est donc représenté par des approximateurs de fonction génériques comme les réseaux de neurones. Avec une quantité de données et un pouvoir de représentation suffisants, ces modèles peuvent représenter un grand nombre de déplacements en interaction. La majorité de ces techniques provient de la communauté scientifique de l'apprentissage automatique et de la vision par ordinateur avant d'avoir été adaptée à la problématique de la prédiction de mouvement. Parmi ces méthodes, nous pouvons distinguer : les modèles prenant en compte le caractère séquentiel des trajectoires et qui supposent que l'état des véhicules à un instant donné est dépendant des

états des véhicules aux instants précédents ; les modèles qui supposent que l'état à l'instant courant ne dépend pas des états précédents, c'est à dire sans imposer le besoin d'historique des états des véhicules pour construire l'état courant.

La plupart des approches non séquentielles sont basées sur le regroupement ou « clustering » de trajectoires observées. Ces différents groupes constituent ensuite un ensemble de trajectoires possibles pour les véhicules [69, 70]. Dans cette thèse, les modèles séquentiels sont préférés de par l'introduction dans leur fonctionnement d'un vecteur d'état (au sens de l'Automatique) plus ou moins explicite décrivant exhaustivement la flotte de véhicules à tout instant. Cette caractéristique donne les clés pour qualifier le fonctionnement interne du réseau vis à vis de la dynamique des véhicules.

Plusieurs méthodes séquentielles récentes utilisent des réseaux neuronaux récurrents pour la prédiction de séries temporelles correspondant aux trajectoires de véhicules en interaction. Cette génération s'effectue en supposant un système non Markovien où la loi de transition de l'état est conditionnée sur les états précédents jusqu'à un certain nombre d'instants passés.

Les approches basées sur les LSTMs sont les plus populaires pour la prédiction du déplacement des véhicules [71, 72, 73, 74]. Cependant, certains modèles adoptent un point de vue Markovien et reposent sur l'utilisation de réseaux non-récurrents. La référence [75] présente un modèle probabiliste hiérarchique basé sur un réseau à propagation avant pour déterminer le futur des véhicules ainsi que des réseaux auto-encodeurs variationnels conditionnels pour générer leurs trajectoires continues, le tout en interaction sur un segment autoroutier. Dans [76], les auteurs soutiennent qu'en raison des faibles dépendances temporelles dans les trajectoires d'un véhicule, un réseau de neurones à propagation avant peut être utilisé à la place de réseaux récurrents.

Dans [77], un unique réseau récurent apprend à générer sur plusieurs instants dans le futur les positions longitudinales et les vitesses latérales d'un véhicule cible. Ce réseau est alimenté par un ensemble de données véhiculaires (positions, vitesses, métriques de collision... etc.) provenant d'un véhicule cible et de son voisinage direct sur un segment autoroutier. Dans [72] les auteurs proposent une architecture basée sur plusieurs LSTMs, chacun apprenant à produire une distribution sur la position future de toute voiture par rapport à son historique et celle de ses voisins. Avec les mêmes données, un autre réseau récurrent se charge de générer les probabilités de manœuvre associées au véhicule cible. Une loi de type mélange sur le futur du véhicule peut ainsi être construite. La référence [71] propose un réseau récurent à base de LSTM qui apprend à prédire les transitions continues des véhicules indépendamment de leurs positions sur le segment routier. Chaque voiture est modélisée individuellement par un réseau LSTM. Les interactions sont prises en compte via l'interconnexion et le rassemblement des états cachés des réseaux LSTMs voisins dans un tenseur social. Initialement prévu pour prédictions de trajectoires piétonnes, ce modèle est également très

utilisé dans la littérature ITS comme étalon de comparaison. La construction du tenseur social peut être adaptée au domaine routier en modifiant la taille de la grille de façon que sa surface recouvre le segment routier considéré et que la taille des cases corresponde à la taille moyenne des véhicules observés.

Plusieurs améliorations du Social-LSTM ont été proposées dans la littérature. Dans [72], les auteurs combinent dans une architecture encodeur-décodeur la génération de trajectoires multimodales de leur précédents travaux [78] avec un tenseur à convolutions inspiré du Social-LSTM. Convoluer et rassembler les états cachés du voisinage des véhicules permet une meilleure généralisation du modèle appris et améliore les performances de prédiction sur des situations routières non observées durant l'apprentissage. La référence [73] s'appuie sur cette méthode pour y incorporer un mécanisme d'attention permettant de prendre en compte les interactions à la fois locales et non locales du véhicule cible. Ce dernier modèle, tout comme pour le CS-LSTM, délivre en sortie une prédiction multimodale capturant explicitement les différentes intentions ou manœuvres des conducteurs.

Les capacités des réseaux récurrents pour la prédiction de séries temporelles peuvent être également mises en place en combinaison avec des architectures neuronales différentes. Dans [79] l'apprentissage des trajectoires est couplé avec l'apprentissage de variables latentes discrètes permettant depuis des données continues non labélisées d'apprendre un ensemble de manœuvres possibles. Dans un même objectif d'interprétabilité, [80] propose une architecture basée sur des réseaux récurrents et un auto-encodeur variationnel conditionnel pour la prédiction de trajectoire en interaction sur rond-point. Comme pour [79], les variables latentes ainsi apprises permettent de structurer la génération des trajectoires afin de mieux l'expliquer. Plusieurs travaux [81, 73] profitent également du pouvoir de représentation des réseaux convolutifs pour introduire et traiter des données issues de la scène routière globale en plus des données trajectoires. Plus récemment, certains travaux utilisent une architecture à base de réseaux de neurones à graphes [82] pour représenter plus fidèlement les interactions entre les véhicules et/ou les autres éléments de la scène routière. Ces réseaux permettent de réaliser de l'inférence sur des données structurées sous la forme de graphe. Dans [74], les auteurs utilisent un graphe pour représenter les relations entre un véhicule cible et les autres véhicules et objet composant la scène. Dans [83], les réseaux de neurones à graphes permettent d'apprendre à prédire le déplacement des véhicules avec des données sémantiques de la scène routière. La représentation en graphe dote cette méthode d'une certaine généricité quant aux scènes routières pouvant être considérées. La référence [84] exploite les données de carte sémantique pour prédire le déplacement de plusieurs agents (piétons, véhicules... etc.) sur la route. Leur réseau à graphes opère directement sur des cartes vectorielles sémantiques modélisant ainsi les interactions entre les agents selon leur proximité sur la portion de route considérée.

Dans les modèles basés sur l'apprentissage, les fonctions de prédiction apprises dépendent plus ou moins du jeu de données fourni pendant l'entraînement. Si par exemple, les données ne contiennent pas d'instance de comportement dangereux ou antagoniste, alors les modèles ainsi créés peuvent faire par construction l'hypothèse que les véhicules observés se déplacent de manière rationnelle et collaborative. Certains jeux de données intègrent des instances de trajectoires conflictuelles pouvant être utilisées pour mieux généraliser les algorithmes proposés aux situations routières dangereuses [85].

3.1.3.3 Modèles basés sur les Réseaux Bayésiens Dynamiques

La scène routière peut être modélisée à l'aide d'une réseau Bayésien dynamique (Dynamic Bayesian Network – DBN). Dans la littérature des ITS, la flexibilité de représentation des DBNs est utilisée pour représenter les interactions entres les véhicules en mouvement. Dans [86], les auteurs utilisent un DBN pour modéliser les manœuvres et les trajectoires des conducteurs sur autoroute ainsi que leurs lien avec les observations. Une série de variables aléatoires représente le contexte d'un véhicule cible, ce contexte incluant l'état de la flotte de véhicules présents dans la scène. Le DBN décrit comment le contexte est relié à la génération des trajectoires. L'inférence est effectuée à l'aide d'un filtre particulaire : il s'agit d'estimer l'état de tout véhicule sur la base d'observations bruitées des véhicules de la scène routière. Après l'estimation, le DBN peut être ensuite utilisé récursivement sans intégration de mesure pour réaliser la prédiction. Lors de travaux plus récents [87], les auteurs proposent de remplacer la construction manuelle des distributions de probabilité associées aux variables aléatoires du DBN par leur apprentissage depuis les données via l'algorithme d'Espérance-Maximisation (Expectation-Maximization – EM) [88]. Les algorithmes proposés par [87] supposent pour chaque instant que la manœuvre et la trajectoire prédites de chaque véhicule sont indépendantes de celles inférées pour les autres véhicules au même instant. Dans [89] cette restriction est levée et le mouvement d'un véhicule est dépendant à tout moment du mouvement et des intentions du trafic environnant. L'inférence sur ce modèle est réalisée via des méthodes variationnelles pour suivre des camions en interaction dans une mine à ciel ouvert à partir de données GPS. Un DBN a également été proposé pour modéliser le déplacement des véhicules lors d'intersections [90]. Contrairement aux approches précédentes, ce travail se concentre sur la détection de situations dangereuses.

Plus récemment, [91] a recours à un DBN pour modéliser les différentes relations entre les conducteurs de la scène et leur dépendance sur la génération des trajectoires. Les auteurs proposent une méthode pour estimer en ligne les hypothèses de trajet en plus des hypothèses de manœuvres pour les véhicules en interaction dans des scénarios routiers génériques. La référence [92] améliore l'efficacité de leur mécanisme d'inférence initialement construit avec un filtre particulaire au moyen

d'un filtre de Kalman « unscented » multi-modèles. Enfin dans [93], les auteurs intègrent dans leur DBN un modèle d'actions appris par des techniques d'apprentissage automatique réduisant ainsi le besoin de construire manuellement certaines dépendances entre le contexte routier interactif et les trajectoires des véhicules.

Les DBNs constituent un paradigme flexible pour modéliser les véhicules en interaction et prédire leur déplacement. Il permettent de modéliser la scène en faisant intervenir des variables de modélisation continues et/ou discrètes provenant d'une connaissance experte ou éventuellement d'un apprentissage. Contrairement aux approches purement à base de techniques d'apprentissage automatique, il est possible d'incorporer directement dans le modèle la prise en compte de la dangerosité de la situation via des variables additionnelles. Cependant, l'inférence sur ces modèles est en général coûteuse en termes de calcul, en particulier dans des situations de trafic complexes avec de nombreux véhicules en interaction. Le choix de la structure d'un DBN peut être également rédhibitoire dans des scénarios routiers ou le nombre et la variété des interactions en jeu sont importants.

3.1.3.4 Modèles basés sur la planification

Les modèles basés sur la planification reposent sur l'hypothèse explicite de rationalité du comportement des conducteurs, contrairement aux modèles basés sur l'apprentissage automatique qui déduisent cette caractéristique implicitement depuis les données observées. Les approches basées sur la planification suivent le paradigme *Percevoir-Raisonner-Agir* souvent rencontré dans la littérature de la robotique. Cette hypothèse impose la prise en compte dans les modèles de l'impact des actions du conducteur sur la scène routière entière. Par conséquent, la plupart des modèles proposés dans la littérature utilise des fonctions objectifs qui minimisent un coût total calculé sur une séquence de mouvements pour la génération de la trajectoire. Il est possible d'introduire dans ce coût la prise en compte des autres conducteurs et de l'environnement. Deux approches sont distinguées dans la littérature. Les approches basées sur la planification directe utilisent une fonction de coût prédéfinie pour prédire le déplacement des véhicules. La majorité des approches considère une planification coopérative dans un espace d'état commun qui inclut tous les agents [94, 47]. Les méthodes de planification inverse supposent que le coût, dépendant du contexte et des choix du conducteur vis-à-vis de celui-ci, peut être appris à partir d'observations. Elles diffèrent des modèles reposant purement sur de l'apprentissage automatique de par la modélisation explicite d'une fonction de coût représentant l'impact positif ou négatif du déplacement des conducteurs dans la scène. La référence [95] formule le problème de prédiction du mouvement comme un problème d'optimisation dont la fonction de coût dépend de trajectoires futures multimodales capturant les interactions entre véhicules et les caractéristiques de l'environnement routier. Cette fonction de coût

est apprise par renforcement inverse via un réseau de neurones récurent [96]. Dans [97], les auteurs proposent un modèle probabiliste basé sur l'apprentissage par renforcement inverse hiérarchique pour la prédiction de trajectoires sur une rampe d'accès autoroutière. Une hiérarchie entre les variables aléatoires mises en jeu permet de modéliser explicitement l'impact du choix de manœuvre des conducteurs dans la réalisation de leurs futures trajectoires continues.

3.2 Évaluation du risque

La notion même de risque n'est pas un concept clairement défini et dépend souvent de l'utilisateur final du système d'évaluation du risque [36]. Dans le cas des véhicules intelligents, toute situation susceptible de nuire au conducteur est définie comme dangereuse. Un nombre important d'articles se concentre uniquement sur la détection des collisions futures, considérant qu'il s'agit de la principale source de risque pour les conducteurs. Dans cette thèse, nous considérons également les collisions comme le seul risque devant être évalué. L'évaluation est le plus souvent composée de deux étapes : une première étape de prédiction des trajectoires en interaction ; puis une détection des collisions depuis toutes les paires de trajectoires générées.

Le risque de collision peut être binaire. Les collisions sont détectées par comparaison à un seuil sur la distance entre deux points de deux trajectoires au même instant [98]. Afin de prendre en compte la forme des véhicules, ce seuil peut être remplacé par une condition sur le chevauchement entre les formes des deux véhicules [45, 99, 100]. Lorsqu'on dispose de la caractérisation probabiliste des incertitudes associées aux états prédits, des calculs d'intersections entre deux ellipses de confiance permettent de déterminer un risque de collision entre deux véhicules [38, 101] associé à une certaine probabilité [42, 50]. Lorsque le mouvement futur d'un véhicule est représenté par une distribution de probabilité sur des trajectoires, il est alors possible de calculer le risque comme la probabilité d'une collision en calculant la somme de toutes les paires de trajectoires prédites comme entrantes en collision avec un véhicule cible [63]. Enfin, il existe des indicateurs de risque indirects pouvant renseigner sur la criticité des collisions possibles entre les véhicules [100]. Il existe dans la littérature des définitions plus larges du risque. Il s'agit par exemple de détecter les manœuvres conflictuelles ou la violation des règles de conduite sans nécessairement modéliser le mouvement de la voiture [102]. Certaines de ces approches reposent sur des techniques d'apprentissage [103, 46].

3.3 Les jeux de données

Comme expliqué dans la partie précédente, de nombreux modèles de prédiction reposent sur le traitement massif de trajectoires véhiculaires. Afin d'évaluer la qualité des prédictions, les états

ou distributions prédits sont généralement comparés à une vérité terrain correspondant à un sous-ensemble des données utilisées pour la construction des modèles. Cette section décrit les principaux jeux de données sur voie publique de trajectoires véhiculaires. S'inscrivant directement dans le contexte de cette thèse, les caractéristiques de deux enregistrements effectués sur autoroutes sont détaillés.

Dans la plupart des enregistrements, les mesures peuvent être obtenues à l'aide d'une caméra vidéo statique capturant la scène depuis un point haut et une vue orientée vers le sol. On retrouve également dans la littérature des lasers ou des capteurs de profondeur amarrés à un véhicule se déplaçant sur le segment routier d'intérêt. Les véhicules détectés dans chaque « image » sont étiquetés par leur identifiant (ID) unique et leur position dans un repère global monde. Ces enregistrements sont également horodatés. Les coordonnées sont souvent complétées par des informations sur l'orientation et la vitesse des véhicules. De plus, des informations supplémentaires comme un étiquetage des manœuvres, la mesure du temps de collision entre les véhicules et d'autres indices contextuels peuvent être fournis. Outre les données brutes, afin d'évaluer (ou d'apprendre) correctement les modèles de prédiction, il est important de connaître le contexte routier dans lequel l'enregistrement a été effectué. Le type de segment routier, le lieu, l'heure et la durée de l'enregistrement, les variations de congestion du segment routier, la présence d'obstacle ou encore la disponibilité d'une carte sémantique sont autant d'informations utiles.

3.3.1 Point de vue statique

3.3.1.1 NGSIM

NGSIM est un jeu de données très utilisé pour la recherche sur l'analyse du trafic et le comportement des conducteurs [104]. Il a été réalisé à partir d'enregistrements vidéo de caméras fixes placées sur divers segments routiers des États-Unis. NGSIM contient quatre lieux d'enregistrement différents : deux enregistrements autoroutiers sur l'autoroute 101 et l'*interstate* 80. Les deux autres jeux de données enregistrent des données véhiculaires, en milieu urbain et dans une zone de sélection de rampe d'accès/sortie sur l'autoroute, respectivement.

Sur chaque site autoroutier, plusieurs caméras vidéo synchronisées ont été installées au sommet de plusieurs immeubles adjacents afin d'enregistrer les différents segments de route. Les points de vue des caméras se chevauchent et couvrent entre 500 et 640m de route. L'alignement des images des caméras inclinées a été réalisé pour recouvrir l'ensemble de la zone d'étude. La totalité des enregistrements a une durée de 90 minutes pour plus de 9000 véhicules capturés. L'autoroute 101 et l'*interstate* 80 sont respectivement dotées de 6 et 5 voies de déplacement dans la même direction. Dans NGSIM, la vitesse moyenne des véhicules ne dépasse pas 75km/h et témoigne d'un segment

autoroutier congestionné.

3.3.1.2 HighD

Plusieurs travaux démontrent l'impossibilité d'utiliser les données NGSIM en l'état pour une analyse fine du déplacement des véhicules [105]. En effet, le bruit induit par la qualité des enregistrements utilisés ainsi que les algorithmes de segmentation utilisés entraîne des faux positifs de collisions entre véhicules ainsi que des vitesses et des accélérations physiquement impossibles. Un exemple de mauvaise annotation depuis une trame vidéo de NGSIM est illustré dans la Figure 3.2. HighD fournit un ensemble de données à grande échelle de trajectoires véhiculaires sur six portions d'autoroute allemande, enregistrées par des caméras aériennes à très haute définition [106]. La position, la vitesse, l'accélération et les caractéristiques physiques des véhicules sont calculées à partir d'une segmentation vidéo réalisée par des méthodes de pointe de la littérature. Les véhicules sont visibles pendant environ 14 secondes sur un segment d'autoroute d'environ 420 mètres. Les données sont enregistrées à une fréquence de 25 images par seconde pendant une durée totale de plus de 16 heures. 110000 véhicules sont ainsi capturés sur l'ensemble du jeu de données. La vitesse moyenne des véhicules est répartie selon une distribution à deux modes, 85 et 120km/h, correspondant respectivement la vitesse typique des camions et des voitures sur un segment autoroutier fluide. Une partie minoritaire des données de HighD a été enregistrée lors de bouchons routiers. HighD offre une précision importante sur la position des véhicules avec une erreur moyenne sur la position du véhicule de moins de 3cm par rapport à des données annotées manuellement. D'un point de vue de l'interaction entre les véhicules, 11000 changements de voies sont enregistrés. Par rapport à NGSIM, le taux de changement de voie est plus de quatre fois plus grand avec 0.45 et 0.10 changement de voie par véhicule pour HighD et NGSIM respectivement. Pour toutes les caractéristiques évoquées dans cette partie, ce jeu de données sera utilisé dans les Chapitres 4 et 5 pour l'apprentissage de notre modèle de dynamique ainsi que pour l'évaluation globale de notre approche.

3.3.1.3 Les autres jeux de données

Il existe d'autres jeu de données « en vue de haut » dans des configurations de conduite plus urbaine ou péri-urbaine. Comme évoqué précédemment, NGSIM contient un autre jeu de données enregistré sur un boulevard californien. Cette configuration comprend une série d'intersections entre plusieurs voies toutes signalées par des feux tricolores. Enregistrés depuis une méthode similaire à celle du jeu de données HighD, RounD [107] et InD [108] sont deux jeux de données véhiculaires provenant de rond-points péri-urbains et d'intersections urbaines. En plus des catégories de véhicules habituels, les trajectoires des usagers vulnérables de la route sont comprises

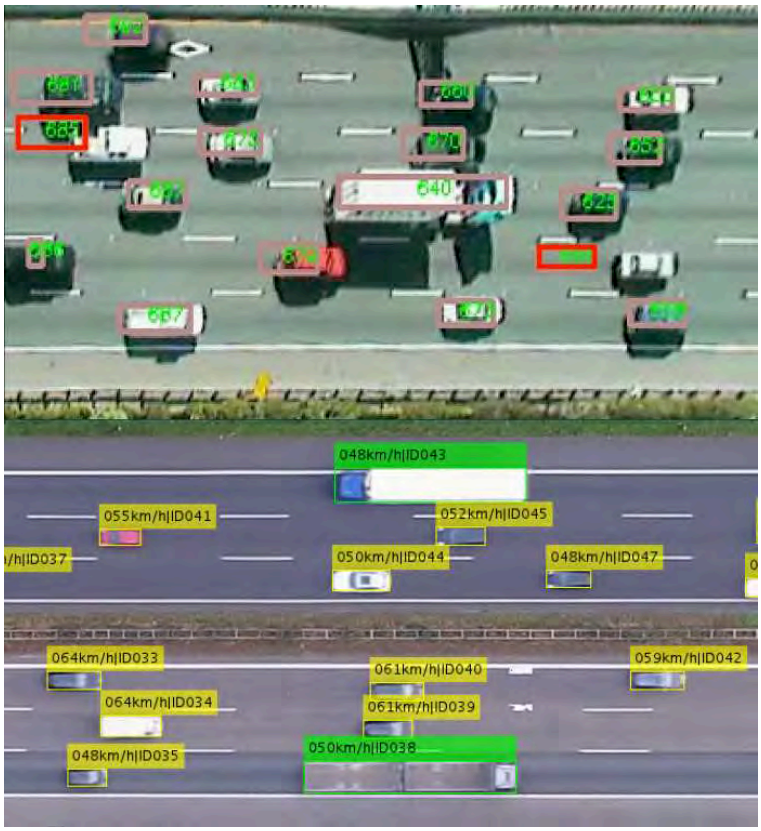


FIGURE 3.2 – (Haut) Les boîtes englobantes de NGSIM recouvrent rarement les véhicules observés. Deux boîtes rouges englobent exclusivement la route. (Bas) Sur cette trame, les boîtes englobantes de HighD recouvrent entièrement les véhicules.

dans les données fournies. Le jeu de données INTERACTION [85] propose des données issues de ronds-points péri-urbains, de rampes d'accès et d'intersections. L'accent ici est mis sur la présence d'un nombre important de trajectoires issues d'une coopération ou d'un conflit évident entre les véhicules évoluant dans la scène. Les auteurs fournissent également des cartes sémantiques haute définition (*HD-maps*) permettant l'ajout d'information haut-niveau sur l'environnement routier.

3.3.2 Point de vue mobile

L'avènement de la recherche sur les systèmes de transport autonomes et le recours massif aux techniques d'apprentissage profond a initié la création d'un nombre important de jeux de données capturés depuis des capteurs embarqués sur un véhicule. Les capteurs utilisés sont souvent une combinaison d'un GPS, de caméras et de scanners laser. Des capteurs proprioceptifs comme la centrale inertielle peuvent être également utilisés. La Figure 3.3 donne un exemple de configuration de capteurs sur un véhicule standard. Dans le paragraphe suivant, un éventail non exhaustif de jeux de données est présenté.

KITTI [109] est un jeu de données précurseur dans la littérature. Il a été utilisé dans de nombreux travaux de recherche sur des applications de conduite autonome, de vision par ordinateur, de détection d'objets ou encore de pistage. Cityscapes [110] puis ApolloScape [111] proposent des jeux de données large échelle avec des données similaires à celle fourni par KITTI. De plus, l'environnement de conduite des véhicules enregistrés est annoté. Pour chaque trame vidéo, l'utilisateur peut extraire par exemple la position des marqueurs de voies, la position du trottoir ou encore le type de véhicules observés. Plus récemment, l'utilisation de caméras 360 degrés permet une meilleure densité des données enregistrées [112]. Dans [112], les auteurs proposent des outils informatiques pour faciliter la mise en relation de cartes sémantiques de la route avec les données obtenues par les capteurs.

FIGURE 3.3 – Configuration de capteurs utilisée dans [109]. Ce véhicule standard comprend un banc stéréo de caméras, un laser et un GPS couplé à une centrale inertielle.

CHAPITRE 4
UNE MÉTHODE DE PRÉDICTION HYBRIDE MÊLANT ESTIMATION STOCHASTIQUE ET APPRENTISSAGE AUTOMATIQUE SUPERVISÉ

4.1 Introduction

Comme en témoigne le grand nombre d'approches présentées dans le Chapitre 3, la prédiction du mouvement de véhicules en interaction est un sujet majeur de la littérature des ITS. Les solutions mises en place pour répondre à cette problématique visent à améliorer la sécurité de la conduite à travers une meilleure compréhension de la scène routière.

Il est difficile de construire manuellement des modèles décrivant le futur de véhicules en interaction étant donnés leurs états courants ou les historiques de leurs états. La complexité de tels modèles est en effet exponentielle en le nombre de manœuvres admissibles de chaque véhicule.

De nombreuses approches récentes reposent sur l'apprentissage d'un modèle de prédiction depuis une quantité importante de données de trajectoires véhiculaires. Ces modèles sont constitués d'approximateurs génériques à base de réseaux de neurones récurrents. Bien qu'ils puissent donner lieu à des inférences très performantes, ils n'en demeurent pas moins difficilement interprétables : ils n'encodent pas explicitement un vecteur d'état et son lien aux données via une loi d'observation comme le font les modèles standard du traitement du signal et de l'automatique ; les trajectoires prédites sont générées de bout en bout au sein d'un processus complexe caché, qui, de plus, dépend fortement de l'étape d'apprentissage ; la confiance pouvant être placée dans ces prédictions est difficile à caractériser lorsque les données véhiculaires sont entachées d'un bruit d'observation dont les statistiques ne sont dévoilées qu'au moment de leur assimilation (*e.g.*, lorsque ces données sont des localisations absolues individuelles obtenues par fusion de modalités sensorielles embarquées sujettes à des défaillances). A contrario, les modèles basés sur les réseaux Bayésiens dynamiques (Dynamic Bayes Networks – DBNs) exhibent naturellement un vecteur d'état ainsi qu'un modèle d'observation. Cependant, les dépendances statistiques entre les variables aléatoires mises en jeu en vue de modéliser la scène routière doivent pour la plupart être encodées manuellement sur la base d'une connaissance experte.

Bien que la littérature fasse état de méthodes de prédiction dotées de performances impressionnantes sur divers cas d'étude, peu d'approches nous semblent adresser le nécessaire compromis entre puissance, flexibilité de modélisation et garanties d'interprétabilité, d'explicabilité et de consistance mathématique. Ce chapitre tente de répondre à cette problématique en définissant une approche unifiée pour l'estimation puis la prédiction de l'état d'une flotte de véhicules en inter-

action. À l'interface entre l'apprentissage automatique et le filtrage stochastique, celle-ci combine plusieurs éléments complémentaires : (1) une représentation d'état stochastique impliquant un vecteur d'état continu dont le sens est intuitif ainsi qu'un modèle d'observation exprimant le lien statique qui le relie à des mesures bruitées ; (2) un modèle de dynamique a priori Markovien établi par des méthodes d'apprentissage automatique, qui, contrairement à un modèle défini manuellement, permet la description des futurs possibles de la flotte tout en capturant les interactions entre véhicules ; (3) une inférence pour l'estimation de l'état de la flotte par deux algorithmes interchangeables de filtrage stochastique adaptés aux espaces de très grande dimension mis en jeu, qui ne pourraient pas être adressés par des techniques conventionnelles. La densité de probabilité postérieure obtenue conditionnellement aux observations ainsi que la loi de dynamique a priori constituent naturellement les ingrédients de la prédiction de l'état de la flotte à moyen terme. Ce chapitre détaille l'ensemble de ces éléments. Il se termine par l'implémentation et l'évaluation de l'approche sur une série de scénarios autoroutiers.

4.1.1 Description détaillée du contexte

Dans cette étude, une flotte de véhicules connectés à un serveur distant de type « Cloud » est considérée. À des intervalles de temps réguliers, chacun d'eux détermine sa position absolue sur la base de ses capteurs embarqués, l'étiquette par son identifiant, puis transmet le tout au serveur. Ces remontées synchrones de données individuelles peuvent être accompagnées des statistiques du bruit de localisation si celles-ci sont établies au fil de l'eau. Disposant de trajectoires de mesures bruitées étiquetées, le serveur doit estimer l'état de la flotte des véhicules. Cette estimation n'implique de traiter aucun problème d'association de données. Pour simplifier, on admet également l'absence de fausses mesures [1]. Sur la base de cette estimation, les mouvements des véhicules sont prédits à moyen terme par le serveur Cloud (sur un horizon de quelques secondes). Les collisions potentielles sont alors caractérisées puis renvoyées aux conducteurs concernés afin qu'ils puissent adapter leur comportement. Comme indiqué précédemment, pour mener à bien sa tâche, le serveur Cloud doit être en mesure de capturer les intentions sous-tendant les manœuvres des conducteurs. Ces processus sont complexes et dépendent d'interactions subtiles entre les véhicules et l'environnement. La dépendance à l'environnement est minimisée ici, en considérant une section d'autoroute droite à trois voies, sans rampe d'accès, sans intersection, sans panneau de signalisation.

1. Les fausses mesures pourraient toutefois être traitées par les méthodes classiques de la littérature couplées aux techniques de filtrage stochastique.

4.2 Schéma global

Les approches proposées dans cette étude sont scindées en deux parties complémentaires : une partie hors ligne qui consiste principalement en l'apprentissage de la dynamique a priori du vecteur d'état caractérisant la flotte de véhicules via un jeu de trajectoires routières ; une partie en ligne réalisant l'estimation et la prédiction de ce vecteur d'état sur la base de mesures bruitées. C'est également en ligne qu'est effectuée la caractérisation du risque de collisions. Ces considérations sont résumées dans la Figure 4.1.

4.2.1 Partie hors ligne

La partie hors ligne est donc principalement constituée de l'apprentissage supervisé de la dynamique a priori des véhicules en interaction. Pour ce faire, un jeu de données labellisées est préalablement construit, via une segmentation guidée par une connaissance experte. Ce jeu sert de base à l'entraînement d'un ensemble de réseaux de neurones, dont le nombre dépend du nombre de manœuvres identifiées lors de la segmentation. Pour notre problème, quatre réseaux sont entraînés depuis un jeu de données grande échelle de trajectoires autoroutières.

4.2.2 Partie en ligne

C'est dans la partie en ligne que se trouve l'estimation puis la prédiction du vecteur d'état de la flotte. Un estimation initiale de ce vecteur est établie par un filtre stochastique information. Partant de cette initialisation, un algorithme de filtrage stochastique permet, par assimilation récursive des observations, l'estimation d'une approximation de la loi postérieure du vecteur d'état. Le modèle de dynamique a priori pré-appris hors ligne intervient naturellement dans le processus. L'estimation s'effectue sur plusieurs instants, entre 3 et 5 secondes pour notre cas. Disposant de l'approximation de la loi (postérieure) de filtrage à la fin de l'horizon d'estimation, la loi de prédiction à moyen terme du vecteur d'état de la flotte est établie, typiquement sur un horizon de l'ordre de 3 secondes. C'est depuis cette prédiction qu'une évaluation du risque de collision peut être envisagée.

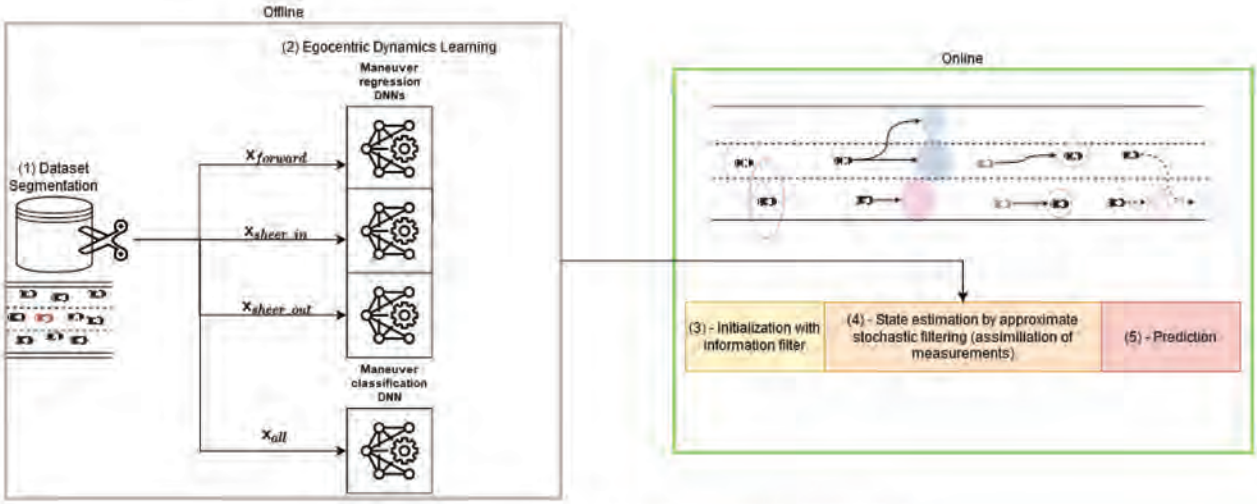


FIGURE 4.1 – Schéma global de l'approche.

4.3 Modélisation

Cette section présente le modèle d'état stochastique de la flotte de véhicules en interaction ainsi que les hypothèses associées.

4.3.1 Un vecteur d'état Markovien

Une flotte $\mathcal{V}$ composée de K véhicules $\mathcal{V}_1, \ldots, \mathcal{V}_K$ est supposée évoluer sur le segment autoroutier considéré. Les intentions des conducteurs dans $\mathcal{V}$ peuvent être grossièrement étiquetées en termes de manœuvres telles que : *conduite_libre*, *suivi*, *freinage*, *dépassement*, etc. À tout instant $n \in \{0, \ldots, N\}$, un vecteur d'état caché $x_n = \left((x_n^1)^T, \ldots, (x_n^K)^T\right)^T$ est défini, avec $.^T$ l'opérateur de transposition. Ce vecteur admet une structure « par véhicule », au sens où tout sous-vecteur x_n^k contient le même type d'information sur $\mathcal{V}_k$. Il est important de noter que x_n doit être suffisamment riche pour résumer correctement le statut de la flotte $\mathcal{V}_1, \ldots, \mathcal{V}_K$ à l'instant n et assurer que le processus aléatoire $x_{0:n} = x_0, \ldots, x_n$ est Markovien de loi de dynamique a priori $p(x_n|x_{n-1})$. La définition provisoire de x_n et l'énoncé de la problématique sont les suivantes.

Définition 1 $x_n = \left((x_n^1)^T, \ldots, (x_n^K)^T\right)^T$ *est à valeurs continues. Chaque sous-vecteur* $x_n^k \in \mathbb{R}^4$ *est constitué de (dans l'ordre) : la position longitudinale (selon l'axe* x*) du véhicule* $\mathcal{V}_k$*; la vitesse longitudinale de* $\mathcal{V}_k$*; la position latérale (selon l'axe* y*) de* $\mathcal{V}_k$*; la vitesse latérale de* $\mathcal{V}_k$*.*

Les dimensions (constantes) W^k et H^k du k ème véhicule selon x et y, respectivement, sont supposées connues.

Définition 2 (Énoncé du problème) *Étant donnée la séquence $z_{0:n}$ des mesures (bruitées) des positions des véhicules, déterminer la densité de filtrage $p(x_n|z_{0:n})$—de moyenne et covariance a posteriori $x_{n|n}, P_{n|n}$—du vecteur d'état de l'ensemble de la flotte $\mathcal{V}$. Ensuite, établir les lois de prédiction $p(x_{n+m}|z_{0:n})$—de moments $x_{n+m|n}, P_{n+m|n}$— aux instants $n+m$, ou les lois $p(z_{n+m}|z_{0:n})$ de prédiction à m pas—de de moments $z_{n+m|n}, S_{n+m|n}$—des observations. Sur la base de ces densités de probabilité,* e.g.*, à partir de leurs ensembles de confiance approximatifs, évaluer le risque de collisions.*

4.3.1.1 Définition mathématique

Les hypothèses suivantes sont en vigueur dans la définition de la loi de dynamique a priori $p(x_n|x_{n-1})$, $n \geq 1$.

Hypothèse 1 *Conditionnellement au vecteur d'état décrivant la flotte à l'instant précédent, tous les sous-vecteurs décrivant les états de chaque véhicule à l'instant courant sont mutuellement indépendants,* i.e.*,*

$$\forall n \geq 1,\ p(x_n|x_{n-1}) = \prod_{k=1}^{k=K} p(x_n^k|x_{n-1}). \tag{4.1}$$

Tout véhicule cible $\mathcal{V}_k$ est désormais désigné par « égo-vehicule ». Sa densité (ou loi) de dynamique a priori $p(x_n^k|x_{n-1})$ est nommée « modèle de transition égocentrique ».

Hypothèse 2 *Le modèle de transition égocentrique $p(x_n^k|x_{n-1})$ entre les instants $n-1$ et n dépend uniquement du sous-vecteur d'état x_{n-1}^k associé au véhicule $\mathcal{V}_k$ à l'instant $n-1$ et des sous-vecteurs d'état décrivant l'ensemble $\mathcal{N}_{n-1}(\mathcal{V}_k)$ des voisins de $\mathcal{V}_k$ à l'instant $n-1$. De manière équivalente, $p(x_n^k|x_{n-1})$ s'écrit comme*

$$p(x_n^k|x_{n-1}) = p\big(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}\big). \tag{4.2}$$

Hypothèse 3 *Le modèle de transition égocentrique $p\big(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}\big)$ est indépendant de l'instant n. Le voisinage $\mathcal{N}_{n-1}(\mathcal{V}_k)$ de l'égo-véhicule à l'instant précédent est lui-même défini comme une fonction du type de $\mathcal{V}_k$.*

Ces hypothèses sont peu restrictives et permettent de capturer une large famille d'interactions entre les véhicules.

Un modèle à mélange de Gaussiennes (Gaussian Mixture Model – GMM) est sélectionné pour la loi de transition égocentrique du vecteur d'état continu défini dans l'Hypothèse 2. En effet, les GMMs approximent une grande famille de densités de probabilité [113]. Nous conjecturons

que leur polyvalence et leur propriété de multimodalité permettent de capturer la diversité intra- et extra-manœuvre inhérente au déplacement de véhicules routiers. La manœuvre (ou intention) engagée par tout conducteur constitue la première source de diversité à prendre en compte. Une seconde source de variabilité provient de l'incertitude dans le déplacement continu d'un véhicule conditionnellement à la manœuvre engagée. Autrement dit, la séquence d'actions mise en place par tout conducteur pour déplacer son véhicule selon la manœuvre qu'il a choisie est sujette à des fluctuations dépendantes du conducteur lui-même et de l'environnement routier dans lequel il est immergé.

Hypothèse 4 *La loi de dynamique a priori égocentrique est définie par*

$$p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}) := \sum_{m=1}^{M} \lambda_m(x_{n-1}^k, \diamond)\mathcal{G}(x_n^k; \mu_m(x_{n-1}^k, \diamond), \Sigma_m(x_{n-1}^k, \diamond)), \tag{4.3}$$

$$\textit{où } \diamond = \{x_{n-1}^l\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}, \tag{4.4}$$

$$\textit{et } \sum_{m=1}^{M} = 1$$

et $\mathcal{G}(x; \mu, \Sigma)$ désigne la densité de probabilité Gaussienne multivariée en le vecteur x, de moyenne μ et de matrice de covariance Σ. Chaque $m^{\text{ième}}$ hypothèse de (4.3)*, $m \in \{1, \ldots, M\}$, peut être vue comme la manœuvre engagée par l'égo-véhicule entre l'instant précédent $n-1$ et l'instant présent n lorsque son état précédent et l'état précédent de ses voisins sont respectivement x_{n-1}^k et $\{x_{n-1}^l\}$. Ces états constituent naturellement des paramètres du poids $\lambda_m \in [0; 1]$, du vecteur moyenne $\mu_m \in \mathbb{R}^4$ et de la matrice de covariance $\Sigma_m \in \mathbb{R}^{4\times4}$ associés à la $m^{\text{ième}}$ hypothèse.*

Hypothèse 5 *Selon la capacité de modélisation souhaitée, les matrices de covariance $\{\Sigma_m\}_{m\in\{1,\ldots,M\}}$ sont soit diagonales, soit pleines,* i.e.*, (avec $\diamond$ défini ci-dessus)*

$$\Sigma_m(\diamond) := \mathrm{diag}\big(\sigma_{m,1}^2(x_{n-1}^k, \diamond), \ldots, \sigma_{m,4}^2(x_{n-1}^k, \diamond)\big) \tag{4.5}$$

$$\textit{ou bien } \Sigma_m(\diamond) := L_m(x_{n-1}^k, \diamond)L_m^T(x_{n-1}^k, \diamond). \tag{4.6}$$

Par conséquent, pour chaque m dans $\{1, \ldots, M\}$, la dépendance aux arguments de la matrice de covariance Σ_m est modélisée soit par les 4 écarts types $\sigma_{m,1}, \ldots, \sigma_{m,4}$, soit par les 4 composantes strictement positives $\{l_{m,ii}\}_{i=1,\ldots,4}$ et les 6 composantes non-nulles $\{l_{m,ij}\}_{i>j}$ de la racine de Cholesky $L_m \in \mathbb{R}^{4\times4}$ (triangulaire inférieure) de Σ_m (supposée définie positive).

4.3.2 Le modèle de mesure

Pour les mesures définies plus haut, les hypothèses suivantes s'appliquent.

Définition 3 *Le vecteur aléatoire d'observation* $z_n = \left((z_n^1)^T, \ldots, (z_n^K)^T\right)^T$ *possède également une structure « par véhicule ». Chaque sous-vecteur* z_n^k *est constitué des mesures des positions du véhicule* $\mathcal{V}_k$ *selon les axes* x *et* y*. Il est entaché d'un bruit additif Gaussien de moyenne nulle et de matrice de covariance* $R_n^k \in \mathbb{R}^{2\times 2}$ *connue (possiblement variante dans le temps et dépendante du véhicule)*[2].

Hypothèse 6 *La loi d'observation* $p(z_n|x_n)$ *présente une structure « par véhicule » et les bruits qui entâchent les mesures liées à des véhicules distincts sont mutuellement indépendants,* i.e.*,*

$$p(z_n|x_n) := \prod_{k=1}^{k=K} p(z_n^k|x_n^k) \tag{4.7}$$

$$\textit{avec } p(z_n^k|x_n^k) := \mathcal{G}(z_n^k; H_n^k x_n^k, R_n^k),\ H_n^k = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right). \tag{4.8}$$

4.4 Apprentissage hors ligne supervisé de la loi de dynamique a priori égocentrique

Les réseaux à densité de mélanges (Mixture Density Neural Networks – MDN) constituent une manière directe d'apprendre le modèle égocentrique GMM de transition (4.3) [114]. Cependant, de par la grande dimension des entrées et sorties induite par la problématique considérée, l'apprentissage des MDNs dégénère souvent. Il résulte en un réseau produisant une densité de probabilité Gaussienne en lieu et place du GMM attendu [115]. Afin d'éviter ce comportement, les paramètres du GMM sont appris séparément par des réseaux de neurones profonds à propagation avant (Feed-Forward Neural Networks – FFNNs) différents. Les étapes de l'apprentissage sont détaillées ci-dessous.

4.4.1 Segmentation des données

Comme les hypothèses constituant le GMM (4.3) se rapportent aux manœuvres, leur ensemble doit être défini au préalable. Un ensemble d'apprentissage doit être défini dans highD et segmenté par un ensemble de règles systématiques provenant d'une connaissance experte, de façon à étiqueter à chaque instant l'évolution de tout véhicule par une des M manœuvres possibles. Cette segmentation conduit aux n-uplets $\left\{\left(\mathsf{x}_{\mathsf{n}-1}^{\mathsf{k}}, \{\mathsf{x}_{\mathsf{n}-1}^{\mathsf{l}}\}_{\mathcal{V}_{\mathsf{l}}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}, \mathsf{m}_{\mathsf{n}}^{\mathsf{k}}, \mathsf{x}_{\mathsf{n}}^{\mathsf{k}}\right)\right\}$, avec m_n^k l'étiquette de la manœuvre du véhicule cible entre les instants $n-1$ et n. Un exemple de segmentation est présenté dans la Figure 4.2. Celle-ci capture l'ambiguïté du mouvement du véhicule cible lors des phases de préparation et de fin d'une manœuvre incluant un changement de voie, en étiquetant des portions de trajectoires en amont et en aval comme appartenant à une même manœuvre. Ceci a pour effet

2. L'hypothèse de connaissance à la volée de la matrice de covariance semble réaliste : cette dernière peut être définie par les processus mis en place pour la fusion ou hybridation de données

de rendre plus floue la frontière entre les différentes manœuvres et permet à l'algorithme de classification de mieux considérer l'ambiguïté dans la réalisation d'une trajectoire. L'estimation et la prédiction considèrent ainsi un futur plus varié pour l'égo-véhicule. Il est important de noter que le besoin d'étiquetage « flou » lors de la phase d'apprentissage est lié à la granularité temporelle sélectionnée pour le modèle de dynamique a priori. Intuitivement, l'étiquette attachée à chaque instance d'apprentissage permet la capture implicite d'une information de plus long-terme dans la réalisation d'une manœuvre spécifique. Les seuils de segmentation sont déterminés empiriquement et dépendent des manœuvres et du jeu de données considérés.

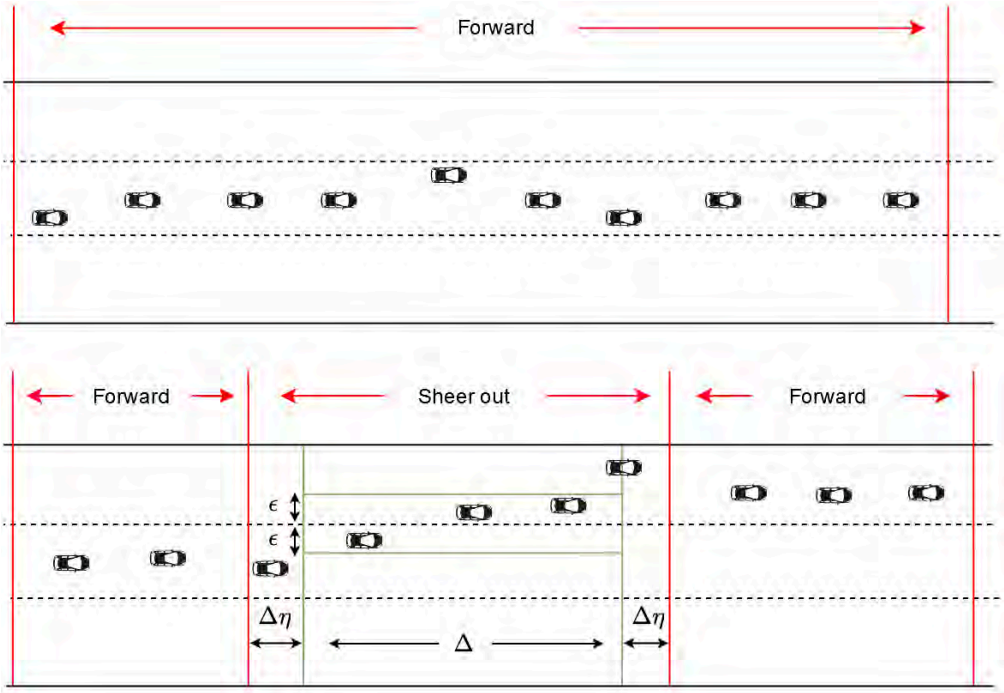


FIGURE 4.2 – Exemples de segmentation de manœuvres pour un véhicule cible. (Haut) Le véhicule cible reste dans sa voie, l'ensemble des données trajectorielles est donc classé comme appartenant à la manœuvre « forward ». (Bas) Plusieurs seuils sont introduits dans le processus de segmentation du véhicule cible pour une manœuvre comprenant un changement de voie (*e.g.*, « sheer-in » et « sheer-out »): la distance longitudinale totale de changement de voie Δ est sélectionnée de telle sorte que la position latérale de l'égo-véhicule se situe dans un couloir à $\pm\epsilon$ des marqueurs de délimitation de voie. La distance à laquelle les véhicules sont en phase de préparation ou de finalisation de manœuvre est délimité à l'aide du facteur η. Cette distance est proportionnelle à Δ.

4.4.2 Un réseau de neurones à propagation avant, de classification, pour l'apprentissage des probabilités des manœuvres de l'égo-véhicule conditionnellement à l'état précédent de la flotte

Chaque fonction scalaire $\lambda_m(.)$ dans (4.3) représente la probabilité d'engager la $m^{\text{ième}}$ manœuvre par le véhicule cible ($m \in \{1, \ldots, M\}$) entre les instants $n-1$ et n, conditionnellement à l'état de la flotte à l'instant $n-1$. Les probabilités de manœuvre $\{\lambda_m(.)\}_{m=1,\ldots,M}$ sont apprises au moyen d'un réseau de neurones profond à propagation avant (Feed-Forward Neural Network – FFNNs) constitué de : une couche d'entrée de taille égale à la somme des tailles des sous-vecteurs d'état relatifs au véhicule cible et à ses voisins ; plusieurs couches cachées entièrement connectées ; une couche de sortie dont les sorties $\{y_m^\lambda \in \mathbb{R}\}_{m=1,\ldots,M}$ sont combinées par la fonction Log-Softmax

$$\log \lambda_m = y_m^\lambda - \log\Big(\sum_{l=1}^M \exp y_l^\lambda\Big) \tag{4.9}$$

de façon à produire des $\{\lambda_m(.)\}_{m=1,\ldots,M}$. Par construction, ces probabilités se somment à 1 tout en assurant la stabilité numérique des calculs. Si le jeu de données est organisé en KN n-uplets $\big\{\big(\mathsf{x}_{\mathsf{n}-1}^{\mathsf{k}}, \{\mathsf{x}_{\mathsf{n}-1}^{\mathsf{l}}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}}, \mathsf{m}_{\mathsf{n}}^{\mathsf{k}}\big)\big\}$ construits à partir des véhicules cibles $\mathsf{k} = 1, \ldots, \mathsf{K}$ sur les instants $\mathsf{n} = 1, \ldots, \mathsf{N}$, où $\mathsf{m}_{\mathsf{n}}^{\mathsf{k}}$ définit la vérité terrain (étiquetage) de la manœuvre, alors le réseau de classification considéré est entraîné de façon à minimiser

$$\text{Loss} := -\sum_{\mathsf{n}=1,\mathsf{k}=1}^{\mathsf{n}=\mathsf{N},\mathsf{k}=\mathsf{K}} \log p\big(\mathsf{m}_{\mathsf{n}}^{\mathsf{k}} | \mathsf{x}_{\mathsf{n}-1}^{\mathsf{k}}, \{\mathsf{x}_{\mathsf{n}-1}^{\mathsf{l}}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}}\big). \tag{4.10}$$

4.4.3 Une banque de réseaux de neurones à propagation avant, de régression, pour l'apprentissage de la loi de dynamique a priori égocentrique conditionnellement à chaque manœuvre

Pour chaque indice $m \in \{1, \ldots, M\}$, un FFNN indépendant est en charge de l'apprentissage du vecteur moyenne $\mu_m(.)$ et de la matrice de covariance $\Sigma_m(.)$ de la loi de dynamique a priori égocentrique conditionnellement à manœuvre, indexée par m, de l'égo-véhicule, *i.e.*, de la loi $\mathcal{G}(x_n^k; \mu_m(x_{n-1}^k, \diamond), \Sigma_m(x_{n-1}^k, \diamond))$ apparaissant dans (4.3). Chaque $m^{\text{ième}}$ FFNN comprend également plusieurs couches cachées entièrement connectées, ainsi qu'une couche de sortie dont les sorties sont : $y_m^\mu \in \mathbb{R}^4$; soit $y_m^{\sigma_1}, \ldots, y_m^{\sigma_4} \in \mathbb{R}$ (pour (4.5)), soit $y_m^{l_{m,ij}} \in \mathbb{R}$, $(i,j) \in \{1, \ldots, 4\} \times \{1, \ldots, 4\}$, $i \geq j$ (pour (4.6)). Afin d'assurer une positivité stricte à $\{\sigma_{m,i}\}_{i=1,\ldots,4}$ et $\{l_{m,ii}\}_{i=1,\ldots,4}$, on pose, pour

$(i,j) \in \{1,\dots,4\} \times \{1,\dots,4\}, i > j,$

$$\text{soit } \mu_m = y_m^{\mu},\ \sigma_{m,i} = \exp(y_m^{\sigma_i}), \tag{4.11}$$

$$\text{soit } \mu_m = y_m^{\mu},\ l_{m,ii} = \exp(y_m^{l_{m,ii}}),\ l_{m,ij} = y_m^{l_{m,ij}}. \tag{4.12}$$

Chaque $m^{\text{ième}}$ réseau est entraîné avec les $\mathsf{K}_m\mathsf{N}_m$ n-uplets $\left\{\left(\mathsf{x}^{\mathsf{k}}_{\mathsf{n}-1,m}, \{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1,m}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}}, \mathsf{x}^{\mathsf{k}}_{\mathsf{n},m}\right)\right\}$ construits autour des K_m véhicules engagés dans la manœuvre indexée par m sur N_m instants. La fonction objectif correspondante s'écrit comme

$$\text{Loss}_m := -\sum_{\mathsf{n}=1,\mathsf{k}=1}^{\mathsf{n}=\mathsf{N}_m,\mathsf{k}=\mathsf{K}_m} \log \mathcal{G}\left(\mathsf{x}^{\mathsf{k}}_{\mathsf{n},m}; \mu_m(x^{\mathsf{k}}_{\mathsf{n}-1,m}, \mathsf{o}), \Sigma_m(x^{\mathsf{k}}_{\mathsf{n}-1,m}, \mathsf{o})\right),\ \text{avec } \mathsf{o} = \{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1,m}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}}. \tag{4.13}$$

4.5 Estimation stochastique en ligne du vecteur d'état de la flotte de véhicules

4.5.1 Introduction

Le filtre particulaire (PF) présenté dans la Section 2.3.5.2 constitue un estimateur récursif de la densité de filtrage du vecteur d'état de tout système Markovien, linéaire ou non, avec bruits additifs Gaussiens ou non. Les lois postérieures de prédiction et filtrage (jointes ou marginales) du vecteur d'état sont approximées par des lois ponctuelles dont le support est aléatoire. En d'autres termes, on construit un nuage de particules par échantillonnage aléatoire d'une loi d'importance définie préalablement, et on pondère chacun de ses éléments de telle sorte que l'échantillonnage selon la loi postérieure d'intérêt puisse être approchée par la sélection d'une particule avec une probabilité égale au poids qui lui est associé. Lorsque l'espace d'état admet une dimension élevée, son échantillonnage peut induire le phénomène de « trous et d'amas » (“gaps and clusters”) [116, 117, 118]. La conséquence est une faible efficacité du filtre pour un coût calculatoire élevé.

Afin de pallier ces défauts tout en conservant la modélisation choisie, deux approches reposant sur des approximations du filtre particulaire « standard » (où la loi d'importance est égale à la loi de dynamique a priori) sont envisagées. Ces deux algorithmes sont présentés et exploités dans le cadre des ITS ci-dessous.

4.5.2 Le Filtre Particulaire Multiple

Le Filtre Particulaire Multiple (Multiple Particle filter – MPF) repose sur la partition du vecteur d'état en sous-vecteurs dont chacun est estimé à l'aide d'un filtre particulaire standard. Des étapes d'échange sont insérées préalablement à l'échantillonnage (« propagation temporelle ») et la

pondération des particules propres à chaque filtre. Le MPF a été utilisé avec succès dans des applications de filtrage en temps réel [119, 120]. Plutôt que d'échanger les moments estimés des lois postérieures marginales « partitionnées » gérées par chaque filtre, la référence [121] propose que les filtres échangent leurs particules. Les caractéristiques du MPF semblent en faire un algorithme de choix pour les besoins du problème ITS considéré dans cette thèse.

On rappelle que la loi de dynamique a priori est le GMM (4.3) (avec (4.5) ou (4.6)) et que le modèle d'observation est donné par (4.7). Les lois marginales partitionnées de filtrage $\{p(x_n^k|z_{0:n})\}_{k=1,\ldots,K}$ à l'instant n sont estimées récursivement au moyen de distributions discrètes dans un filtre particulaire en charge de chaque $k^{\text{ième}}$ partition, de telle sorte que

$$p(x_n^k|z_{0:n}) \approx \sum_{m=1}^{M} w_n^{k(m)} \delta(x_n^k - x_n^{k(m)}). \tag{4.14}$$

Les étapes d'échantillonnage (« propagation temporelle ») et de pondération de ses particules (indexées par m) sont les suivantes :

$$x_n^{k(m)} \sim q(x_n^k|x_{0:n-1}^{k(m)}, z_{0:n})\,;\; w_n^{k(m)} \propto w_{n-1}^{k(m)} \frac{p(z_n|x_{0:n}^{k(m)}, z_{0:n-1}) p(x_n^{k(m)}|x_{0:n-1}^{k(m)}, z_{0:n-1})}{q(x_n^{k(m)}|x_{0:n-1}^{k(m)}, z_{0:n})}. \tag{4.15}$$

Il convient de remarquer que, du fait que $x_\cdot^k$ ne désigne qu'une portion du vecteur d'état $x_\cdot$, $p(x_n^k|x_{0:n-1}^k, z_{0:n-1}) \neq p(x_n^k|x_{n-1}^k)$ et $p(z_n|x_{0:n}^k, z_{0:n-1}) \neq p(z_n|x_n^k)$. La référence [121] [3] propose les deux hypothèses et simplifications suivantes, le complément dans $x_\cdot$ de sa $k^{\text{ième}}$ partition $x_\cdot^k$ étant désigné par $x_\cdot^{\{k-\}}$:

— la fonction d'importance est définie comme la distribution marginale de prédiction, *i.e.*,

$$q(x_n^k|x_{0:n-1}^k, z_{0:n}) = p(x_n^k|x_{0:n-1}^k, z_{0:n-1}) = \int p(x_n^k|x_{n-1}) p(x_{n-1}^{\{k-\}}|x_{0:n-1}^k, z_{0:n-1}) \mathrm{d}x_{n-1}^{\{k-\}} \tag{4.16}$$

couplée à l'approximation

$$p(x_{n-1}^{\{k-\}}|x_{0:n-1}^k, z_{0:n-1}) \approx \prod_{l \neq k} p(x_{n-1}^l|z_{0:n-1}); \tag{4.17}$$

3. Nous proposons dans ce document une interprétation que nous espérons plus claire de celle présentée dans cette référence. De plus, l'équation de l'article définissant la distribution marginale de prédiction utilisée pour la propagation des particules par filtre comporte une erreur, elle est corrigée ici.

— la vraisemblance conditionnelle par partition

$$p(z_n|x_{0:n}^k, z_{0:n-1}) = \int p(z_n|x_n)p(x_n^{\{k-\}}|x_{0:n}^k, z_{0:n-1})\mathrm{d}x_n^{\{k-\}} \tag{4.18}$$

est approchée par

$$p(x_n^{\{k-\}}|x_{0:n}^k, z_{0:n-1}) \approx \prod_{l\neq k} p(x_n^l|z_{0:n-1}). \tag{4.19}$$

Pour mettre en pratique ces idées, deux étapes d'échange sont insérées dans chaque $k^{\text{ième}}$ filtre particulaire « partitionné » : une première avant l'étape de propagation, une autre avant l'étape de pondération. Ces échanges prennent en compte les densités approchées par les autres $l^{\text{ièmes}}$ filtres particulaires partitionnés comme suit :

— pour implémenter l'échantillonnage par importance selon $p(x_n^k|x_{0:n-1}^{k(m)}, z_{0:n-1})$ avec (4.16)-(4.17), des approximations de Monte Carlo des lois marginales de filtrage partitionnées $\{p(x_{n-1}^l|z_{0:n-1})\}_{l\neq k}$ à l'instant précédent sont utilisées ;

— pour implémenter $p(z_n|x_{0:n}^{k(m)}, z_{0:n-1})$ dans la pondération des particules (4.18)-(4.19), des approximations de Monte Carlo des lois marginales de prédiction partionnées $\{p(x_n^l|z_{0:n-1})\}_{l\neq k}$ à l'instant courant sont utilisées.

L'algorithme complet du MPF [121] est détaillé dans la Figure 4.3. Son application à des véhicules en interaction avec des dynamiques a priori interactives respectant l'Hypothèse 4 et le modèle de mesure décrit dans l'Hypothèse 3 est simple : le vecteur d'état est partitionné par véhicule, de sorte que K représente à la fois le nombre de partitions et le nombre total de véhicules de la scène routière.

4.5.3 Le Filtre Particulaire Multiple Variationnel Bayésien

Dans cette section, un autre algorithme de filtrage particulaire multiple est présenté : le Filtre Particulaire Multiple Variationnel Bayésien (Variational Bayesian Multiple Particle Filter – VBMPF)[122]. Le VBMPF est voisin du MPF présenté dans la section précédente dont il cherche à améliorer les performances en combinant des approximations variationnelles et des approximations de Monte Carlo des densités de probabilités mises en jeu dans le processus d'estimation stochastique. À chaque instant, la distribution postérieure de l'état du système est approchée analytiquement par une loi séparable, *i.e.*, un produit de densités marginales indépendantes. Chaque distribution marginale est estimée par une loi ponctuelle, comme dans un filtre particulaire standard. Comparativement au MPF, le VBMPF fonde la séparabilité des lois de prédiction et de filtrage sur l'optimisation de divergences de Kullback-Leibler pour l'obtention d'approximations variationnelles Bayésiennes de certaines distributions intermédiaires. Dans la veine du MPF, un échange de

particules entre les partitions de l'état intervient au cours de l'estimation, mais via un mécanisme différent plus efficace sur le plan calculatoire.

À tout instant n, la densité de filtrage $p(x_n|z_{0:n})$ est donc approximée par la forme séparable $q(x_n|z_{0:n}) := \prod_{k=1}^{K} q(x_n^k|z_{0:n})$. La référence [122] établit une stratégie de filtrage particulaire « multiple » efficace d'un point de vue calculatoire, qui, grâce à des approximations variationnelles Bayésiennes ("VB Approximations"), permet une solution basée sur K filtres particulaires traitant séparément les partitions respectives du vecteur d'état caché, mais interagissant entre eux. Les caractéristiques suivantes sont attrayantes pour notre problématique, l'algorithme étant instancié sur la Figure 4.4.

— Chaque facteur de l'approximation séparable $q(x_n|z_{0:n}) := \prod_{k=1}^{K} q(x_n^k|z_{0:n})$ de la densité de filtrage $p(x_n|z_{0:n})$ est lui-même estimé par une densité discrète $\hat{q}(x_n^k|z_{0:n}) = \sum_{s=1}^{S} \rho_n^{k,(s)} \delta(x_n^k - \tilde{x}_n^{k,(s)})$.

— Le point clé (Partie II/2a) repose sur une approximation $q(x_n^k|x_{n-1}^k, z_{0:n-1})$ de la loi de dynamique a priori partitionnée $p(x_n^k|x_{n-1}) = p(x_n^k|x_{n-1}, z_{0:n-1})$ définie de façon à éliminer la dépendance statistique entre x_n^k et x_{n-1}^{k-} (le complément de x_{n-1}^k dans x_{n-1}) conditionnellement à x_{n-1}^k et $z_{0:n-1}$. La recherche de l'approximation variationnelle Bayésienne (VB) conduit à la « loi de dynamique a priori partitionnée VB »

$$q(x_n^k|x_{n-1}^k, z_{0:n-1}) \propto \exp\big(\mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}[\ln p(x_n^k|x_{n-1})]\big). \tag{4.20}$$

— L'utilisation de (4.20) comme fonction d'importance du filtre particulaire traitant chaque partition $\{x_n^k\}_{k=1,\ldots,K}$, du vecteur d'état caché x_n permet le calcul récursif de l'approximation particulaire partitionnée $\hat{q}(x_n^k|z_{0:n})$. C'est pourquoi l'algorithme complet se présente comme un ensemble de K filtres particulaires séparés « par partition ».

— Toutefois, un mécanisme d'interaction de ces filtres particulaires séparés doit être mis en place (voir Partie III). En effet, pour chaque $k^{\text{ième}}$ partition du vecteur d'état, la définition et l'échantillonnage approché de la loi de dynamique a priori partitionnée VB $q(x_n^k|x_{n-1}^k, z_{0:n-1})$ donnée par (4.20) nécessite le calcul d'espérances de $\ln p(x_n^k|x_{n-1})$ selon l'approximation séparable de la loi postérieure des autres partitions.

Comme pour le MPF, le vecteur d'état caché est partitionné par véhicule, de sorte que K représente à la fois le nombre de partitions et le nombre de véhicules. Cependant, comme indiqué sur la Partie I de la Figure 4.4, [122] suppose une loi de dynamique a priori par partition $p(x_n^k|x_{n-1})$ Gaussienne dont la matrice de covariance Q_n^k est indépendante de x_{n-1} alors que dans le contexte des véhicules en interaction, la loi de dynamique a priori égocentrique (4.2)–(4.3) est un mélange de Gaussiennes dont les covariances et les poids dépendent de x_{n-1}.

La résolution analytique de (4.20) pour $p(x_n^k|x_{n-1})$ définie en (4.2)–(4.3) semble être un problème difficile. Bien que la littérature traite de l'approximation de la divergence de Kullback-Leibler entre deux GMMs [123, 124], l'obtention d'une approximation GMM de la dynamique a priori égocentrique VB demeure un problème ouvert et complexe : difficultés d'exploitation des bornes supérieure et inférieure ; conservatisme de ces bornes ; non-invariance de l'approximation par permutation circulaire des hypothèses des GMM ; etc. Néanmoins, une solution conservative peut être établie en approximant la loi de dynamique a priori égocentrique GMM (4.2)–(4.3) par la loi Gaussienne admettant les mêmes moyenne et matrice de covariance (« moment-matched Gaussian approximation »)

$$\begin{aligned} p(x_n^k|x_{n-1}^k,\diamond) &= \textstyle\sum_{m=1}^M \lambda_m(x_{n-1}^k,\diamond)\mathcal{G}(x_n^k;\mu_m(x_{n-1}^k,\diamond),\Sigma_m(x_{n-1}^k,\diamond)), \\ &\approx \bar{p}(x_n^k|x_{n-1}^k,\diamond) := \mathcal{G}(x_n^k;\mu(x_{n-1}^k,\diamond),\Sigma(x_{n-1}^k,\diamond)), \\ \text{avec } \mu(.) &= \textstyle\sum_{m=1}^M \lambda_m(.)\mu_m(.) \\ \Sigma(.) &= \textstyle\sum_{m=1}^M \lambda_m(.)\left(\Sigma_m(.) + (\mu_m(.)-\mu(.))(\mu_m(.)-\mu(.))^T\right). \end{aligned} \tag{4.21}$$

puis en étendant [122] de façon à permettre la dépendance en le vecteur d'état précédent x_{n-1} de la matrice de covariance Σ obtenue. Afin de traiter le cas où $Q_n(x_n) := \mathrm{blkdiag}(Q_n^1(x_n),\ldots,Q_n^K(x_n))$ dans la Partie I de la Figure 4.4, il suffit de remplacer dans le premier item de la Partie III de la Figure 4.5 $q(x_n^k|x_{n-1}^k,z_{0:n-1})$ par

$$\begin{aligned} q(x_n^k|x_{n-1}^k,z_{0:n-1}) &= \mathcal{G}(x_n^k;m_n^k(x_{n-1}^k),\Lambda_n^k(x_{n-1}^k)), \\ \text{avec } \Lambda_n^k(x_{n-1}^k) &:= \left(\mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\left[\left(Q_{n-1}^k(x_{n-1})\right)^{-1}\right]\right)^{-1} \\ \text{et } m_n^k(x_{n-1}^k) &:= \Lambda_n^k(x_{n-1}^k)\mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\left[(Q_{n-1}^k(x_{n-1}))^{-1}f_{n-1}^k(x_{n-1})\right], \end{aligned} \tag{4.22}$$

où $f_n^k(x_n)$ est la fonction de transition à l'instant n du sous-vecteur d'état de la partition k. La preuve de ce résultat est contenue dans la Figure 4.6. Les approximations Monte Carlo de $m_n^k(x_{n-1}^k), \Lambda_n^k(x_{n-1}^k)$ peuvent toujours être obtenues dans la veine de la méthode présentée dans la Partie III. Les équations (4.22) montrent encore une fois que même si le VBMPF permet une approximation séparable de la loi postérieure, celle-ci ne s'effectue pas au détriment de la capture des interactions entre les véhicules.

4.5.4 Initialisation de l'estimateur

A l'instant $n = 0$, La densité de probabilité initiale de l'état $p(x_0)$ est inconnue. Elle est déterminée simplement par assimilation des mesures z_{-1}, z_0 dans un filtre d'information [125] commençant à $n = -1$ avec une matrice inverse de covariance nulle et en supposant une dynamique à

vitesse constante entre les instants $n = -1$ et $n = 0$.

I - Objectif de chaque $k^{\text{ième}}$ filtre particulaire « partitionné », *i.e.*, propre au sous-vecteur $x_{\cdot}^{k}$

- Étant donnée la distribution discrète $\hat{p}(x_{n-1}^{k}|z_{0:n-1}) = \sum_{m=1}^{M} w_{n-1}^{k(m)} \delta(x_{n-1}^{k} - x_{n-1}^{k(m)})$ qui approche $p(x_{n-1}|z_{0:n-1})$ à l'instant $n-1$, déterminer l'approximation $\hat{p}(x_{n}^{k}|z_{0:n}) = \sum_{m=1}^{M} w_{n}^{k(m)} \delta(x_{n}^{k} - x_{n}^{k(m)})$ de $p(x_n|z_{0:n})$.

II - ALGORITHME PRINCIPAL

— [Si $n = 0$, alors remplacer les Étapes 1 et 2 ci-dessous par : Échantillonner $\{x_0^{k,(m)}\}_{m=1,\ldots,M}$ i.i.d. selon $p(x_0^k)$ et définir $J = 1$.] [Puis, aller à l'Étape 3.]

— Pour chaque $k = 1, \ldots, K$,

1. [ÉCHANGE-I] Pour chaque particule $x_{n-1}^{k(m)}$—dont le poids est en fait $w_{n-1}^{k(m)} = \frac{1}{M}$—tirer les J particules compagnes $\{x_{n-1}^{\{k-\}(m,j)}\}_{j=1,\ldots,J}$ selon $\prod_{l \neq k} p(x_{n-1}^{l}|z_{0:n-1})$ donné dans (4.17) en échantillonnant chaque $l^{\text{ième}}$ partition $x_{n-1}^{l(m,j)}$ de $x_{n-1}^{\{k-\}(m,j)}$ selon l'approximation de la densité marginale de filtrage à l'instant précédent $\hat{p}(x_{n-1}^{l}|z_{0:n-1}) = \sum_{r=1}^{M} w_{n-1}^{l(\mu_{l,r})} \delta(x_{n-1}^{l} - x_{n-1}^{l(\mu_{l,r})})$ produite par le $l^{\text{ième}}$ filtre particulaire partitionné; en d'autres termes, pour chaque $l \in \{1, \ldots, K\} \backslash \{k\}$, sélectionner indépendamment avec remise chaque $x_{n-1}^{l(m,j)}$, $j = 1, \ldots, J$, dans l'ensemble des M éléments $\{x_{n-1}^{l(\mu_{l,r})}\}_{r=1,\ldots,M}$ avec une probabilité égale au poids respectif dans $\{w_{n-1}^{l(\mu_{l,r})}\}_{r=1,\ldots,M}$, *i.e.*, avec la probabilité $\frac{1}{M}$.
2. [PROPAGATION] Constituer l'ensemble des MJ éléments $\{x_n^{k(m,j)}\}_{m=1,\ldots,M;\, j=1,\ldots,J}$ en échantillonnant indépendamment $x_n^{k(m,j)} \sim p(x_n^k|x_{n-1}^{k(m)}, x_{n-1}^{\{k-\}(m,j)})$. Compte tenu de l'étape d'échange ci-dessus, pour un m fixé, chaque sous-ensemble $\{x_n^{k(m,j)}\}_{j=1,\ldots,J}$ peut être vu comme un ensemble d'échantillons indépendants de l'approximation (4.16) de $p(x_n^k|x_{0:n-1}^{k(m)}, z_{0:n-1})$, et une approximation de Monte Carlo de la densité marginale de prédiction par partition à l'instant courant $p(x_n^k|z_{0:n-1})$ s'écrit $\hat{p}(x_n^k|z_{0:n-1}) = \sum_{m=1}^{M} \sum_{j=1}^{J} w_{n-1}^{k(m,j)} \delta(x_n^k - x_n^{k(m,j)})$ avec $w_{n-1}^{k(m,j)} = \frac{1}{MJ}$.

— Pour chaque $k = 1, \ldots, K$,

1. [ÉCHANGE-II] Tirer les I particules compagnes $\{x_n^{\{k-\}(i)}\}_{i=1,\ldots,I}$ selon $\prod_{l \neq k} p(x_n^l|z_{0:n-1})$ donné dans (4.19) en échantillonnant indépendamment chaque $l^{\text{ième}}$ partition $x_n^{l(i)}$ de $x_n^{\{k-\}(i)}$ selon l'approximation de Monte Carlo de la loi de prédiction $\hat{p}(x_n^l|z_{0:n-1}) = \sum_{r=1}^{M} \sum_{s=1}^{J} w_{n-1}^{l(\mu_{l,r},\nu_{l,s})} \delta(x_n^l - x_n^{l(\mu_{l,r},\nu_{l,s})})$à l'instant courant produite par le $l^{\text{ième}}$ filtre particulaire partitionné après son étape de PROPAGATION, avec $\forall (r,s) \in \{1, \ldots, M\} \times \{1, \ldots, J\}$, $w_{n-1}^{l(\mu_{l,r},\nu_{l,s})} = \frac{1}{MJ}$; en d'autres termes, pour chaque $l \in \{1, \ldots, K\} \backslash \{k\}$, sélectionner indépendamment avec remise chaque $x_n^{l(i)}$, $i = 1, \ldots, I$, dans l'ensemble des MJ éléments $\{x_n^{l(\mu_{l,r},\nu_{l,s})}\}_{r=1,\ldots,M;\, s=1\ldots,J}$ avec la probabilité $\frac{1}{MJ}$.
2. [PONDÉRATION] Associer à chaque particule $x_n^{k(m,j)}$, $m = 1, \ldots, M$, $j = 1, \ldots, J$, le poids non normalisé $w_n^{k(m,j)} = \frac{1}{I} \sum_{i=1}^{I} p(z_n|x_n^{k(m,j)}, x_n^{\{k-\}(i)})$. Cela correspond bien au schéma de pondération (4.15), avec $q(x_n^{k(m)}|x_{0:n-1}^{k(m)}, z_{0:n}) = p(x_n^{k(m)}|x_{0:n-1}^{k(m)}, z_{0:n-1})$, $w_{n-1}^{k(m)} = \frac{1}{M}$, et avec une approximation de Monte Carlo de la vraisemblance conditionnelle (4.18) exploitant l'étape d'échange ci-dessus.
3. [RE-NORMALISATION DES POIDS] Pour chaque $m \in \{1, \ldots, M\}$, $j \in \{1, \ldots, J\}$, définir $w_n^{k(m,j)} = \frac{w_n^{k(m,j)}}{\sum_{r=1}^{M} \sum_{s=1}^{J} w_n^{k(r,s)}}$. Les moments de la distribution marginale de filtrage par partition $p(x_n^k|z_{0:n})$ peuvent être approchés depuis $\{w_n^{k(m,j)}, x_n^{k(m,j)}\}_{m=1,\ldots,M;\, j=1\ldots,J}$ à l'aide de formules standards.
4. [SOUS-ÉCHANTILLONNAGE] Rapporter l'ensemble $\{x_n^{k(m,j)}\}_{m=1,\ldots,M;\, j=1,\ldots,J}$ des MJ particules à un ensemble final de particules à M éléments $\{x_n^{k(m)}\}_{m=1,\ldots,M}$ en échantillonnant indépendamment avec remise chaque $x_n^{k(m)}$ dans $\{x_n^{k(r,s)}\}_{r=1,\ldots,M;\, s=1,\ldots,J}$ avec une probabilité égale au poids respectif dans $\{w_n^{k(r,s)}\}_{r=1,\ldots,M;\, s=1,\ldots,J}$. Ainsi, $p(x_n^k|z_{0:n}) \approx \hat{p}(x_n^k|z_{0:n}) = \sum_{m=1}^{M} w_n^{k(m)} \delta(x_n^k - x_n^{k(m)})$ avec $w_n^{k(m)} = \frac{1}{M}$ est obtenu.

FIGURE 4.3 – Filtre Particulaire Multiple (MPF) selon [121].

I - PROBLÈME

— Hypothèses
  - $x_{n+1} = \begin{pmatrix} x_{n+1}^1 \\ \vdots \\ x_{n+1}^K \end{pmatrix} = f_n(x_n) + w_n = \begin{pmatrix} f_n^1(x_n)+w_n^1 \\ \vdots \\ f_n^K(x_n)+w_n^K \end{pmatrix}$, $w_n \sim \mathcal{G}(0, Q_n := \mathrm{blkdiag}(Q_n^1, \ldots, Q_n^K))$, $w_{0:n}$ blanc ;
  - $z_n = \begin{pmatrix} z_n^1 \\ \vdots \\ z_n^K \end{pmatrix} = Hx_n + v_n = \begin{pmatrix} H_n^1 x_n^1 + v_n^1 \\ \vdots \\ H_n^K x_n^K + v_n^K \end{pmatrix}$, $v_n \sim \mathcal{G}(0, R_n := \mathrm{blkdiag}(R_n^1, \ldots, R_n^K))$, $v_{0:n}$ blanc indépendant de $w_{0:n}$ et x_0.

— Objectif du filtre particulaire traitant chaque $k^{\text{ième}}$ partition
  - Étant donnée la loi discrète $\hat{q}(x_{n-1}^k|z_{0:n-1}) = \sum_{s=1}^S \lambda_{n-1}^{k,(s)} \delta(x_{n-1}^k - \tilde{x}_{n-1}^{k,(s)})$ qui approche $q(x_{n-1}^k|z_{0:n-1})$ telle que $q(x_{n-1}|z_{0:n-1}) := \prod_{k=1}^K q(x_{n-1}^k|z_{0:n-1}) \approx p(x_{n-1}|z_{0:n-1})$, déterminer l'approximation $\hat{q}(x_n^k|z_{0:n}) = \sum_{s=1}^S \lambda_n^{k,(s)} \delta(x_n^k - \tilde{x}_n^{k,(s)})$ de $q(x_n^k|z_{0:n})$ qui satisfait $q(x_n|z_{0:n}) := \prod_{k=1}^K q(x_n^k|z_{0:n}) \approx p(x_n|z_{0:n})$.

II - ALGORITHME PRINCIPAL

— [Si $n = 0$, alors : remplacer les étapes 1 et 2 ci-dessous par l'échantillonnage de $\{x_0^{k,(s)}\}_{s=1,\ldots,S}$ i.i.d. selon $p(x_0)$, puis effectuer l'étape 2b.]

— Pour chaque $k = 1, \ldots, K$,
  1. Échantillonner $\{x_{n-1}^{k,(s)}\}_{s=1,\ldots,S}$ i.i.d. selon $\hat{q}(x_{n-1}^k|z_{0:n-1})$ (*i.e.*, rééchantillonner l'ensemble des particules $\{(\tilde{x}_{n-1}^{k,(s)}, \lambda_{n-1}^{k,(s)})\}_{s=1,\ldots,S}$ en $\{(x_{n-1}^{k,(s)}, \frac{1}{S})\}_{s=1,\ldots,S}$).
  2. Pour chaque $s \in \{1, \ldots, S\}$,
     (a) Échantillonner $\tilde{x}_n^{k,(s)} \sim q(x_n^k|x_{n-1}^{k,(s)}, z_{0:n-1})$, où $q(x_n^k|x_{n-1}^k, z_{0:n-1})$ est l'approximation VB de la loi de dynamique a priori partitionnée $p(x_n^k|x_{n-1})$ (égale à $p(x_n^k|x_{n-1}, z_{0:n-1})$) de façon à supprimer la dépendance statistique entre x_n^k et x_{n-1}^{k-} conditionnellement à x_{n-1}^k et $z_{0:n-1}$;
     (b) Définir $\lambda_n^{k,(s)} = p(z_n^k|\tilde{x}_n^{k,(s)})$.
  3. Re-normaliser les poids, *i.e.*, pour chaque $s \in \{1, \ldots, S\}$, définir $\lambda_n^{k,(s)} = \frac{\lambda_n^{k,(s)}}{\sum_{r=1}^S \lambda_n^{k,(r)}}$.
  4. Sur la base de la loi discrète $\hat{q}(x_n^k|z_{0:n}) = \sum_{s=1}^S \lambda_n^{k,(s)} \delta(x_n^k - \tilde{x}_n^{k,(s)})$, qui approche $q(x_n^k|z_{0:n})$, déduire les approximations de la moyenne et covariance postérieures $\hat{x}_{n|n}^k = \sum_{s=1}^S \lambda_n^{k,(s)} \tilde{x}_n^{k,(s)}$ et $\hat{P}_{n|n}^k = \sum_{s=1}^S \lambda_n^{k,(s)} (\tilde{x}_n^{k,(s)} - \hat{x}_{n|n}^k)(\tilde{x}_n^{k,(s)} - \hat{x}_{n|n}^k)^T$ de la $k^{\text{ième}}$ partition de $p(x_n|z_{0:n})$.

FIGURE 4.4 – Filtre Particulaire Multiple Variationnel Bayésien [122] pour un problème dont le modèle d'observation est simplifié.

III - DÉTAILS DE L'ÉTAPE 2a : COMMENT APPROXIMER LA LOI DE DYNAMIQUE A PRIORI PARTITIONNÉE VB $q(x_n^k|x_{n-1}^{k,(s)}, z_{0:n-1})$?

— L'équation (4.20) mène à $q(x_n^k|x_{n-1}^k, z_{0:n-1}) = \mathcal{G}(x_n^k; m_n^k(x_{n-1}^k), Q_{n-1}^k)$ avec $m_n^k(x_{n-1}^k) := \mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\big[f_{n-1}^k(x_{n-1})\big]$.

— $m_n^k(x_{n-1}^k)$ ne peut pas être calculé analytiquement, mais une approximation de Monte Carlo $\hat{m}_n^k(x_{n-1}^k) = \frac{1}{J}\sum_{j=1}^J f_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_j)})$ peut être obtenue facilement en définissant $\mathcal{X}_{k,n-1}^{(s_j)} := \big((x_{n-1}^{1,(s_j)})^T, \ldots, (x_{n-1}^{k-1,(s_j)})^T, (x_{n-1}^k)^T, (x_{n-1}^{k+1,(s_j)})^T, \ldots, (x_{n-1}^{K,(s_j)})^T\big)^T$, où, pour chaque $r \neq k$: $\forall j = 1, \ldots, J,\ x_{n-1}^{r,(s_j)} \overset{\text{i.i.d.}}{\sim} \sum_{s=1}^S \frac{1}{S}\delta(x_{n-1}^r - x_{n-1}^{r,(s)})$, *e.g.*, avec $J = S$. En d'autres termes, pour chaque $k^{\text{ième}}$ partition, $\hat{m}_n^k(x_{n-1}^k)$ est calculé comme la moyenne de $\{f_{n-1}^k(\mathcal{X}_{k,n-1}^{(s_j)})\}_{j=1,\ldots,J}$, où : pour tout $j = 1, \ldots, J$, le $k^{\text{ième}}$ sous-vecteur de $\mathcal{X}_{k,n-1}^{(s_j)}$ est défini par la variable x_{n-1}^k ; chaque autre $r^{\text{ième}}$ sous-vecteur de $\mathcal{X}_{k,n-1}^{(s_j)}$, $r \neq k$, est obtenu en échantillonnant l'ensemble $\{x_{n-1}^{r,(1)}, \ldots, x_{n-1}^{r,(S)}\}$ obtenu après l'étape 1, ce processus étant bien sûr répété pour $j = 1, \ldots, J$.

FIGURE 4.5 – Mécanisme d'interaction des filtres particulaires partitionnés constituant le VBMPF.

PREUVE DE (4.22)

Quelle que soit la fonction $\phi(x_{n-1})$ de x_{n-1}, avec x_{n-1}^k un sous-vecteur de x_{n-1} et x_{n-1}^{k-} son complément dans x_{n-1}, soit $\overline{\phi}(x_{n-1}^k) := \mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}[\phi(x_{n-1})]$. L'approximation VB de $p(x_n^k|x_{n-1}) = \mathcal{G}(x_n^k; f_{n-1}^k(x_{n-1}), Q_{n-1}^k(x_{n-1}))$ selon (4.20) s'écrit comme suit, avec $C_1, C_2, \ldots$ des termes indépendants de x_n^k :

$$\ln q(x_n^k|x_{n-1}^k, z_{0:n-1}) = \mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\Big[\frac{1}{2}(x_n^k - f_{n-1}^k(x_{n-1}))^T(Q_{n-1}^k(x_{n-1}))^{-1}(x_n^k - f_{n-1}^k(x_{n-1}))\Big] + C_1 \tag{4.23}$$

$$= \mathbb{E}_{q(x_{n-1}^{k-}|z_{0:n-1})}\Big[\frac{1}{2}(x_n^k)^T\underbrace{(Q_{n-1}^k(x_{n-1}))^{-1}}_{=:I_{n-1}^k(x_{n-1})}(x_n^k) - (x_n^k)^T\underbrace{(Q_{n-1}^k(x_{n-1}))^{-1}f_{n-1}^k(x_{n-1})}_{=:J_{n-1}^k(x_{n-1})}\Big] + C_2 \tag{4.24}$$

$$= \frac{1}{2}(x_n^k)^T\Big(\overline{I_{n-1}^k}(x_{n-1}^k)\Big)(x_n^k) - (x_n^k)^T\Big(\overline{J_{n-1}^k}(x_{n-1})\Big) + C_2 = \ln \mathcal{G}(x_n^k; m_n^k(x_{n-1}^k), \Lambda_n^k(x_{n-1}^k)) \tag{4.25}$$

avec $\Lambda_n^k(x_{n-1}^k) = \Big(\overline{I_{n-1}^k}(x_{n-1}^k)\Big)^{-1}$ et $\Big(\Lambda_n^k(x_{n-1}^k)\Big)^{-1}m_n^k(x_{n-1}^k) = \overline{J_{n-1}^k}(x_{n-1}^k)$, ce qui complète la preuve.

FIGURE 4.6 – L'extension requise du VBMPF.

4.6 Prédiction et évaluation du risque

A tout moment, l'estimation peut être arrêtée et la prédiction peut être lancée : aucune mesure n'est incorporée et les états des véhicules sont prédits de manière itérative en échantillonnant de nouveaux ensembles de particules sur m-pas dans le futur selon le modèle de transition. Les approximations discrètes obtenues des densités de prédiction d'état ou de mesure permettent d'évaluer les collisions possibles entre véhicules.

L'approche élémentaire suivante est considérée, sur la base des K ensembles de particules pondérées associées aux densités de filtrage $\{p(x_n^k|z_{0:n})\}_{k=1,\ldots,K}$ des véhicules à l'instant n. Tout d'abord, ces ensembles aux poids inégaux sont rééchantillonnés. Puis, leur produit Cartésien est calculé, conduisant à un ensemble unique $\{x_n^{(i)}\}_{i=1,\ldots,I}$ tel que $\hat{p}(x_n|z_{0:n}) := \frac{1}{I}\sum_{i=1}^{I}\delta(x_n - x_n^{(i)})$ approche la densité de filtrage $p(x_n|z_{0:n})$ de la flotte. Ci-dessus, I peut-être fixé à une valeur élevée de façon à augmenter la variété dans le mélange des particules. Alors, une approximation ponctuelle de la distribution de prédiction $p(x_{n+1}|z_{0:n})$ s'écrit $\hat{p}(x_{n+1}|z_{0:n}) := \frac{1}{I}\sum_{i=1}^{I}\delta(x_{n+1} - x_{n+1}^{(i)})$ où chaque particule $x_{n+1}^{(i)}$ est l'empilement des sous-vecteurs par véhicule $x_{n+1}^{1,(i)}, \ldots, x_{n+1}^{k,(i)}, \ldots, x_{n+1}^{K,(i)}$ échantillonnés indépendemment selon la dynamique égocentrique apprise (4.2)–(4.3), *i.e.*, selon $\forall k \in \{1,\ldots,K\}$, $x_{n+1}^{k,(i)} \sim p(x_{n+1}^k|x_n^{k,(i)}, \{x_n^{l,(i)}\}_{\mathcal{V}_l \in \mathcal{N}_n(\mathcal{V}_k)})$. Ce processus est répété jusqu'à la valeur voulue de m pour obtenir l'approximation ponctuelle $\hat{p}(x_{n+m}|z_{0:n})$ de la prédiction à m instants dans le futur de la densité de prédiction $p(x_{n+m}|z_{0:n})$, voir la Définition 1. Les approximations des moyennes et covariances $\hat{x}_{n+m|n}, \hat{P}_{n+m|n}$ sont ensuite calculées directement.

En supposant qu'aucune collision entre véhicules ne soit présente dans le jeu de données ayant servi pour l'apprentissage, les moments de prédiction calculés décrivent les distributions futures pour chaque véhicule lors d'un déroulement nominal de la scène routière. Les approximations des ellipses de confiance à 99% de probabilité associées à chaque véhicule peuvent être facilement obtenues depuis les partitions $\hat{x}_{n+m|n}^k, \hat{P}_{n+m|n}^k$. de $\hat{x}_{n+m|n}, \hat{P}_{n+m|n}$, $k = 1,\ldots,K$. Depuis ces moments, il est également possible de calculer ceux correspondant aux lois de prédiction des mesures à tout instant dans le futur pour chaque véhicule. En comparant, pour chaque véhicule, la loi de prédiction du sous-vecteur de mesure à l'instant futur $n+m$ avec les observations associées, il est possible, il est possible de détecter un comportement non nominal signalant une collision potentielle. Le calcul de l'innovation quadratique relative à chaque véhicule normalisée par l'inverse de sa matrice de covariance théorique permet d'établir si la mesure reçue de chaque véhicule constitue un échantillon plausible de la loi de prédiction du sous-vecteur de mesure associé conditionnellement aux observations passées. Ainsi, pour chaque véhicule $k = 1,\ldots,K$ et à tout instant futur

$n+m$, on vérifie si l'innovation quadratique normalisée à m pas dans le futur,

$$r_{n+m}^{k}{}^{T}(H_{n+m}^{k}{}^{T}\hat{P}_{n+m|n}^{k\,-1}H_{n+m}^{k}+R_{n+m}^{k})^{-1}r_{n+m}^{k},\ \text{avec}\ r_{n+m}^{k}:=z_{n+m}^{k}-H_{n+m}^{k}\hat{x}_{n+m|n}^{k} \tag{4.26}$$

est un échantillon plausible d'une loi du khi-deux χ_2^2 à 2 degrés de liberté [4]. Si ce test n'est pas vérifié pour un des véhicules, alors une alerte peut être levée indiquant une situation potentiellement dangereuse car absente de la base de données ayant servi pour les apprentissages des dynamiques a priori nominales. Par la suite, ce même test peut être effectué entre les observations de tout égo-vehicule et les lois de prédiction des mesures des véhicules avoisinants pour affirmer (ou infirmer) un risque de collision. Si la mesure d'un des véhicules devient incompatible avec sa propre loi de prédiction de mesure ou que celle-ci devienne compatible avec la loi de prédiction de la mesure d'un autre véhicule, alors une alerte peut être levée. Afin d'optimiser le nombre de tests à réaliser, une heuristique peut être appliquée pour choisir les véhicules à tester, *i.e*, le véhicule le plus proche précédant l'égo-véhicule sur le segment routier.

4.7 Implémentation et étude de cas

4.7.1 Apprentissage

4.7.1.1 Données et segmentation

Les données d'apprentissage proviennent toutes du jeu de données HighD [106]. Enregistré en conditions réelles depuis des caméras haute-définition aéroportées, HighD fournit un grand ensemble de trajectoires véhiculaires autoroutières. La position et la vitesse des véhicules sont calculées à partir d'une segmentation vidéo réalisée par des algorithmes récents d'apprentissage profond. Les véhicules sont visibles sur un segment autoroutier d'environ 420 mètres. Pour notre problème, un sous-échantillonage par un facteur de 13 est appliqué aux données enregistrées initialement à 25 images par seconde. Ceci donne comme pas final entre chaque point de donnée : $T_s = 0.52$s. Cette résolution a été choisie empiriquement pour permettre à la fois de recueillir un nombre suffisant de mesures lors du processus de filtrage et par ailleurs de permettre l'identification de manœuvres dans le mouvement des véhicules. Un pas trop petit aura tendance à noyer dans le bruit les distinctions du mouvement par rapport aux manœuvres ; à l'inverse, un pas trop grand réduirait drastiquement le nombre de mesures agrégées sur le segment considéré tout en empêchant de découper le mouvement des véhicules en manœuvres distinctes. HighD est jugé suffisamment précis pour être utilisé

4. De manière très classique, en supposant la blancheur de l'innovation, la somme des innovations quadratiques normalisées sur une fenêtre glissante de taille L pourrait être testée relativement à une loi du khi-deux χ_{2L}^2 à $2L$ degrés de liberté afin de limiter des fausses alarmes, sans dégrader la capacité de détection pour L suffisamment faible.

en tant que substitut pour le jeu de données des véritables états cachés. Les positions et les vitesses longitudinales et latérales des véhicules sont collectées, ainsi que leurs largeurs et hauteurs [5].

Un total de 14 300 000 instances de données sont extraites de highD et sont séparées aléatoirement à 80% pour l'ensemble d'entraînement et à 20% pour celui de validation. L'ensemble est ensuite segmenté manuellement selon la section 4.4.1 pour M = 3 manœuvres : `forward`, `sheer-in` et `sheer-out`. De par leur méthode de segmentation, les instances de trajectoires correspondant aux deux dernières manœuvres comprennent trois phases. La préparation : tout en restant dans la même voie, le véhicule cible commence à moduler sa vitesse et sa position pour changer de voie ; le changement de voie ; la stabilisation : le véhicule reprend une trajectoire stable dans sa nouvelle voie. Les seuils de segmentation sont définis empiriquement depuis l'analyse des données par une connaissance experte. Ils sont choisis de façon que les trois phases puissent être correctement identifiées pour la majorité des trajectoires de HighD. Il est important de noter que selon notre segmentation, HighD est un dataset fortement déséquilibré avec 95% d'instances correspondant à la manœuvre `forward` et les derniers 5% répartis équitablement entre les manœuvres `sheer-in` et `sheer-out`. Le processus de segmentation conduit à l'ensemble $\left\{\left(x^k_{n-1}, \{x^l_{n-1}\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)}, m^k_n, x^k_n\right)\right\}$ utilisé dans l'apprentissage du modèle de transition égocentrique comme décrit dans la Section 4.4.

4.7.1.2 Un réseau de neurones pour la classification de manœuvre

Le FFNN de classification de manœuvre est composé de 2 couches cachées composées de 256 neurones chacune, avec Tanh comme fonction d'activation. Les entrées de ce réseau sont comme suit :

— sans perte de généralité, du fait qu'aucune rampe ou intersection n'est considérée, la position longitudinale passée $x^k_{n-1,\text{long_pos}}$ de l'égo-véhicule est fixée à 0 ; de manière équivalente, elle pourrait être fixée à la position passée absolue —au lieu de la position relative—afin de prendre en compte l'influence de la route sur la dynamique a priori de l'égo-véhicule (courbes, signalisation, etc.) ;

— la position latérale passée de l'égo-véhicule $x^k_{n-1,\text{lat_pos}}$;

— les vitesses longitudinales et latérales passées de l'égo-véhicule $x^k_{n-1,\text{vel}}$;

— les positions relatives par rapport à l'égo-véhicule de ses voisins $\{x^l_{n-1,\text{pos}} - x^k_{n-1,\text{pos}}\}_{\mathcal{V}_l \in \mathcal{N}_n(\mathcal{V}_k)}$;

— les vitesses absolues des voisins de l'égo-véhicule $\{x^l_{n-1,\text{vel}}\}_{\mathcal{V}_l \in \mathcal{N}_n(\mathcal{V}_k)}$.

5. Les largeurs et les hauteurs doivent être interprétées d'un point de vue aérien, *i.e.*, les largeurs sont longitudinales tandis que les hauteurs sont latérales.

Les sorties de ce réseau sont les probabilités manœuvre $\lambda_1, \ldots, \lambda_M$ conditionnellement aux entrées. Les poids du FFNN de classification sont mis à jour durant l'apprentissage depuis la vérité terrain $\{\mathsf{m}_\mathsf{n}^\mathsf{k}\}$ selon (4.10).

Lors de l'apprentissage de ce réseau avec l'ensemble du jeu de données HighD, de mauvaises performances de classification sont observées. Une précision équilibrée d'environ 49% est observée pour la classification des manœuvres lors de l'entraînement. Ce problème connu [126] découle de la propension qu'a la direction du gradient à se diriger vers des valeurs qui réduisent l'erreur de la classe majoritaire, délaissant ainsi la minimisation de l'erreur sur les classes minoritaires. Deux stratégies sont mises en place pour pallier les mauvaises performances du réseau induites par ce déséquilibre : il est possible de transformer l'objectif de classification en y ajoutant une pondération sur le coût des instances d'entraînements selon leur classe. En sur-pondérant les instances de `sheer-in` et de `sheer-out` par rapport aux instances de `forward`, la direction du gradient est ainsi rééquilibrée et ne privilégie aucune classe. Dans la pratique, plusieurs configurations de pondérations sont testées avec des résultats moyens sur l'amélioration des performances de classification. Un gain de 15 % est observé après une recherche empirique des meilleurs facteurs de pondération pour notre problème. Une autre solution repose sur le rééchantillonage des données d'entraînement. Dans notre cas, les données de la classe dominante sont sous-échantillonnées de 85%. Les autres classes sont sur-échantillonnées de 25%. De nouvelles données synthétiques sont créées depuis les véritables instances d'entraînement auxquelles un bruit Gaussien blanc est ajouté. La variance de ce bruit est très inférieure aux variances qui seront utilisées pour la génération des mesures de position des véhicules. Expérimentalement, cette méthode se montre très efficace avec un gain de 48%, portant ainsi la précision équilibrée du modèle à 97%. L'ensemble des performances est calculé depuis une validation croisée à 5 blocs sur les données.

4.7.1.3 Une banque de réseaux de neurones pour la régression des trajectoires

Après l'apprentissage du modèle de classification, M réseaux de neurones sont entraînés en parallèle. Un réseau est affecté à chaque manœuvre considérée. Ces réseaux partagent la même architecture et sont composés de deux couches cachées. La première et la deuxième couche sont composées de 256 et 128 neurones respectivement, avec comme fonction d'activation la fonction ReLU. Les entrées du FFNN sont similaires à celles du réseau de classification des manœuvres. Les sorties de chaque $m^{\text{ième}}$ réseau permettent d'obtenir la moyenne μ_m et la matrice de covariance Σ_m (paramétrées par les valeurs passées du véhicule cible et des états de ses voisins) de l'hypothèse Gaussienne conditionnellement à la manœuvre qui constitue le modèle de transition égocentrique. Les poids de ces FFNNs de régression sont mis à jour au cours de la phase d'entraînement en utilisant les $\mathsf{K}_m\mathsf{N}_m$ n-uplets $\left\{\left(\mathsf{x}_{\mathsf{n}-1,m}^\mathsf{k}, \{\mathsf{x}_{\mathsf{n}-1,m}^\mathsf{l}\}_{\mathcal{V}_{\mathsf{l}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_\mathsf{k})}}, \mathsf{x}_{\mathsf{n},m}^\mathsf{k}\right)\right\}$ de la vérité terrain construits autour

des K_m véhicules engagés dans la manœuvre indexée par m sur N_m instants selon (4.13).

Dans le cadre des objectifs d'explicabilité revendiqués, on remarquera que pendant l'entraînement de chaque FFNN de régression correspondant à une manœuvre, un test du khi-deux sur l'erreur de transition d'état quadratique normalisée indique si l'état actuel du véhicule cible peut être considéré comme un échantillon de la densité de probabilité Gaussienne apprise conditionnellement à la manœuvre et paramétrée par les états passés du véhicule cible et de ses voisins. Pour chaque manœuvre $m = 1, \ldots, M$, on vérifie que

$$\mathsf{r}^{\mathsf{k}}_{\mathsf{n},m}{}^{T}\Sigma_m^{-1}\mathsf{r}^{\mathsf{k}}_{\mathsf{n},m}, \text{ avec } \mathsf{r}^{\mathsf{k}}_{\mathsf{n},m} := \mathsf{x}^{\mathsf{k}}_{\mathsf{n},m} - \mu_m(\mathsf{x}^{\mathsf{k}}_{\mathsf{n}-1,m}, \{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1}\}_{\mathcal{V}_{\mathsf{l}}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})})$$

est un échantillon plausible d'une loi de khi-deux χ^2_4 à 4 degrés de liberté. Une moyenne de 95% de tests positifs est trouvée pour les trois réseaux. De plus, l'erreur en position et en vitesse des FFNNs de régression est submétrique. La Figure 4.7 donne les racines de l'erreur quadratique moyenne (Root Mean Square Error – RMSE) en position et en vitesse pour chaque FFNN associé aux trois manœuvres considérées dans notre étude de cas.

	RMSE x (m)	RMSE y (m)	RMSE vx (m.s^{-1})	RMSE vy (m.s^{-1})
FFNN `forward`	0.084	0.020	0.210	0.050
FFNN `sheer-in`	0.105	0.035	0.218	0.094
FFNN `sheer-out`	0.084	0.034	0.194	0.089

FIGURE 4.7 – RMSE en position et en vitesse obtenues après validation croisée sur les données HighD.

4.7.1.4 Implémentation

L'ensemble des FFNNs est entraîné sur une carte graphique dédiée en utilisant l'algorithme d'optimisation Adam avec un taux d'apprentissage de $\alpha = 0.01$ et une taille de batchs de 512 pour le réseau de classification de manœuvre et un taux de $\alpha = 0.001$ avec des batchs de taille 128 pour les réseaux de régression. Ces taux d'apprentissage sont réduits linéairement pendant l'entraînement. Les modèles sont implémenté en Python avec la bibliothèque PyTorch [127].

4.7.2 Estimation et prédiction sur scénarios autoroutiers difficiles

4.7.2.1 Description des scénarios choisis

HighD est un jeu de données très déséquilibré, avec une proportion importante de manœuvres *forward* facile à estimer et prédire. Dans le but de mettre à l'épreuve notre approche, douze scénarios de difficultés variables sont extraits des données, pour une durée totale de plus de deux minutes.

Les scénarios choisis manuellement ont pour but de représenter des situations typiques de trafic en interaction sur autoroute :

— Le *scénario 1* est composé de 8 véhicules circulant sur l'ensemble du segment d'autoroute sans changer de voie.

— Le *scénario 2* met en scène 6 véhicules avec plusieurs manœuvres de changement de voie : un véhicule s'insère d'abord entre deux autres, puis, un autre véhicule le dépasse et se place sur la voie la plus à droite.

— Dans le *scénario 3*, 2 véhicules dépassent un autre.

— Le *scénario 4* est un dépassement conventionnel entre deux véhicules.

— Le *scénario 5* et le *scénario 6* impliquent de multiples dépassements et changements de voie entre 5 et 6 véhicules, respectivement. La densité de circulation est fluide pour les deux scénarios.

— Le *scénario 7* et *scénario 8* sont composés d'une flotte de 5 et 6 véhicules respectivement. Dans chacun d'eux, la circulation y est dense et un véhicule de la flotte effectue un dépassement dans son entièreté.

— Dans le *scénario 9* et le *scénario 10*, un véhicule traverse en largeur le segment routier depuis la voie la plus a gauche vers celle la plus à droite tout en dépassant un ou plusieurs véhicules.

— Enfin, le *scénario 11* et le *scénario 12* sont constitués de 2 et 3 véhicules respectivement. Dans chaque scénario, l'un des véhicules se rabat sur sa voie de droite. Les manœuvres de dépassement qui pourraient avoir précédé les rabattements ne sont pas incluses.

4.7.2.2 Génération des mesures

L'ensemble des trajectoires de mesure $\{z_{0:n}^k\}$ est également dérivé des données HighD par l'ajout aux composantes de position des vecteurs d'état des véhicules un bruit de mesure blanc Gaussien à moyenne nulle et de variance donnée selon le modèle (4.8), avec

$$R_n^k = \begin{pmatrix} \sigma_{k(x)}^2 & 0 \\ 0 & \sigma_{k(y)}^2 \end{pmatrix}. \tag{4.27}$$

Deux valeurs des variances $\sigma_{k(x)}^2, \sigma_{k(y)}^2$ des bruits selon les axes x et y sont choisies. Elles émulent un intervalle de confiance à 99% de probabilité dont l'amplitude est $\pm\frac{1}{4}$ ou $\pm\frac{1}{2}$ de la taille d'un véhicule standard.

4.7.2.3 Description des stratégies comparées

Afin de mettre à l'épreuve l'approche proposée dans cette étude, plusieurs stratégies de prédiction et, le cas échéant, d'estimation prenant en compte les interactions sont implémentées et sont comparées sur les scénarios décrit précédemment. Les approches d'intérêt sont les suivantes.

— Notre approche reposant sur le Multiple Particle Filter (MPF) ou le Variational Bayesian Multiple Particle Filter (VBMPF) utilisant comme dynamique a priori un mélange de Gaussiennes dont les paramètres ont été appris comme décrit dans la Section 4.7.1. La Figure 4.8 illustre le modèle de transition utilisé pour les scénarios considérés. Afin de comparer à la fois le niveau de performance du VBMPF par rapport au MPF et les performances des deux algorithmes par rapport aux méthodes de la littérature présentées plus loin, le VBMPF est lancé avec 100 particules et 250 particules. Le nombre de particules pour le MPF est fixé à 100. La complexité du MPF induite par ce paramétrage est similaire à celle du VBMPF pour 250 particules.

— Afin de mettre en exergue les capacités de filtrage d'un état de grande dimension de notre approche, un Filtre Particulaire (PF) standard est implémenté. Il utilise le même modèle de dynamique que le MPF et le VBMPF. Pour rester dans des ordres de complexité comparables, le filtre utilise un ensemble de 10000 particules.

— Le modèle Social-LSTM [71] et le modèle similaire, le Convolutional Social Pooling LSTM [72] sont implémentés et configurés pour un apprentissage sur les données HighD. Des performances similaires à celles présentes dans la littérature sont observées dans le cas de trajectoires non bruitées. S-LSTM produit une prédiction sous la forme d'une distribution Gaussienne unimodale sur la position future des véhicules. Pour CS-LSTM, cette distribution est un mélange de Gaussiennes dont les poids sont issus d'un premier réseau récurrent et les paramètres de chaque hypothèse sont donnés par un second réseau appris en parallèle. Les dimensions de la grille requise dans les deux algorithmes sont adaptées en longueur et en largeur afin de capturer un nombre de véhicule voisins proche de celui de la méthode présentée dans cette thèse pour la portion d'autoroute considérée.

— Enfin, pour étudier l'impact du modèle de transition appris depuis des données véhiculaires, une comparaison est effectuée avec un Filtre de Kalman Étendu à Modèles Multiples Interactifs (Interactive Multiple Model Kalman Extended Filter – IMM EKF) [48] où les probabilités de transitions entre un modèles à vitesse constante et un modèle à vitesse de rotation et accélération constantes sont établies manuellement. La configuration des bruits de mesure et de dynamique ainsi que les probabilités de transition entre les modèles sont déterminées empiriquement de manière à obtenir des performances satisfaisantes lors de l'utilisation sur

les scénarios extraits de HighD. Pour les deux modèles considérés, l'ordre du vecteur d'état n'est pas identique pour tous les modes et l'algorithme de l'IMM-EKF ne peut donc pas être utilisé en l'état. L'implémentation d'un IMM-EKF spécifique à notre problème est détaillée dans l'Annexe A.

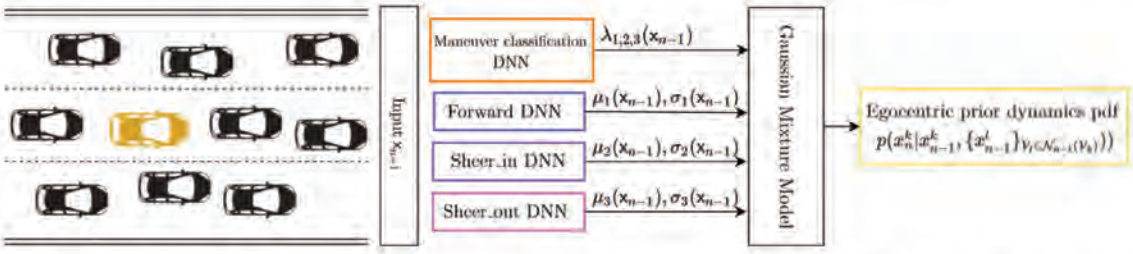


FIGURE 4.8 – Vue schématique du modèle de transition considéré pour les scénarios autoroutiers présentés. Les entrées du véhicule cible jaune et de ses voisins alimentent le réseau de classification et des trois réseaux de régression pour générer les paramètres du modèle de mélange de Gaussiennes (GMM) de la distribution a priori de la dynamique.

4.7.3 Résultats

4.7.3.1 Estimation

	Durée du filtrage ≈ 5s				
	RMSE x	RMSE y	RMSE v_x	RMSE v_y	# divergence
MPF GMM	0.03/0.13/0.22	0.06/0.30/0.49	0.17/0.23/0.40	0.23/0.48/0.72	0/0/0
VBMPF GMM (250 part.)	0.02/0.14/0.21	0.06/0.28/0.52	0.18/0.23/0.39	0.21/0.44/0.80	0/0/0
VBMPF GMM (100 part.)	0.06/0.15/0.28	0.07/0.31/0.62	0.19/0.29/0.42	0.23/0.47/0.83	0/0/0
IMM EKF	0.05/0.34/0.68	0.04/0.12/0.27	0.22/0.23/0.39	0.28/0.43/0.67	- / - / -
PF GMM	1.98/2.12/2.08	6.12/5.31/4.85	1.52/1.51/1.46	3.70/2.92/3.12	12/12/12

FIGURE 4.9 – Performances moyennes en estimation des algorithmes comparés. Les résultats sont calculés sur tous les scénarios avec trois niveaux de bruit différents sur les positions observées : aucun bruit / intervalle de confiance à 99% du bruit égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule). Pour chacun des filtres, les niveaux de bruit de mesure utilisés sont les suivants : intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)

Quatre des six algorithmes comparés produisent des estimés de l'état des véhicules, à savoir, l'ensemble des déclinaisons du filtre particulaire et le filtre de Kalman étendu. La Figure 4.9 résume les performances des 4 stratégies envisagées, sur la base des distributions Gaussiennes bivariées

qu'elles génèrent. Ainsi, les moyennes estimées servent à calculer les racines de l'erreur quadratique moyenne (RMSE), *i.e.*, les racines des moyennes des distances quadratiques par rapport à la vérité terrain. Le nombre de particules efficaces des filtres est surveillé. Lorsque ce nombre tombe à 1, ce phénomène est désigné sous le vocable « divergence ».

Les estimés produits par le MPF et le VBMPF sont dans leur ensemble très satisfaisants et peuvent servir de base pour la prédiction. Les stratégies employées pour gérer le filtrage d'un vecteur d'état de grande dimension se révèlent efficaces, et ce même pour des scénarios impliquant un nombre important de véhicules avec un bruit conséquent sur les mesures. Les performances entre le MPF et le VBMPF avec 250 particules sont sensiblement les mêmes avec un léger avantage pour ce premier lors de l'estimation avec le plus fort niveau de bruit. Même si les performances sont un peu en deçà de celles du MPF, les estimations du VBMPF avec 100 particules sont obtenues pour un temps d'exécution 10 fois moins important en moyenne sur les scénarios. Afin d'étudier l'impact des approximations VB, c'est cette version qui sera utilisée pour l'étape de prédiction. De l'autre côté, le PF souffre énormément de l'inefficacité d'échantillonner le vecteur d'état de grande dimension et diverge après quelques instants pour tous les scénarios considérés. Ces estimés ne peuvent pas être considérés comme exploitables pour notre application en raison de leur trop grande différence avec le véritable état caché. Enfin, l'estimation par l'IMM EKF est assez bonne en moyenne, mais elle s'avère trop pessimiste. Ces conclusions sont valables même si les bruits dynamiques et de mesure sont supposés être des processus aléatoires Gaussiens blancs indépendants de moyenne nulle et de faible variance. Ce pessimisme rend les estimés de ce filtre peu informatifs et donc difficilement exploitables pour la prédiction.

4.7.3.2 Prédiction

Dans cette partie, les six approches de prédiction de mouvement avec prise en compte des interactions sont comparées. Dans le cas du MPF et du VBMPF avec le modèle de transition égocentrique GMM appris, la prédiction est obtenue en invoquant séquentiellement la dynamique a priori sur le dernier instant de filtrage afin d'obtenir les lois de prédiction de plus en plus loin dans le futur. Pour le VBMPF, le nombre de particules choisi est de 100. La même remarque sur le processus itératif de la prédiction peut être faite pour le PF, qui utilise le même modèle de transition, et l'IMM EKF qui repose également sur l'utilisation de modèles à sauts Markoviens.

Les approches à base de réseaux récurrents fonctionnent de « séquence-à-séquence » : ils sont alimentés par une séquence de mesures d'entrée et produisent une séquence de paramètres décrivant les densités de probabilité sur les mesures futures par le biais d'un mécanisme interne de récursion [31]. La séquence d'entrée est construite sur la base de la même série d'observations utilisée dans les méthodes basées filtrage stochastique. La Figure 4.11 présente les résultats en pré-

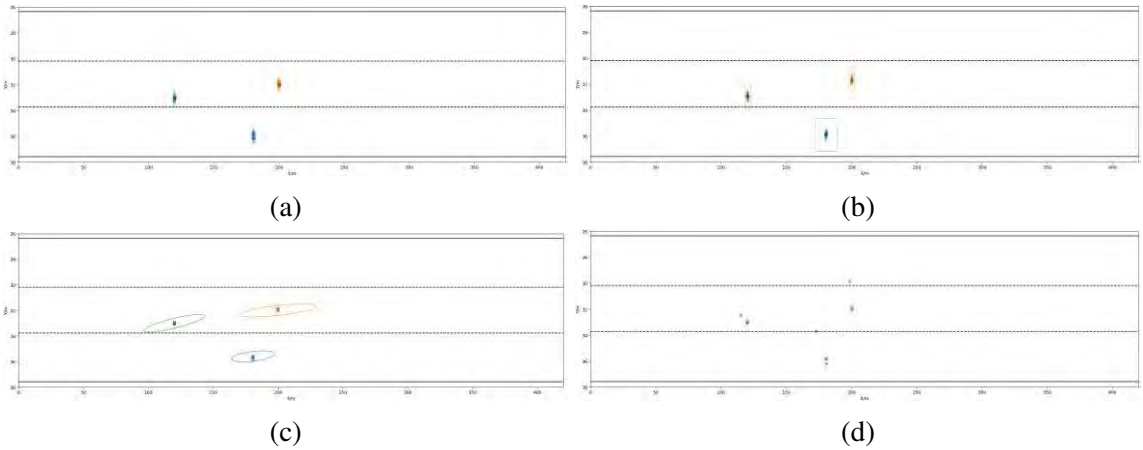


FIGURE 4.10 – Vue schématique aérienne de l'application des quatre algorithmes de filtrage sur un scénario routier pour lequel l'intervalle de confiance à 99% de probabilité du bruit de mesure est de $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). Chaque graphe correspond au quatrième instant du processus d'estimation pour lequel la position latérale (ordonnée) et longitudinale (abscisse) est affichée. Chaque véhicule est représenté par une couleur distincte. Les estimations sont représentées par des marqueurs ronds. Les ellipses de confiance qui leur sont associées devraient entourer les valeurs de la vérité terrain (marqueurs carrés) avec une probabilité de 99%. Les croix de couleur représentent les particules des filtres. Les positions mesurées et le bruit associé sont représentés par des étoiles et des ellipses en pointillés. Les voies sont droites et délimitées par des lignes pointillées. (a-b) Le MPF et le VBMPF estiment de manière satisfaisante la position de chaque véhicule dans la scène. (c) L'IMM EKF produit des estimés qui sont satisfaisants en termes de moyennes sur la position des véhicules. Cependant, la covariance d'erreur associées à ces estimés est trop conservative. (d) Quant au PF, les estimés produits sont mauvais car trop loin de la vérité terrain. A cet instant, l'algorithme diverge et le nombre de particules efficaces du filtre tend vers 1.

diction à moyen terme des algorithmes comparés sur l'ensemble des scénarios autoroutiers. Dans la même veine que pour l'estimation, ce sont les distributions Gaussiennes en prédiction qui permettent de calculer ces métriques de performance. Cependant les erreurs ici sont calculées dans l'espace des mesures et seulement en position. En plus des RMSEs, le taux de succès du *test du* χ^2 sur l'innovation quadratique normalisée par sa covariance théorique illustre la consistance de la prédiction. Enfin, le pourcentage de voie correcte (*% voie correcte*) indique si la position latérale prédite se trouve sur la voie de circulation convenable.

Après une estimation correcte, les prédictions en mesure du MPF et du VBMPF sont satisfaisantes et suffisamment informatives pour l'évaluation de risque, même durant les scénarios qui mettent en scène un nombre important de véhicules effectuant des manœuvres en interaction complexes. Comme illustré pour deux scénarios spécifiques dans la Figure 4.13, les bonnes estimations données par le MPF permettent une prédiction cohérente de l'état de la flotte avec de bons résultats en termes de *tests du* χ^2 sur les distributions produites et de tests de *voie correcte*. Par rapport au

MPF, les performances du VBMPF sont inférieures. Les approximations effectuées dans le processus de filtrage de l'algorithme le rendent plus efficace mais semblent impacter négativement la qualité des estimations et, par effet de bord, les performances en prédiction. De plus, les erreurs en prédiction pour le VBMPF sont concentrées sur l'axe latéral de la route. Sur un segment autoroutier, les changements de position sur cet axe induisent potentiellement de fortes interactions entre les véhicules. Il est donc crucial d'obtenir de bonnes performances sur les estimations et les prédictions latérales pour effectuer une évaluation du risque pertinente. Des visualisations supplémentaires des résultats en estimation et en prédiction pour différents scénarios sont présentées dans l'Annexe D. L'estimation et la prédiction basées sur le filtre particulaire standard sont négativement impactées par la grande dimension du vecteur d'état. Les divergences du filtre conduisent à de mauvaises estimations qui donnent ensuite des prédictions erronées avec de faibles taux de réussite des *% tests du* χ^2 et de *% de voie correcte*. Dans le cas sans bruit, le S-LSTM et le CS-LSTM produisent les meilleures prédictions de position. Cependant, lorsqu'il sont alimentés par des observations entachées de bruit, les performances des deux algorithmes diminuent considérablement. Ce phénomène, illustré dans la Figure 4.12 pour le LSTM, les rend inadaptés à l'évaluation des risques basée sur la prédiction du mouvement. Le MPF et le VBMPF parviennent à produire de bonnes prédictions pour toutes les différentes configurations de bruit. Enfin les performances en prédiction de l'IMM EKF sont moyennes et comme pour l'estimation, les intervalles de confiance sur la position future des véhicules sont beaucoup trop grands pour être utilisés pour l'évaluation du risque.

Afin d'étudier l'impact des données bruitées sur le S-LSTM et le CS-LSTM, plusieurs apprentissages sont effectués, chacun avec des niveaux de bruit différents sur les données d'entraînement. Ces différents modèles appris sont ensuite utilisés pour prédire les trajectoires des véhicules de la même manière que pour les approches présentées dans cette section. L'ensemble des résultats est présenté dans la Figure 4.14. Lorsque l'apprentissage est réalisé avec des entrées bruitées, on observe une amélioration importante des performances des deux algorithmes d'apprentissage profond. L'impact positif de l'ajout de bruit dans les données d'entraînement est un phénomène déjà observé dans la littérature [128]. Le bruit interviendrait comme un régularisateur [129] de la fonction de coût du réseau [130], limitant ainsi le sur-apprentissage et améliorant la généralisation du réseau. Le choix du niveau de bruit est primordial lors de l'apprentissage. Trop peu de bruit conduit à une généralisation non-satisfaisante comme observé pour le Social-LSTM entraîné sur les données entachées avec le premier niveau de bruit. Trop de bruit entraînerait un sous-apprentissage et des performances de prédiction en position dégradées. Dans la référence [72], les auteurs suggèrent l'utilisation de couches de convolution dans le tenseur social pour le rendre plus robuste aux instances de test non-observées pendant l'apprentissage, notamment pour le cas où les voitures

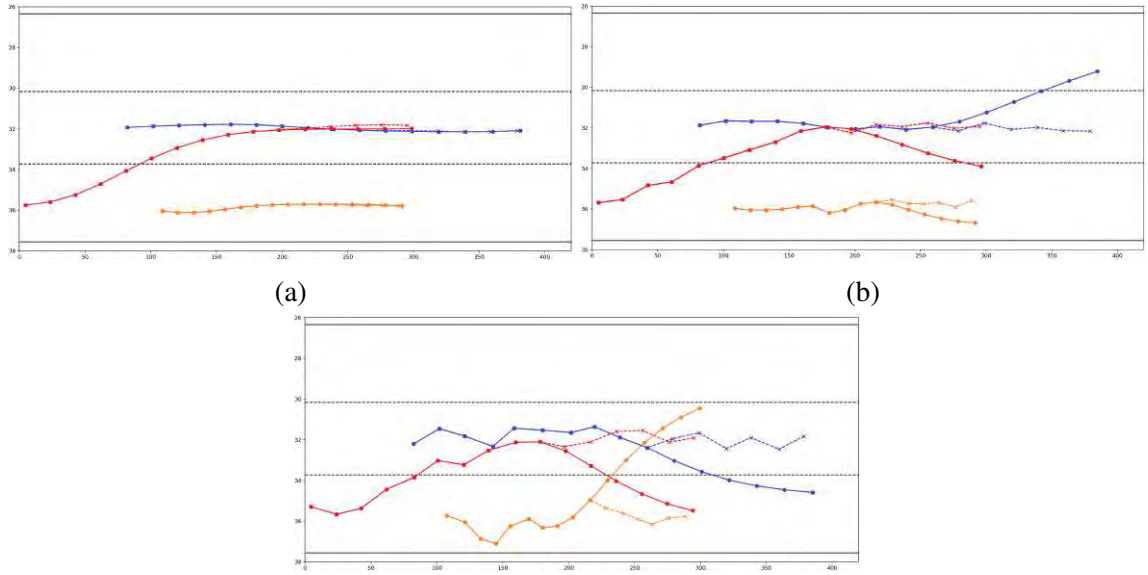
ont des configurations proches en termes de position mais une répartition sur la grille différente. A travers nos expériences, le pouvoir supplémentaire de généralisation du CS-LSTM serait vérifié. Il semblerait qu'à partir d'un certain niveau de bruit sur les données d'apprentissage, le réseau parvient à être performant en prédiction et ce même pour des données entachées d'un bruit beaucoup plus important que celui utilisé pour apprendre le réseau. Nous ne sommes pas parvenu à expliquer rigoureusement un tel changement de performance en prédiction du réseau lors du passage d'un apprentissage sans bruit à un apprentissage avec des trajectoires bruitées. Enfin, ce dernier point est à nuancer avec la nécessité de trouver manuellement ou par le biais d'apprentissages répétés, le niveau de bruit optimal pour les données observées. Ceci rend ainsi la mise en place du modèle plus fastidieuse et moins évolutive par rapport aux approches reposant sur le VBMPF et MPF.

	Prédiction sur un horizon de 3s	
	% de test du χ^2	% voie correcte
MPF GMM	99.13/99.13/98.53%	92/91.02/91.02%
VBMPF GMM	98.35/96,22/95.61%	87.14/84.64/80.98%
KFE IMM	98.22/96.12/97.21%	96.78/94.43/78.86%
PF GMM	0/2.50/13.17%	30.83/19.50/18.33%
S-LSTM	97.13/74.92/57.71%	95.14/93.86/90.47%
CS-LSTM	98.22/45.61/22.92%	99.13/71.56/49.54%

	MPF GMM		VBMPF GMM		IMM EKF	
Horizon de prédiction (s)	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE
1	0.41/0.60/1.23	0.31/0.39/0.55	0.47/0.80/0.84	0.32/0.45/0.78	0.40/0.68/1.36	0.41/0.55/0.63
2	0.99/1.07/1.59	0.51/0.58/0.66	1.05/1.36/1.62	0.62/0.83/1.01	1.08/1.73/1.98	0.55/0.67/0.85
3	1.86/1.53/2.60	0.71/0.80/0.72	1.80/2.11/2.66	0.83/0.99/1.17	2.12/2.44/3.19	0.85/1.00/1.63

	PF GMM		S-LSTM		CS-LTSM	
Horizon de prédiction (s)	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE
1	26.98/23.7/28.42	3.30/2.82/3.89	0.36/1.54/1.85	0.04/0.39/0.48	0.15/0.75/1.52	0.02/0.34/0.55
2	28.12/22.52/29.98	3.22/4.27/3.93	1.01/3.32/5.67	0.17/0.68/0.72	0.52/2.34/5.31	0.12/0.98/1.00
3	39.58/17.28/32.07	4.47/5.11/4.95	1.68/6.46/9.51	0.36/0.85/0.93	0.84/4.60/9.09	0.23/1.41/1.89

FIGURE 4.11 – Performances moyennes en prédiction des algorithmes comparés. Les résultats sont calculés sur tous les scénarios avec trois niveaux de bruit différents sur les positions observées : aucun bruit / intervalle de confiance à 99% du bruit égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule). Pour chacun des filtres, les niveaux de bruit de mesure utilisés sont les suivants : intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)



(a) (b)

(c)

FIGURE 4.12 – Vue aérienne d'une prédiction à trois secondes du Social-LSTM sur le Scénario 3 (trois véhicules) pour divers niveaux de bruit de mesure. (a) : Aucun bruit de mesure. (b)-(c) : Un intervalle de confiance sur le bruit de mesure à 99% de probabilité égal à $\pm\frac{1}{4} \times$ (la taille standard d'un véhicule) ou $\pm\frac{1}{2} \times$ (la taille d'un véhicule standard), respectivement. Les mesures des véhicules correspondant à la vérité terrain sont représentées par des lignes en tirets et des croix. Les mesures prédites du Social-LSTM sont représentées par des lignes continues et des marqueurs ronds. Avant la prédiction, les deux trajectoires se superposent. Les voies sont délimitées par des lignes pointillées. Les résultats sans bruit (a) sont entièrement satisfaisants, alors que le Social-LSTM souffre beaucoup du bruit ajouté aux observations (b-c).

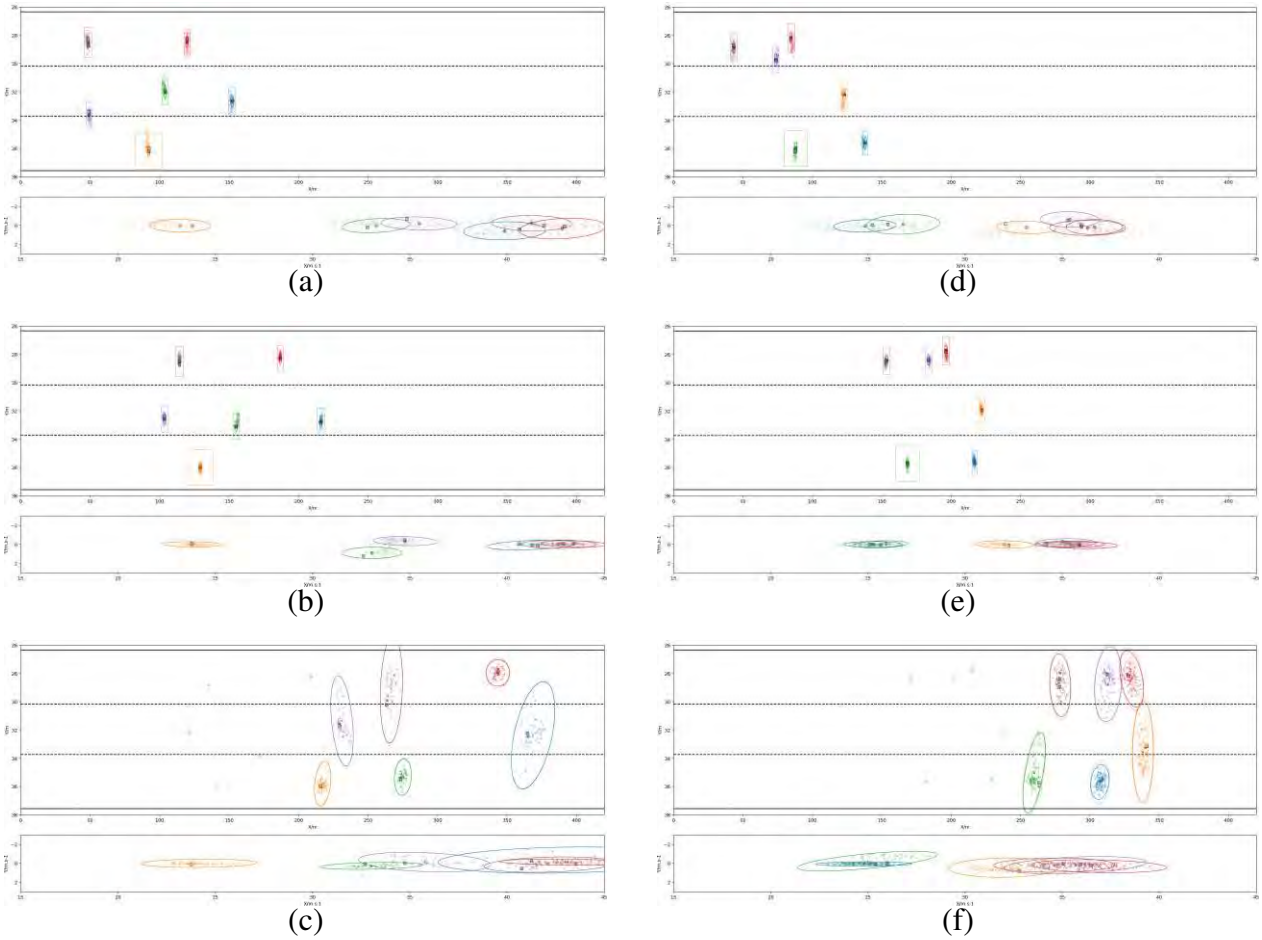


FIGURE 4.13 – Vue schématique aérienne de l'application du MPF sur deux scénarios autoroutiers pour lesquels l'intervalle de confiance à 99% de probabilité du bruit de mesure est de $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). Chaque paire de graphes étiquetée par (a),(b),. . . ,(f) affiche (haut) la position latérale en ordonnée et longitudinale en abscisse ; (bas) La vitesse est affichée de la même manière. Chaque véhicule est représenté par une couleur distincte. Les estimations sont représentées par des marqueurs ronds. Les ellipses de confiance qui leur sont associées devraient entourer les valeurs de vérité terrain (marqueurs carrés) avec une probabilité de 99%. Les boîtes de délimitation centrées sur les estimations de position représentent la taille des véhicules correspondants. Les croix de couleur représentent les particules du filtre. Les positions mesurées et le bruit associé sont représentés par des étoiles et des ellipses en pointillés. Les voies sont droites et délimitées par des lignes pointillées. (a-b-d-e) : Les six véhicules du Scénario 5 et du Scénario 6 sont correctement estimés sur l'ensemble du segment routier. (c-f) : Après avoir été estimé, l'état des véhicules est correctement prédit sur un horizon de 3s. Les moyennes prédites sont sur les bonnes voies et les covariances incluent la vérité terrain. Il convient de noter que les boîtes de délimitation apparaissent déformées en raison du rapport long/lat important de la vue. Pour faciliter la visualisation, le nombre de particules affichées a été réduit.

	prédiction sur un horizon de 3s	
	% de test du χ^2	% voie correcte
S-LSTM - données bruitées $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule)	76.21/96.48/93.33%	92.98/96.7/96.20%
S-LSTM - données bruitées $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)	75.35/95.23/94.11%	87.23/96.31/95.87%
CS-LSTM - données bruitées $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule)	95.24/96.49/91.81%	96.18/96.66/95.63%
CS-LSTM - données bruitées $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)	95.67/94.87/94.34%	94.11/93.79/93.96%

	S-LSTM - données bruitées $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule)		S-LSTM - données bruitées $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)	
Horizon de prédiction (s)	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE
1	1.68/0.22/0.32	0.23/0.16/0.17	0.65/0.29/0.34	0.32/0.23/0.27
2	2.19/1.01/1.39	0.41/0.22/0.34	0.97/1.16/1.31	0.49/0.36/0.34
3	3.46/1.36/1.71	0.60/0.35/0.40	2.77/1.74/1.89	0.75/0.50/0.51

	CS-LSTM - données bruitées $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule)		CS-LSTM - données bruitées $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule)	
Horizon de prédiction (s)	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE
1	0.19/0.22/0.31	0.13/0.16/0.18	0.40/0.29/0.32	0.26/0.27/0.29
2	0.65/0.74/0.88	0.22/0.20/0.28	1.01/3.32/5.67	0.42/0.43/0.45
3	1.21/1.36/1.83	0.37/0.35/0.42	1.46/1.28/1.46	0.60/0.56/0.57

FIGURE 4.14 – Performances moyennes de prédiction pour le S-LSTM et CS-LSTM entraînés avec deux niveaux de bruit différents sur les données d'entrée. Les résultats sont calculés sur tous les scénarios avec trois niveaux de bruit de mesure différents : aucun bruit de mesure / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule).

CHAPITRE 5
APPRENTISSAGE NON SUPERVISÉ DE LA LOI DE DYNAMIQUE A PRIORI ÉGOCENTRIQUE PAR UNE ARCHITECTURE NEURONALE INTERPRÉTABLE

5.1 Introduction

Le chapitre précédent a proposé un apprentissage supervisé de la dynamique a priori d'un égo-véhicule en interaction. Les paramètres d'un modèle à mélange de Gaussiennes (GMM) ont été établis par une série de réseaux de neurones profonds à propagation avant (FFNNs). Pour les besoins de cet apprentissage, une segmentation manuelle préalable des données était nécessaire. Celle-ci fait appel à une connaissance experte pour étiqueter les trajectoires des véhicules en un ensemble prédéfini de manœuvres. Ce processus peut s'avérer fastidieux à mesure que l'environnement où circulent les véhicules se complexifie et que le nombre de manœuvres admissibles augmente. Afin de permettre l'application de méthodes d'estimation et de prédiction sans nécessiter une segmentation préalable, l'apprentissage non supervisé de la dynamique égo-centrique est abordé ci-dessous, constituant ainsi une étape vers l'utilisation potentielle de notre méthode d'estimation-prédiction dans des environnements plus complexes tels que les intersections ou les ronds-points. Les nouvelles méthodes et les nouveaux modèles d'apprentissage automatique étudiés doivent respecter les objectifs d'interprétabilité établis précédemment, et doivent pouvoir être utilisés dans un schéma d'estimation stochastique impliquant un vecteur d'état défini explicitement.

Ce chapitre traite presque intégralement de l'évolution de la partie hors-ligne du schéma global détaillé dans la section 4.2. Nous nous concentrons donc sur l'apprentissage non supervisé hors-ligne de la dynamique a priori de véhicules en interaction sur un segment autoroutier. Pour ce faire, un type particulier de réseau de neurones est sélectionné afin de remplacer les FFNNs utilisés auparavant. Plusieurs apprentissages sont effectués, à la fois sur des données synthétiques et sur les données HighD. Dans le cadre de notre problématique ITS, l'interprétabilité et les performances en prédiction de ces nouveaux modèles sont étudiées en détail. La fin du chapitre propose l'incorporation, dans l'un des deux schémas de filtrage, de la loi de dynamique a priori égocentrique ainsi apprise de manière non supervisée, ainsi que l'évaluation du schéma d'estimation-prédiction sur certains scénarios autoroutiers.

5.2 Une dynamique a priori interprétable apprise de manière non supervisée

Un système de prédiction du déplacement d'une flotte de véhicules pour l'évaluation du risque est critique, au sens où les résultats produits peuvent engendrer des conséquences importantes pour ses utilisateurs. Afin de justifier de son bon fonctionnement, un premier prérequis semble être la capacité à interpréter le modèle sous-jacent pour comprendre le processus génératif des données de sortie. Autrement dit, pour notre application, il doit être possible d'expliquer les facteurs génératifs du mouvement futur de l'égo-véhicule.

La loi de dynamique a priori égocentrique de type GMM sélectionnée dans le chapitre 4, apprise par des techniques neuronales, présente un certain degré d'interprétabilité. En effet, bien que la contribution des poids et biais des FFNNs sur la production des sorties soit difficile à caractériser, le rôle de chaque FFNN dans l'apprentissage des paramètres du GMM (poids, moyennes, covariances) est très clair : les poids des hypothèses du GMM capturent la probabilité que l'égo-véhicule engage telle ou telle manœuvre ; les moyenne et covariance de chaque hypothèse décrivent la distribution du futur à un pas de l'égo-véhicule conditionnellement à la manœuvre associée.

L'objectif est donc d'apprendre de manière non supervisée une loi de dynamique a priori égocentrique, comme introduit dans la section 2.4. Un modèle discriminatif (voir Section 2.4.4) qui relierait directement l'état de la flotte à un futur possible de l'égo-véhicule sans possibilité d'interprétation du modèle n'a que peu d'intérêt pour notre problème. Nous nous orientons donc naturellement vers les modèles génératifs profonds, qui estiment la distribution de probabilité des données sur la base d'un nombre fini d'échantillons pour permettre ensuite la génération de nouvelles instances. Parmi la multitude de réseaux génératifs profonds proposés dans la littérature, deux grandes familles sont particulièrement plébiscitées pour leurs performances : les Réseaux Génératifs Antagonistes (Generative Adversarial Networks – GANs) [131] et les Auto-Encodeurs Variationnels, conditionnels ou non, ((Conditional) Variational AutoEncoders – (C)VAEs) [34, 35]. Les GANs sont ancrés dans la théorie des jeux alors que les (C)VAEs reposent sur l'inférence Bayésienne variationnelle. Une des différences fondamentales entre les deux modèles réside dans la présence (ou non) de variables latentes. Les VAEs cherchent à encoder les facteurs génératifs d'un ensemble de données (supposées i.i.d. selon une loi inconnue) dans des variables latentes aléatoires, et établissent la loi jointe des données et des variables latentes dans un schéma où : la loi a priori des variables latentes est en général prédéfinie (typiquement une loi Gaussienne standard multivariée), plus rarement apprise ; la génération des données conditionnellement aux variables latentes est capturée dans un modèle dit « génératif » ; la loi postérieure des variables latentes conditionnellement aux données est approximée par un modèle dit « de reconnaissance » ; les modèles de reconnaissance et génératif consistent en des lois de probabilité dont les paramètres sont délivrés par des

réseaux de neurones entraînés conjointement et respectivement désignés par « (réseau) encodeur » et « (réseau) décodeur ». Les CVAEs s'inscrivent dans la même veine, mais l'objectif est de caractériser des données de sortie conditionnellement à des données d'entrée. Par rapport au VAE, cette caractéristique est intéressante. Pour le problème traité dans ce chapitre, les données de sortie et d'entrée représentent respectivement l'état futur de l'égo-véhicule et l'état de la flotte à l'instant courant. Dans leur forme standard, les GANs ne permettent pas de mettre en correspondance des variables latentes internes avec des données. De plus, la fonction objectif impliquée dans l'entraînement des (C)VAEs étant une approximation de la vraisemblance (par rapport aux données) des paramètres des réseaux encodeur et décodeur, il est possible de l'utiliser pour comparer différents résultats d'apprentissage. Ce n'est pas le cas pour les GANs de par la forme de leur fonction objectif qui impose de mesurer leur performance via une étude de la qualité de leur prédictions (ou de leur reconstructions). La mesure de la performance des GANs est un sujet actif de la littérature [132]. Enfin, les GANs requièrent un nombre conséquent de données pour converger [133]. Comme expliqué tout au long de ce chapitre, l'interprétabilité d'un (C)VAE repose précisément sur la structuration de son espace latent.

Les (C)VAEs ont été utilisés dans de nombreuses applications où une modélisation explicite de la distribution des données est nécessaire, comme en vision par ordinateur [134], en traitement du langage naturel [135] ou encore dans la littérature des ITS [136]. Les performances des (C)VAEs en terme de prédiction ainsi que leur interprétabilité potentielle semblent en faire des candidats de choix pour l'apprentissage de la dynamique a priori égocentrique de véhicules en interaction.

5.2.1 Un auto-encodeur variationnel conditionnel pour modéliser la dynamique a priori

Les variables latentes du CVAE peuvent servir l'objectif d'interprétabilité recherché dans cette étude. Pour notre application, ces variables doivent encoder l'information sur les manœuvres associée aux trajectoires continues sans qu'il soit nécessaire de segmenter et étiqueter préalablement le jeu de données. Autrement dit, le CVAE viendrait remplacer le réseau de classification accompagné des réseaux de régression présentés dans le chapitre précédent pour appendre la dynamique a priori de l'égo-véhicule. Si les variables latentes du CVAE vivent dans un espace de faible dimension et encodent de manière démêlée les facteurs génératifs des manœuvres de l'égo-véhicule, alors un opérateur humain pourrait interpréter leurs valeurs de façon à exhiber la structure profonde des trajectoires constituant l'ensemble de données.

5.2.2 Apprentissage de représentation et variables latentes démêlées

L'apprentissage de représentation (representation learning) [137] est un domaine de l'apprentissage automatique qui étudie les techniques permettant d'extraire automatiquement de l'information

pertinente des données observées.

La Figure 5.1 illustre le concept d'apprentissage de représentation pour un VAE dans le cadre d'un apprentissage sur des données images.

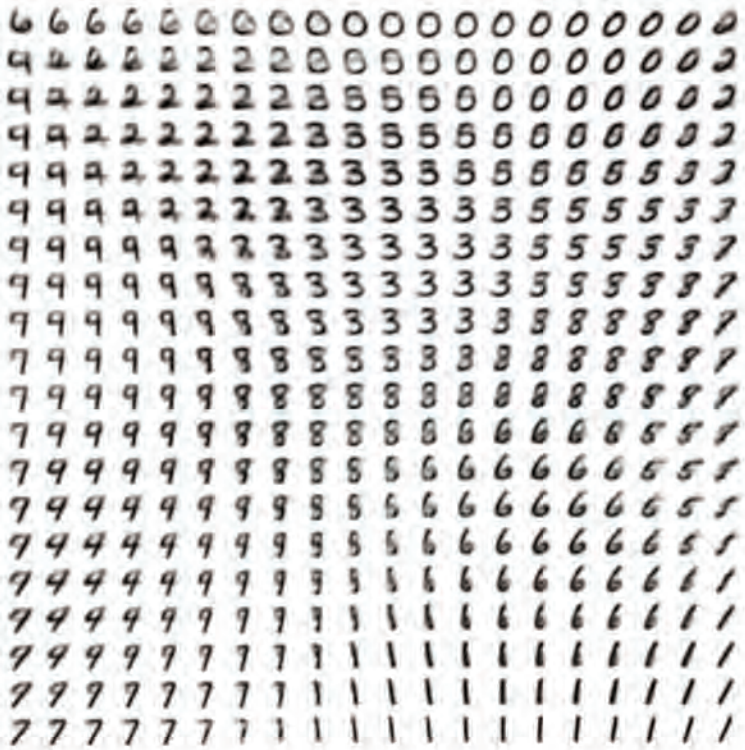


FIGURE 5.1 – D'après [34]. Visualisation de la représentation apprise depuis l'entraînement d'un VAE avec un espace latent à deux dimensions sur les données images MNIST de chiffres manuscrits [138]. Afin d'exhiber la représentation démêlée obtenue, on considère des échantillons de z Gaussien bi-dimensionnel et de composantes mutuellement indépendantes. Conditionnellement à chacun de ces derniers, des échantillons du décodeur sont tirés et affichés. Leur position sur les deux axes est donnée par la valeur échantillonnée de chaque composante de z. La variable latente semble donc ici avoir encodé le chiffre écrit.

Une représentation impliquant des facteurs génératifs mutuellement indépendants est qualifiée de représentation démêlée ou désenchevêtrée (« disentangled representation ») [137]. Dans le cas général, une représentation démêlée peut être définie comme une représentation où les unités latentes d'un modèle sont sensibles aux changements de facteurs génératifs uniques, tout en étant relativement invariantes aux changements sur les autres facteurs. L'apprentissage non supervisé d'une représentation démêlée sur la base de données réelles est un défi majeur [139]. De multiples auteurs [137, 139, 140, 141] défendent le fait que de telles représentations amélioreraient les performances des modèles d'apprentissage automatique dans les cas où un transfert de connaissance (Transfer Learning – TL) est nécessaire, lorsque peu d'instances d'entraînement sont disponibles

(Zero- ou Few-Shot Learning), ou bien pour la détection d'anomalies.

Pour notre problème, une représentation démêlée idéale pourrait par exemple impliquer : une première variable latente encodant le degré d'appartenance de la trajectoire à un ensemble de manœuvres possibles ; une deuxième variable, indépendante de la première, encodant la variabilité des comportements des conducteurs dans la réalisation de leur trajectoire, *e.g.*., permettant de distinguer les conducteurs agressifs des conducteurs prudents.

5.2.3 CVAE *Vanilla*

5.2.3.1 Introduction

Le premier modèle implémenté pour représenter la dynamique a priori égocentrique est le CVAE sous sa forme originale [35], ci-dessous désigné par CVAE *Vanilla*. Des éléments importants pour la suite sont rappelés ci-dessous.

5.2.3.2 Modélisation

On considère un ensemble de données $\mathcal{X}_{\text{dataset}} = \{x^{(i)}, y^{(i)}\}_{i=1}^{D}$ où chaque $y^{(i)}$ est un échantillon d'une variable aléatoire y dont la loi est fonction de $x^{(i)}$. On note leur loi jointe $p_{\mathcal{D}}(x, y)$. Les $x^{(i)}$ et les $y^{(i)}$ sont considérées comme les entrées et les sorties du réseau respectivement. Le processus de génération des données se déroule comme suit : une instance de $z^{(i)}$ est obtenue par échantillonnage de la distribution a priori $p_\theta(z|x^{(i)})$ puis $y^{(i)}$ est obtenue par échantillonnage d'une autre distribution notée $p_\theta(y|x^{(i)}, z^{(i)})$. Il est important de remarquer que $y^{(i)}$ et $z^{(i)}$ sont toutes les deux conditionnées par une entrée connue. La loi jointe des vecteurs de sortie et des variables latentes s'écrit donc

$$p_\theta(y, z|x) = p_\theta(y|x, z)p_\theta(z|x). \tag{5.1}$$

Les densités de probabilité $p_\theta(z|x)$ et $p_\theta(y|x, z)$ sont spécifiées via des fonctions du vecteur θ des paramètres d'un réseau de neurones faisant l'objet d'un apprentissage.

En marginalisant par z dans (5.1), on obtient donc le modèle génératif des sorties conditionnellement aux entrées :

$$p_\theta(y|x) = \int p_\theta(y|x, z)p_\theta(z|x)dz. \tag{5.2}$$

5.2.3.3 Estimation des paramètres via la borne inférieure variationnelle de l'évidence

Afin de produire des trajectoires vraisemblables quant au jeu de données, il est nécessaire d'estimer les paramètres θ du modèle. L'équation (5.2) met en exergue deux problèmes liés à ce processus : comment inférer la distribution de la variable latente z capturant l'information sur le processus

de génération de y conditionnellement à x ; comment approcher efficacement l'intégrale sur les z, celle-ci étant intractable de par le nombre de valeurs que peut prendre cette dernière.

Le premier problème implique d'être capable de calculer, pour un θ donné, la loi postérieure de z telle que $p_\theta(z|x,y) = \frac{p_\theta(y|x,z)p_\theta(z|x)}{p_\theta(y|x)}$. L'intégrale dans (5.2), rend cette quantité également intractable. Le deuxième problème suggère d'être en mesure d'obtenir efficacement des échantillons vraisemblables de z par rapport à y conditionnellement aux entrées x pour pouvoir calculer (5.2).

Le CVAE contourne ces deux obstacles conjointement en introduisant une approximation $q_\phi(z|x,y)$ de la loi postérieure $p_\theta(z|x,y)$ du vecteur de variables latentes, appelée « modèle de reconnaissance ». Cette distribution est définie de sorte à rendre possible le calcul de la loi postérieure (approchée) de z. Les échantillons obtenus de z sont ensuite utilisés pour calculer efficacement une approximation de $p_\theta(y|x)$ et de permettre une optimisation des paramètres du modèle.

Tout comme la loi $p_\theta(z|x)$ de z conditionnellement aux entrées et le modèle génératif $p_\theta(y|x,z)$, le modèle $q_\phi(z|x,y)$ est une distribution de probabilité spécifiée via des fonctions du vecteur ϕ des paramètres d'un réseau de neurones. Les vecteurs θ, ϕ font alors l'objet d'un apprentissage conjoint. Le modèle de reconnaissance $q_\phi(z|x,y)$ est naturellement appelé « encodeur », et le réseau de neurones associé est dit « réseau encodeur ». De même, le modèle génératif $p_\theta(y|x,z)$ et le réseau de neurones associé sont dits « décodeur » et « réseau décodeur ». La Figure 5.2 présente les relations entres les différentes variables impliquées dans le CVAE. La contrainte de dépendance de $p_\theta(z|x)$ aux entrées est généralement relaxée, et cette loi est souvent définie comme une Gaussienne multivariée standard, de sorte que $p_\theta(z|x) = p_\theta(z) = p(z)$.

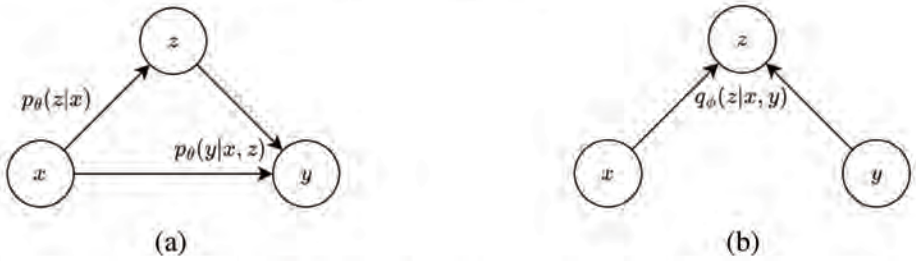


FIGURE 5.2 – Représentation en graphes (a) du processus de génération des variables de sortie y et (b) du processus d'encodage des variables latentes z.

La fonction objectif d'optimisation du CVAE est la maximisation via les paramètres du réseaux de l'espérance sur les données de sa log-évidence conditionnelle $\log p_\theta(y|x)$:

$$\arg\max_\theta \mathbb{E}_{x,y\sim p_\mathcal{D}(x,y)}[\log(p_\theta(y|x))] \tag{5.3}$$

De par l'impossibilité de la calculer cette quantité analytiquement, les paramètres du CVAE sont

estimés, via une approximation. Les équations suivantes sont en vigueur

$$\log p_\theta(y|x) = \mathcal{L}(x, y, \theta, \phi) - D_{KL}(q_\phi(z|x, y)||p_\theta(z|x, y)), \tag{5.4}$$

$$\text{avec } \mathcal{L}(x, y, \theta, \phi) = \mathbb{E}_{z\sim q_\phi(z|x,y)}[\log p_\theta(y|x, z)] - D_{KL}(q_\phi(z|x, y)||p_\theta(z|x)). \tag{5.5}$$

Le second terme de (5.4) est la divergence de KullBack-Leibler (KLD) de la loi postérieure du vecteur des variables latentes vers le modèle décodeur. En raison de la positivité de toute KLD, le premier terme de (5.4) constitue une borne inférieure de la log-évidence, appelée « borne inférieure de l'évidence » (Evidence Lower BOund – ELBO). Cette borne et ses dérivées par rapport aux paramètres θ et ϕ sont estimés à l'aide de l'algorithme de descente de gradient stochastique Bayésien variationnel (Stochastic Gradient Variational Bayes – SGVB) [34]. Ces estimés servent ensuite à maximiser l'objectif d'apprentissage du CVAE. L'équation (5.5) développe l'ELBO comme la différence de deux termes. Le second, qui est la KLD de la loi a priori du vecteur des variables latentes vers le modèle encodeur, peut être exprimé analytiquement dès lors que ces lois sont Gaussiennes. Le premier terme, généralement appelé « erreur de reconstruction », peut être approché en tirant aléatoirement L échantillons $z^{(l)}$, $(l = 1, 2, \ldots, L)$ i.i.d. selon $q_\phi(z|x, y)$, ce qui conduit à l'ELBO empirique

$$\tilde{\mathcal{L}}(x, y, \theta, \phi) = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(y|x, z^{(l)}) - D_{KL}(q_\phi(z|x, y)||p(z|x)), \tag{5.6}$$

$$\text{où } z^{(l)} = g_\phi(x, y, \epsilon^{(l)}), \text{ et } \epsilon^{(1)}, \ldots, \epsilon^{(L)} \overset{\text{i.i.d.}}{\sim} p(\epsilon). \tag{5.7}$$

La méthode (5.7) d'obtention des échantillons $z^{(l)}, \ldots, z^{(L)}$ est appelée « astuce de reparamérisation » (« reparametrization trick »). Elle nécessite naturellement la définition d'une loi $p(\epsilon)$ et d'une fonction $g_\phi(.)$ adéquates, ce qui ne pose pas de problème si le modèle encodeur est Gaussien. Si $g_\phi(.)$ est déterministe et différentiable, alors le gradient de l'ELBO empirique par rapport aux vecteurs de paramètres θ, ϕ peut être calculé et permettre un entraînement des réseaux encodeur et décodeur. En pratique, lors de l'apprentissage, le nombre d'échantillons L est fixé à 1.

Le processus d'apprentissage des CVAEs peut être vu comme l'estimation des paramètres optimaux du modèle qui reconstruisent le plus fidèlement les sorties y conditionnellement aux entrées x et à un « code » latent représenté par les valeurs possibles de z. En pratique, la dimension choisie pour le codage latent est très inférieure à celle des entrées et sorties. Les deux termes de la fonction objectif (5.5) peuvent être interprétés comme deux sous-objectifs en compétition, au sens où $D_{KL}(q_\phi(z|x, y)||p_\theta(z|x))$ agit comme un régularisateur de l'objectif de reconstruction $\mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(y|x, z)$, enjoignant l'encodeur à se rapprocher de la loi a priori du vecteur des

variables latentes.Une fois les paramètres du réseau appris, l'inférence peut être réalisée. Durant ce processus, le modèle encodeur n'est pas utilisé. La variable latente z est échantillonnée selon $p_\theta(z|x)$ ou $p_\theta(z)$ (ci-après désignés par l'anglicisme « prior ») puis le décodeur $p_\theta(y|x,z)$ génère ensuite une instance de la variable de sortie y.

5.2.3.4 Le cas d'un CVAE avec encodeur et décodeur Gaussiens et un prior non-appris

Par la suite, nous supposerons que les réseaux de neurones encodeur et décodeur sont des réseaux de neurones à propagation avant (FFNNs) qui produisent tous les deux en sortie les moyennes et matrices de covariance diagonales de distributions Gaussienne multivariées, respectivement :

$$q_\phi(z|x,y) = \mathcal{N}(z; \mu_{\text{enc}}(x,y;\phi), \text{diag}(\sigma^2_{\text{enc}}(x,y;\phi))) \tag{5.8}$$

$$\text{et } p_\theta(y|x,z) = \mathcal{N}(y; \mu_{\text{dec}}(x,z;\theta), \text{diag}(\sigma^2_{\text{dec}}(x,z;\theta))). \tag{5.9}$$

où l'opérateur $\text{diag}(\cdot)$ est défini tel que $\text{diag}(v) = \boldsymbol{I}_N \odot (v^T u)$ avec $\boldsymbol{I}_N$ la matrice identité de dimension $N \times N$, $v = [v_1, \ldots, v_N] \in \mathbb{R}^N$ et $u = [1, \ldots, 1] \in \mathbb{R}^N$ et où $\odot$ indique l'opération du produit terme à terme. μ_{enc}, μ_{dec}, σ^2_{enc} et σ^2_{dec} sont tous des vecteurs. Par abus de notation, σ^2_{enc} et σ^2_{dec} désigne ici le vecteur des carrés terme à terme de σ_{enc} et σ_{dec} respectivement.

Le réseau encodeur paramétré par ϕ établit la loi du vecteur $z \in \mathbb{R}^N$ de N variables latentes Gaussiennes mutuellement indépendantes, de moyennes et variances respectives $\mu_{1,\text{enc}}, \ldots, \mu_{N,\text{enc}}$ et $\sigma^2_{1,\text{enc}}, \ldots, \sigma^2_{N,\text{enc}}$ (composantes des vecteurs μ_{enc} et σ^2_{enc} dans (5.8)) si ses sorties $y^{\mu_1}_{enc}, \ldots, y^{\mu_N}_{enc}$ et $y^{\sigma^2_1}_{enc}, \ldots, y^{\sigma^2_N}_{enc}$ sont unies à ces quantités par

$$\mu_{i,\text{enc}} = y^{\mu_i}_{enc} \text{ et } \sigma_{i,\text{enc}} = \exp\left(y^{\sigma^2_i}_{enc}\right),\ i = 1, \ldots, N. \tag{5.10}$$

Les sorties du réseau décodeur sont définies de la même manière pour produire le vecteur de moyennes μ_{dec} et de variances σ^2_{dec} dans (5.9). Le prior est la distribution Gaussienne multivariée standard $p_\theta(z) \sim \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_N)$ avec $\mathbf{0}_N = [0, \ldots, 0]^T \in \mathbb{R}^N$. L'astuce de reparamétrisation sous-jacente à l'obtention des échantillons $z^{(1)}, \ldots, z^{(L)}$ est telle que $z^{(l)} = g_\phi(x, y, \epsilon^{(l)}) = \mu_{\text{enc}}(x,y) + \sigma_{\text{enc}}(x,y) \odot \epsilon^{(l)}$ où $\epsilon^{(l)} \sim \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_N)$. De plus, puisque le prior et l'encodeur sont tous les deux Gaussiens, la KLD dans l'ELBO peut être calculée analytiquement, de sorte qu'elle

s'écrive

$$\tilde{\mathcal{L}}(x, y, \theta, \phi) = \frac{1}{N}\sum_{l=1}^{L} \log \mathcal{N}(y; \mu_{\text{dec}}(x, z^{(l)}; \theta), \text{diag}(\sigma^2_{\text{dec}}(x, z^{(l)}; \theta))) \\ + \frac{1}{2}\sum_{i=1}^{N}(1 + \log\left(\sigma^2_{i,\text{enc}}(x, y; \phi)\right) - \mu_{i,\text{enc}}(x, y; \phi)^2 - \sigma^2_{i,\text{enc}}(x, y; \phi)), \tag{5.11}$$

où L est le nombre d'échantillons de z. Enfin, le processus génératif des données de l'équation (5.2) peut être réécrit comme suit :

$$p_\theta(y|x) = \int \mathcal{N}(y; \mu_{\text{dec}}(x, z; \theta), \text{diag}(\sigma^2_{\text{dec}}(x, z; \theta)))\mathcal{N}(z; \mathbf{0}_N, \boldsymbol{I}_N)dz. \tag{5.12}$$

Cette forme met en évidence la capacité du CVAE à apprendre une fonction qui relie des données d'entrée à une distribution potentiellement multimodale sur les données de sortie (« one-to-many mapping »). Ici, les y sont générées via un mélange de Gaussiennes infini : le prior sur les variables latentes fait office de distribution des probabilités des hypothèses (en nombre infini) du mélange ; les paramètres de chaque hypothèse sont définis par le décodeur et naturellement conditionnés par z. C'est sous cette forme particulière que le CVAE est considéré dans les parties suivantes.

Si le réseau de vecteur de paramètres θ est également en charge de l'apprentissage du prior $p_\theta(z|x)$, alors, conditionnellement aux entrées, une plus grande fidélité peut être envisagée dans la génération des sorties relativement à la vérité terrain. Cependant, afin de renforcer la dépendance du décodeur aux variables latentes, celles-ci doivent contenir de l'information utile à la génération de y qui n'est pas déjà présente dans les entrées x. En procédant à l'apprentissage du prior, il est possible que les variables latentes dépendent uniquement des entrées x. Dans cette étude afin de favoriser au maximum l'encodage d'information par le CVAE, la distribution a priori des variables latentes ne sera jamais apprise.

5.2.3.5 La distribution postérieure agrégée des variables latentes

La distribution postérieure agrégée des variables latentes (latent aggregated posterior distribution) appelée aussi encodeur agrégé (aggregated encoder). Elle est définie par

$$q_\phi^{\text{agg}}(z) = \mathbb{E}_{x,y\sim p_{\mathcal{D}}(x,y)} q_\phi(z|x, y) \approx \frac{1}{M}\sum_{i=1}^{M} q_\phi(z|x^{(i)}, y^{(i)}), \tag{5.13}$$

avec $x^{(i)}, y^{(i)}$ représentant le $i^{\text{ième}}$ échantillon i.i.d. de x, y distribués selon $p_{\mathcal{D}}(x, y)$. Le nombre total d'échantillons utilisé pour approcher l'espérance est donné par M. L'analyse de la distribution de

$q_\phi^{\text{agg}}(z)$ est primordiale dans le cadre des objectifs d'interprétabilité considérés ici et ce pour au moins deux raisons. En théorie, à l'issue de l'apprentissage, la distribution postérieure agrégée et le prior des variables latentes doivent être similaires. Cependant, dans la pratique, il est possible que ça ne soit pas le cas comme ce sera vu tout au long du chapitre pour plusieurs problèmes considérés. Dans notre cas où le prior n'est pas appris, la distribution $q_\phi^{\text{agg}}(z)$ doit être comparée avec $\mathbb{E}_{x\sim p_{\mathcal{D}}(x,y)}[p(z|x,y)] = p(z)$. Via cette comparaison, il est donc possible de s'assurer que les codes latents produits respectent la structure imposée par le choix du prior. De plus, la distribution postérieure agrégée renseigne sur la capacité des variables latentes à être interprétées. En effet, l'équation (5.13) définit un mélange de Gaussiennes fini représentant la distribution moyenne de l'encodeur dont la répartition de ses composants équipondérés partitionnent l'espace latent de façon à ce qu'un opérateur humain puisse en extraire de l'information.

5.2.4 Validation du CVAE *Vanilla* sur l'apprentissage de données synthétiques de dynamique

Afin d'illustrer le mécanisme d'apprentissage du CVAE, tout en validant son implémentation, une évaluation du modèle est effectuée sur des données synthétiques.

5.2.4.1 Les données synthétiques

Les données sont générées de façon à représenter une dynamique simplifiée facile à interpréter tout en restant pertinente par rapport à celle d'un véhicule en interaction sur un segment routier. Elles représentent des positions dans un système d'axe bi-dimensionnel et ensemble elles constituent un problème de régression de trajectoires à un pas combiné à un problème sous-jacent de classification vis-à-vis des catégories de ces mêmes trajectoires. Il est important de rappeler que dans le cas général, les classes ne sont pas disponibles. Le jeu de données synthétiques est composé d'un ensemble de paires (entrées, sorties) et i.i.d. selon la loi jointe $p_{\mathcal{D}}(x,y)$ définie par

$$x \sim p_{\mathcal{D}}(x) = \mathcal{N}\left(x; \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_x^2 & 0\\ 0 & \sigma_x^2\end{bmatrix}\right) \tag{5.14}$$

$$y \sim p_{\mathcal{D}}(y|x) = \frac{1}{2}\mathcal{N}\left(y; \begin{bmatrix}\mu_{y_{d1}}(x)\\ \mu_{y_{d1}}(x)\end{bmatrix}, \Sigma_1(x)\right) + \frac{1}{2}\mathcal{N}\left(y; \begin{bmatrix}\mu_{y_{d2}}(x)\\ \mu_{y_{d2}}(x)\end{bmatrix}, \Sigma_2(x)\right), \tag{5.15}$$

où σ_x^2 fixé, conditionne l'étalement des données d'entrée autour de l'origine. Les échantillons de y sont construits à partir de ceux de x, une symétrie centrale se réalisant sur le point échantillon x. Les deux moyennes $\mu_{y_{d1}}$ et $\mu_{y_{d2}}$ sont sélectionnées de façon à être d'un côté ou de l'autre de la symétrie à une distance choisie aléatoirement de la moyenne de x. Les matrices de covariances

$\Sigma_1(x)$ et $\Sigma_2(x)$ sont choisies de façon à étaler plus ou moins les données de sortie le long de deux axes. Le premier axe est la droite reliant le centre de symétrie et les moyennes des deux hypothèses constituant (5.15). Le second axe est perpendiculaire à ce dernier. L'étalement sur le premier axe est proportionnel à la distance entre la moyenne x et les moyennes des deux hypothèses de y. L'étalement sur le second axe est fixé. La Figure 5.3 représente un jeu de données construit depuis un ensemble restreint de 5 tirages différents pour les données en entrée. Les données sont approximativement i.i.d. et le nombre total d'instances de couples entrée-sortie est de 10000. C'est ce jeu de données qui sera utilisé par la suite pour valider et étudier les différentes variations de CVAE. De la même façon que pour les données réelles, le prior des variables latentes n'est pas appris et sa forme est également celle d'une distribution Gaussienne standard vivant bivariée.

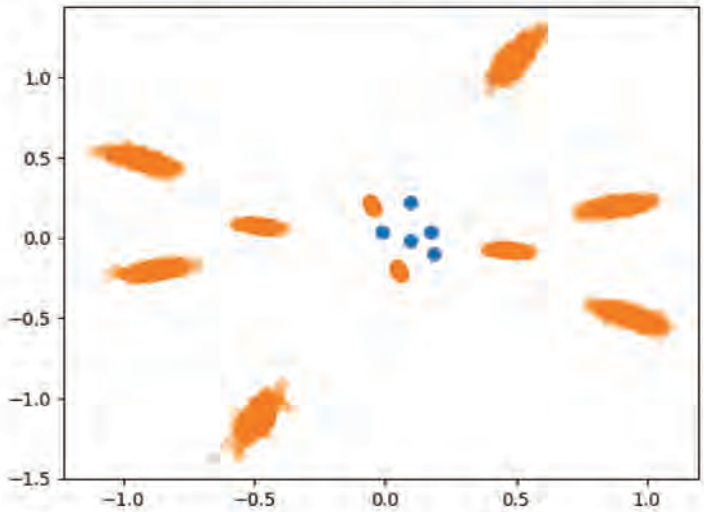


FIGURE 5.3 – Jeu de données synthétiques obtenu depuis 5 échantillons d'entrée x (en bleu) associé pour chacun d'eux à 2000 échantillons de la variable de sortie y (en orange). Les entrées et sorties sont des positions dans un système d'axe bi-dimensionnel.

5.2.4.2 Apprentissage

Le CVAE utilisé pour l'apprentissage de la dynamique a priori synthétique est composé d'un encodeur et d'un décodeur tous les deux Gaussiens avec une covariance diagonale et représentés par des FFNs composés eux-mêmes de 3 couches cachées de 64 neurones. La fonction d'activation utilisée pour l'ensemble des couches cachées est LEAKYRELU [142]. Les réseaux encodeur et décodeur sont entraînés avec l'algorithme d'optimisation ADAM avec un taux d'apprentissage de $\alpha = 0.001$ linéairement décroissant au fil des époques. La taille des *batchs* est de 128.

5.2.4.3 Résultats

La Figure 5.4 illustre les résultats du CVAE obtenus depuis l'apprentissage sur données synthétiques. Une RMSE de 8.10^{-2} est obtenue sur la prédiction en position. En plus des résultats corrects en prédiction du modèle appris, il est intéressant d'analyser les prédictions y_{pred} en fonction de la valeur échantillonnée de $z = (z_1, z_2)$ pour mettre en évidence la façon dont les variables latentes encodent ou non les facteurs génératifs des données. La Figure 5.4c présente une structure évidente quant à la génération des données par rapport à la valeur de la variable latente z_2. Cette variable semble avoir encodé l'axe horizontal. Afin de rapprocher le problème de l'apprentissage sur données synthétiques de l'apprentissage des trajectoires véhiculaires, le dégradé continu allant du jaune au violet peut être interprété comme représentant le degré d'appartenance d'un point prédit à une classe de manœuvre spécifique. Dans ce cas, deux manœuvres hypothétiques sont repérées par les couleurs jaune-vert et bleu-violet. Depuis cette structure, il est possible de générer des prédictions pour une des classes de manœuvre extraites depuis l'apprentissage de représentation.

A l'inverse, les sorties du CVAE selon z_1 sont plus difficilement interprétables et ne mettent pas en évidence une structure saillante dans les données. En effet, dans la Figure 5.4b, le dégradé de couleur continu s'est transformé en un nuage dont la couleur de chacun des points est indépendante de leur position et de la valeur prise par cette composante latente. L'analyse de l'approximation de la distribution postérieure agrégée des variables latentes $q_\phi^{\text{agg}}(z)$ explique cette différence de structure dans les composantes de z. A partir de la Figure 5.4d. en supposant une marginalisation de l'encodeur agrégé par z_2, la distribution obtenue ferait correspondre un nombre important d'instances de données d'entrée-sortie à des valeurs similaires de z_1. En d'autres termes, lors de l'apprentissage, l'encodeur a encodé peu d'information sur les données dans l'axe représenté par le domaine de z_1. Pour deux couples $(x_{\text{pred}}, y_{\text{pred}})$ résultant de la prédiction du CVAE et similaires en position d'entrée et de sortie, la valeur pouvant être prise par z_1 pour chacun des couples s'étend aléatoirement sur de longs segments. En revanche, en supposant une marginalisation de z selon z_1, la distribution obtenue serait un mélange de Gaussiennes équipondéré qui couvre partiellement le domaine de la distribution a priori $p(z_2)$. Les groupes de moyennes et de covariances de l'encodeur sur l'axe de z_2 représentent la structure apprise. Plusieurs regroupements distincts de codes latents sont présents sur le graphe de l'encodeur agrégé et il est donc possible d'identifier deux manœuvres hypothétiques depuis ces données non-labélisées. En considérant conjointement les deux composantes de la distribution du postérieure agrégé des variables latentes, il semble possible de séparer l'espace latent d'un axe allant du nord-est au sud-ouest.

Il est important de noter que, bien qu'une des variables latentes semble avoir encodé de l'information, la forme de l'encodeur agrégé n'est pas satisfaisante avec une faible couverture du domaine de la distribution a priori sur les variables latentes. De plus, l'encodeur produit des distributions

proches de la dégénérescence où les ellipses de covariance associées sont écrasées sur une de leurs dimensions. Ces aspects de couverture et de recouvrement résultant de la distribution apprise de l'encodeur sont abordés dans les sections suivantes.

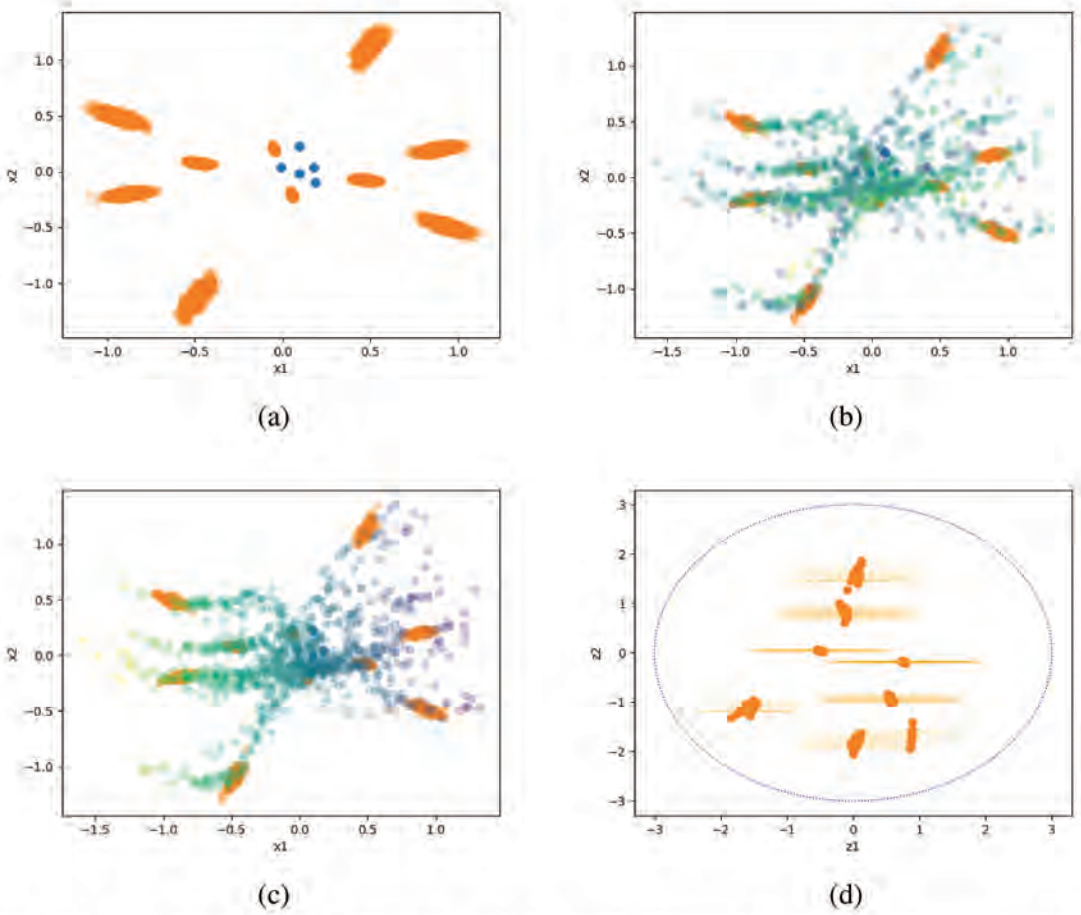


FIGURE 5.4 – (a) Données synthétiques utilisées pour l'apprentissage. Les positions en entrée et sortie sont représentées par des cercles bleus et orange respectivement. (b-c) Sorties du CVAE obtenues depuis 1000 échantillons de la variable latente z. Dans chaque graphe, les sorties sont représentées par des cercles et sont superposées aux données d'entraînement. Leurs couleurs s'étendent graduellement du violet vers le jaune, selon les valeurs prises par la première (graphe (b)) ou la seconde (graphe (c)) composante de z. (d) Approximation de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ pour 1000 échantillons de (x, y). Pour chaque échantillon de la paire (x, y), la densité de probabilité de l'encodeur est représentée par l'ellipse de confiance associée à une probabilité de 99%, elle-même étant centrée sur l'espérance de l'encodeur, représentée par un cercle. L'ellipse de confiance bleue représente la probabilité à 99% de la distribution a priori $p(z) \sim \mathcal{N}(\mathbf{0}_N, \mathbf{1}_N)$.

5.2.5 Apprentissage de la dynamique égocentrique de véhicules en interaction avec le CVAE *Vanilla*

5.2.5.1 Les données HighD

Une fois l'apprentissage du CVAE *Vanilla* validé sur des données synthétiques, un nouvel apprentissage est effectué sur des données réelles de trajectoires avec le jeu de données HighD qui sont censées être des trajectoires du vecteur d'état, dans la même veine que dans le Chapitre 4. En reprenant les notations choisies dans la section précédente, le vecteur d'entrée x représente l'état de la flotte de véhicules à l'instant $n-1$. Le vecteur de sorte y désigne le sous-vecteur caractérisant l'état de l'égo-véhicule à l'instant n de façon que le CVAE permette d'apprendre de manière non supervisée le modèle de transition égocentrique. L'ensemble $\left\{\left(\mathsf{x}^{\mathsf{k}}_{\mathsf{n}-1}, \{\mathsf{x}^{\mathsf{l}}_{\mathsf{n}-1}\}_{\mathcal{V}_{\mathsf{l}}\in\mathcal{N}_{\mathsf{n}-1}(\mathcal{V}_{\mathsf{k}})}, \mathsf{x}^{\mathsf{k}}_{\mathsf{n}}\right)\right\}$ extrait depuis HighD constitue les données d'apprentissage du CVAE. Les entrées et sorties du CVAE sont similaires à celles utilisées pour les réseaux de régression présentés dans la section 4.7.1.1. Cependant ici, aucune information sur la manœuvre discrète relative à la trajectoire continue n'est établie de manière supervisée. Le modèle de dynamique a priori présenté dans l'équation (4.3) s'écrit désormais

$$p(x_n^k|x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l\in\mathcal{N}_{n-1}(\mathcal{V}_k)}) = \int \mathcal{N}(x_n^k; \mu_{\text{dec}}(x_{n-1}^k, \diamond, z), \text{diag}(\sigma^2_{\text{dec}}(x_{n-1}^k, \diamond, z)))\mathcal{N}(z; \mathbf{0}_N, \boldsymbol{I}_N)dz, \quad \text{où } \diamond = \{x_{n-1}^l\}_{\mathcal{V}_l\in\mathcal{N}_{n-1}(\mathcal{V}_k)}. \tag{5.16}$$

En conséquence de ces notations, l'ELBO dans (5.11) utilisée dans la fonction objectif s'écrit désormais

$$\begin{aligned}\tilde{\mathcal{L}}(x_n^k, (x_{n-1}^k, \{x_{n-1}^l\}_{\mathcal{V}_l\in\mathcal{N}_{n-1}(\mathcal{V}_k)}), \theta, \phi) := \\ \frac{1}{N}\sum_{l=1}^{L}\log\mathcal{N}(x_n^k; \mu_{\text{dec}}(x_{n-1}^k, \diamond, z^{(l)}; \theta), \text{diag}(\sigma^2_{\text{dec}}(x_{n-1}^k, \diamond, z^{(l)}; \theta))) \\ + \frac{1}{2}\sum_{i=1}^{N}(1 + \log(\sigma^2_{i,\text{enc}}(x_n^k, x_{n-1}^k, \diamond; \phi)) - \mu_{i,\text{enc}}(x_n^k, x_{n-1}^k, \diamond; \phi)^2 - \sigma^2_{i,\text{enc}}(x_n^k, x_{n-1}^k, \diamond; \phi)), \\ \text{où } \diamond = \{x_{n-1}^l\}_{\mathcal{V}_l\in\mathcal{N}_{n-1}(\mathcal{V}_k)}.\end{aligned} \tag{5.17}$$

Dans les deux équations précédentes, z fait bien entendu référence aux variables aléatoires latentes internes au modèle utilisé pour représenter la dynamique a priori et non aux variables observées dans le cadre d'un schéma d'estimation stochastique comme présenté au chapitre précédent.

Dans cette première tentative d'apprentissage de la dynamique en interaction d'un égo-véhicule, deux apprentissages sont réalisés à des pas de temps différents, $T_s = 0.53$s et $T_s = 1.06$s. Utilisés dans le contexte d'estimation sur les données HighD, ces deux pas temporels permettent de filtrer l'état de la flotte pendant suffisamment d'instants avant de réaliser la prédiction.

Comme pour l'apprentissage sur données synthétiques, le CVAE *Vanilla* utilisé est composé d'un encodeur et d'un décodeur tous les deux Gaussiens impliquant des FFNNs. Les dimensions des couches d'entrée et de sortie des réseaux sont adaptées aux nouvelles données utilisées. De par l'augmentation des dimensions de x et y, la capacité du CVAE est augmentée avec pour chacune des couches cachés 128 neurones. La fonction d'activation utilisée pour l'ensemble des couches cachées est LEAKYRELU. La distribution a priori des variables latentes n'est pas apprise et sa forme est celle d'une distribution Gaussienne standard bivariée. Par la suite, sauf si précisé, la distribution a priori des variables latentes pour l'apprentissage sur les données HighD restera identique.

Enfin, afin de prendre en compte la topologie et la sémantique de la route sur laquelle les véhicules se déplacent, un réseau de neurones supplémentaire est utilisé et entraîné en même temps que le CVAE. Ce dernier repose sur l'utilisation de couches de convolution réalisant l'extraction automatique d'information sur l'environnement depuis un patch d'image capturé autour de l'égo-véhicule. L'Annexe B présente le fonctionnement de l'extraction de l'information sémantique pour notre étude. Par la suite, le vecteur d'information produit par le réseau est concaténé au vecteur d'entrée du réseau.

5.2.5.2 Résultats

Après l'apprentissage du CVAE *Vanilla* sur les données HighD, il est intéressant de comparer les résultats obtenus en prédiction de la même manière que pour les données synthétiques. La Figure 5.5 illustre les résultats en prédiction sur une scène autoroutière aléatoire obtenus par le CVAE entraîné avec des données dont le pas de temps est égal à $T_s = 0.53$s. L'encodeur agrégé de ce même réseau est représenté dans la Figure 5.6. La loi de l'encodeur est ici indépendante du couple de données sur lequel elle est conditionnée, ce qui constitue naturellement un problème. Afin de valider les modèles appris sur ces données, le calcul des RMSEs en position et vitesse est effectué. Comme expliqué au chapitre précédent (Section 4.7.1.3), dans le cadre des objectifs d'explicabilité revendiqués, un test du khi-deux « approximatif » sur l'erreur de transition d'état quadratique normalisée est effectué pendant l'apprentissage de tous les CVAEs présentés. Un test positif indique que l'état actuel du véhicule cible peut être considéré comme un échantillon de la densité de probabilité Gaussienne produite par le réseau à l'aide d'un seul échantillon de z. Les données de test sont obtenues sur un ensemble de trajectoires qui ne servent pas à l'apprentissage du réseau. Enfin, un test du khi2 « approximatif » plus élaboré pourrait être effectué pour détecter

un éventuel échec d'apprentissage de la part du CVAE. En effet, la loi de dynamique égocentrique apprise s'écrivant comme un mélange infini d'hypothèses gaussiennes dont les poids sont les valeurs de la densité de probabilité des variables latentes, un tel test approximatif pourrait consister en : l'échantillonnage d'un nombre fini J d'échantillons de z ; l'approximation de la loi de dynamique égocentrique comme une GMM de J lois de type décodeur ; l'approximation de cette GMM par la loi Gaussienne de mêmes moments (« moment-matched ») ; l'implémentation d'un test de chi2 sur la base de cette loi. Dans ce manuscrit seul le premier test du khi2 présenté est effectué. Le test du khi2 « approximatif » réalisé avec plusieurs échantillons de z est laissé pour des travaux futurs.

Pour le CVAE $T_s = 0.53$s, les RMSEs en position (m) et en vitesse (m.s$^{-}1$) sont de 0.02 et 0.07 respectivement. Le pourcentage de tests corrects du khi-deux est lui de 91%. Pour le CVAE $T_s = 1.06$s, les RMSEs en position et en vitesse sont de 0.06 et 0.08 respectivement. Le pourcentage de tests corrects du khi-deux est lui de 89%. Bien que les métriques de performances pour les deux modèles avec les deux pas de temps soient bonnes, l'encodage appris n'est pas satisfaisant. En effet, comme l'illustrent les Figures 5.5 et 5.6, aucune information sur les données de trajectoires ne semble être encodée dans les variables latentes. La prédiction en position et en vitesse ne dépend pas de la valeur prise par z. Les paramètres de la distribution de la trajectoire y sont entièrement déterministes et semblent dépendre uniquement des entrées x du réseau. Cette conclusion est identique quel que soit le pas de temps considéré. Deux remarques peuvent être établies concernant le processus génératif du CVAE pour notre problème. De bonnes RMSEs, de bons pourcentages de tests du khi-deux et un décodeur indépendant de z semblent indiquer que la dynamique a priori de l'égo-véhicule n'est pas un mélange infini de Gaussiennes et peut se modéliser avec une seule Gaussienne. Enfin, aux pas de temps considérés, il est peut-être difficile pour le réseau d'apprendre une distribution multimodale via les variables latentes de par le peu de variété dans les états futurs des véhicules conditionnellement aux manœuvres *i.e.* le futur à un pas des véhicules peut être représenté facilement par une Gaussienne multivariée, de fait unimodale.

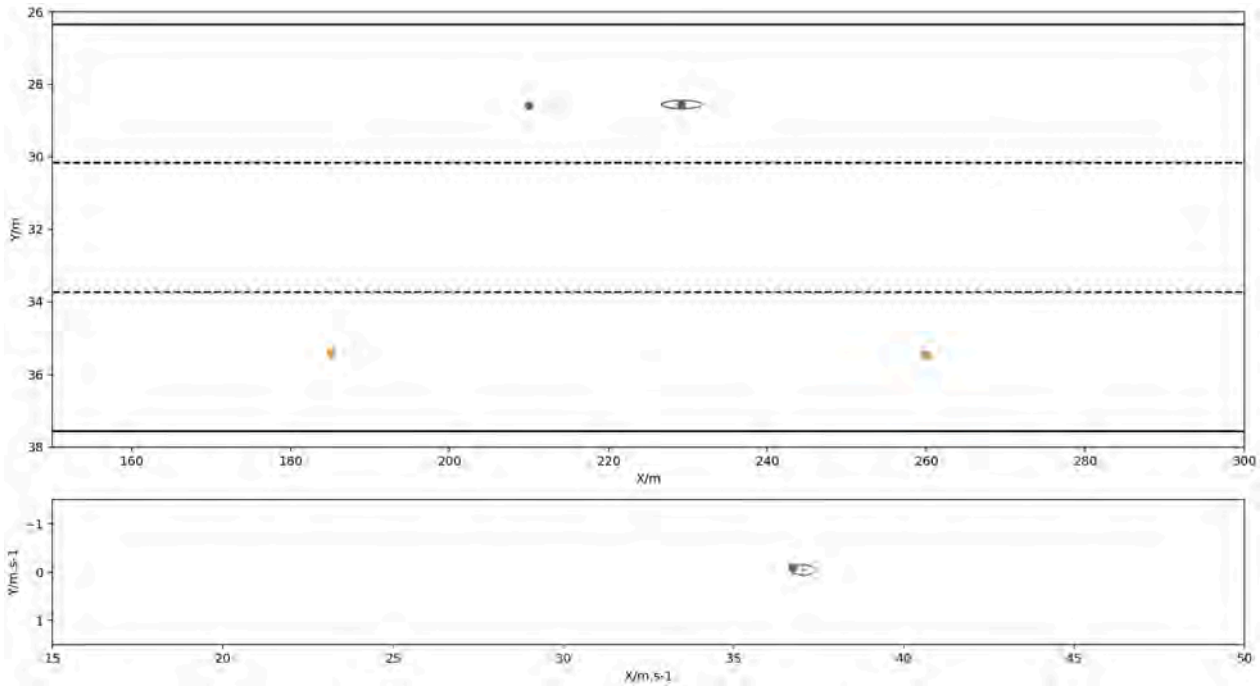


FIGURE 5.5 – Sorties du CVAE *Vanilla* avec $T_s = 0.53$s obtenues pour une instance des données HighD. Le cercle bleu représente la position de l'égo-véhicule en entrée du réseau à l'instant courant (la vitesse n'est pas représentée). Le point vert représente la vérité terrain en termes de position et vitesse futures. Les points orange sont les positions des véhicules appartenant au voisinage de l'égo-véhicule en entrée du réseau à l'instant courant. Pour un vecteur d'entrée et pour différentes réalisations du vecteur des variables latentes dans son prior, les ellipses de confiance à 99% associées au décodeur sont affichées. La couleur de ces éléments graphiques dépend de la valeur prise par une des composantes du vecteur des variables latentes. Ici, seules les prédictions selon les valeurs de z_1 sont affichées. Les couleurs choisies couvrent le domaine de la variable latente Gaussienne en allant uniformément du violet au jaune. Les ellipses affichées sont confondues et indiquent que le décodeur ne dépend pas de z_1.

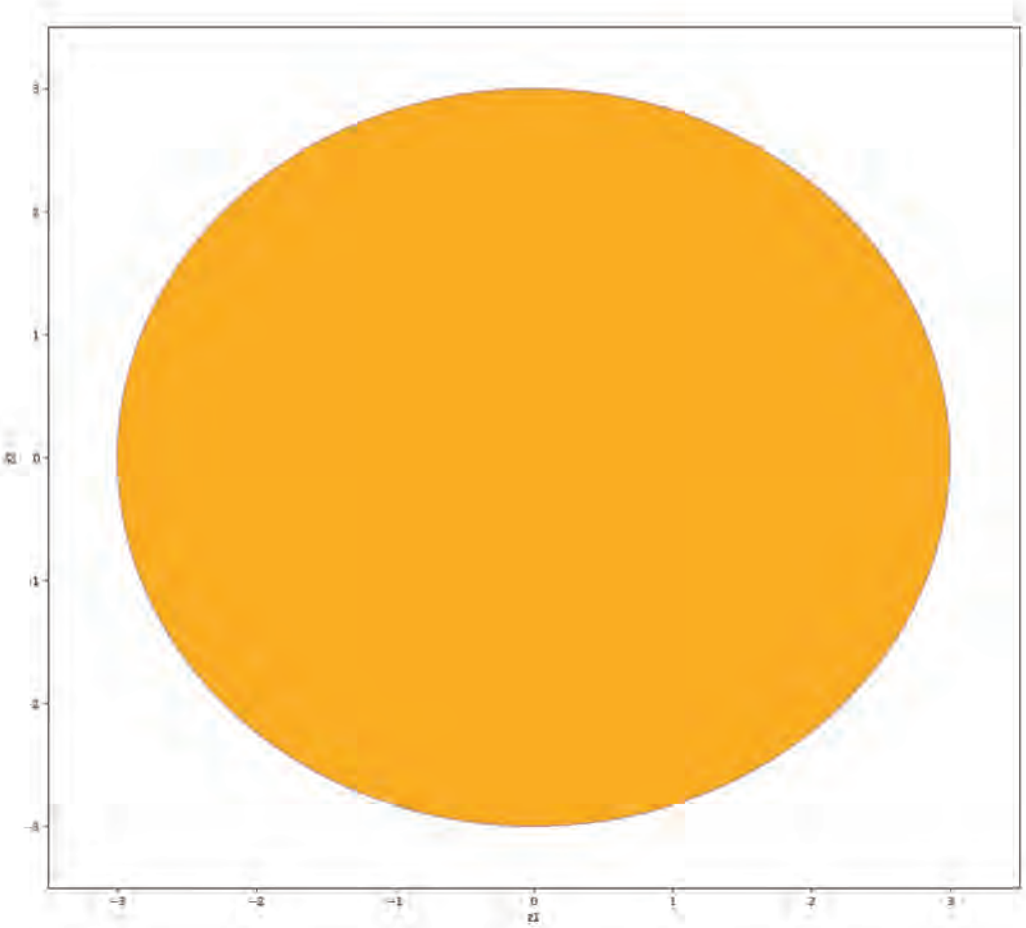


FIGURE 5.6 – Approximation de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ d'un CVAE *Vanilla* avec $T_s = 0.53$s pour le tirage de 1000 échantillons des couples entrée-sortie affichés en orange sur le graphe. La densité de l'encodeur pour chaque échantillon est représentée par l'ellipse de confiance associée à une probabilité de 99%, elle-même étant centrée sur l'espérance de l'encodeur, représenté par un cercle. Ici, les ellipses sont identiques et se superposent toutes parfaitement. L'ellipse de confiance bleue en périphérie de la superposition d'ellipses, représente la probabilité à 99% de la distribution a priori $p(z) \sim \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_N)$. La distribution de l'encodeur est indépendante des valeurs prises par les $x^{(i)}$ et $y^{(i)}$.

5.2.6 Modification des données d'entrée et de sortie du CVAE *Vanilla*

Afin de s'assurer que les capacités d'encodage des variables latentes ne sont pas entravées par le choix du vecteur d'état, plusieurs apprentissages sont réalisés avec des vecteurs d'entrées-sorties

x et y différents de ceux utilisés dans la section précédente. Ces différentes expérimentations sont décrites dans les sous-parties suivantes.

5.2.6.1 Apprentissage faiblement supervisé avec équilibrage des données à partir de la segmentation en manœuvres

En utilisant les données segmentées avec la méthode de la section 4.4.1, il est possible d'équilibrer les données de trajectoires en termes de fréquence des manœuvres présentes dans le jeu de données non étiquetées. Dans ce cas, l'apprentissage est défini comme faiblement supervisé : le modèle n'a pas accès à l'information sur les manœuvres mais le jeu de données est prétraité avec cette information. Le fort déséquilibre dans la fréquence des classes de manœuvres observé dans les données HighD peut être à l'origine des mauvaises performances du CVAE *Vanilla* [143]. Dans [144], les auteurs ont recours à un rééquilibrage des données couplé à l'utilisation d'un modèle issu de la fusion d'un CVAE et d'un réseau antagoniste génératif (Generative Adversarial Network – GAN) pour partitionner l'espace latent selon les classes d'un problème d'apprentissage supervisé. Afin de mitiger simplement ce déséquilibre, comme pour l'apprentissage supervisé, les futurs entraînements sont effectués avec le même profil de rééchantillonnage des classes de trajectoires : la classe `forward` est sous-échantillonnée de 85% alors que les classes `sheer_in` et `sheer_out` sont toutes les deux sur-échantillonnées de 25%.

Dans les sections suivantes, les résultats présentés par défaut sont obtenus depuis un apprentissage sur le jeu de données équilibré. Les réseaux reportant de bonnes performances en termes de séparation des variables latentes seront ensuite entraînés avec l'ensemble des données HighD pour être ensuite ré-évalués sur le jeu de données équilibré afin de s'assurer que sur des scénarios routiers complexes, avec beaucoup de changements de voie, les performances des modèles appris restent acceptables.

5.2.6.2 Ajout des accélérations au sous-état des véhicules

Ici, l'accélération longitudinale a_x et l'accélération latérale a_y sont ajoutées aux sous-vecteurs d'état associés à chaque véhicule. Un apprentissage du CVAE *Vanilla* est réalisé avec la même paramétrisation présentée dans la section précédente. Les résultats en prédiction obtenus après l'apprentissage du CVAE *Vanilla* avec ces nouvelles données sont similaires à ceux obtenus précédemment. Les variables latentes ne sont pas utilisées pour la génération structurée des sorties. Un score de 63% est obtenu pour le test du khi-deux et les RMSEs sont de 4.10^{-2} en position, 0.01 en vitesse et 0.03 en accélération pour $T_s = 0.53$s. Au regard du test du khi2 effectué, les distributions prédites sont trop optimistes vis-à-vis de la véritable distribution future de l'égo-véhicule et ceci pourrait entraver les performances des algorithmes de prédiction présentées au chapitre 4

en limitant les futurs considérés pour l'état de l'égo-véhicule lors de l'étape de propagation des particules.

5.2.6.3 Ajout de l'orientation et de la vitesse angulaire au sous-état des véhicules

Afin de faciliter la comparaison avec les méthodes de prédiction trajectoire-vers-trajectoire présentes dans l'état de l'art [71, 72, 73] l'état de chaque véhicule n'inclut aucune information sur son orientation, les véhicules sont considérés comme omnidirectionnels. Ceci peut être également justifié par le besoin d'un sous-état simplifié mais également de par le segment autoroutier considéré où les véhicules évoluent avec des vitesses angulaires faibles.

Un nouvel apprentissage est réalisé en incluant dans le sous-vecteur d'état associé à chaque véhicule son orientation et sa vitesse angulaire. En prenant en compte l'orientation, le CVAE *Vanilla* ne parvient toujours pas à encoder d'information pertinente sur la trajectoire de l'égo-véhicule. Un score de 58% est obtenu pour le test du khi-deux et les RMSEs sont de 0.02 en position, 0.06 en vitesse et 6.10^{-4} en vitesse angulaire pour $T_s = 0.53$s. Comme pour l'apprentissage avec les accélérations, ces résultats sont trop optimistes en terme de dynamique a priori égocentrique pour que le modèle appris soit exploitable.

5.2.6.4 Prise en compte des contraintes de non-holonomie des véhicules

Une nouvelle modification repose sur la prise en compte dans le modèle des contraintes cinématiques de non-holonomie, qui rendent compte du fait que chaque véhicule roule sans glisser en demeurant tangent à sa trajectoire. L'objectif ici est d'apprendre, via un CVAE, à relier l'état de la flotte de véhicules à l'instant courant à une commande de l'égo-véhicule permettant d'obtenir son état à l'instant suivant tout en respectant la contrainte de non-holonomie sur son mouvement. Pour ce faire, un modèle cinématique unicycle étendu [145] (ce modèle s'apparente également à un modèle à accélération et à vitesse de rotation constantes (Constant Turn Rate and Acceleration – CTRA)) est adopté pour représenter la dynamique de l'égo-véhicule. Celui-ci s'écrit

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ \omega \\ a \end{bmatrix}, \tag{5.18}$$

où x et y représentent la position du véhicule, v sa vitesse curviligne et ϕ son orientation. Le vecteur de commande est constitué de l'accélération curviligne a et de la vitesse de rotation ω de l'égo-véhicule. Comme la loi de dynamique est exprimée entre deux instants, une discrétisation de la

dynamique est effectuée en considérant constantes les commandes entre chaque instant. Le pas de temps est égal à $T_s = 0.53$s. Le CVAE appris fait donc correspondre l'état de la flotte de véhicules à un instant avec la distribution de probabilité du vecteur de commande qui amène le véhicule dans l'état suivant. L'Annexe C détaille comment obtenir le modèle à temps discret par discrétisation du modèle à temps continu (5.18) sous l'hypothèse de blocage à l'ordre 0 de la commande de l'égo-véhicule. En dépit de cette modification, le CVAE *Vanilla* ne parvient toujours pas à encoder de l'information dans la distribution de z. Une hypothèse peut être faite sur le pas de temps considéré rendant encore plus difficile la séparation des commandes qui différencient le déplacement des véhicules en termes de manœuvres. En effet, à un pas de temps faible, la différence de commande parmi les manœuvres est faible. En ce qui concerne la prédiction, après intégration des commandes pour obtenir l'état à l'instant suivant, le pourcentage de tests positifs du khi-deux est inférieur à 75%.

5.2.6.5 Conclusion

Comme vu à travers les modifications précédentes, il semble que ni l'augmentation du vecteur d'état des véhicules ni la prise en compte des contraintes sur le mouvement des véhicules n'ait d'impact positif significatif quant à la structuration des variables latentes apprise. Et ce, même dans le cas où le jeu de données d'apprentissage est rééquilibré. Ce phénomène limitant pour notre objectif d'interprétabilité est exploré en détails dans les parties suivante. Par la suite, et ce jusqu'à la section 5.3.4.2, seul le pas de temps $T_s = 1.06$s est appliqué pour la construction de l'ensemble des données de trajectoire HighD. Ceci est fait afin de maximiser les différences dans les données entre les différentes manœuvres tout en gardant un pas de temps raisonnable pour effectuer l'estimation et la prédiction des trajectoires.

5.2.7 L'« effondrement du postérieur » dans le CVAE *Vanilla*

Pendant l'apprentissage, le réseau encodeur cherche à modéliser, au travers des variables latentes, des facteurs permettant une interprétation du processus génératif des données. Malheureusement, la qualité et le nombre de ces facteurs latents appris sont négativement impactés par un phénomène connu sous le nom d'effondrement du postérieur (posterior collapse) [146, 147, 148], dans lequel le modèle génératif apprend à ignorer un sous-ensemble (ou l'ensemble complet) des variables latentes. Ici le terme « postérieur » désigne la loi postérieure des variables latentes représenté par l'encodeur et ne doit pas être compris en tant qu'encodeur agrégé.

Dans le cas du CVAE *Vanilla*, le problème d'effondrement du postérieur intervient lorsque la procédure d'apprentissage fait correspondre le postérieur des variables latentes à leur distribution a

priori de sorte que

$$q_\phi(z|x,y) \approx p_\theta(z). \tag{5.19}$$

Ce comportement est indésirable puisque depuis cette solution dégénérée aucune interprétation des variables latentes ne peut être effectuée. Dans ce cas où aucune information n'est distillée dans les variables latentes, le décodeur les ignore pour la prédiction/reconstruction. Dans les cas où le décodeur ignore les variables latentes, pour une instance en entrée fixée x et deux échantillons distincts z_i et z_j, le processus génératif du CVAE *Vanilla* vérifie

$$p_\theta(y_i|x,z_i) \approx p_\theta(y_j|x,z_j). \tag{5.20}$$

Une des premières méthodes de résolution de la littérature reposaient sur une diminution de la capacité du décodeur [149, 150]. Des tests préliminaires ont été effectués pour notre problème avec des CVAEs *Vanilla* et des décodeurs dotés d'un faible pouvoir de représentation avec un nombre de paramètres réduit. Les résultats en prédiction obtenus ne sont pas acceptables et le gain en structure dans les variables latentes n'est pas suffisant.

Il existe donc un nombre important de travaux dans la littérature qui cherche à diminuer l'effondrement du postérieur en modulant l'effet du terme de divergence dans l'ELBO [146, 151, 152, 153]. Ces travaux suggèrent que l'effondrement postérieur est causé par le terme de divergence KL dans la fonction objectif représenté par l'ELBO. En effet, celui-ci encouragerait directement la distribution de l'encodeur à correspondre à celle du prior. Afin d'éliminer l'effondrement du postérieur, différentes modifications du CVAE *Vanilla* sont considérées par la suite.

5.3 Améliorations et modifications du CVAE *Vanilla* pour limiter l'effondrement du postérieur

5.3.1 Discrete-CVAE

5.3.1.1 Présentation du modèle

Une première modification est entreprise sur le CVAE *Vanilla*. Celle-ci consiste à remplacer les variables latentes continues par des variables latentes discrètes. Le modèle de transition appris sera ainsi un mélange de Gaussiennes avec un nombre fini d'hypothèses K. Le prior choisi pour les variables latentes est une distribution uniforme discrète, soit $p_\theta(z) = \mathcal{U}(1, K)$. Intuitivement, la conduite peut être vue comme une série de choix discrets de manœuvres dont la réalisation forme une trajectoire continue. Imposer au réseau une forme discrète pour les variables latentes peut être vue comme une modification simple du CVAE *Vanilla* permettant l'apprentissage d'une dynamique a priori avec une structure pertinente pour notre problème. Compte tenu du prior choisi,

les hypothèses sur les manœuvres pour la génération sont équipondérées. Soit Z le vecteur de variables latentes discrètes du Discrete-CVAE, l'encodeur s'écrit maintenant

$$q_\phi(Z|x,y) = \text{Cat}(Z|\pi_{enc}(x,y;\phi)) \tag{5.21}$$

où Cat désigne la distribution catégorielle. Il implique un réseau de neurones paramétré par ϕ dont le vecteur des sorties est constitué des approximations des probabilités a posteriori des K valeurs admissibles de Z notées $\pi_{1,\text{enc}}, \ldots, \pi_{K,\text{enc}}$. Le décodeur reste défini comme pour le cas du CVAE *Vanilla*. L'optimisation des paramètres du réseaux se fait via la moyenne sur le jeu de données de la fonction de coût suivante

$$\begin{aligned}\mathcal{L}_{\text{discrete}}(x,y,\theta,\phi) = &\sum_{k=1}^{K} \log \mathcal{N}(y;\mu_{\text{dec}}(x,Z=k), \text{diag}(\sigma^2_{\text{dec}}(x,Z=k))q_\phi(Z=k|x,y) \\ &+ \sum_{k=1}^{K} \big\{ \big(\log K - \log\left(\pi_{k,\text{enc}}(x,y;\phi)\right)\big)\pi_{k,\text{enc}}(x,y;\phi)\big\}.\end{aligned} \tag{5.22}$$

L'espérance selon l'encodeur du log-pdf du modèle de reconnaissance est maintenant calculée directement sans nécessiter l'astuce de reparamétrisation n'est donc plus nécessaire. Le premier terme dans l'équation (5.22) est calculé en alimentant le décodeur avec un vecteur d'encodage 1 parmi n représentant la valeur prise par Z. Le deuxième terme calcule analytiquement la divergence KL entre l'encodeur et le prior sur Z. La fonction objectif de ce modèle peut être calculée analytiquement en un temps acceptable lorsque le nombre de valeurs que peut prendre Z est inférieur à 100. Le Discrete-CVAE présenté ici peut être vu comme un cas simplifié de l'approche présentée dans [154]. Dans [155] les auteurs proposent des modèles de CVAEs avec des variables latentes discrètes sans fixer d'a priori sur la cardinalité du domaine de Z. L'implémentation du Discrete-CVAE a été préalablement validée sur des données synthétiques.

5.3.1.2 Apprentissage et résultats sur les données HighD

Les résultats avec ce nouveau modèle sur les données HighD sont décevants. Plusieurs configurations d'apprentissage sont effectuées avec Z pouvant prendre 3, 5, 10 ou 25 valeurs. Pour tous les apprentissages des résultats similaires sont obtenus : les variables latentes discrètes sont ignorées. Lors de la génération, une seule des valeurs possibles de Z produit des résultats suffisamment proches en position et vitesse de la vérité terrain. La Figure 5.7 illustre pour une instance de données d'entrée, la loi de transition égocentrique apprise. Sa pertinence est évaluée vis-à-vis de l'instance de sortie correspondante. Le Discrete-CVAE est également entraîné de façon à produire des distributions de probabilité des commandes de l'égo-véhicule comme présenté dans la

section 5.2.6.2. Dans ce cas également, les variables latentes discrètes sont ignorées. Par la suite, seules des variables latentes continues seront utilisées. L'effondrement du postérieur dans le cadre de l'apprentissage de VAEs à variables discrètes ne sera pas traité dans le cadre de cette étude. Cette problématique particulière est abordée dans [155, 156].

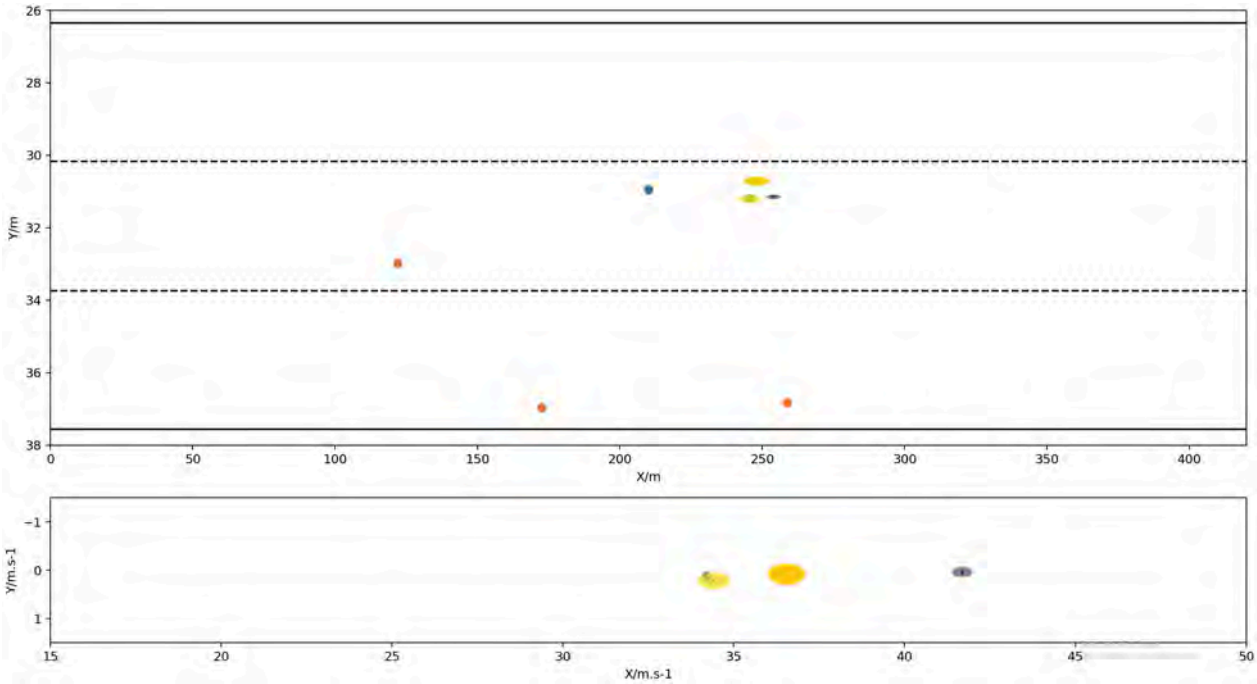


FIGURE 5.7 – Sorties du Discrete-CVAE obtenues pour une instance aléatoire des données HighD. La loi de transition en position et en vitesse selon les valeurs de Z est représentée par les points et les ellipses associées colorées. Pour ce modèle, Z peut prendre 3 valeurs différentes. La distribution du décodeur associée à chacune d'elles est colorisée en jaune, orange et violet. Seule la prédiction associée à la couleur jaune englobe la vérité terrain. Celle correspondant à la couleur violette est assez éloignée de la véritable trajectoire en termes de vitesse. Enfin, pour ce scénario, aucun recouvrement selon les variables latentes n'est présent dans la loi de transition.

5.3.2 Le Beta-CVAE

5.3.2.1 Présentation du modèle

Le Beta-CVAE [151] est une modification simple du CVAE *Vanilla*. Celle-ci implique l'ajout d'un facteur au terme de divergence KL dans l'ELBO. L'ajout de cet unique hyperparamètre β modifie la fonction objectif telle que

$$\mathcal{L}_\beta(x, y, \theta, \phi) = \mathbb{E}_{z\sim q_\phi(z|x,y)}[\log p_\theta(y|x, z)] - \beta D_{KL}(q_\phi(z|x, y)||p_\theta(z)). \tag{5.23}$$

Pour notre problème ce nouveau modèle possède des caractéristiques attrayantes. En gardant comme distribution a priori une distribution Gaussienne multivariée standard, le Beta-CVAE ne fait pas d'hypothèse supplémentaire sur les données. L'ajout d'un seul hyperparamètre facilite le réglage du réseau lors de la phase d'apprentissage. Via une inspection visuelle des résultats, il est possible d'optimiser β pour moduler l'extraction d'information sur la structure des données observées.

Faire varier β revient à appliquer plus ou moins de pression pendant l'apprentissage sur le sous-objectif d'encodage d'information sur les variables latentes, encourageant ainsi l'apprentissage de représentation. L'objectif du CVAE original est obtenu pour $\beta = 1$. Dans [151], les auteurs préconisent de fixer $\beta > 1$ pendant toute la durée de l'apprentissage pour permettre l'apprentissage d'une représentation démêlée des facteurs génératifs indépendants conditionnellement aux données. Par rapport au CVAE original, ce nouvel objectif encourage l'indépendance conditionnelle des composantes z_1, z_2 du vecteur des variables latentes de telle sorte que $q_\phi(z_1|x, y) \perp\!\!\!\perp q_\phi(z_2|x, y)$ et pousse le réseau à apprendre une représentation plus efficace et pertinente du processus génératif des données. Pour notre problème, la plus grande contrainte sur les variables latentes pourrait empirer l'effondrement du postérieur. Comme vu par la suite, considérer des valeurs pour β avec $\beta < 1$, permet de relâcher cette contrainte et d'atténuer ce phénomène.

5.3.2.2 Apprentissage et résultats sur les données synthétiques

Afin de valider l'implémentation, le Beta-CVAE est entraîné sur les données synthétiques de dynamique. Ce modèle reprend les mêmes caractéristiques que pour le CVAE Vanilla, seule la fonction objectif est changée. Une brève optimisation de β est effectuée via une inspection visuelle des sorties du Beta-CVAE. La Figure 5.8 illustre les résultats obtenus sur les données synthétiques pour un Beta-CVAE avec $\beta = 2.5$. Lors de cette apprentissage, la RMSE est meilleure que pour le CVAE Vanilla avec 5.10^{-2}. Après l'apprentissage, plusieurs différences sont présentes au niveau de l'encodeur agrégé. Avec ce paramétrage, les échantillons produits par l'encodeur recouvrent mieux le prior $p(z)$, ce qui a pour effet de réduire l'erreur en prédiction lors du processus génératif des données. De plus, les covariances produites par l'encodeur ne sont plus dégénérées et leur surface individuelle est plus importante. Cette modification de l'encodeur agrégé impacte le processus génératif des données : comme pour le CVAE *Vanilla*, aucune structure n'est encodée dans z_1. Ici, les couleurs associées aux points pour cette composante sont totalement aléatoires. Pour z_2, la résolution de la structure présente dans l'encodeur agrégé semble avoir diminué. En effet, par l'inspection visuelle des couleurs associées aux points générés, on peut observer que par rapport au CVAE *Vanilla*, un nombre réduit de couleurs encode la position des points par rapport à l'axe est-ouest. Ceci est dû à la position et au recouvrement des ellipses dans l'encodeur agrégé. Il semblerait que la symétrie entre les échantillons de y et le centre représenté par ceux de la variable x soit

encodée via deux pôles présents le long de l'axe du domaine de z_2. De par la superposition des points et de leur couleurs, on peut observer un mélange de couleur unique pour chacun des côtés de cette symétrie centrale.

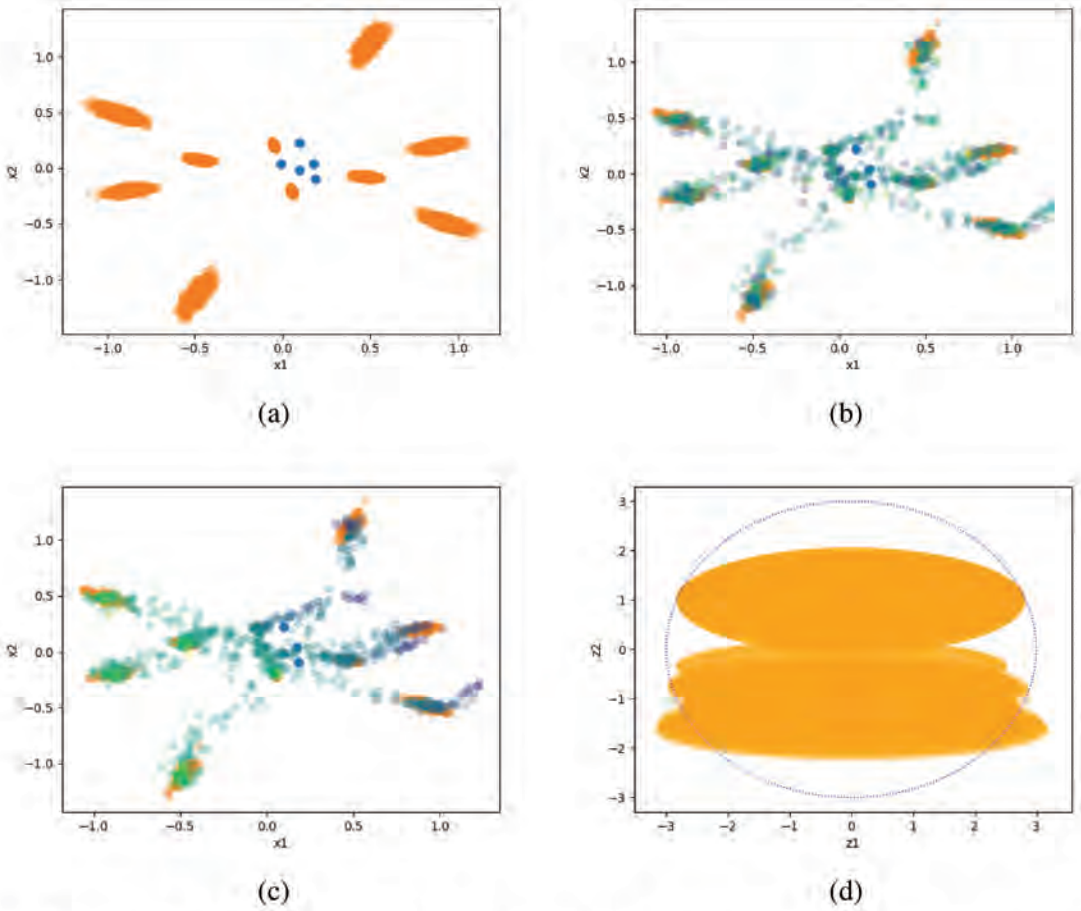
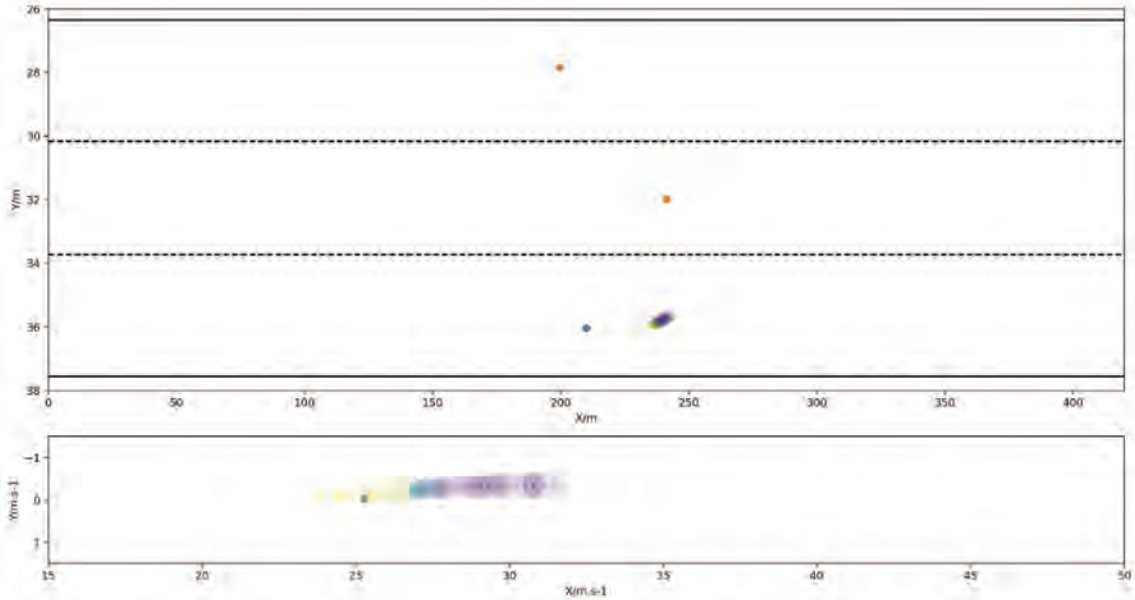
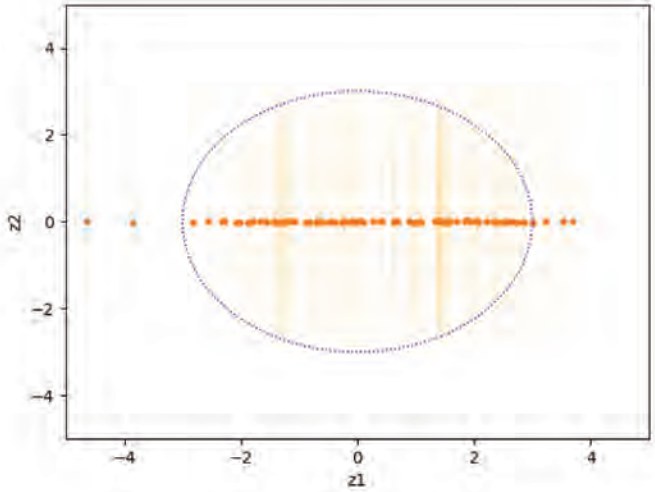


FIGURE 5.8 – (a) Données synthétiques utilisées pour l'apprentissage. (b-c) Sorties du Beta-CVAE obtenues pour 1000 échantillons de la variable latente z. Seule la composante z_2 encode de l'information sur le processus génératif des données. Pour chaque côté de la symétrie centrale présente dans les données, les points sont respectivement de couleur verte ou bleu-violet (d) Approximation du postérieur agrégé $q_\phi^{\text{agg}}(z)$ depuis 1000 échantillons des couples entrée-sortie. L'espace latent semble pouvoir être partitionné horizontalement en deux moitiés représentant chacune un côté de la symétrie.

5.3.2.3 *Apprentissage et résultats sur les données HighD*



(a)



(b)

FIGURE 5.9 – (a) Sorties du Beta-CVAE entraîné avec $\beta = 0.1$ pour une instance des données où l'égo-véhicule effectue la manœuvre `Sheer_out`. Une structure peut être appréciée via les couleurs des points et ellipses associées représentant les distributions de prédiction selon les valeurs de la variable latente z_1. Cette structure semble permettre la génération d'une manœuvre spécifique avec une vitesse longitudinale plus ou moins grande. (b) Approximation du postérieur agrégé $q_\phi^{\text{agg}}(z)$ depuis 1000 échantillons des couples entrée-sortie. Seule la composante z_1 encode de l'information sur la trajectoire de l'égo-véhicule.

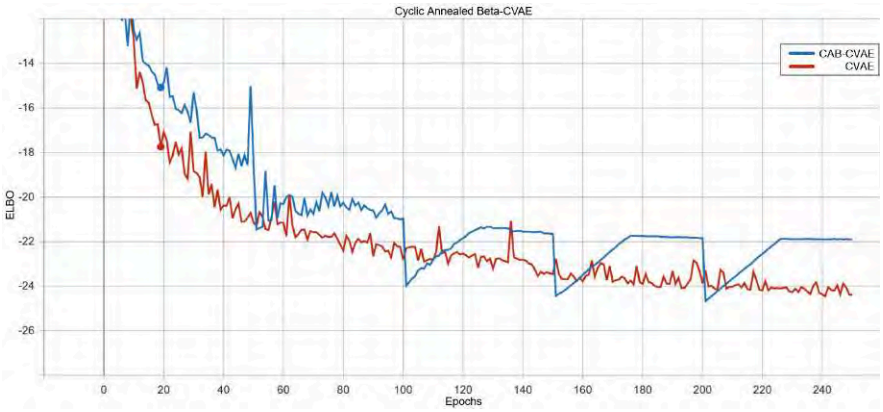
Comme pour le CVAE *Vanilla*, le modèle est maintenant appris depuis les données de trajectoire HighD. Une série d'entraînements est réalisée avec $\beta \geq 1$. Pour tous les apprentissages, le réseau ignore les variables latentes et aucune structure interprétable ne peut être dégagée de celles-ci. Certains travaux de la littérature ITS [80, 157] ont recours à une architecture à base d'auto-encodeurs avec un objectif modifié comme pour le Beta-CVAE. Curieusement, la valeur choisie pour β est telle que $\beta < 1$ sans justification aucune quant aux aspects d'apprentissage de représentation du modèle neuronal. Cependant, nous pouvons supposer que cette modification permet d'atténuer le problème potentiel d'effondrement du postérieur des variables latentes. Pour notre cas, un apprentissage avec les mêmes conditions que dans les sections précédentes est effectué avec $\beta = 0.1$. Ce seuil pour β est obtenu empiriquement en examinant les sorties du modèles en termes de structuration des variables latentes par rapport à la qualité de la prédiction. Une instance de résultats obtenu avec les données d'un égo-véhicule effectuant une manœuvre de `Sheer_out` est présentée dans la Figure 5.9a. Comme observé, la qualité de la prédiction est médiocre surtout en termes de prédiction de la vitesse future de l'égo-véhicule. Après une analyse détaillée de la représentation apprise, celle-ci semble avoir encodé, sur une seule des composantes de z, les vitesses longitudinales possiblement prises par l'égo-véhicule lors de la réalisation d'une trajectoire particulière. La composante z_2 n'est pas structurée. Ceci est vérifié par la forme de la distribution de l'encodeur agrégé dans la Figure 5.9b. Par exemple, pour la Figure 5.9a, lors de la réalisation d'un dépassement, le réseau génère via des tirages différents de z des trajectoires plus ou moins agressives en terme de vitesse à laquelle le véhicule commence à se diriger vers la voie du milieu. Après un entraînement avec un jeu de données équilibré, les RMSEs en position et vitesse sont de 0.45 et 0.15 respectivement Le pourcentage de tests positifs du khi-deux s'élève à seulement 73%. Bien qu'une structure soit extraite de l'apprentissage, ces métriques de performance sont trop faibles pour suggérer une utilisation de ce modèle dans le cadre de notre méthode de prédiction présentée au chapitre précédent.

5.3.2.4 Recuit cyclique de β

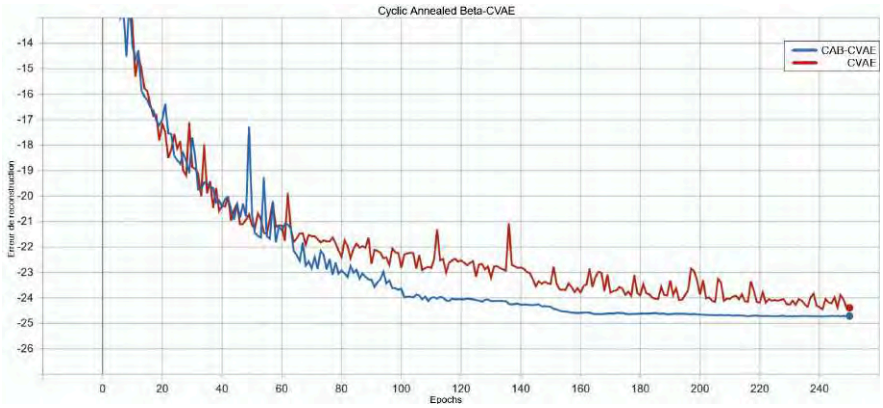
Dans la section précédente, β est fixé pendant toute la durée de l'apprentissage. Dans [158], les auteurs proposent de faire varier ce paramètre de manière à maximiser la quantité d'information introduite dans les variables latentes tout en gardant de bonnes performances de reconstruction. Le nouveau modèle introduit est appelé CVAE avec recuit simulé de beta (Cyclic Annealing Beta-CVAE – CAB-CVAE). En effet, un des problèmes majeurs dans l'apprentissage des (C)VAEs est la disparition au fil des époques du terme de divergence de KullBack-Leibler du prior vers l'encodeur dans la fonction objectif. Cette disparition pourrait être causée par la faible quantité d'information encodée dans z ce qui rend par la suite plus difficile la prédiction (ou reconstruction)

du décodeur [158] et qui incite donc le décodeur à ignorer les variables latentes. Ce phénomène est accentué par la capacité de représentation du décodeur par rapport au problème d'apprentissage considéré. En augmentant progressivement le facteur sur la divergence KL plusieurs fois pendant l'apprentissage, le code latent est appris progressivement et le réseau a moins tendance à converger vers la solution triviale qui consiste à prédire la sortie en ignorant ce code.

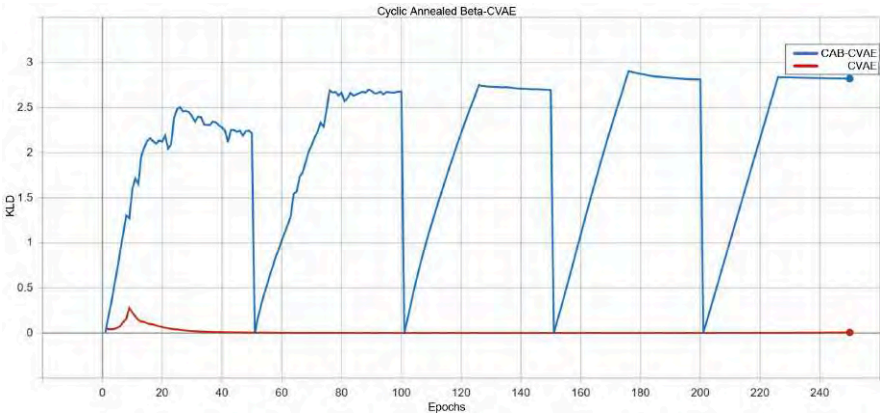
L'hyperparamètre β est donc modulé à travers les époques de façon à alterner cycliquement entre une valeur minimale et une valeur maximale. Les courbes d'apprentissage du CVAE *Vanilla* et du Cyclic Annealing Beta-CVAE pour l'ELBO, l'erreur de reconstruction et le terme de divergence KL (KLD) du prior des variables latentes vers l'encodeur sont affichées dans la Figure 5.10. Pour commencer, l'erreur de reconstruction obtenue pour le Cyclic Annealing Beta-CVAE est inférieure à celle du CVAE tout en gardant le terme de divergence KL supérieur tout au long de l'apprentissage. Pour le CVAE *Vanilla*, ce dernier s'effondre dès les premières époques indiquant que le réseau ignore ce terme et par extension, ignore les variables latentes. Les performances du CAB-CVAE s'améliorent au fil des trois premiers cycles de variation de β avec une amélioration franche lors du début du deuxième, à la $60^{\text{ième}}$ itération. En effet, à partir de ce moment dans l'apprentissage, l'erreur de reconstruction diminue fortement et sa valeur est significativement en dessous de celle du CVAE *Vanilla*. Comme dans [158], ceci suggère que le code latent est construit progressivement au fil des cycles, et est utilisé pour améliorer les performances de prédictions du réseau tout au long de l'apprentissage. En ce qui concerne le CVAE *Vanilla*, une disparition du terme de la divergence KL est observée, l'ELBO est égale à l'erreur de reconstruction. Très rapidement au cours de l'apprentissage, le décodeur ignore les variable latentes pour générer y.



(a)



(b)



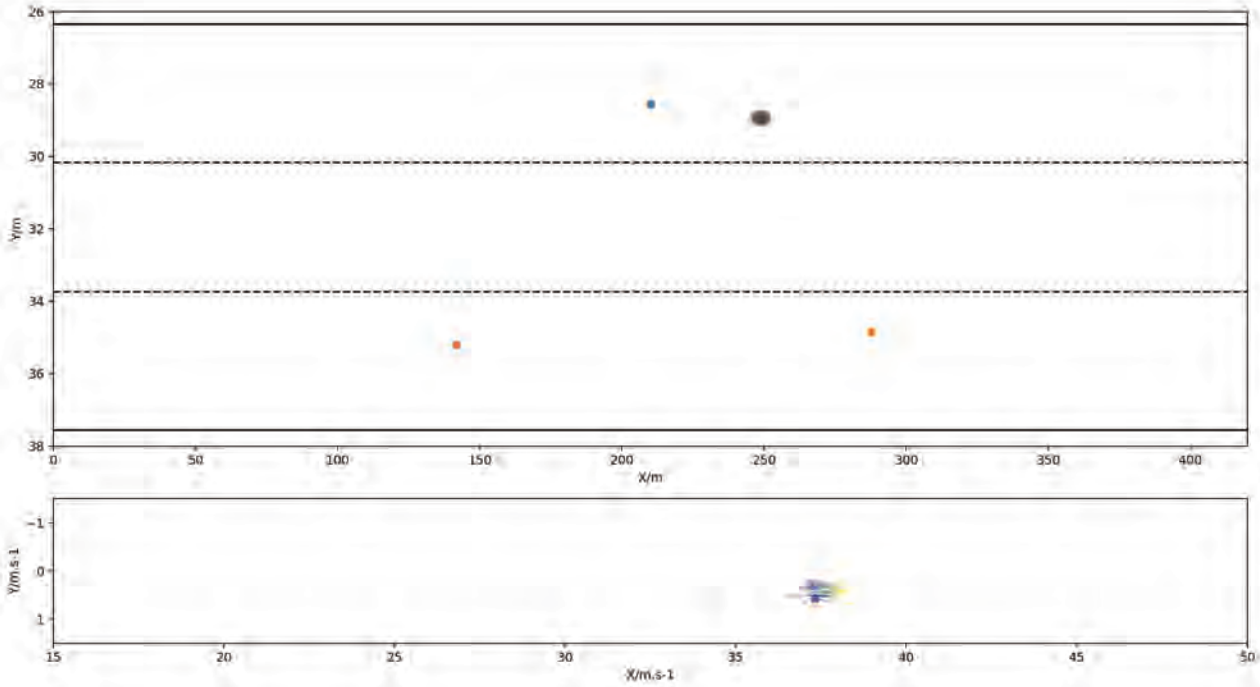
(c)

FIGURE 5.10 – Courbes d'apprentissage obtenues au fil de l'entraînement d'un CVAE *Vanilla* et d'un CAB-CVAE sur les données HighD. Trois valeurs sont affichées : (a) l'ELBO, (b) l'erreur de reconstruction et (c) la valeur du terme de divergence KL.

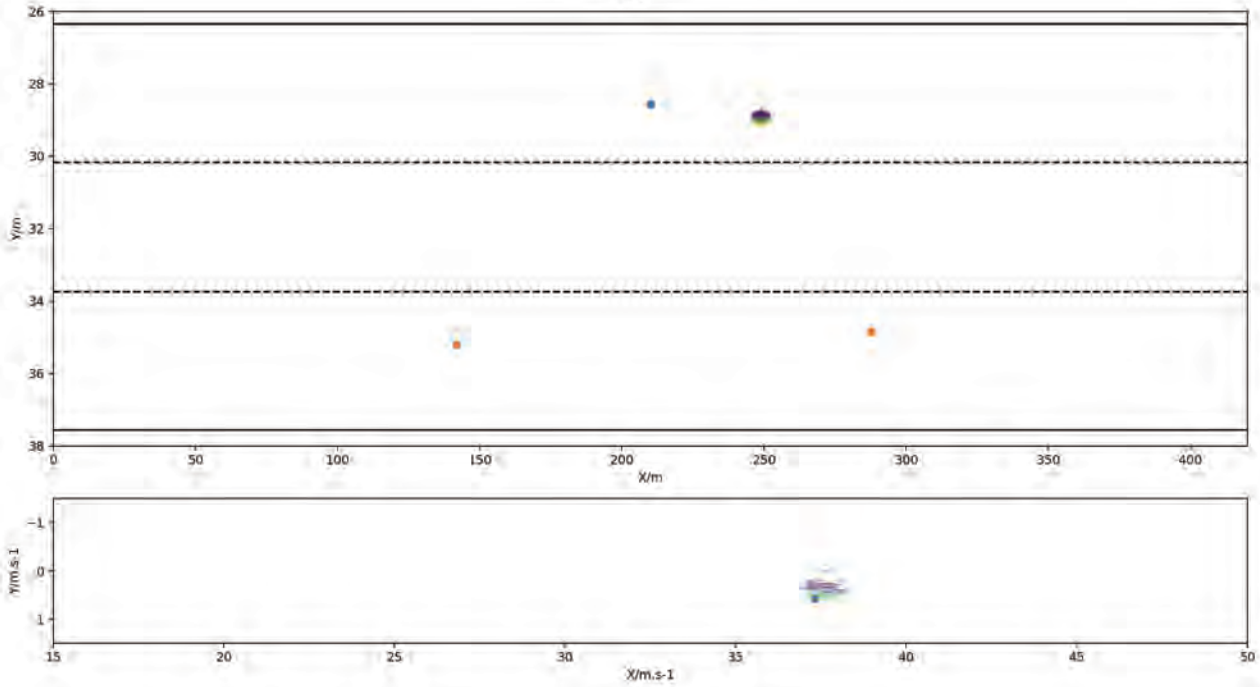
Les résultats obtenus avec le CAB-CVAE sur les données HighD sont illustrés avec un exemple de prédiction dans la Figure 5.11. L'encodeur agrégé associé est présenté dans la Figure 5.12. Le modèle produit lors de cet apprentissage parvient à extraire des données une structure pouvant per-

mettre une interprétation du processus de génération des trajectoires de l'égo-véhicule. Cependant, comme montré dans la Figure 5.13, il est difficile de partitionner l'espace latent en zones associées à chaque manœuvre recherchée. Pour cet apprentissage, il est difficile de générer des trajectoires ou d'interpréter des trajectoires selon les manœuvres présentées au chapitre précédent. Bien que la plupart des ellipses de confiance associées aux instances de l'encodeur soient de petites tailles, cela ne signifie pas que la représentation apprise est dans le cas dégénéré d'une table de correspondance comme présenté plus tard dans la Section 5.3.3. En effet, les points proches associés à des régions de l'espace latent sont associés à des futurs de l'égo-véhicule proches en termes de position et de vitesse. De plus, en ignorant les valeurs aberrantes, l'encodeur couvre raisonnablement le domaine du prior en produisant à la fois des codes latents ne se recouvrant pas totalement.

Des RMSEs de 0.05 et de 0.06 sont obtenues en position et en vitesse respectivement. Le test du khi-deux est positif pour 90% des données de validation. Il est important de rappeler que ces métriques sont obtenues lors d'un apprentissage faiblement supervisé avec des données équilibrées en termes de fréquence de manœuvres. Le même apprentissage est effectué avec l'intégralité des données highD non-équilibrées. Les nouvelles métriques de performance obtenues sur l'ensemble de validation sont sensiblement les mêmes. Afin de mesurer les performances en prédiction pour chaque manœuvre identifiée au chapitre précédent avec le modèle appris de manière non supervisé, les métriques sont calculées une nouvelle fois sur un jeu de validation provenant des données équilibrées. Dans ce cas, les RMSEs ne varient pas significativement mais le pourcentage positif de tests du khi-deux n'est plus que de 81%.



(a)



(b)

FIGURE 5.11 – Sorties du CAB-CVAE obtenues pour une instance aléatoire des données HighD. Les distributions de prédiction en position et en vitesse sont obtenues avec 1000 échantillons de z et sont représentées par les points et les ellipses associées colorées. Chaque graphe (a-b) illustre respectivement le processus de génération selon la valeur de chacune des deux composantes de z.

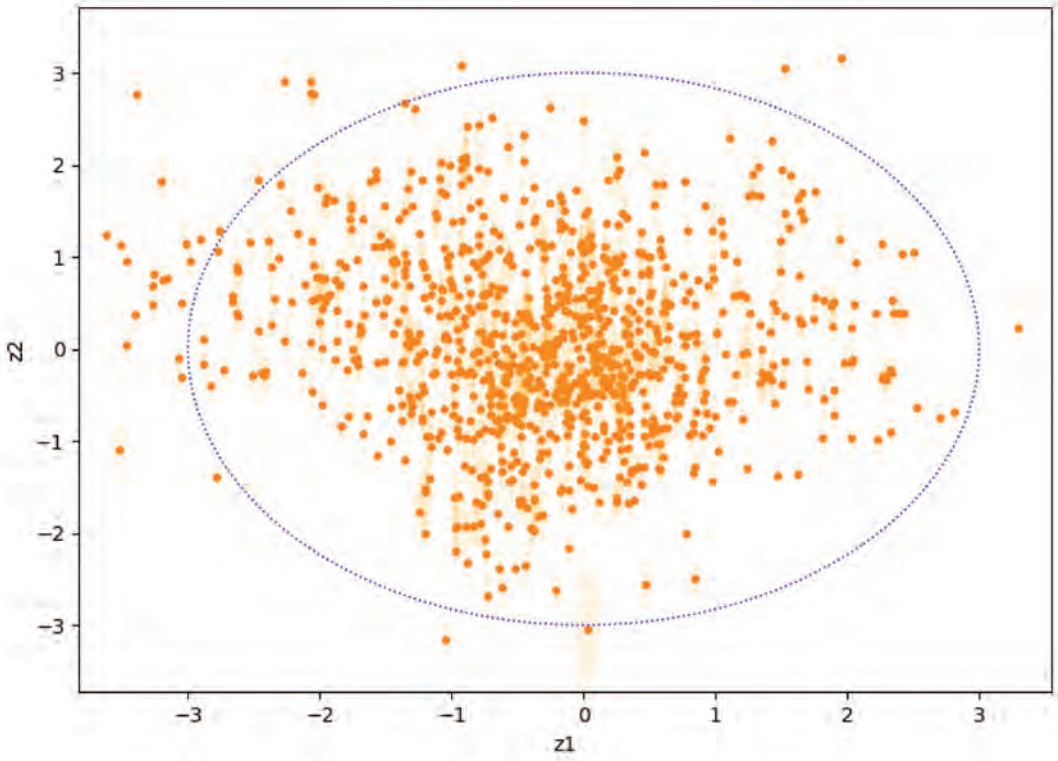
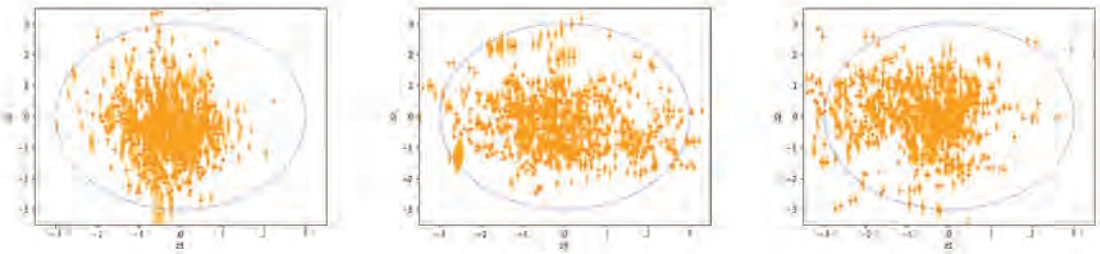


FIGURE 5.12 – Approximations de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ d'un CAB-CVAE entraîné sur les données HighD pour 1000 échantillons des couples entrée-sortie. Le réseau semble produire des codes latents répartis convenablement sur le domaine du prior avec un recouvrement raisonnable des instances des encodeurs. On observe toutefois quelques codes latents dans les queues de leur prior.



(a) (b) (c)

FIGURE 5.13 – (a-b-c) Approximations de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ d'un CAB-CVAE entraîné sur les données HighD pour 1000 échantillons des couples entrée-sortie sélectionnées manuellement comme correspondant aux manœuvres `Forward`, `Sheer_in` et `Sheer_out` respectivement pour chaque graphe. La répartition observée des codes latents ne permet pas l'interprétation du modèle de transition égocentrique en termes de manœuvre réalisée. Il est difficile d'isoler sur le domaine de z des zones correspondant aux diverses manœuvres considérées.

5.3.3 L'Info-CVAE

Dans cette section un nouveau modèle est introduit, l'Info-CVAE. Avec ce modèle, nous cherchons à introduire un nouveau paramétrage du CVAE permettant de mieux contrôler la forme prise par l'encodeur agrégé lors de son apprentissage non supervisé.

5.3.3.1 Décomposition du code latent

Dans [159], les auteurs introduisent la notion de décomposition comme une généralisation du concept de démêlage de l'espace latent présenté précédemment en la définissant comme la réalisations de deux objectifs contradictoires : les codes latents doivent avoir un niveau approprié de superposition et l'agrégation de ses codes doit se conformer à une structure désirée imposé par le choix d'une distribution a priori. La démêlage des variables latentes est donc un cas spécifique de le décomposition. Dans le contexte des auto-encodeurs variationnels, le démêlage des variables latentes revient à apprendre un modèle dont les composantes des variables latentes représentent chacune indépendamment un facteur génératif des données. Dans notre cas, un des facteurs génératifs d'intérêt est la manœuvre entreprise par le conducteur pour réaliser la trajectoire continue. Selon [159], la notion de démêlage des variables latentes est trop restrictive car il existe peu de jeux de données réelles exhibant un ensemble de facteurs génératifs indépendants. Pour le cas particulier des CVAEs, la décomposition caractérise le fait que chaque valeur prise par z doit encoder une plage de données (x, y) d'une taille raisonnable pour le problème considéré.De plus, au terme de l'apprentissage, l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ doit correspondre au prior $p(z)$ afin de respecter la structure attendue sur les codes latents.

La Figure 5.14 illustre le concept de décomposition pour un VAE, pour une distribution a priori de z en forme de croix. Dans le cas des CVAEs, le premier objectif de superposition peut être facilement expliqué en considérant les deux cas extrêmes où : lors d'une superposition trop faible, l'encodeur devient une simple table de correspondance de (x, y) vers z ; lors d'une superposition trop importante, un nombre important de valeurs de z correspondent aux même point x, y. Dans les deux cas l'encodage ne capture pas l'information adéquate sur les données observées. L'objectif de correspondance entre l'encodeur agrégé et le prior sur les variables latentes permet de contraindre simplement notre réseau à générer des données selon une structure choisie. C'est la forme de la distribution a priori qui représente l'objectif de dépendance ou d'indépendance entre les dimensions de z.

Afin d'éviter de tomber dans un des deux cas où les codes latents appris ne sont pas, ou sont peu informatifs, et pour remplir l'objectif de décomposition présenté, [159] propose de modifier la fonction objectif du CVAE de façon à pouvoir contrôler en même temps la superposition des codes latents et le respect de la forme du prior de la distribution de l'encodeur agrégé. Ce nouvel objectif s'écrit tel que

$$\begin{aligned}\mathcal{L}_{\beta,\gamma}(x, y, \theta, \phi) = \mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(y|x, z) \\ - \beta D_{KL}(q_\phi(z|x, y)||p_\theta(z)) \\ - \gamma \mathbb{D}(q_\phi^{\text{agg}}(z), p_\theta(z))\end{aligned} \tag{5.24}$$

où γ et β contrôlent respectivement le respect envers le prior de l'encodeur agrégé et la superposition des codes latents respectivement. En plus de l'altération de l'objectif du CVAE *Vanilla*, les auteurs proposent de réaliser l'entraînement avec des formes spécifiques de distribution a priori des variables latentes encourageant l'apprentissage d'une représentation démêlée, partitionnée ou sporadique.

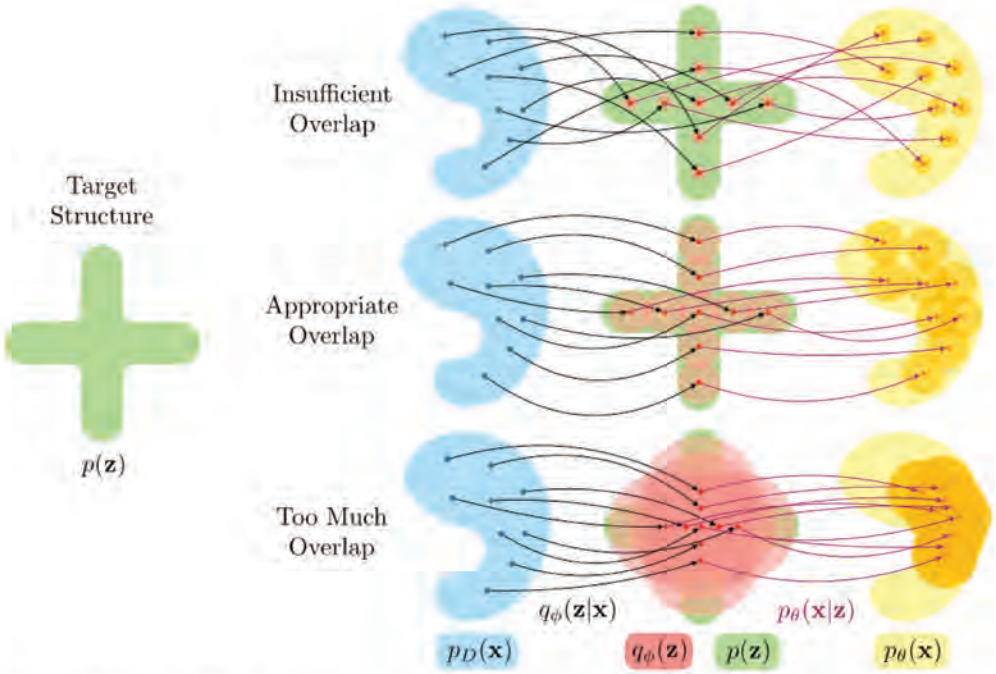


FIGURE 5.14 – D'après [159]. Illustration du concept de décomposition pour l'apprentissage d'un VAE où la structure désirée pour les variables latentes est imposée par une distribution a priori $p(z)$ en forme de croix. (Haut) Dans le cas où l'encodeur produit des codes qui se superposent trop peu, un phénomène de table de correspondance se produit. L'encodeur associe des points x à z de manière non structurée. En conséquence, le décodeur génère des points proches dans l'espace des données mais qui ne le sont pas dans l'espace latent. Les variables latentes n'encodent donc pas d'information sur les données. (Bas) La situation inverse se produit ici. Les codes latents se recouvrent trop entre eux et dans ce cas également les variables latentes encodent trop peu d'information. Un point des données peut appartenir à un nombre trop important de codes différents. (Milieu) Ici, l'objectif de décomposition est atteint. Les codes latents ont une superposition adéquate et la forme de l'encodeur agrégé respecte celle du prior.

5.3.3.2 Présentation du modèle

Après avoir établi le besoin de contrôler plus finement l'encodeur, une instanciation des considérations exposées dans la section précédente est présentée : l'Info-CVAE [160]. Ce modèle encourage la décomposition des variables latentes du CVAE en introduisant un terme supplémentaire dans sa fonction objectif en plus d'un facteur sur le terme de divergence. Ce modèle possède donc deux degrés de liberté régissant le respect du prior par l'encodeur et la couverture par l'encodeur agrégé de ce même prior sur les variables latentes. Le cadre présenté dans [160] est utilisé dans un contexte ITS dans [161] où les auteurs ont recours à un auto-encodeur variationnel conditionnel pour la prédiction de trajectoires avec un apprentissage sur des données routières hétérogènes. L'Info-CVAE est construit de manière similaire au CVAE *Vanilla*. Sa fonction objectif différente

permet de réaliser 5.24, et est définie comme

$$\begin{aligned}\mathcal{L}_{\alpha,\lambda}(x,y,\theta,\phi) = \mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(y|x,z) \\ - (1-\alpha) D_{KL}(q_\phi(z|x,y)||p_\theta(z)) \\ - (\alpha+\lambda-1) D_{KL}(q_\phi(z)||p_\theta(z))\end{aligned} \tag{5.25}$$

Les deux premiers termes correspondent à ceux dans l'équation (5.5) et sont optimisés de la même manière que pour le CVAE *Vanilla*. Le dernier terme est plus difficile à calculer car il est impossible d'évaluer $\log(q_\phi(z))$ de façon tractable. Dans [160], les auteurs démontrent que leur modèle reste applicable si ce terme est remplacé par n'importe quelle autre divergence stricte. La notion de divergence stricte implique pour notre cas que $D(q_\phi(z)||p_\theta(z)) = 0$ ssi $q_\phi(z) = p_\theta(z)$. Plusieurs familles de divergences sont proposées dans l'article. En considérant le cadre établi par l'utilisation d'un prior Gaussien, notre objectif de modularité et de performances calculatoires, nous optons pour l'utilisation de l'Écart Maximum-Moyen (Maximum-Mean Discrepancy – MMD) comme fonction de divergence. En supposant un noyau $k(.,.)$ défini positif, la MMD entre deux distributions p et q s'écrit

$$\begin{aligned}D_{MMD}(q||p) = \mathbb{E}_{p(z),p(z')}[k(z,z')] - 2\mathbb{E}_{q(z),p(z')}[k(z,z')] \\ + \mathbb{E}_{q(z),p(z')}[k(z,z')]\end{aligned} \tag{5.26}$$

avec $D_{MMD} = 0$ si et seulement si $p = q$. Ici, le noyau utilisé est le noyau Gaussien paramétré par σ^2_{kernel} défini comme suit :

$$k_{\mathcal{G}}(z,z') = \exp\left(-\frac{|z-z'|^2}{2\sigma^2_{\text{kernel}}}\right). \tag{5.27}$$

Les espérances dans le termes de la MMD ne peuvent pas être calculés analytiquement et sont donc approchées via la méthode de Monte Carlo.

5.3.3.3 Apprentissage et résultats sur les données synthétiques

Tout comme pour les modèles précédents, l'implémentation de l'Info-CVAE est validée sur les données d'une dynamique synthétique. Le réseau est construit avec la même structure pour l'encodeur et le décodeur que pour le CVAE *Vanilla*. Afin de démontrer la possibilité, via la fonction objectif, de contrôler sur deux degrés de liberté la forme de l'espace latent pendant l'apprentissage, des premiers résultats sont proposés dans la Figure 5.15. Ici le réseau est appris en fixant $\alpha = 0$ et $\lambda = 8$. Pour cet apprentissage, une RMSE de 0.18 est obtenu. C'est une valeur élevée au regard des résultats obtenus par les modèles précédents sur les mêmes données. Ceci peut être expliqué

avec la représentation apprise via les variables latentes. La visualisation de l'encodeur agrégé et la distribution des points de couleurs révèle que l'information encodée dans les variables latentes z_1 et z_2 représente respectivement pour chacune des deux variables la position sur l'axe vertical et l'axe horizontal délimités tous les deux approximativement par les instances aux extrémités de l'ensemble des données. Les covariances de l'encodeur recouvrent trop peu de points dans l'espace latent. Le réseau a finalement appris une représentation qui correspond a une rotation et à une échelle près à la position 2D des points y sur l'image. Bien que l'objectif de l'Info-CVAE semble promouvoir une répartition des moyennes de l'encodeur sur une zone raisonnablement étendue, cette hyper-paramétrisation ne produit pas une représentation adéquate car non informative quant aux facteurs génératifs des données. Les données générées en sortie s'étendent uniformément sur quasiment l'entièreté de l'espace 2D. Cette dernière remarque explique les faibles performances du modèle en termes de RMSEs. La Figure 5.16 illustre un apprentissage depuis les mêmes données avec une hyper-paramétrisation différente : $\alpha = -0.35$ et $\lambda = 8$. Ces nouveaux paramètres mettent plus de poids sur le terme de la divergence de Kullback-Leibler du prior vers l'encodeur qui incite l'encodeur de produire des distributions proches d'une Gaussienne multivariée standard. Les moyennes de l'encodeur sont réparties de façon à couvrir une zone d'une taille acceptable dans l'espace latent. Par rapport au premier apprentissage, les covariances produites ne sont plus aussi étriquées et se recouvrent suffisamment entre elles pour faire correspondre des points proches dans l'espace des données à des points proches dans l'espace latent. En visualisant la répartition des couleurs des points générés par rapport à z_1, les couleurs allant du jaune au vert-bleu semblent encoder la partie de l'espace latent correspondant à un des deux côtés de la symétrie centrale de l'ensemble d'apprentissage. De plus, l'espace latent peut être divisé en deux (du nord-est au sud-ouest) pour séparer, d'après la valeur de z_1 et z_2, les données y générées aléatoirement conditionnellement aux entrées x sur deux extrémités différentes. Ces deux apprentissages illustrent la possibilité de l'objectif de l'Info-CVAE de contrôler sur deux degrés de liberté la forme des codes latents. Ces résultats semblent donc valider l'implémentation réalisée de l'Info-CVAE pour cette étude. Il convient de noter tout de même que la recherche des « bons » hyperparamètres du réseau n'est pas évidente.

5.3.3.4 Apprentissage et résultats sur les données HighD

Comme pour la validation du modèle sur les données synthétiques, l'architecture de l'Info-CVAE ne diffère pas de celle du CVAE *Vanilla*. La Figure 5.17 présente l'encodeur agrégé pour trois apprentissages différents de l'Info-CVAE avec respectivement $\lambda = 10, 100$ et 1000. Pour ces apprentissages, $\alpha = 0$. Au fur est à mesure que le facteur sur le terme de distance MMD augmente, la taille des covariances de l'encodeur diminue. Les moyennes se situent toutes sur une des deux coordonnées de l'espace latent : $(-1, 1)$ ou $(1, -1)$. Cette structure sur z pourrait s'apparenter au

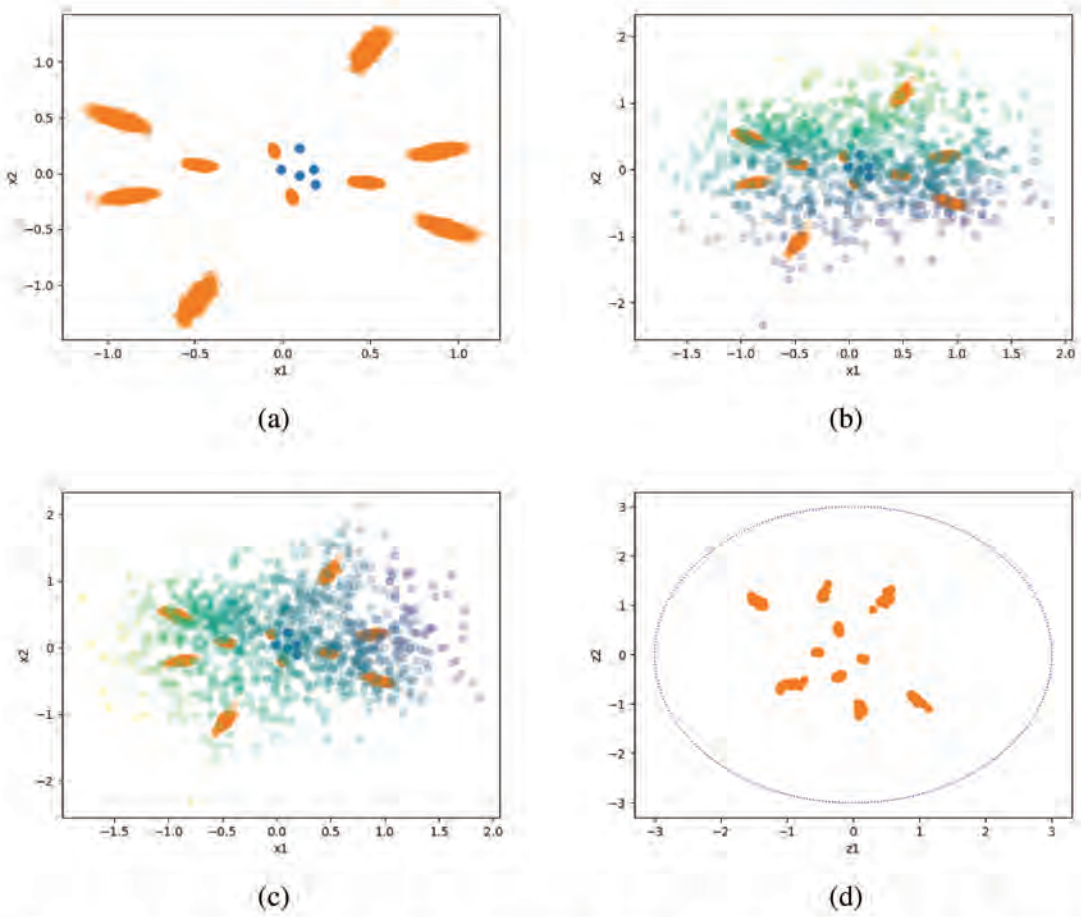


FIGURE 5.15 – Résultats obtenus pour l'apprentissage de l'Info-CVAE avec comme hyper-paramètres $\alpha = 0$ et $\lambda = 8$. (a) Données synthétiques utilisées pour l'apprentissage. (b-c) Sorties de l'Info-CVAE obtenues avec 1000 échantillons de la variable latente z. (d) Approximation du postérieur agrégé $q_\phi^{\text{agg}}(z)$ pour 1000 échantillons des couples entrée-sortie.

code latent d'une variable discrète pouvant prendre deux valeurs. Cependant, quelles que soient les valeurs de x et y, l'encodeur produit des moyennes et covariances identiques suggérant une absence de prise en compte des données. La Figure 5.18 illustre le résultat sur une instance des données HighD depuis un apprentissage de l'Info-CVAE avec $\lambda = 100$. Étonnamment, bien que l'espace latent soit séparé en deux pôles distincts, ceux-ci font correspondre les codes latents à des points extrêmement similaires dans l'espace des données que ce soit en position ou en vitesse. Une superposition des ellipses de prédiction jaunes et violettes en position et en vitesse est observée. Pour cette solution particulière, la forme de l'encodeur agrégé dépend uniquement du poids du terme de divergence KL par rapport à celui de la distance MMD et non des données en entrées et sorties du réseau.

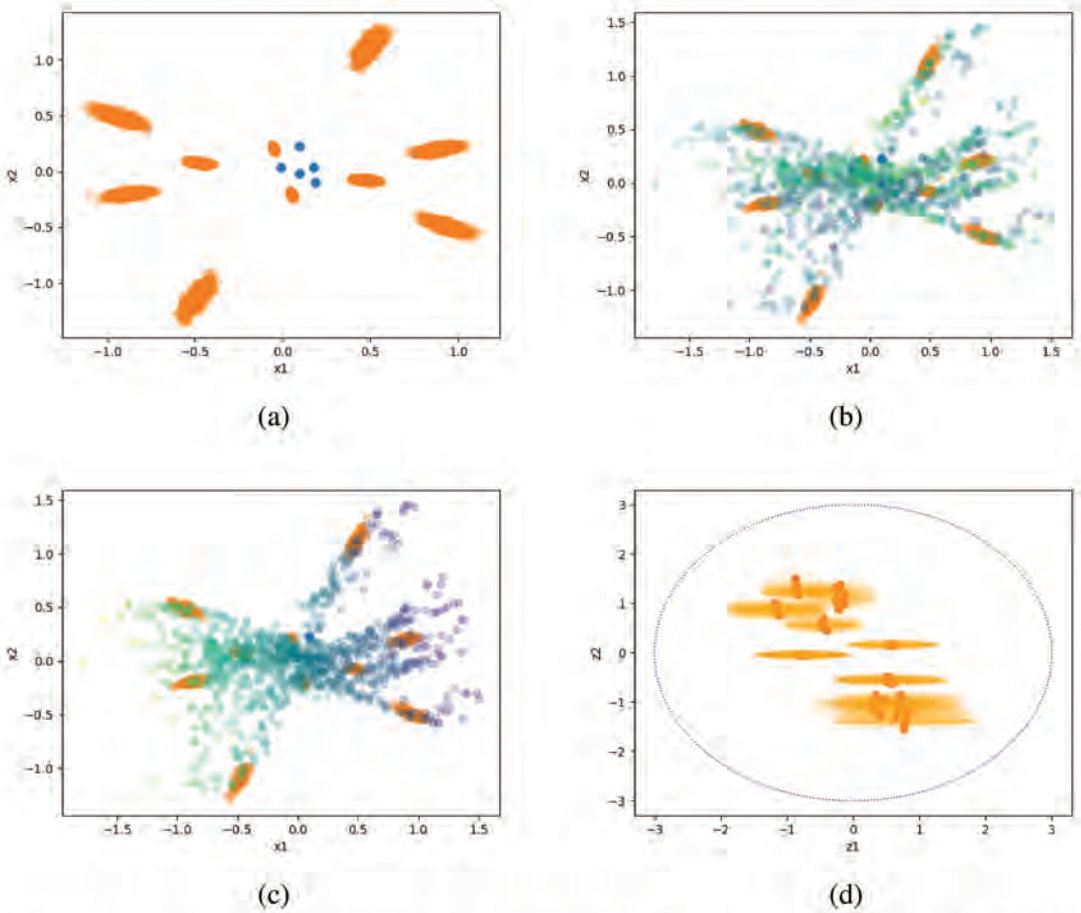
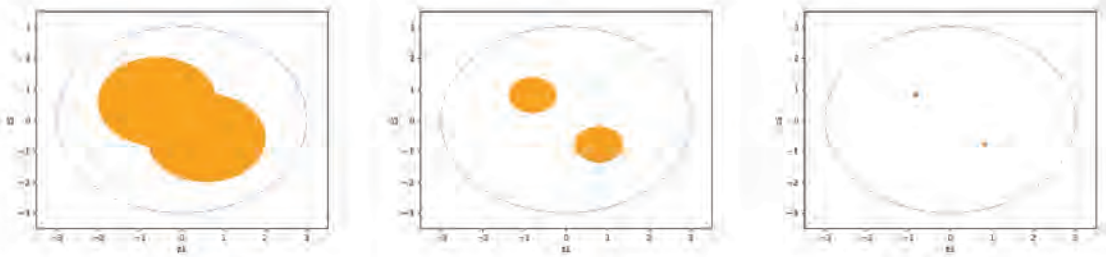


FIGURE 5.16 – Résultats obtenus depuis l'apprentissage de l'Info-CVAE avec pour hyperparamètres $\alpha = -0.35$ et $\lambda = 8$. (a) Données synthétiques utilisées pour l'apprentissage. (b-c) Sorties de l'Info-CVAE obtenues depuis 1000 échantillons de la variable latente z. (d) Approximation du postérieur agrégé $q_\phi^{\text{agg}}(z)$ pour 1000 échantillons des couples entrée-sortie.

Pour cette apprentissage réalisé de manière intégralement non supervisé, des RMSEs de 0.09 et de 0.23 sont obtenues en position et en vitesse respectivement. Le pourcentage de tests postif du khi-deux s'élève à 84%. Mesurées sur des données non équilibrées, les performances en prédiction se détériorent, le même apprentissage produit des RMSEs de 0.12 et 0.26 en position et vitesse respectivement, avec un score de tests du khi-deux de 80%. Bien que les RMSEs soient acceptables, les résultats pour le test du khi-deux ne semblent pas être suffisamment bons pour pouvoir exploiter le modèle dans le cadre présenté au chapitre précédent.

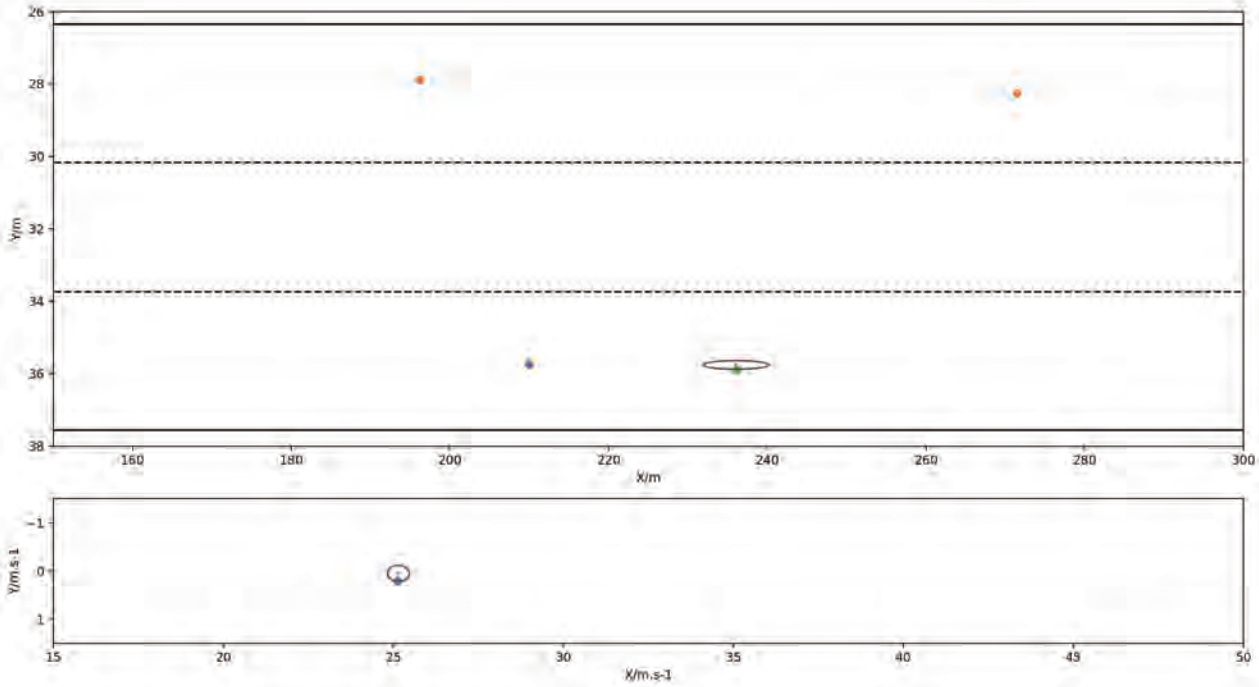
Un nombre conséquent d'apprentissages est réalisé pour rechercher une paramétrisation adéquate de l'Info-CVAE sur les données HighD. Après les premiers résultats obtenus en faisant varier λ, c'est au tour de l'hyperparamètre α d'être modulé. Comme pour les données synthétiques, plu-

sieurs apprentissages sont effectués pour tenter de trouver la valeur α permettant un équilibre entre le recouvrement des codes latents et la couverture par l'encodeur agrégé de $p(z)$. Les apprentissages sont répartis équitablement entre des apprentissages non supervisés et des apprentissages faiblement supervisés. Malheureusement, aucune paramétrisation du réseau ne parvient à produire un résultat convenable en termes de structure de l'espace latent et de performances de prédiction (RMSEs en position et vitesse avec test du khi-deux) sur les trajectoires. Du fait de sa fonction objectif et des hyperparamètres du réseaux, l'Info-CVAE semble être un modèle difficile à entraîner pour notre problème.



(a) $\lambda = 10$ (b) $\lambda = 100$ (c) $\lambda = 1000$

FIGURE 5.17 – (a-b-c) Approximations de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$ depuis 1000 échantillons de l'encodeur depuis l'apprentissage d'un Info-CVAE pour $\lambda = 10, 100$ et 1000 sur les données HighD. Pour les données HighD, le réseau semble produire des codes se concentrant sur deux pôles. Au fur et à mesure que λ augmente, les codes latents sont répartis aux mêmes endroits sur l'espace latent mais les ellipses de probabilité à 99% associées rétrécissent jusqu'à disparaître entièrement. Dans les trois cas, l'apprentissage semble avoir produit un encodeur dégénéré n'encodant pas d'information sur le processus de génération des trajectoires. En effet, quelle que soit la paramétrisation du réseau, les échantillons sont tous regroupés parfaitement autour de deux coordonnées indépendamment des données x, y.



(a)

FIGURE 5.18 – Sorties obtenues de l'Info-CVAE appris avec $\lambda = 100$ pour une instance aléatoire des données HighD. Les distributions de prédiction en position et en vitesse sont obtenues depuis 1000 échantillons de z et sont représentées par les points et les ellipses associées colorées. Pour faciliter la lecture, seuls les contours des ellipses sont affichés. De plus, seul le processus de génération selon la composante z_1 est illustré ici. Une superposition des distributions de trajectoires prédites est observée quelle que soit la valeur de z_1. Le même constat est fait pour z_2.

5.3.4 Un modèle inspiré des auto-encodeurs variationnels dynamiques

5.3.4.1 Introduction

Dans [162], les auteurs proposent une revue et une présentation unifiée d'une sous catégorie de modèles génératifs profonds ; les Réseaux Bayésiens Dynamiques Profonds (Deep Dynamical Bayesian Network – DDBN). Ces modèles sont construits sur la base des Réseaux Bayésiens Dynamiques présentés dans la section 2.2.4 combinés à l'utilisation de réseaux de neurones profonds pour la modélisation des distributions de probabilités des variables aléatoires du réseau Bayésien. Cette combinaison donne le potentiel aux DDBNs de promouvoir une forme d'interprétabilité en modélisant les dépendances entre les variables aléatoires mises en jeu via des distributions de probabilité tout en profitant de la capacité de représentation des réseaux de neurones profonds. Tous les modèles de CVAEs présentés précédemment sont fondamentalement des Réseaux Bayesiens profonds qui n'incluent pas la la notion de temporalité, que ce soit au niveau de leur entrées/sorties

ou de leur variables latentes. Deux instances consécutives dans le temps de données x, y qui formeraient une trajectoire sont traitées sans considérer de dépendance temporelle par le modèle appris. Il n'y également pas de mise en correspondance d'une séquence de données avec une variable latente ou une séquence de variables latentes. Dans [162], des extensions du VAE sont proposées. Elles permettent de prendre en compte certaines dépendances temporelles absentes du (C)VAE standard.

5.3.4.2 L'Auto-Encodeur Variationnel Récurent Conditionnel

Une des variations présentées s'avère particulièrement intéressante pour notre étude : l'Auto-Encodeur Variationnel Récurent Conditionnel (Conditional Variational Recurrent Autoencoder – CVRAE) [163]. Un CVRAE encode des séquences de données d'entrée et de sortie $(x_{1:\tau-1}, y_{\tau:T})$, réalisations de processus aléatoires, dans un vecteur latent statique z. Le processus génératif du CVRAE s'écrit comme suit :

$$p_\theta(y_{\tau:T}, z|x_{1:\tau-1}) = p_\theta(z) \prod_{t=\tau}^{T} p_\theta(y_t|y_{\tau:t-\tau}, z, x_{1:\tau-1}). \tag{5.28}$$

Comme pour les variations de CVAE précédentes, le prior est une Gaussienne bivariée standard. Dans cette instanciation d'auto-encodeur Variationnel Récurent Conditionnel, les variables latentes ne sont pas exprimées sous la forme d'une séquence. Le décodeur $p_\theta(y_t|y_{\tau:t-1}, z, x_{1:\tau-1})$ et l'encodeur associés $q_\phi(z|x_{1:\tau-1}, y_{\tau:T})$ sont tous les deux implémentés au moyen de réseaux de neurones récurrents (voir section 2.4.3). Le réseau décodeur admet en entrée z et $x_{1:\tau-1}$ pour calculer un premier état caché h_τ^{dec}. Chaque état caché suivant $h_{\tau+t}^{dec}$ est obtenu itérativement depuis le précédent en admettant en entrée $y_{\tau+t-1}$ et permet de produire les paramètres de la distribution de $y_{\tau+t}$. L'encodeur établit les paramètres de l'approximation de la loi postérieure du vecteur z des variables latentes à partir d'un état caché final h_T^{enc} obtenu itérativement similairement au décodeur. À la différence que pour l'encodeur, le premier état caché h_τ^{enc} est obtenu avec en entrée uniquement la séquence $x_{1:\tau-1}$. Le CVRAE établi donc deux modèles probabilistes liant une séquence d'entrée à un code latent et un code latent à une séquence décodée de sortie. Dans [162], des similarités sont exhibées entre le CVRAE et les réseaux récurrents stochastiques (STOchastic Recurrent Networks – STORN). Une étude détaillée des équations régissant ce dernier modèle est proposée dans cette référence.

5.3.4.3 *Instanciation d'un Auto-Encodeur Variationnel Récurent Conditionnel pour l'apprentissage du modèle de transition égocentrique de véhicules en interaction*

Le choix d'un modèle neuronal incluant des dépendances temporelles est motivé par l'utilisation du jeu de données de trajectoire HighD pour l'apprentissage. Dans la Section 4.4.1, l'information long terme sur le futur de l'égo-véhicule à un pas est représentée par l'étiquette de manœuvre associée après une segmentation considérant la trajectoire complète de l'égo-véhicule sur le segment routier. Dans le cas d'un apprentissage non-supervisé en manœuvre cette information est inexistante et doit être extraite automatiquement. Cependant, en alimentant le modèle neuronal avec seulement des futurs à un pas, il semble être difficile d'apprendre à différencier des futurs appartenant à une trajectoire différente en termes de manœuvres et ce, même avec un vecteur d'état suffisamment riche contenant des informations sur l'ensemble de la flotte de véhicules. Comme illustré dans la Figure 5.19, considérer la trajectoire complète pour extraire l'information de manœuvre pourrait permettre de mieux distinguer des futurs différents et ainsi encourager l'apprentissage d'une représentation où les manœuvres peuvent être encodées.

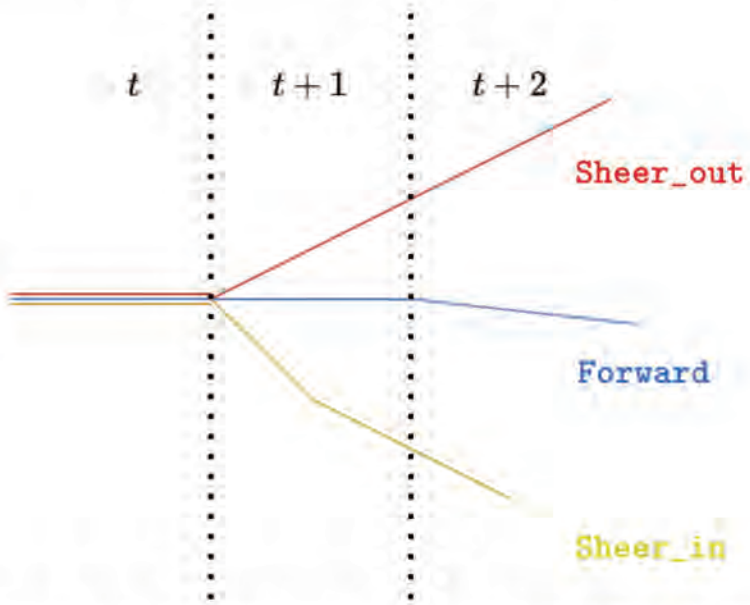
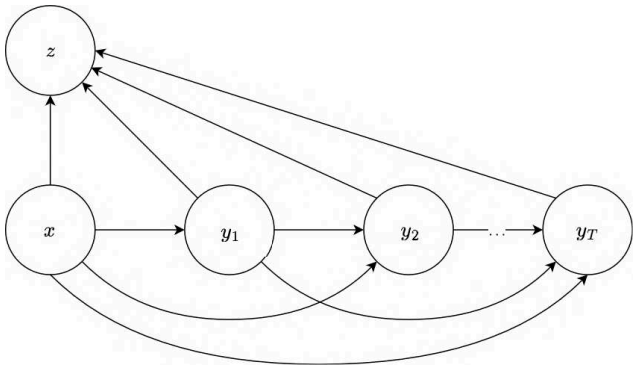


FIGURE 5.19 – Pour des pas de temps assez courts, une hypothèse peut être faite sur l'impossibilité d'observer depuis les données HighD des différences suffisamment importantes entre des manœuvres pour permettre leur identification de manière non supervisée.

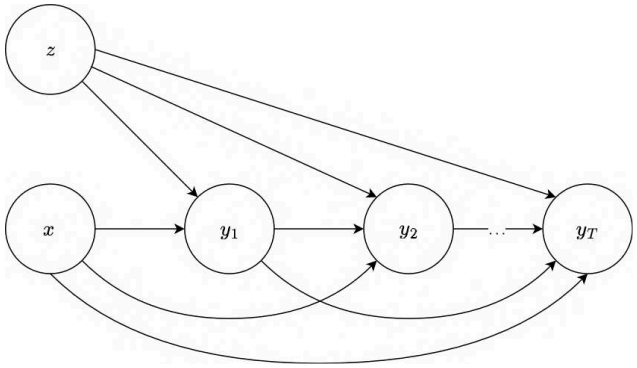
Dans le cadre de notre approche présentée au chapitre 4, nous avons fait l'hypothèse d'un vecteur d'état suffisamment riche pour permettre la description de la flotte et être Markovien ; l'objectif étant d'établir un modèle de transition Markovien, il n'est pas question de remettre en cause le fait que conditionnellement à sa valeur à l'instant courant, le vecteur d'état à un pas dans le futur est

indépendant des vecteurs d'état passés. Il est également impossible d'augmenter suffisamment le pas de temps sur lequel les données sont échantillonnées. En effet, un pas trop grand limiterait le nombre de mesures observées avant la prédiction et ceci dégraderait la performance des stratégies de filtrage présentées au chapitre précédent. De plus, le tronçon autoroutier considéré est trop court pour à la fois augmenter le pas de temps et établir une prédiction long terme des véhicules. Les mêmes remarques quant au pas de temps peuvent être faites pour les méthodes trajectoire-vers-trajectoire.

Pour nos besoins, le modèle CVRAE est donc modifié de telle sorte que le processus d'encodage des variables latentes repose sur l'utilisation d'un réseau récurent alimenté seulement avec le vecteur d'état complet de la flotte de véhicules à l'instant courant et une séquence des sous-vecteurs d'état futurs relatifs à l'égo-véhicule. Dans la même veine, le décodeur est alimenté par la variable latente et le vecteur d'état complet de la flotte de véhicules à l'instant courant pour produire une séquence de vecteurs d'état futurs de l'égo-véhicule. La Figure 5.20 représente les modèles graphiques du décodeur et de l'encodeur associé pour le réseau utilisé dans l'apprentissage de la dynamique de l'égo-véhicule. La forme de z est identique à celle utilisée pour les modèles CVAEs précédents. Le processus génératif est donc identique à celui de l'équation (5.28) où la séquence d'entrée $x_{1:\tau}$ est remplacée par l'entrée unique x. La séquence de sortie $y_{\tau:T}$ est maintenant simplement notée $y_{1:T}$.



(a)



(b)

FIGURE 5.20 – Modèles graphiques compacts du processus de reconnaissance (a) et du processus génératif (b) du CVRAE utilisés pour l'apprentissage de la dynamique de l'égo-véhicule. En reprenant les notations de notre modèle de transition égocentrique de la Section 5.2.5, sur ce graphe, $x = (x^k_{n-1}, \{x^l_{n-1}\}_{\mathcal{V}_l \in \mathcal{N}_{n-1}(\mathcal{V}_k)})$ et $y_1, y_2 \ldots, y_T = x^k_n, x^k_{n+1}, \ldots, x^k_T$.

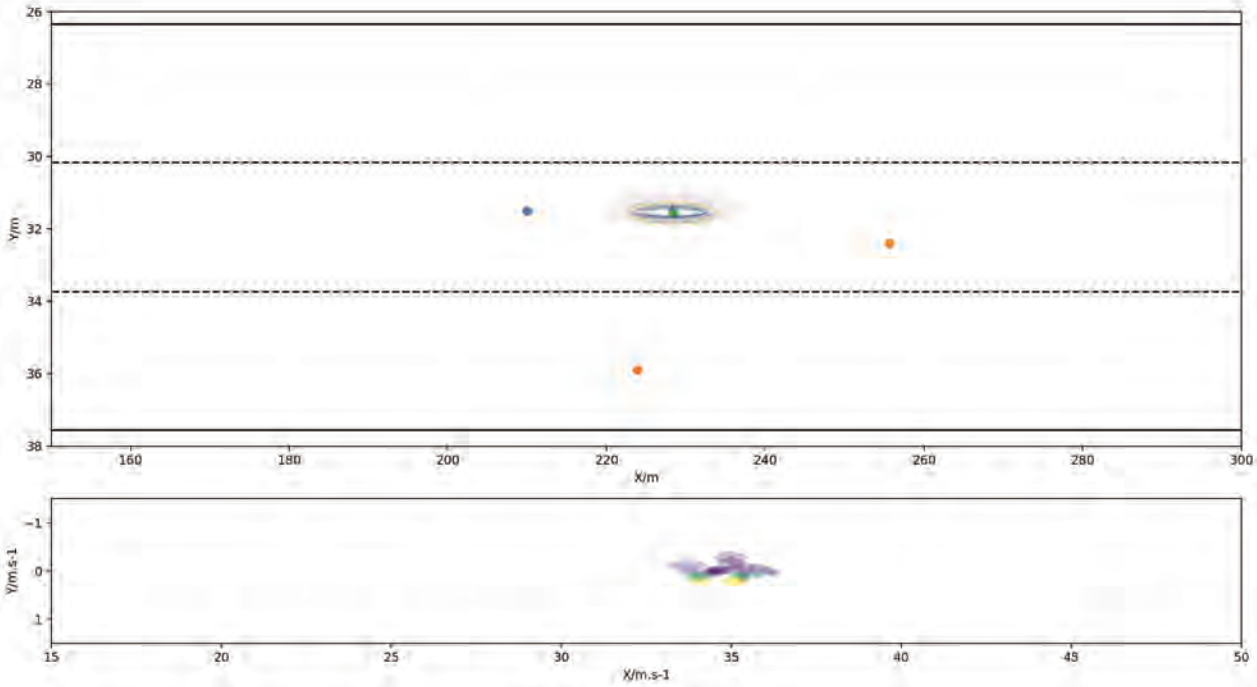
5.3.4.4 Apprentissage et résultats sur les données HighD

L'instance de CVRAE utilisé repose sur l'utilisation de deux LSTMs standards pour l'apprentissage du décodeur et de l'encodeur associé. Ces deux réseaux sont paramétrés identiquement avec une taille pour les couches cachées de 128 unités (ou cellules). L'apprentissage est effectué sur les mêmes données que pour les modèles présentés aux sections précédentes. Cependant, contrairement aux modèles non séquentiels, chaque instance d'apprentissage est maintenant composée d'un vecteur d'entrée x composé de l'état de la flotte à l'instant courant et d'un vecteur de sortie $y_{1:T}$ construit depuis une trajectoire (en position et vitesse) future de l'égo-véhicule s'étendant sur T instants. L'apprentissage est effectué avec une taille de séquence égale à $T = 3$s et avec un pas entre les instants de $T_s = 1.06$s. Le nombre total d'instances de données est maintenant de 12 000 000 avec un déséquilibre dans la distribution des trois classes de manœuvres similaire à celui observé dans les données non séquentielles.

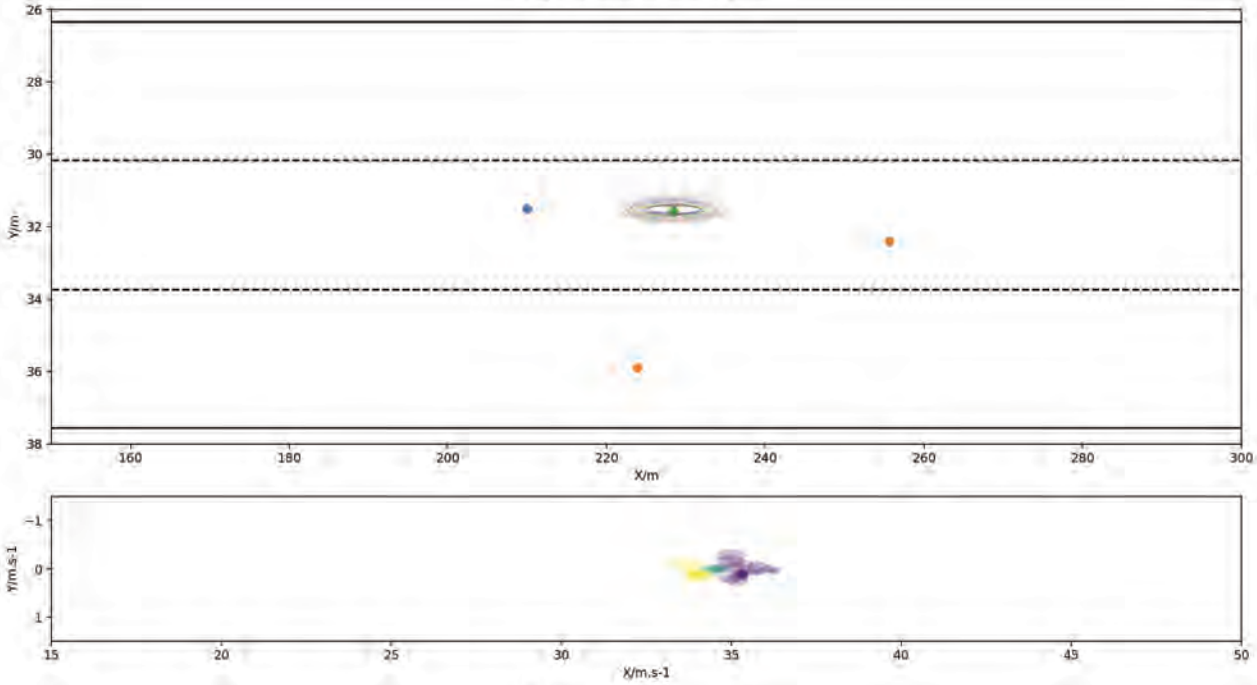
Les modèles de prédiction reposant sur l'utilisation de réseaux récurrents nécessitent un nombre supérieur d'instances de données pour faire converger l'apprentissage comparés aux réseaux Feed-Forward. Ceci est causé par le nombre plus important de paramètres à entraîner et de la présence de dépendances temporelles dans le modèle. Le CVRAE modifié est entraîné avec l'ensemble des données de trajectoire de manière non supervisée. La fonction objectif du CVRAE est définie comme suit :

$$\mathcal{L}_{\text{CVRAE}}(x, y_{1:T}, \theta, \phi) = \sum_{t=1}^{T} \mathbb{E}_{z \sim q_\phi(z|x, y_{1:T})}[\log p_\theta(y_t | y_{1:t-1}, x, z)] - D_{KL}(q_\phi(z|x, y_{1:T}) || p_\theta(z)), \tag{5.29}$$

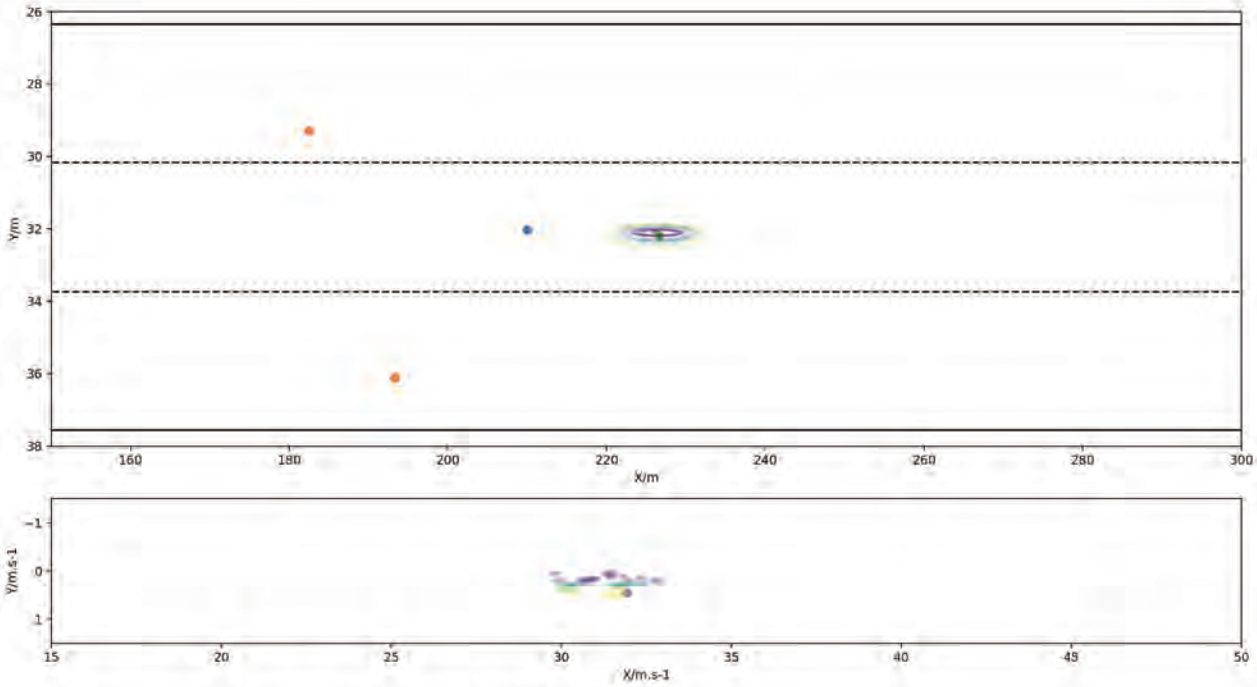
où le terme de reconstruction est calculé depuis la somme de tous les instants de prédiction comme dans l'équation (2.54). Afin de pouvoir comparer le CVRAE aux autres modèles, les métriques, ainsi que les visualisations suivantes, sont établies seulement sur la base de la prédiction à un pas. Le reste de la séquence prédite n'est pas considéré. Sur les données non équilibrées, le modèle obtient une RMSE en position de 0.09 et une RMSE en vitesse de 0.12. Les tests du khi-deux sont validés à 94%. Comme pour les modèles précédents, afin de mesurer les performances en prédiction pour chaque manœuvre identifiée et s'assurer de bonnes performances lors de la prédiction dans les scénarios sélectionnés au chapitre précédent, les métriques sont recalculées sur un jeu de validation provenant des données équilibrées. Dans ce cas, la RMSE en position augmente légèrement à 0.11. L'erreur en vitesse est plus que doublée et passe à 0.26. Le nombre de tests du khi-deux valides n'est plus que de 83%. Les résultats obtenus en terme de prédiction sont illustrés sur la Figure 5.21. L'encodeur agrégé est représenté sur la Figure 5.22.



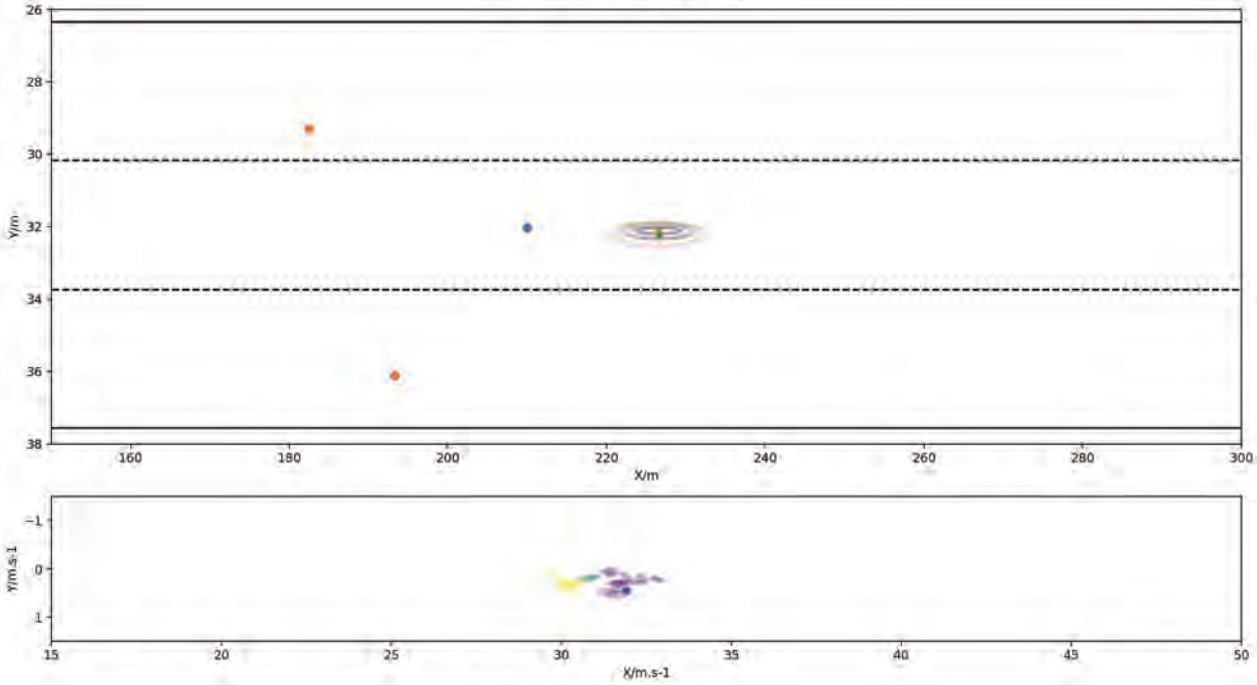
(a) Forward $- z_1$



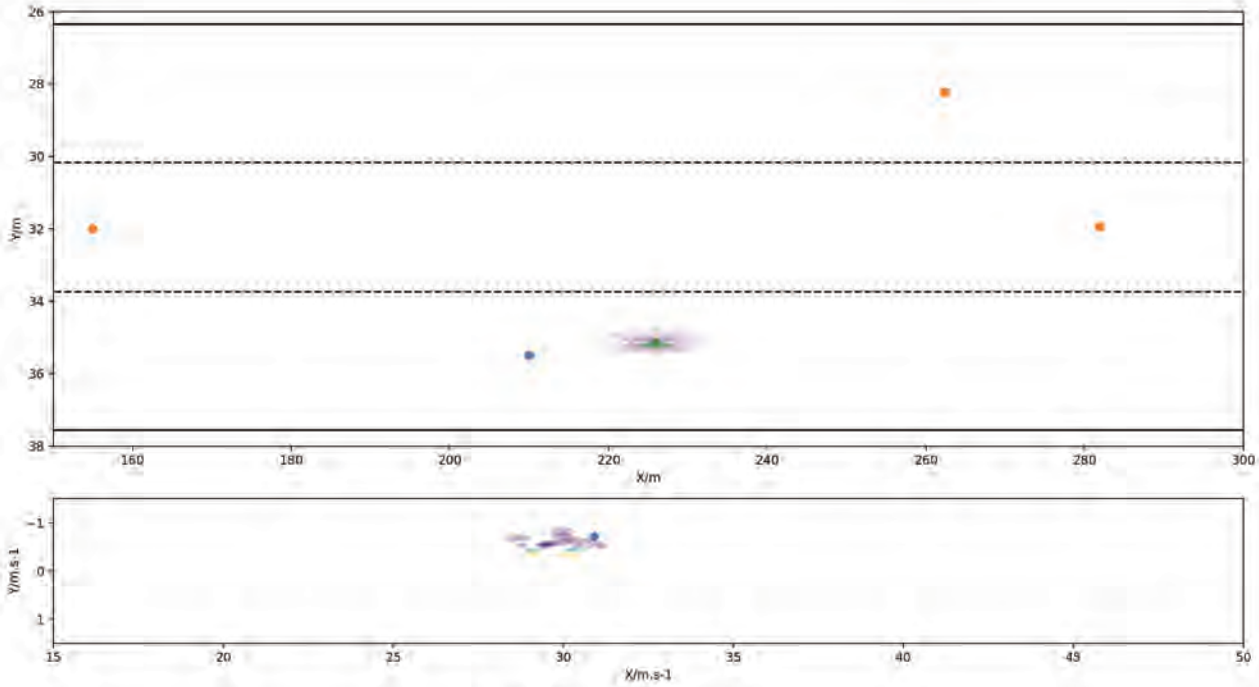
(b) Forward $- z_2$



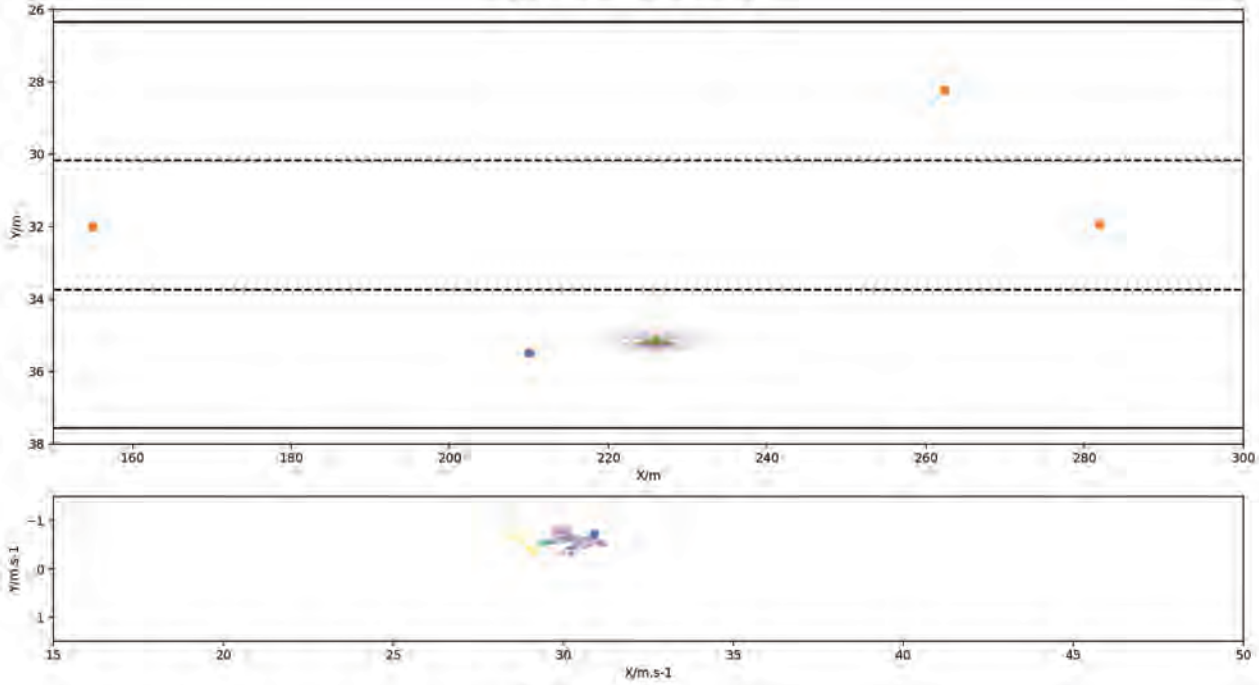
(c) Sheer_in − z_1



(d) Sheer_in − z_2



(e) `Sheer_out` – z_1



(f) `Sheer_out` – z_2

FIGURE 5.21 – Sorties du CVRAE obtenues pour une instance des données HighD mettant en scène l'égo-véhicule effectuant les manœuvres (a-b) `Forward`, (c-d) `Sheer_in` et (e-f) `Sheer_out` respectivement pour chaque graphe. Les distributions de prédiction en position et en vitesse sont obtenues depuis 1000 échantillons de z et sont représentées par les points et les ellipses associées colorées.

FIGURE 5.21 – (Suite) Pour les trois manœuvres considérées, les distributions générées par le décodeur en position et vitesse sont affichées selon les valeurs de z_1 et de z_2. En comparant la vérité terrain aux distributions produites avec des valeurs différentes de z_1, il est possible de différencier chacune des manœuvres. De plus, pour cette composante, les vitesses et les positions futures sont cohérentes avec la manœuvre entreprise par le véhicule. Pour z_2, le résultat en termes de séparation est plus mitigé. La répartition via les valeurs de des vitesses générées semble encoder une certaine modulation de la vitesse longitudinale de l'égo-véhicule. Néanmoins, les ellipses en position font correspondre de manière indifférenciée des points similaires à des codes latents éloignés selon z_2.

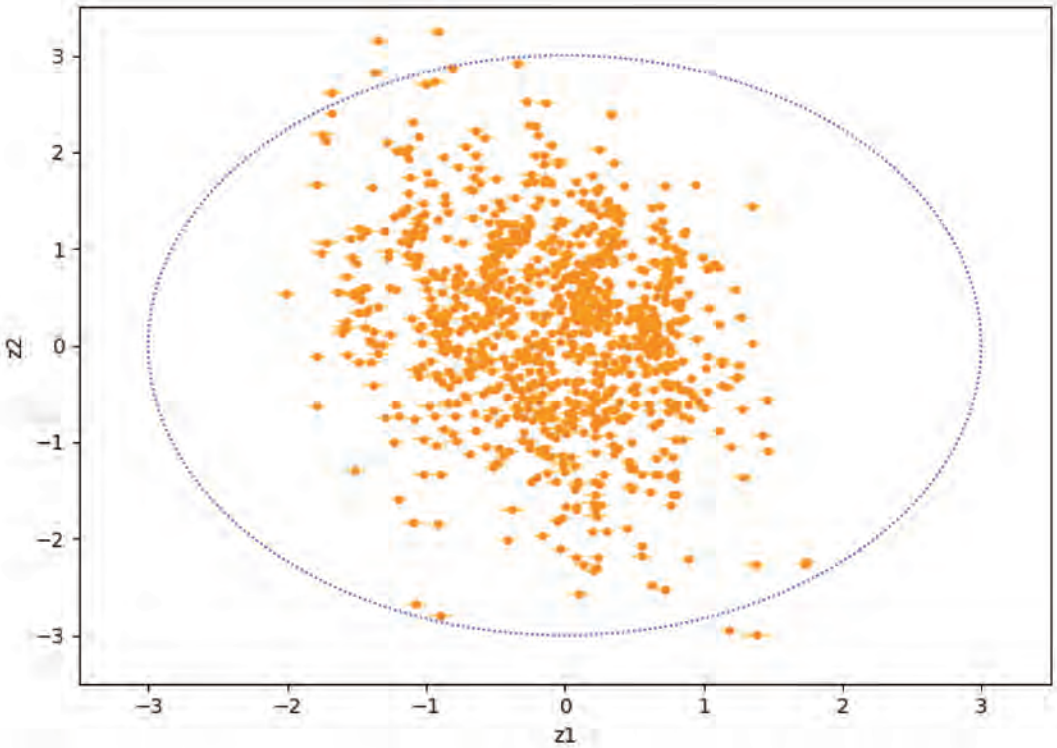
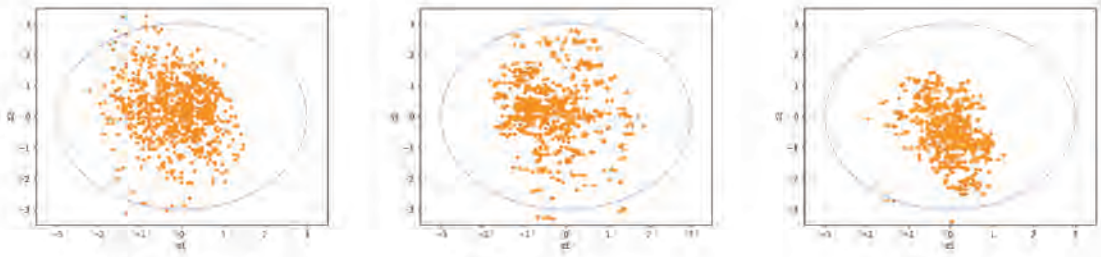


FIGURE 5.22 – Approximation de l'encodeur agrégé d'un CVRAE $q_\phi^{\text{agg}}(z)$ obtenu depuis 1000 échantillons de couples entrée-sortie des données HighD. La couverture des codes latents par rapport à $p(z)$ est acceptable. Les ellipses à 99% de probabilité associées aux moyennes sont assez resserrées. Cependant, le recouvrement entre les codes latents semble satisfaisant avec l'absence de pôles concentrant une majeure partie des échantillons sur une zone latente réduite.



(a) Forward (b) Sheer_in (c) Sheer_out

FIGURE 5.23 – (a-b-c) Approximations de l'encodeur agrégé d'un CVRAE $q_\phi^{\text{agg}}(z)$ obtenus avec 1000 échantillons de couples entrée-sortie des données HighD correspondant respectivement aux manœuvres Forward, Sheer_in et Sheer_out. Une répartition des codes latents différente pour chaque manœuvre respective est observée permettant une interprétation du processus de génération du réseau. Cependant, les codes latents ne sont pas entièrement démêlés en termes de manœuvres : plusieurs manœuvres se partagent des parties communes de l'espace latent. Les manœuvres Sheer_in et Sheer_out peuvent être séparées depuis les valeurs de la composante z_1. Il est plus difficile d'isoler la classe Forward. Par rapport aux deux autres manœuvres, les codes latents semblent être plus concentrés au centre dans la partie nord du graphe de $q_\phi^{\text{agg}}(z)$.

Dans un premier temps, compte-tenu des résultats et de la forme de l'encodeur agrégé obtenus, l'ajout des trajectoires futures de l'égo-véhicule semble être bénéfique pour l'apprentissage de représentation effectué sur les données HighD. Plusieurs remarques peuvent être faites concernant la génération des trajectoires par rapport aux variables latentes et la représentation apprise pour chaque manœuvre considérée. La Figure 5.21 illustre le résultat en prédiction du modèle par rapport aux trois manœuvres identifiées précédemment. A travers la variable latente z_1, pour ces instances de données, la représentation apprise du CVRAE semble permettre une certaine classification des trajectoires observées. En ne regardant que la vitesse, pour la manœuvre Forward, la majorité des prédictions qui englobent la vérité terrain se trouvent autour d'un intervalle de valeur pour z_1 représenté par des couleurs proches du bleu. La même observation peut être faite pour les manœuvres Sheer_in et Sheer_out avec leur intervalle et couleur associée (autour du jaune ou du violet respectivement). En position ce constat est un peu plus flou : les ellipses associées grossièrement aux trois manœuvres recouvrent pour beaucoup des zones en commun. Il est difficile d'isoler en position des zones de prédiction avec des manœuvres distinctes. Cependant, on observe que pour les instances faiblement classifiées en Sheer_in ou Sheer_out les ellipses en position tendent à couvrir des zones cohérentes avec le mouvement futur de l'égo-véhicule. La variable latente z_2 semble avoir encodé de l'information permettant de générer une prédiction pouvant être modulée dans sa vitesse longitudinale. En position, z_2 ne semble pas encoder d'information pertinente. En ce qui concerne la forme de l'encodeur agrégé $q_\phi^{\text{agg}}(z)$, celle-ci ne suit pas tout a fait celle du prior

attendu pour la variable latente z_1. En effet, ici, ils se distribue principalement dans l'intervalle $[-1, 1]$. La distribution de z_2 est assez proche de celle souhaitée. Aussi, en ne considérant pas les valeurs aberrantes, les deux composantes semblent indépendantes et aucune corrélation empirique n'est observé. La Figure 5.23 représente une estimation de l'encodeur agrégé pour chacune des trois manœuvres présentées précédemment. Ils sont tous les trois obtenus avec des échantillons de couples d'entrée-sortie (x, y) correspondant à la manœuvre souhaitée. Depuis ces visualisations, il est possible d'observer directement les zones qui définissent grossièrement les différentes manœuvres auxquelles les données de sorties y peuvent appartenir. Pour les manœuvres `Sheer_in` et `Sheer_out`, les zones associées observées se situent à l'ouest et au sud-est respectivement de la zone formée par z. Pour `Forward` cette zone est assez étendue et semble se concentrer au centre du graphe dans sa partie haute. La représentation apprise permet de différencier les manœuvres à travers les des zones grossières de l'espace latent. Celle-ci n'est pas entièrement démêlée et fait correspondre une grande partie de l'espace latent associée à la manœuvre `Forward` aux deux autres manœuvres. Ce phénomène peut être expliqué par la continuité de par la méthode de segmentation employée.

Par rapport au CAB-CVAE, l'incertitude associée aux prédictions en position et en vitesse est plus importante et cela gène à l'interprétation de la représentation apprise des données. Il est cependant important de rappeler que ce réseau apprend depuis une fonction objectif standard sur des données non supervisées et sans besoin de termes ou d'une hyper-paramétrisation supplémentaires visant à contrôler plus finement l'apprentissage de représentation. L'ajout de dépendances temporelles dans les données de sortie facilite donc la séparation des trajectoires dans l'espace latent. Sans dépendance temporelle passée au delà de l'instant courant, ce CVRAE modifié peut être utilisé dans le cadre de la méthode de prédiction présentée au chapitre 4 pour représenter la dynamique a priori des véhicules dans la scène.

5.4 Application aux algorithmes de filtrage stochastique

Dans le chapitre précédent, deux algorithmes de filtrage stochastique sont présentés. Avec l'apprentissage de la dynamique de l'égo-véhicule, ils forment notre cadre d'estimation et de prédiction de véhicules en interaction. L'objectif principal de ce chapitre est l'identification de technique d'apprentissage non supervisé pouvant produire un modèle performant et interprétable pour représenter cette dynamique. Compte-tenu des résultats obtenus pour les variations de CVAEs étudiées, deux modèles se démarquent : le Cyclic Annealing Beta-CVAE et le CVRAE. Tous les deux entraînés avec la position et la vitesse de chaque véhicule de la scène, ces deux modèles neuronaux sont évalués sur les scénarios routiers précédemment sélectionnés.

En ce qui concerne leur intégration aux algorithmes de filtrage, l'utilisation d'une nouvelle

dynamique a priori portée par un CVAE (ou une de ses variations) dans le cadre du MPF (voir section 4.5.2) est directe. Le modèle remplace simplement les trois réseaux utilisés pour produire les paramètres de la GMM. Lors de l'étape de propagation, les particules représentant la distribution de l'état prédit sont obtenues en deux étapes : pour chaque particule un échantillon de $p(z)$ est obtenu ; cet échantillon est ensuite utilisé dans le décodeur pour obtenir les paramètres d'une Gaussienne qui sera également à son tour échantillonnée pour obtenir une prédiction sur la particule du sous-vecteur d'état lié à l'égo-véhicule.

L'application au VBMPF (voir section 4.5.3) de la dynamique nouvellement apprise n'est pas possible en l'état. Dans le cas du CVAE, le modèle de dynamique est un mélange infini de lois Gaussiennes dont les moments sont délivrés par un réseau de neurones, les poids de ce mélange étant eux-mêmes distribués selon une loi Gaussienne, il n'est plus possible d'appliquer l'équation (4.20) conduisant à une approximation variationnelle de la dynamique a priori. Par la suite, les résultats présentés sont obtenus depuis l'application de l'algorithme MPF associé au CAB-CVAE ou au CVRAE.

5.4.1 Résultats sur scénarios autoroutiers

Les deux modèles sélectionnés sont évalués sur les scénarios présentés dans la section 4.7.2.1. La Figure 5.24 reprend les performances en estimation des deux modèles. Ces résultats sont obtenus en moyenne sur 10 réalisations de chaque scénario. En estimation, le MPF couplé au CAB-CVAE produit des estimés acceptables pouvant être utilisés par la suite pour la prédiction. Cependant, le filtre voit son nombre de particules efficaces s'effondrer pour un des scénarios extrait des données HighD. Pour le CVRAE, les résultats en termes de position sont correctes et le filtre ne diverge par pour l'ensemble des scénarios considérés. Les estimés en vitesse sont assez éloignés des valeurs de la vérité terrain. Comme constaté au chapitre précédent, une bonne estimation de la vitesse est fondamentale pour produire une prédiction long terme cohérente des trajectoires des véhicules. La Figure 5.25 présente les résultats moyens en prédiction du MPF avec le CAB-CVAE et le CVRAE. Le CAB-CVAE produit des trajectoires futures raisonnablement proches en termes de RMSEs de la vérité terrain. Cependant, les performances de cet apprentissage sont en dessous de la plupart des méthodes de prédiction présentées au chapitre 4 et notamment de celles basées sur la représentation d'une dynamique a priori depuis une GMM dont les paramètres sont appris de manière supervisée. Bien que les prédictions se retrouvent souvent sur la bonne voie, les distributions prédites sont souvent trop optimistes. Ce phénomène est d'autant plus fréquent que le bruit de mesure augmente. En prédiction les résultats du CVRAE sont décevants : les ellipses de prédiction sont mal placées et bien que déjà très étendues celles-ci sont trop optimistes quant à la position et à la vitesse de la vérité terrain. Les mauvaises performances en estimation et en prédiction de la vitesse peuvent être

expliquées par un biais affectant le réseau. Celui-ci produit pour la plupart des données d'entrée des vitesses qui tendent à se rapprocher de celle d'un véhicule effectuant une manœuvre `Forward` à une vitesse longitudinale de « croisière » (autour de $30m.s^1$). Une hypothèse peut être faite sur le déséquilibre des données impactant négativement la variété des prédictions du réseau.

En reprenant la vue aérienne adoptée dans le chapitre précédent, les Figures 5.26 et 5.27 illustrent sur deux scénarios différents l'utilisation du MPF avec respectivement le CAB-CVAE et le CVRAE pour représenter la dynamique a priori. Bien que ces modèles offrent une solution d'apprentissage non supervisé avec des possibilités d'interprétation du processus de génération des trajectoires, en prenant en compte les performances en estimation et en prédiction, il est difficile de les considérer comme appropriés pour l'évaluation du risque de situations routières. Les performances moyennes en termes de RMSEs et de test du khi-deux obtenus lors de l'apprentissage individuel des deux réseaux impactent directement la qualité des estimations et des prédictions une fois couplé à un algorithme de filtrage stochastique. Afin d'être utilisés plus efficacement, une paramétrisation différente de l'apprentissage de ces réseaux de neurones doit permettre d'obtenir des modèles plus performants tout en préservant une structure sur la distribution apprise des variables latentes. L'objectif du CVRAE pourrait être modifié afin de mieux contrôler l'équilibre entre l'objectif de reconstruction et l'apprentissage de représentation via les variables latentes comme pour le Beta-CVAE ou l'Info-CVAE. Des visualisations supplémentaires des résultats en estimation et en prédiction pour différents scénarios sont présentées dans l'Annexe D.

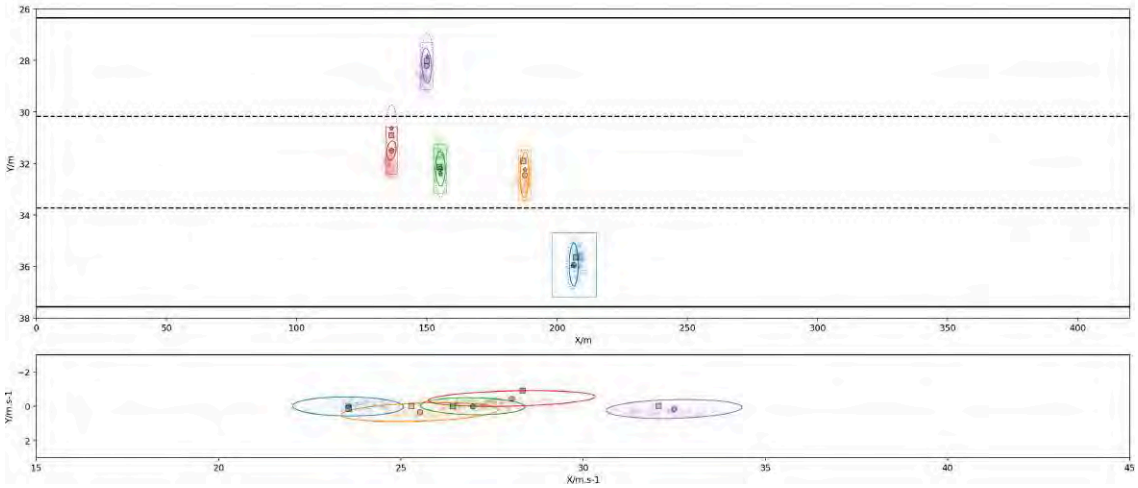
	Durée du filtrage ≈ 5s				
	RMSE x	RMSE y	RMSE v_x	RMSE v_y	# divergence du filtre
MPF CAB-CVAE	0.05/0.29/0.54	0.09/0.14/0.29	0.54/0.57/0.71	0.18/0.20/0.32	1/1/1
MPF CVRAE	0.08/0.32/0.65	0.07/0.12/0.22	1.81/1.85/1.99	0.23/0.28/0.40	0/0/0

FIGURE 5.24 – Performances moyennes en estimation du MPF avec une dynamique a priori apprise depuis soit depuis un CAB-CVAE ou un CVRAE. Les résultats sont calculés sur tous les scénarios avec trois niveaux de bruit de mesure différents : aucun bruit de mesure / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule).

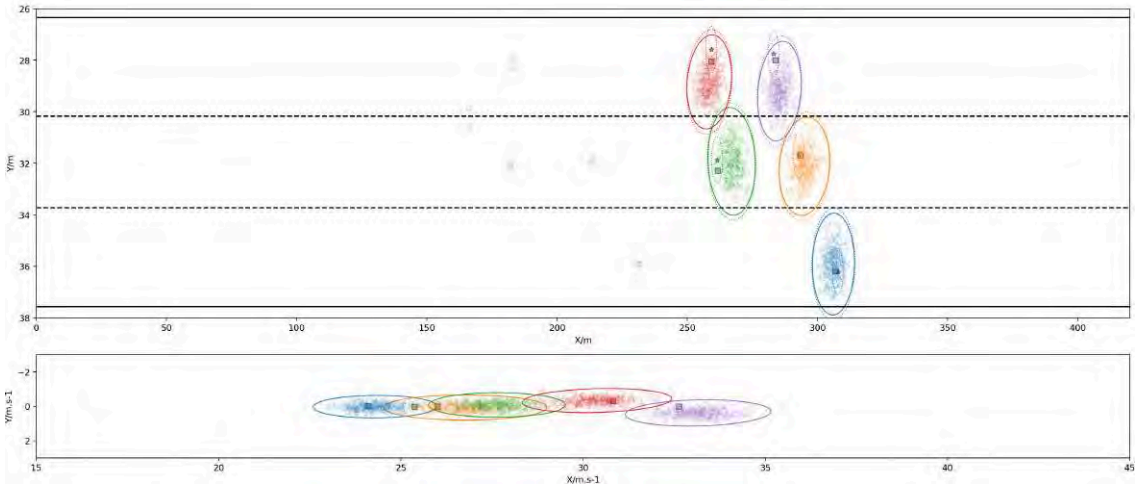
	Prédiction pour un horizon de 3s	
	% xy χ^2 test	% correct lane
MPF CAB-CVAE	86.18/77.85/60.07%	91.67/91.67/93.06%
MPF CVRAE	25.97/24.31/22.71%	87.57/86.18/86.18%

	MPF CAB-CVAE		MPF CVRAE	
Horizon de prédiction (s)	Long. RMSE	Lat. RMSE	Long. RMSE	Lat. RMSE
1	1.03/1.15/1.33	0.40/0.37/0.58	2.93/3.01/3.46	0.42/0.52/0.61
2	2.22/2.18/2.19	0.62/0.60/0.83	6.51/6.50/7.02	0.75/0.78/0.94
3	3.64/3.70/3.60	0.77/0.76/0.93	1.80/2.11/2.66	10.62/10.63/10.96

FIGURE 5.25 – Mesures des performances moyennes en prédiction des algorithmes comparés. Les résultats sont calculés sur tous les scénarios avec trois niveaux de bruit de mesure différents : aucun bruit de mesure / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{4}\times$ (la taille standard d'un véhicule) / intervalle de confiance à 99% du bruit de mesure égal à $\pm\frac{1}{2}\times$ (la taille standard d'un véhicule).

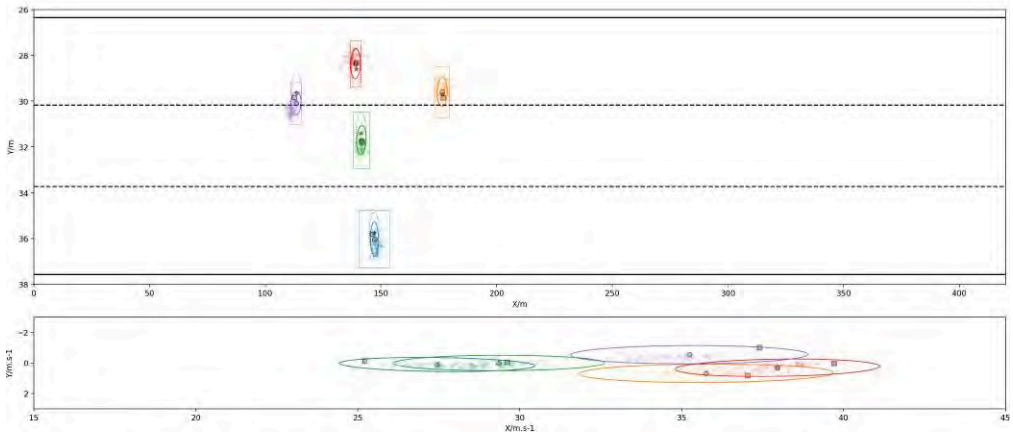


(a)

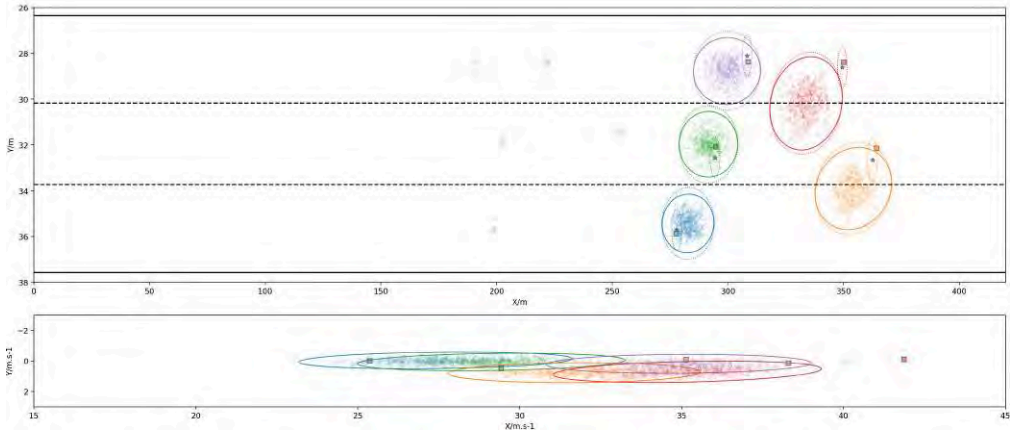


(b)

FIGURE 5.26 – Vue schématique aérienne de l'application du MPF couplé au CAB-CVAE sur un scénario autoroutier pour lequel l'intervalle de confiance à 99% de probabilité du bruit de mesure est de $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). (a) Après plusieurs instants l'état de chaque véhicule est raisonnablement bien estimé. (b) Pour cette configuration, Les ellipses de prédiction à 3s englobent toutes la vérité terrain de tous les véhicules.



(a)



(b)

FIGURE 5.27 – Vue schématique aérienne de l'application du MPF couplé au CVRAE sur un scénario autoroutier pour lequel l'intervalle de confiance à 99% de probabilité du bruit de mesure est de $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). (a) Après plusieurs instants l'état de chaque véhicule est correctement estimé. (b) Les distributions de prédiction à 3s sont peu informatives et pour deux des cinq véhicules celle-ci sont trop optimistes par rapport à la véritable position des véhicules.

CHAPITRE 6
CONCLUSION

Cette conclusion commence par un bref résumé des chapitres précédents. Les différentes conclusions tirées de cette thèse sont ensuite présentées. Enfin, les perspectives de travaux futurs viennent clore cette étude.

6.1 Résumé

Cette thèse traite de la prédiction du mouvement avec prise en compte des interactions de véhicules dans le but d'établir une évaluation du risque liée à leur évolution conjointe sur un segment autoroutier.

Le Chapitre 1 présente le contexte de cette étude en introduisant le domaine des Systèmes de Transport Intelligents et les différentes fonctionnalités nécessaires d'un véhicule pour le qualifier comme appartenant à ce domaine. Un accent est mis sur les systèmes de télécommunication, notre méthode de prédiction supposant une interface de communication commune et centrale de type « Cloud » permettant de recueillir et d'échanger les données en temps réel des véhicules. L'estimation, la prédiction et l'évaluation du risque de scènes autoroutières complexes constituent les trois objectifs principaux de cette étude. Les approches modulaires, interprétables et explicables pour répondre à la problématique sont préférées afin de pouvoir garantir du bon fonctionnement d'un tel système directement responsable de la sécurité des usagers de la route.

Le Chapitre 2 présente les concepts et outils théoriques utilisés dans les chapitres suivants. Des notions de modélisation probabiliste sont introduites, elles nous servent de cadre pour gérer les multiples sources d'incertitude qui composent la problématique traitée. Les techniques d'estimation stochastique récursive sont ensuite abordées. Des exemples de filtres stochastiques sont détaillés comme des solutions à l'estimation Bayésienne récursive. Certaines d'entre elles font l'usage de techniques d'approximation permettant l'application des filtres dans des cas d'usage réels complexes. L'estimation et la prédiction nécessitent un modèle de transition permettant de relier un état courant à son état futur. La complexité induite par les scènes routières traitées suggère d'apprendre le modèle de transition avec des techniques d'apprentissage automatique. Une introduction au domaine conclut ce chapitre.

Le Chapitre 3 propose un état de l'art en trois parties. La première est une analyse détaillée des approches de prédiction de mouvement de la littérature. Celles prenant en compte les interactions sont les seules capables de répondre à notre problématique de prédiction à long terme. Une partie

considérable de la littérature se concentre sur des méthodes reposant totalement ou en partie sur des techniques d'apprentissage pour réaliser la prédiction en s'abstenant de décrire manuellement l'évolution des véhicules en interaction. Une deuxième partie présente brièvement les définitions et méthodes d'évaluation du risque présentes dans la littérature ITS. De nombreuses méthodes tirent parti de la quantité massive de données de trajectoires véhiculaires disponible publiquement. La dernière partie de ce chapitre présente donc les principaux jeux de données de trajectoires existants. Ceux-ci sont également utilisés pour évaluer et comparer les méthodes de prédiction du mouvement.

Le Chapitre 4 détaille la première contribution de ce manuscrit, un cadre pour l'estimation et la prédiction d'une flotte de véhicules basée sur l'estimation stochastique couplé à de l'apprentissage automatique supervisé. Le *Multiple Particle Filter* et le *Variational Bayesian Multiple Particle Filter* sont deux filtres stochastiques approximatifs instanciés pour réaliser à partir d'observations bruitées, l'estimation en-ligne du vecteur d'état supposé Markovien de la flotte de véhicules. L'estimation et la prédiction reposent sur l'utilisation d'un modèle de transition égocentrique multimodale prenant en compte les interactions entre véhicules. Ce modèle appris est représenté par une banque de réseaux de neurones à propagation avant, entraînés hors-ligne sur des données de trajectoires préalablement segmentées en manœuvres depuis une connaissance experte. Le schéma modulaire proposé est motivé par l'apport de garanties de bon fonctionnement, d'arguments d'explicabilité et d'interprétabilité sur le système ITS développé. Celui-ci est évalué sur des scénarios autoroutiers difficiles.

Le Chapitre 5 présente la deuxième contribution de cette étude, l'apprentissage non-supervisé en manœuvres de la loi de dynamique égocentrique. L'Auto-Encodeur Variationnel Conditionnel est choisi comme modèle génératif profond interprétable pour représenter cette loi. Il est appris sur la base de trajectoires non-segmentées en termes de manœuvres. Les variables latentes du CVAE permettent d'exhiber une structure sur les données et d'extraire ainsi l'information sur les manœuvres automatiquement. Le phénomène d'« effondrement du postérieur » des variables latentes est observé dans le cadre de l'apprentissage du modèle de transition. Ce dernier met en péril les capacités d'interprétation et d'extraction de structure sur les données de trajectoire. Tout au long du chapitre, des techniques pour mitiger ce phénomène sont présentées. Enfin, après validation sur des données de dynamique synthétique et réelle, plusieurs modèles de transition a priori égocentriques représentés par des variations de CVAEs sont intégrés à notre schéma d'estimation et de prédiction pour être ensuite évalués sur scénarios autoroutiers.

6.2 Conclusions

Plusieurs conclusions peuvent être tirées de cette thèse :

— Mêler estimation stochastique et apprentissage automatique permet de proposer une approche hybride modulaire considérant les multiples sources d'incertitudes du problème tout en offrant des résultats intermédiaires permettant de s'assurer de son bon fonctionnement. Ces résultats servent également à produire des explications et des interprétations du processus ayant généré les prédictions et l'évaluation du risque. Réaliser l'estimation d'un vecteur d'état supposé Markovien préalablement à la prédiction, permet de s'adapter en ligne aux différents bruits de mesure associés aux observations.

— Apprendre le modèle de transition via des méthodes d'apprentissage automatique supervisées est une façon de modéliser le déplacement complexe d'une flotte de véhicules en interaction. Ces techniques nous permettent d'éviter de spécifier manuellement les règles qui régissent le déplacement des véhicules sur le segment routier. Le jeu de données HighD est utilisé avec succès comme substitut à des séquences de vecteurs d'état caché des véhicules pour l'apprentissage du modèle de transition. Pour garantir le fonctionnement nominal du processus d'estimation et proposer des prédictions suffisamment riches pour l'évaluation du risque, il est nécessaire de fournir un modèle de transition multimodale prenant en compte la variété du mouvement inter- et intra- manœuvres des véhicules. La segmentation en manœuvres des futurs à un pas des véhicules est donc primordiale car les étiquettes associées renseignent sur la trajectoire long terme et permettent ainsi au modèle de transition, d'être plus varié dans ses prédictions.

— Le VBMPF et le MPF sont deux filtres stochastiques approximatifs adaptés aux besoins de filtrage en ligne du vecteur d'état de haute dimension des véhicules. Les techniques reposant sur le filtrage particulaire classique ne permettent pas de produire des estimés convenables en temps quasi réel. Bien que plus efficace en temps de calcul, le VBPF ne permet pas l'utilisation de modèles de transition de forme quelconque comme le permet le MPF. Les deux filtres produisent des estimés suffisamment informatifs pour permettre par la suite une prédiction moyen terme performante du futur de la scène routière.

— L'apprentissage non-supervisé en termes de manœuvres du modèle de transition est possible via l'utilisation de modèles génératifs profonds appelés CVAEs. L'utilisation de ces architectures neuronales permet de se passer du besoin de segmentation des trajectoires en proposant des méthodes d'extraction automatique de structure dans les données traitées améliorant ainsi la généricité de l'approche proposée.

— L'extraction automatique d'une structure sur les futurs pas à pas est entravée par un phéno-

mène connu sous le nom d'« effondrement du postérieur ». Dans cette étude, nous établissons le besoin de contrôler plus finement l'objectif du CVAE *Vanilla* pour notre besoin d'apprentissage de représentation. Le CAB-CVAE et le CVRAE sont deux variations du CVAE original permettant de mitiger ce phénomène et ainsi d'extraire une structure des futurs à un pas de l'égo-véhicule pouvant être interprétée en termes de manœuvres.

— L'hypothèse adoptée d'un modèle Markovien où le vecteur d'état riche capture suffisamment d'information sur la flotte de véhicule pour permettre un apprentissage efficace de la dynamique de l'égo-véhicule est partiellement vérifiée. Pour l'apprentissage supervisé, les résultats en termes de prédiction sont honorables mais mitigés par rapport à ceux de littérature. En non supervisé, les performances médiocres en estimation et en prédiction lors de l'utilisation de ces modèles de transition témoignent d'une certaine difficulté à entraîner ce type de réseau génératif pour notre problème. En dépit d'un vecteur d'état de grande dimension, les manoeuvres sont difficiles à discerner dans le mouvement de l'égo-véhicule pour des pas de temps faibles. Cependant, conserver un tel système Markovien « court-terme » couplé à une information long terme fournie de manière supervisée ou découverte de manière non supervisée peut permettre la mise en place d'un schéma explicable de prédiction.

— La manière générale d'évaluer les algorithmes de la littérature peut aussi être reconsidérée. Le déséquilibre des manoeuvres présentes dans les jeux de données autoroutiers sont rarement pris en compte pour l'évaluation des méthodes de prédiction. Dans cette thèse, nous nous sommes focalisés sur des scénarios difficiles. De plus, les algorithmes présentés peuvent être implémentés efficacement pour un fonctionnement en ligne mais, par essence, ils sont beaucoup moins adaptés au traitement d'un jeu de données de trajectoires complet les rendant plus difficiles à évaluer à l'aide des méthodes standards.

6.3 Perspectives

Pour finir cette thèse, les perspectives de travaux futurs sont présentés :

Améliorations de l'apprentissage non supervisé Dans cette étude, la plupart des verrous de l'apprentissage non supervisé à l'aide de CVAEs sont identifiés. Par manque de temps, la modification de l'objectif de l'instance de CVRAE utilisé ainsi que l'utilisation de cartes sémantiques plus complètes n'ont pas été effectuées et constituent une prospective de finalisation du sujet.

Améliorations des stratégies de filtrage Plusieurs modifications aux schémas de filtrage peuvent être apportées : le changement des lois d'importance ; la suppression des multiples étapes de rééchantillonnage ; la gestion de GMM finies (ou infinies) par le VBMPF ; l'utilisation de méthodes de quasi Monte-Carlo [164, 165].

Environnements routiers additionnels Les méthodes d'estimation et de prédiction de cette thèse pourraient être adaptées afin d'être déployées dans des environnements routiers différents comme les intersections et les ronds-points. L'apprentissage de la loi de dynamique devra être faite en conséquence avec un jeu de trajectoires correspondant à l'environnement choisi [107, 108]. L'extraction automatique d'une structure dans les données serait particulièrement intéressante pour ce type d'environnements routiers où le nombre de manœuvres possibles est grand et où chaque trajectoire est difficile à segmenter manuellement. Les arrêts et les redémarrages des véhicules dans la scène devront également être pris en compte. Enfin, faute de jeu de données adéquats, il n'a pas été possible de considérer des contextes favorables avec un pas de temps suffisamment grand et de longues trajectoires. Les résultats obtenus dans ce manuscrit sont donc à compléter pour conclure définitivement.

Gestion des obstacles La méthode proposée ici fait l'hypothèse d'une carte routière statique connue où les seuls objets dynamiques sont les véhicules, tous connectés entre eux. La méthode d'évaluation du risque proposée ainsi que des capteurs extéroceptifs embarqués sur les véhicules pourraient servir à détecter la présence ou non d'obstacles statiques sur la voie.

Application au pistage et gestion des usagers de la route non connectés Dans son implémentation actuelle, la phase d'estimation ne peut gérer les usagers de la route non-connectés à l'interface « Cloud ». Dans la même veine que pour la gestion des obstacles statiques, notre méthode d'estimation pourrait être adaptée pour le pistage d'objets dynamiques. Ces objets pouvant être soit des obstacles dynamiques sur la voie, soit des usagers de la route non-connectés. Le cadre du pistage-par-détection (*tracking-by-detection*) [166] multi-objets pourrait traiter cette problématique additionnelle. Afin de créer un tel pisteur, deux étapes préliminaires supplémentaires devront être ajoutées avant l'estimation : la détection qui construit les observations des obstacles et des usagers de la route sur la base des données capteurs brutes ; l'association qui relie les observations aux états estimés correspondant. L'ajout de ces deux dernières étapes nécessiterait la prise en compte d'incertitudes supplémentaires liées aux problèmes de fausse détection et de mauvaise association des observations. Il serait également nécessaire de fournir des modèles de transition adaptés aux différents objets pistés. Compte tenu des modèles de transition et des algorithmes de filtrage utilisés prenant en compte les interactions, il semble être possible d'imaginer un pistage centralisé recevant des observations à des fréquences plus faibles que celles communément utilisées dans le domaine.

Modèles de transition multiple Dans cette étude, une hypothèse forte sur le comportement des conducteurs est faite. Ceux-ci sont supposés rationnels et collaboratifs de par le jeu de données utilisé pour apprendre les modèles de transition. Exploiter plusieurs modèles de transition associés à des comportements d'usagers de la route différents *e.g.*, normal, conducteur débutant, conducteur agressif... etc., pourrait améliorer les performances des étapes d'estimation et de prédiction en les

rendant plus robustes aux différents types de conducteurs observés. Les capacités d'évaluation du risque du système d'aide à la conduite pourraient être étendues en considérant les comportements antagonistes. Une étape d'estimation du type du comportement de l'usager devra être réalisée avant l'application du modèle de transition lui correspondant. Différents jeux de données réelles [85] ou synthétiques [167] pourraient servir à construire de tels modèles de transition. Enfin, séparer et spécialiser les modèles de transition par comportement promouvrait une interprétabilité accrue du système d'assistance.

REFERENCES

[1] "World health statistics 2018 : monitoring health for the SDGs, sustainable development goals," World Health Organization, 2018.

[2] M. GUÉRIAU, "Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic," thèse de doct., Université de Lyon I Claude Bernard, 2016.

[3] S. LEFÈVRE, "Estimation du risque aux intersections pour applications sécuritaires avec véhicules communicants," thèse de doct., Grenoble, 2012.

[4] N.-E. EL FAOUZI, H. LEUNG et A. KURIAN, "Data fusion in intelligent transportation systems : Progress and challenges–A survey," *Information Fusion*, t. 12, n° 1, p. 4-10, 2011.

[5] N.-E. EL FAOUZI et L. A. KLEIN, "Data fusion for ITS : techniques and research needs," *Transportation Research Procedia*, t. 15, p. 495-512, 2016.

[6] P. PAPADIMITRATOS, A. DE LA FORTELLE, K. EVENSSEN, R. BRIGNOLO et S. COSENZA, "Vehicular communication systems : Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communication Magazine*, t. 47, n° 11, p. 84-95, 2009.

[7] I. BARABÁS, A. TODORUŢ, N CORDOŞ et A. MOLEA, "Current challenges in autonomous driving," in *IOP conference series : materials science and engineering*, IOP Publishing, t. 252, 2017, p. 012 096.

[8] K. GOLESTAN, R. SOUA, F. KARRAY et M. S. KAMEL, "Situation awareness within the context of connected cars : A comprehensive review and recent trends," *Information Fusion*, t. 29, p. 68-83, 2016.

[9] D. GRUYER, V. MAGNIER, K. HAMDI, L. CLAUSSMANN, O. ORFILA et A. RAKOTONIRAINY, "Perception, information processing and modeling : Critical stages for autonomous driving applications," *Annual Reviews in Control*, t. 44, p. 323-341, 2017.

[10] D. L. HALL et J. LLINAS, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, t. 85, n° 1, p. 6-23, 1997.

[11] S. K. DAS, *High-level data fusion*. Artech House, 2008.

[12] D. V. CARVALHO, E. M. PEREIRA et J. S. CARDOSO, “Machine learning interpretability : A survey on methods and metrics,” *Electronics*, t. 8, n° 8, p. 832, 2019.

[13] I. GOODFELLOW, Y. BENGIO et A. COURVILLE, *Deep learning*. MIT press, 2016.

[14] C. M. BISHOP et N. M. NASRABADI, *Pattern recognition and machine learning*, 4. Springer, 2006, t. 4.

[15] T. HASTIE, R. TIBSHIRANI, J. H. FRIEDMAN et J. H. FRIEDMAN, *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2009, t. 2.

[16] A. PAPOULIS, *Probability and statistics*. Prentice-Hall, Inc., 1990.

[17] P. DANÈS, *Notes de cours : estimation stochastique*, 2018.

[18] R. E. KALMAN, “A new approach to linear filtering and prediction problems,” 1960.

[19] H. SORENSON, *Kalman Filtering : Theory and Application*, sér. IEEE Press selected reprint series. IEEE Press, 1985, ISBN : 9780879421915.

[20] S. J. JULIER et J. K. UHLMANN, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, t. 92, n° 3, p. 401-422, 2004.

[21] S. JULIER et J. K. UHLMANN, “A general method for approximating nonlinear transformations of probability distributions,” 1996.

[22] A. DOUCET, S. GODSILL et C. ANDRIEU, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and computing*, t. 10, n° 3, p. 197-208, 2000.

[23] V. ŠMÍDL et A. QUINN, *The variational Bayes method in signal processing*. Springer Science & Business Media, 2006.

[24] K. HORNIK, M. STINCHCOMBE et H. WHITE, “Multilayer feedforward networks are universal approximators,” *Neural networks*, t. 2, n° 5, p. 359-366, 1989.

[25] D. E. RUMELHART, G. E. HINTON et R. J. WILLIAMS, “Learning representations by back-propagating errors,” *nature*, t. 323, n° 6088, p. 533-536, 1986.

[26] F. BASTIEN, P. LAMBLIN, R. PASCANU, J. BERGSTRA, I. GOODFELLOW, A. BERGERON, N. BOUCHARD, D. WARDE-FARLEY et Y. BENGIO, “Theano : new features and speed improvements,” *arXiv preprint arXiv :1211.5590*, 2012.

[27] R. J. WILLIAMS et D. ZIPSER, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, t. 1, n° 2, p. 270-280, 1989.

[28] J. CHUNG, C. GULCEHRE, K. CHO et Y. BENGIO, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv :1412.3555*, 2014.

[29] S. HOCHREITER et J. SCHMIDHUBER, “Long short-term memory,” *Neural computation*, t. 9, n° 8, p. 1735-1780, 1997.

[30] A. GRAVES, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv :1211.3711*, 2012.

[31] A. GRAVES, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv :1308.0850*, 2013.

[32] I. SUTSKEVER, O. VINYALS et Q. V. LE, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, t. 27, 2014.

[33] A. NG et M. JORDAN, “On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes,” *Advances in neural information processing systems*, t. 14, 2001.

[34] D. P. KINGMA et M. WELLING, “Auto-encoding variational bayes,” *arXiv preprint arXiv :1312.6114*, 2013.

[35] K. SOHN, H. LEE et X. YAN, “Learning structured output representation using deep conditional generative models,” *Advances in neural information processing systems*, t. 28, 2015.

[36] S. LEFÈVRE, D. VASQUEZ et C. LAUGIER, “A survey on motion prediction and risk assessment for intelligent vehicles,” *ROBOMECH journal*, t. 1, n° 1, p. 1-14, 2014.

[37] A. RUDENKO, L. PALMIERI, M. HERMAN, K. M. KITANI, D. M. GAVRILA et K. O. ARRAS, “Human motion trajectory prediction : A survey,” *The International Journal of Robotics Research*, t. 39, n° 8, p. 895-935, 2020.

[38] S. AMMOUN et F. NASHASHIBI, “Real time trajectory prediction for collision risk estimation between vehicles,” in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, IEEE, 2009, p. 417-422.

[39] R. MILLER et Q. HUANG, “An adaptive peer-to-peer collision warning system,” in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No. 02CH37367)*, IEEE, t. 1, 2002, p. 317-321.

[40] N. KAEMPCHEN, K. WEISS, M. SCHAEFER et K. C. DIETMAYER, “IMM object tracking for high dynamic driving maneuvers,” in *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, 2004, p. 825-830.

[41] A. POLYCHRONOPOULOS, M. TSOGAS, A. J. AMDITIS et L. ANDREONE, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, t. 8, n° 3, p. 549-562, 2007.

[42] M. ALTHOFF et A. MERGEL, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Transactions on Intelligent Transportation Systems*, t. 12, n° 4, p. 1237-1247, 2011.

[43] L. SPINELLO, M. LUBER et K. O. ARRAS, "Tracking people in 3D using a bottom-up top-down detector," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, p. 1304-1310.

[44] E. REHDER et H. KLOEDEN, "Goal-directed pedestrian prediction," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, p. 50-58.

[45] M. BRÄNNSTRÖM, E. COELINGH et J. SJÖBERG, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, t. 11, n° 3, p. 658-669, 2010.

[46] G. S. AOUDE, V. R. DESARAJU, L. H. STEPHENS et J. P. HOW, "Driver behavior classification at intersections and validation on large naturalistic data set," *IEEE Transactions on Intelligent Transportation Systems*, t. 13, n° 2, p. 724-736, 2012.

[47] J. SCHULZ, K. HIRSENKORN, J. LÖCHNER, M. WERLING et D. BURSCHKA, "Estimation of collective maneuvers through cooperative multi-agent planning," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, p. 624-631.

[48] E. MAZOR, A. AVERBUCH, Y. BAR-SHALOM et J. DAYAN, "Interacting multiple model methods in target tracking : a survey," *IEEE Transactions on aerospace and electronic systems*, t. 34, n° 1, p. 103-123, 1998.

[49] R. TOLEDO-MOREO, J. SANTA et M ZAMORA-IZQUIERDO, "A cooperative overtaking assistance system," *Planning, Perception and Navigation for Intelligent Vehicles (PPNIV)*, t. 50, 2009.

[50] D. GREENE, J. LIU, J. REICH, Y. HIROKAWA, A. SHINAGAWA, H. ITO et T. MIKAMI, "An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists," *IEEE Transactions on intelligent transportation systems*, t. 12, n° 4, p. 942-953, 2011.

[51] M. G. ORTIZ, J. FRITSCH, F. KUMMERT et A. GEPPERTH, "Behavior prediction at multiple time-scales in inner-city scenarios," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, p. 1068-1073.

[52] S. KLINGELSCHMITT, M. PLATHO, H.-M. GROSS, V. WILLERT et J. EGGERT, "Combining behavior and situation information for reliably estimating multiple intentions," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, p. 388-393.

[53] B. MORRIS, A. DOSHI et M. TRIVEDI, "Lane change intent prediction for driver assistance : On-road design and evaluation," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, p. 895-901.

[54] H. M. MANDALIA et M. D. D. SALVUCCI, "Using support vector machines for lane-change detection," in *Proceedings of the human factors and ergonomics society annual meeting*, SAGE Publications Sage CA : Los Angeles, CA, t. 49, 2005, p. 1965-1969.

[55] A. ZYNER, S. WORRALL et E. NEBOT, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robotics and Automation Letters*, t. 3, n° 3, p. 1759-1764, 2018.

[56] H. BERNDT, J. EMMERT et K. DIETMAYER, "Continuous driver intention recognition with hidden markov models," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2008, p. 1189-1194.

[57] D. MEYER-DELIUS, C. PLAGEMANN et W. BURGARD, "Probabilistic situation recognition for vehicular traffic scenarios," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, p. 459-464.

[58] A. TAMKE, T. DANG et G. BREUEL, "A flexible method for criticality assessment in driver assistance systems," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, p. 697-702.

[59] C. WILLIAMS et C. RASMUSSEN, "Gaussian processes for regression," *Advances in neural information processing systems*, t. 8, 1995.

[60] Q. TRAN et J. FIRL, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *2014 ieee intelligent vehicles symposium proceedings*, IEEE, 2014, p. 918-923.

[61] S. M. LAVALLE et J. J. KUFFNER, "Rapidly-Exploring Random Trees : Progress and Prospects : Steven M. LaValle, Iowa State University, A James J. Kuffner, Jr., University of Tokyo, Tokyo, Japan," *Algorithmic and Computational Robotics*, p. 303-307, 2001.

[62] G. AOUDE, J. JOSEPH, N. ROY et J. HOW, "Mobile agent trajectory prediction using Bayesian nonparametric reachability trees," in *Infotech@ Aerospace 2011*, 2011, p. 1512.

[63] T. C. T. CHRISTOPHER, "Analysis of dynamic scenes : Application to driving assistance," thèse de doct., Institut National Polytechnique de Grenoble-INPG, 2009.

[64] M. TREIBER, A. HENNECKE et D. HELBING, “Congested traffic states in empirical observations and microscopic simulations,” *Physical review E*, t. 62, n° 2, p. 1805, 2000.

[65] D. HELBING, I. J. FARKAS, P. MOLNAR et T. VICSEK, “Simulation of pedestrian crowds in normal and evacuation situations,” *Pedestrian and evacuation dynamics*, t. 21, n° 2, p. 21-58, 2002.

[66] J. SORSTEDT, L. SVENSSON, F. SANDBLOM et L. HAMMARSTRAND, “A new vehicle motion model for improved predictions and situation assessment,” *IEEE Transactions on Intelligent Transportation Systems*, t. 12, n° 4, p. 1209-1219, 2011.

[67] A. LAWITZKY, D. ALTHOFF, C. F. PASSENBERG, G. TANZMEISTER, D. WOLLHERR et M. BUSS, “Interactive scene prediction for automotive applications,” in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, p. 1028-1033.

[68] W. SCHWARTING et P. PASCHEKA, “Recursive conflict resolution for cooperative motion planning in dynamic highway traffic,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2014, p. 1039-1044.

[69] M. SURAJ, H. GRIMMETT, L. PLATINSKỲ et P. ONDRÚŜKA, “Predicting trajectories of vehicles using large-scale motion priors,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, p. 1639-1644.

[70] E. KÄFER, C. HERMES, C. WÖHLER, H. RITTER et F. KUMMERT, “Recognition of situation classes at road intersections,” in *2010 ieee international conference on robotics and automation*, IEEE, 2010, p. 3960-3965.

[71] A. ALAHI, K. GOEL, V. RAMANATHAN, A. ROBICQUET, L. FEI-FEI et S. SAVARESE, “Social LSTM : Human trajectory prediction in crowded spaces,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 961-971.

[72] N. DEO et M. M. TRIVEDI, “Convolutional social pooling for vehicle trajectory prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, p. 1468-1476.

[73] K. MESSAOUD, N. DEO, M. M. TRIVEDI et F. NASHASHIBI, “Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation,” in *2021 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2021, p. 165-170.

[74] X. LI, X. YING et M. C. CHUAH, “Grip : Graph-based interaction-aware trajectory prediction,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, p. 3960-3966.

[75] Y. HU, W. ZHAN et M. TOMIZUKA, “A framework for probabilistic generic traffic scene prediction,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, p. 2790-2796.

[76] M. BAHARI et A. ALAHI, “Feed-forwards meet recurrent networks in vehicle trajectory prediction,” in *Swiss Transport Research Conference (STRC)*, 2019.

[77] F. ALTCHÉ et A. de LA FORTELLE, “An LSTM network for highway trajectory prediction,” in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, IEEE, 2017, p. 353-359.

[78] N. DEO et M. M. TRIVEDI, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, p. 1179-1184.

[79] C. TANG et R. R. SALAKHUTDINOV, “Multiple futures prediction,” *Advances in Neural Information Processing Systems*, t. 32, 2019.

[80] Y. HU, W. ZHAN, L. SUN et M. TOMIZUKA, “Multi-modal probabilistic prediction of interactive behavior via an interpretable model,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, p. 557-563.

[81] T. ZHAO, Y. XU, M. MONFORT, W. CHOI, C. BAKER, Y. ZHAO, Y. WANG et Y. N. WU, “Multi-agent tensor fusion for contextual trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, p. 12 126-12 134.

[82] F. SCARSELLI, M. GORI, A. C. TSOI, M. HAGENBUCHNER et G. MONFARDINI, “The graph neural network model,” *IEEE transactions on neural networks*, t. 20, n° 1, p. 61-80, 2008.

[83] Y. HU, W. ZHAN et M. TOMIZUKA, “Scenario-transferable semantic graph reasoning for interaction-aware probabilistic prediction,” *arXiv preprint arXiv :2004.03053*, 2020.

[84] J. GAO, C. SUN, H. ZHAO, Y. SHEN, D. ANGUELOV, C. LI et C. SCHMID, “Vectornet : Encoding hd maps and agent dynamics from vectorized representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, p. 11 525-11 533.

[85] W. ZHAN, L. SUN, D. WANG, H. SHI, A. CLAUSSE, M. NAUMANN, J. KUMMERLE, H. KONIGSHOF, C. STILLER, A. de LA FORTELLE et al., “Interaction dataset : An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps,” *arXiv preprint arXiv :1910.03088*, 2019.

[86] T. GINDELE, S. BRECHTEL et R. DILLMANN, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” in *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2010, p. 1625-1631.

[87] T. GINDELE, S. BRECHTEL et R. DILLMANN, “Learning driver behavior models from traffic observations for decision making and planning,” *IEEE Intelligent Transportation Systems Magazine*, t. 7, n° 1, p. 69-79, 2015.

[88] A. P. DEMPSTER, N. M. LAIRD et D. B. RUBIN, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society : Series B (Methodological)*, t. 39, n° 1, p. 1-22, 1977.

[89] G. AGAMENNONI, J. I. NIETO et E. M. NEBOT, “Estimation of multivehicle dynamics by considering contextual information,” *IEEE Transactions on robotics*, t. 28, n° 4, p. 855-870, 2012.

[90] S. LEFÈVRE, C. LAUGIER et J. IBAÑEZ-GUZMÁN, “Risk assessment at road intersections : Comparing intention and expectation,” in *2012 ieee intelligent vehicles symposium*, IEEE, 2012, p. 165-171.

[91] J. SCHULZ, C. HUBMANN, J. LÖCHNER et D. BURSCHKA, “Interaction-aware probabilistic behavior prediction in urban environments,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, p. 3999-4006.

[92] J. SCHULZ, C. HUBMANN, J. LÖCHNER et D. BURSCHKA, “Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, p. 1467-1474.

[93] J. SCHULZ, C. HUBMANN, N. MORIN, J. LÖCHNER et D. BURSCHKA, “Learning interaction-aware probabilistic driver behavior models from urban scenarios,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, p. 1326-1333.

[94] M. BAHRAM, C. HUBMANN, A. LAWITZKY, M. AEBERHARD et D. WOLLHERR, “A combined model-and learning-based framework for interaction-aware maneuver prediction,” *IEEE Transactions on Intelligent Transportation Systems*, t. 17, n° 6, p. 1538-1550, 2016.

[95] N. LEE, W. CHOI, P. VERNAZA, C. B. CHOY, P. H. TORR et M. CHANDRAKER, “Desire : Distant future prediction in dynamic scenes with interacting agents,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, p. 336-345.

[96] A. Y. NG, S. RUSSELL et al., “Algorithms for inverse reinforcement learning.,” in *Icml*, t. 1, 2000, p. 2.

[97] L. SUN, W. ZHAN et M. TOMIZUKA, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, p. 2111-2117.

[98] G. S. AOUDE, B. D. LUDERS, K. K. LEE, D. S. LEVINE et J. P. HOW, "Threat assessment design for driver assistance system at intersections," in *13th international ieee conference on intelligent transportation systems*, IEEE, 2010, p. 1855-1862.

[99] A. EIDEHALL et L. PETERSSON, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on intelligent transportation systems*, t. 9, n° 1, p. 137-147, 2008.

[100] N. KAEMPCHEN, B. SCHIELE et K. DIETMAYER, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, t. 10, n° 4, p. 678-687, 2009.

[101] T. BATZ, K. WATSON et J. BEYERER, "Recognition of dangerous situations within a cooperative group of vehicles," in *2009 IEEE Intelligent Vehicles Symposium*, IEEE, 2009, p. 907-912.

[102] S. WORRALL, D. ORCHANSKY, F. MASSON et E. NEBOT, "Improving vehicle safety using context based detection of risk," in *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2010, p. 379-385.

[103] J. SALAS, H. JIMÉNEZ, J. GONZÁLEZ et J. HURTADO, "Detecting unusual activities at vehicular intersections," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, 2007, p. 864-869.

[104] V. PUNZO, M. T. BORZACCHIELLO et B. CIUFFO, "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data," *Transportation Research Part C : Emerging Technologies*, t. 19, n° 6, p. 1243-1262, 2011.

[105] B. COIFMAN et L. LI, "A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset," *Transportation Research Part B : Methodological*, t. 105, p. 362-377, 2017.

[106] R. KRAJEWSKI, J. BOCK, L. KLOEKER et L. ECKSTEIN, "The HighD dataset : A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, p. 2118-2125.

[107] R. KRAJEWSKI, T. MOERS, J. BOCK, L. VATER et L. ECKSTEIN, "The round dataset : A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, p. 1-6.

[108] J. BOCK, R. KRAJEWSKI, T. MOERS, S. RUNDE, L. VATER et L. ECKSTEIN, "The ind dataset : A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020, p. 1929-1934.

[109] A. GEIGER, P. LENZ, C. STILLER et R. URTASUN, "Vision meets robotics : The kitti dataset," *The International Journal of Robotics Research*, t. 32, n° 11, p. 1231-1237, 2013.

[110] M. CORDTS, M. OMRAN, S. RAMOS, T. REHFELD, M. ENZWEILER, R. BENENSON, U. FRANKE, S. ROTH et B. SCHIELE, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 3213-3223.

[111] X. HUANG, P. WANG, X. CHENG, D. ZHOU, Q. GENG et R. YANG, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, t. 42, n° 10, p. 2702-2719, 2019.

[112] M.-F. CHANG, J. LAMBERT, P. SANGKLOY, J. SINGH, S. BAK, A. HARTNETT, D. WANG, P. CARR, S. LUCEY, D. RAMANAN et al., "Argoverse : 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, p. 8748-8757.

[113] K. N. PLATANIOTIS et D. HATZINAKOS, "Gaussian mixtures and their applications to signal processing," *Advanced signal processing handbook*, p. 89-124, 2017.

[114] C. M. BISHOP, "Mixture density networks," 1994.

[115] O. MAKANSI, E. ILG, O. CICEK et T. BROX, "Overcoming limitations of mixture density networks : A sampling and fitting framework for multimodal future prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, p. 7144-7153.

[116] F. DAUM et J. HUANG, "Curse of dimensionality and particle filters," in *2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652)*, IEEE, t. 4, 2003, 4_1979-4_1993.

[117] C. SNYDER, T. BENGTSSON, P. BICKEL et J. ANDERSON, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, t. 136, n° 12, p. 4629-4640, 2008.

[118] T. BENGTSSON, P. BICKEL, B. LI et al., "Curse-of-dimensionality revisited : Collapse of the particle filter in very large scale systems," *Probability and statistics : Essays in honor of David A. Freedman*, t. 2, p. 316-334, 2008.

[119] S. NANNURU, Y. LI, Y. ZENG, M. COATES et B. YANG, "Radio-frequency tomography for passive indoor multitarget tracking," *IEEE Transactions on Mobile Computing*, t. 12, n° 12, p. 2322-2333, 2012.

[120] S. NANNURU, Y. LI, M. COATES et B. YANG, “Multi-target device-free tracking using radio frequency tomography,” in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, IEEE, 2011, p. 508-513.

[121] P. M. DJURIĆ et M. F. BUGALLO, “Particle filtering for high-dimensional systems,” in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, IEEE, 2013, p. 352-355.

[122] B. AIT-EL-FQUIH et I. HOTEIT, “A variational Bayesian multiple particle filtering scheme for large-dimensional systems,” *IEEE Transactions on Signal Processing*, t. 64, n° 20, p. 5409-5422, 2016.

[123] J. R. HERSHEY et P. A. OLSEN, “Approximating the Kullback Leibler divergence between Gaussian mixture models,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*, IEEE, t. 4, 2007, p. IV-317.

[124] J.-L. DURRIEU, J.-P. THIRAN et F. KELLY, “Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian mixture models,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 2012, p. 4833-4836.

[125] B. D. ANDERSON et J. B. MOORE, *Optimal filtering*. Courier Corporation, 2012.

[126] Y. SUN, A. K. WONG et M. S. KAMEL, “Classification of imbalanced data : A review,” *International journal of pattern recognition and artificial intelligence*, t. 23, n° 04, p. 687-719, 2009.

[127] A. PASZKE, S. GROSS, S. CHINTALA, G. CHANAN, E. YANG, Z. DEVITO, Z. LIN, A. DESMAISON, L. ANTIGA et A. LERER, “Automatic differentiation in pytorch,” 2017.

[128] R. M. ZUR, Y. JIANG, L. L. PESCE et K. DRUKKER, “Noise injection for training artificial neural networks : A comparison with weight decay and early stopping,” *Medical physics*, t. 36, n° 10, p. 4810-4818, 2009.

[129] J. KUKAČKA, V. GOLKOV et D. CREMERS, “Regularization for deep learning : A taxonomy,” *arXiv preprint arXiv :1710.10686*, 2017.

[130] C. M. BISHOP, “Training with noise is equivalent to Tikhonov regularization,” *Neural computation*, t. 7, n° 1, p. 108-116, 1995.

[131] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. COURVILLE et Y. BENGIO, “Generative adversarial nets,” *Advances in neural information processing systems*, t. 27, 2014.

[132] A. BORJI, “Pros and cons of GAN evaluation measures,” *Computer Vision and Image Understanding*, t. 179, p. 41-65, 2019.

[133] M. ARJOVSKY et L. BOTTOU, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv :1701.04862*, 2017.

[134] Y. PU, Z. GAN, R. HENAO, X. YUAN, C. LI, A. STEVENS et L. CARIN, “Variational autoencoder for deep learning of images, labels and captions,” *Advances in neural information processing systems*, t. 29, 2016.

[135] M. J. KUSNER, B. PAIGE et J. M. HERNÁNDEZ-LOBATO, “Grammar variational autoencoder,” in *International conference on machine learning*, PMLR, 2017, p. 1945-1954.

[136] B. IVANOVIC, K. LEUNG, E. SCHMERLING et M. PAVONE, “Multimodal deep generative models for trajectory prediction : A conditional variational autoencoder approach,” *IEEE Robotics and Automation Letters*, t. 6, n° 2, p. 295-302, 2020.

[137] Y. BENGIO, A. COURVILLE et P. VINCENT, “Representation learning : A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, t. 35, n° 8, p. 1798-1828, 2013.

[138] Y. LECUN, L. BOTTOU, Y. BENGIO et P. HAFFNER, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, t. 86, n° 11, p. 2278-2324, 1998.

[139] B. M. LAKE, T. D. ULLMAN, J. B. TENENBAUM et S. J. GERSHMAN, “Building machines that learn and think like people,” *Behavioral and brain sciences*, t. 40, 2017.

[140] L. TRAN, X. YIN et X. LIU, “Disentangled representation learning GAN for pose-invariant face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, p. 1415-1424.

[141] J. D. LEE, Q. LEI, N. SAUNSHI et J. ZHUO, “Predicting what you already know helps : Provable self-supervised learning,” *Advances in Neural Information Processing Systems*, t. 34, p. 309-323, 2021.

[142] A. L. MAAS, A. Y. HANNUN, A. Y. NG et al., “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, Citeseer, t. 30, 2013, p. 3.

[143] B. KRAWCZYK, “Learning from imbalanced data : open challenges and future directions,” *Progress in Artificial Intelligence*, t. 5, n° 4, p. 221-232, 2016.

[144] T. GUO, X. ZHU, Y. WANG et F. CHEN, “Discriminative sample generation for deep imbalanced learning,” in *Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}*, International Joint Conferences on Artificial Intelligence Organization, 2019.

[145] S. M. LAVALLE, *Planning algorithms*. Cambridge university press, 2006.

[146] S. R. BOWMAN, L. VILNIS, O. VINYALS, A. M. DAI, R. JOZEFOWICZ et S. BENGIO, "Generating sentences from a continuous space," *arXiv preprint arXiv :1511.06349*, 2015.

[147] J. LUCAS, G. TUCKER, R. GROSSE et M. NOROUZI, "Understanding posterior collapse in generative latent variable models," 2019.

[148] J. HE, D. SPOKOYNY, G. NEUBIG et T. BERG-KIRKPATRICK, "Lagging inference networks and posterior collapse in variational autoencoders," *arXiv preprint arXiv :1901.05534*, 2019.

[149] I. GULRAJANI, K. KUMAR, F. AHMED, A. A. TAIGA, F. VISIN, D. VAZQUEZ et A. COURVILLE, "Pixelvae : A latent variable model for natural images," *arXiv preprint arXiv :1611.05013*, 2016.

[150] Z. YANG, Z. HU, R. SALAKHUTDINOV et T. BERG-KIRKPATRICK, "Improved variational autoencoders for text modeling using dilated convolutions," in *International conference on machine learning*, PMLR, 2017, p. 3881-3890.

[151] I. HIGGINS, L. MATTHEY, A. PAL, C. BURGESS, X. GLOROT, M. BOTVINICK, S. MOHAMED et A. LERCHNER, "beta-vae : Learning basic visual concepts with a constrained variational framework," 2016.

[152] C. K. SØNDERBY, T. RAIKO, L. MAALØE, S. K. SØNDERBY et O. WINTHER, "Ladder variational autoencoders," *Advances in neural information processing systems*, t. 29, 2016.

[153] C.-W. HUANG, S. TAN, A. LACOSTE et A. C. COURVILLE, "Improving explorability in variational inference with annealed variational objectives," *Advances in Neural Information Processing Systems*, t. 31, 2018.

[154] D. P. KINGMA, S. MOHAMED, D. JIMENEZ REZENDE et M. WELLING, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, t. 27, 2014.

[155] A. VAN DEN OORD, O. VINYALS et al., "Neural discrete representation learning," *Advances in neural information processing systems*, t. 30, 2017.

[156] A. RAZAVI, A. v. d. OORD, B. POOLE et O. VINYALS, "Preventing posterior collapse with delta-vaes," *arXiv preprint arXiv :1901.03416*, 2019.

[157] Y. HU, L. SUN et M. TOMIZUKA, "Generic prediction architecture considering both rational and irrational driving behaviors," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, p. 3539-3546.

[158] H. FU, C. LI, X. LIU, J. GAO, A. CELIKYILMAZ et L. CARIN, “Cyclical annealing schedule : A simple approach to mitigating kl vanishing,” *arXiv preprint arXiv :1903.10145*, 2019.

[159] E. MATHIEU, T. RAINFORTH, N. SIDDHARTH et Y. W. TEH, “Disentangling disentanglement in variational autoencoders,” in *International Conference on Machine Learning*, PMLR, 2019, p. 4402-4412.

[160] S. ZHAO, J. SONG et S. ERMON, “Infovae : Balancing learning and inference in variational autoencoders,” in *Proceedings of the aaai conference on artificial intelligence*, t. 33, 2019, p. 5885-5892.

[161] T. SALZMANN, B. IVANOVIC, P. CHAKRAVARTY et M. PAVONE, “Trajectron++ : Dynamically-feasible trajectory forecasting with heterogeneous data,” in *European Conference on Computer Vision*, Springer, 2020, p. 683-700.

[162] L. GIRIN, S. LEGLAIVE, X. BIE, J. DIARD, T. HUEBER et X. ALAMEDA-PINEDA, “Dynamical variational autoencoders : A comprehensive review,” *arXiv preprint arXiv :2008.12595*, 2020.

[163] O. FABIUS et J. R. VAN AMERSFOORT, “Variational recurrent auto-encoders,” *arXiv preprint arXiv :1412.6581*, 2014.

[164] M. FONTMARTY, F. LERASLE et P. DANES, “Towards real-time markerless human motion capture from ambiance cameras using an hybrid particle filter,” in *2008 15th IEEE International Conference on Image Processing*, IEEE, 2008, p. 709-712.

[165] M. FONTMARTY, P. DANÈS et F. LERASLE, “Quasi Monte Carlo partitioned filtering for visual human motion capture,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2009, p. 2553-2556.

[166] W. LUO, J. XING, A. MILAN, X. ZHANG, W. LIU et T.-K. KIM, “Multiple object tracking : A literature review,” *Artificial Intelligence*, t. 293, p. 103 448, 2021.

[167] A. DOSOVITSKIY, G. ROS, F. CODEVILLA, A. LOPEZ et V. KOLTUN, “CARLA : An open urban driving simulator,” in *Conference on robot learning*, PMLR, 2017, p. 1-16.

[168] R. LOPEZ, P. DANÈS et F. ROYER, “Extending the IMM filter to heterogeneous-order state space models,” in *49th IEEE Conference on Decision and Control (CDC)*, IEEE, 2010, p. 7369-7374.

[169] H. G. SEIF et X. HU, “Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry,” *Engineering*, t. 2, n° 2, p. 159-162, 2016.

[170] A. GARCIA-GARCIA, S. ORTS-ESCOLANO, S. OPREA, V. VILLENA-MARTINEZ et J. GARCIA-RODRIGUEZ, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv :1704.06857*, 2017.

ANNEXE A
ESTIMATION DE L'ÉTAT D'UN VÉHICULE AVEC UN IMM-EKF D'ORDRE NON-HOMOGÈNE

A.1 Introduction

L'objectif est de présenter un Filtre de Kalman étendu à modèles multiples interactifs (Interacting Multiple Models Extended Kalman Filter – IMM EKF) dans le cas où l'ordre du vecteur d'état n'est pas identique pour tous les modèles [168]. Les deux modèles utilisés sont le modèle à vitesse constante curviligne est inconnue et le modèle à vitesse curviligne et angulaire constantes inconnues.

A.2 Modélisation

A.2.1 Modèle à vitesse constante

Le premier modèle (1) représente l'état avec le vecteur $\boldsymbol{x}^{(1)} = (x, y, \theta, v)$ contenant la position, l'orientation et la vitesse curviligne du véhicule. Il est défini par

$$\boldsymbol{x}_k^{(1)} = f^{(1)}(\boldsymbol{x}_{k-1}^{(1)}) + \boldsymbol{w}^{(1)} \qquad \text{avec } \boldsymbol{w}^{(1)} \sim \mathcal{N}(0, \boldsymbol{Q}^{(1)}) \tag{A.1}$$

$$\boldsymbol{z} = h^{(1)}(\boldsymbol{x}_{k-1}^{(1)}) + \boldsymbol{v} \qquad \text{avec } \boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R}) \tag{A.2}$$

où

$$f^{(1)}(\boldsymbol{x}_{k-1}^{(1)}) = \begin{cases} x_k = x_{k-1} + v_{k-1}\cos(\theta_{k-1})\Delta t \\ y_k = y_{k-1} + v_{k-1}\sin(\theta_{k-1})\Delta t \\ \theta_k = \theta_{k-1} \\ v_k = v_{k-1} \end{cases} \tag{A.3}$$

avec Δt le pas de temps choisi lors de la discrétisation du modèle. Le modèle d'observation est défini simplement par $h^{(1)}(\boldsymbol{x}_{k-1}^{(1)}) = H^{(1)}\boldsymbol{x}_{k-1}^{(1)}$ avec $H^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$. La covariance du bruit de mesure $\boldsymbol{R}$ est définie comme dans (4.27). La covariance du bruit de dynamique est défini comme

suit

$$Q^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & D_\theta & 0 \\ 0 & 0 & 0 & D_v \end{pmatrix}, \tag{A.4}$$

où D_θ et D_v sont les deux variances du bruit de dynamique sur le cap et la vitesse respectivement. La matrice Jacobienne de la fonction $f^{(1)}$ en $\boldsymbol{x}$ est donnée par

$$\frac{\partial f^{(1)}(\boldsymbol{x})}{\partial \boldsymbol{x}^T} = \begin{pmatrix} 1 & 0 & -v\sin(\theta)\Delta t & \cos(\theta)\Delta t \\ 0 & 1 & -v\cos(\theta)\Delta t & \sin(\theta)\Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{A.5}$$

A.2.2 Modèle à vitesses curviligne et angulaire constantes

Le deuxième modèle (2) représente l'état avec le vecteur $\boldsymbol{x}^{(2)} = (x, y, \theta, v, \omega)$ contenant la position, l'orientation, la vitesse curviligne et la vitesse angulaire du véhicule. Il est défini comme suit

$$\boldsymbol{x}_k^{(2)} = f^{(2)}(\boldsymbol{x}_{k-1}^{(2)}) + \boldsymbol{w}^{(2)} \qquad \text{avec } \boldsymbol{w}^{(2)} \sim \mathcal{N}(0, \boldsymbol{Q}^{(2)}) \tag{A.6}$$

$$\boldsymbol{z} = H^{(2)}(\boldsymbol{x}_{k-1}^{(2)}) + \boldsymbol{v} \qquad \text{avec } \boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R}) \tag{A.7}$$

où

$$f^{(2)}(\boldsymbol{x}_{k-1}^{(2)}) = \begin{cases} x_k = x_{k-1} + \frac{v_{k-1}}{\omega_{k-1}}\big[\sin(\theta_{k-1} + \omega_{k-1}\Delta t) - \sin(\theta_{k-1})\big] \\ y_k = y_{k-1} + \frac{v_{k-1}}{\omega_{k-1}}\big[\cos(\theta_{k-1} + \omega_{k-1}\Delta t) - \cos(\theta_{k-1})\big] \\ \theta_k = \theta_{k-1} + \omega_{k-1}\Delta t \\ v_k = v_{k-1} \\ \omega_k = \omega_{k-1} \end{cases} \tag{A.8}$$

Le modèle d'observation est défini par $h^{(2)}(\boldsymbol{x}_{k-1}^{(2)}) = H^{(2)}\boldsymbol{x}_{k-1}^{(2)}$ avec $H^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$. La covariance du bruit de mesure $\boldsymbol{R}$ est définie comme dans (4.27). La covariance du bruit de

dynamique est défini comme suit

$$Q^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_v & 0 \\ 0 & 0 & 0 & 0 & D_\omega \end{pmatrix}, \tag{A.9}$$

où D_v et D_ω sont les deux variances du bruit de dynamique sur la vitesse curviligne et la vitesse angulaire respectivement. La matrice Jacobienne de la fonction $f^{(2)}$ en $\boldsymbol{x}$ est donnée par

$$\frac{\partial f^{(2)}(\boldsymbol{x})}{\partial \boldsymbol{x}^T} = \begin{pmatrix} 1 & 0 & F_{13} & F_{14} & F_{15} \\ 0 & 1 & F_{23} & F_{24} & F_{25} \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{A.10}$$

et $F_{13} = \frac{v}{\omega}\big[\cos(\theta + \omega\Delta t) - \cos\theta\big]$; $F_{14} = \frac{1}{\omega}\big[\sin(\theta + \omega\Delta t) - \sin\theta\big]$; $F_{15} = \frac{-v\big[\sin(\theta+\omega\Delta t)-\sin\theta\big]}{\omega^2}$; $F_{23} = \frac{v}{\omega}\big[\sin(\theta + \omega\Delta t) - \sin\theta\big]$; $F_{24} = \frac{1}{\omega}\big[\cos(\theta + \omega\Delta t) - \cos\theta\big]$; $F_{25} = \frac{v\big[\cos(\theta+\omega\Delta t)-\cos\theta\big]}{\omega^2} + \frac{v\Delta t}{\omega}\sin(\omega + \omega\Delta t)$. Le filtre IMM met à jour tous les modèles simultanément et passe stochastiquement d'un modèle à l'autre en fonction d'un poids pour chaque modèle noté $\mu_k^{(i)}$. On suppose que les probabilités a priori de transition entre les modèles $\pi_{11}, \pi_{12}, \pi_{22}$ et π_{21} sont connues. A l'initialisation, pour $k = 0$, on suppose que les probabilités $\mu_0^{(1)}$ et $\mu_0^{(2)}$ d'être dans l'un des deux modèles tel que $m_0 = (1)$ ou (2) respectivement, sont données. De plus, on suppose que $\hat{\boldsymbol{x}}_{0|0}^{(1)}$, $\boldsymbol{P}_{0|0}^{(1)}$ et $\hat{\boldsymbol{x}}_{0|0}^{(2)}$, $\boldsymbol{P}_{0|0}^{(2)}$, moments de $p(\boldsymbol{x}_0|m_0 = 1)$ et $p(\boldsymbol{x}_0|m_0 = 2)$ sont données.

A.3 Estimation

Par la suite, pour tous les i modèles, on cherche à obtenir récursivement $\mu_k^{(i)}$, $\hat{\boldsymbol{x}}_{k|k}^{(i)}$ et $\boldsymbol{P}_{k|k}^{(i)}$ depuis $k-1$ pour calculer l'estimé global du véhicule à l'instant k. L'algorithme A.1 présente une solution via un IMM-EKF pour le problème d'estimation considéré.

Algorithme 2 IMM EKF non homogène

1: Définir $\boldsymbol{E} = \big(\mathbf{1}_{4\times4}|\mathbf{0}_{4\times1}\big)$ et $\boldsymbol{D} = \big(\mathbf{0}_{1\times4}|1\big)$.
2: **pour** $i \in \{1,2\}$ **faire**
3: **pour** $j \in \{1,2\}$ **faire** Calculer la probabilité de prédiction du $i^{\text{ème}}$ mode $\mu^{(i)}_{k|k-1} = \sum_{j=1}^{2} \pi_{ji}\mu^{(j)}_{k-1}$
4: **pour** $i \in \{1,2\}$ **faire**
5: **pour** $j \in \{1,2\}$ **faire** Déterminer les probabilités de mixage $\mu^{j|i}_{k-1} = \frac{\pi_{ji}\mu^{(j)}_{k-1}}{\mu^{(i)}_{k|k-1}}$
6: Calculer l'estimé a priori et sa covariance pour les deux modes possibles:
a: $\bar{\boldsymbol{x}}^{(1)}_{k-1|k-1} = \sum_{j=1}^{2} \mu^{j|1}_{k-1}\boldsymbol{E}^{(j|1)}\hat{\boldsymbol{x}}^{(j)}_{k-1|k-1}$ où $\boldsymbol{E}^{1|1} = \mathbf{1}_{4\times4}$ et $\boldsymbol{E}^{2|1} = \boldsymbol{E}$
b: $\bar{\boldsymbol{P}}^{(1)}_{k-1|k-1} = \sum_{j=1}^{2} \mu^{j|1}_{k-1}\big[\boldsymbol{E}^{j|1}\boldsymbol{P}^{(j)}_{k-1|k-1}(\boldsymbol{E}^{j|1})^T + (\boldsymbol{E}^{j|1}\hat{\boldsymbol{x}}^{(j)}_{k-1|k-1} - \bar{\boldsymbol{x}}^{(1)}_{k-1|k-1})(\diamond)^T\big]$
c: $\bar{\boldsymbol{x}}^{(2)}_{k-1|k-1} = \begin{pmatrix}\bar{\boldsymbol{\xi}}_{k-1|k-1}\\ \bar{\boldsymbol{\omega}}_{k-1|k-1}\end{pmatrix}$ avec $\bar{\boldsymbol{\xi}}_{k-1|k-1} = \sum_{j=1}^{2} \mu^{j|2}_{k-1}\boldsymbol{E}^{(j|2)}\hat{\boldsymbol{x}}^{(j)}_{k-1|k-1}$ où
$\boldsymbol{E}^{1|2} = \boldsymbol{E}^{1|1}$, $\boldsymbol{E}^{2|2} = \boldsymbol{E}^{2|1}$ et $\bar{\boldsymbol{\omega}}_{k-1|k-1} = \boldsymbol{D}\hat{\boldsymbol{x}}^{(2)}_{k-1|k-1}$
d: $\bar{\boldsymbol{P}}^{(2)}_{k-1|k-1} = \begin{pmatrix}\bar{\boldsymbol{\Xi}}_{k-1|k-1} & \bar{\boldsymbol{\Sigma}}_{k-1|k-1}\\ \bar{\boldsymbol{\Sigma}}^T_{k-1|k-1} & \bar{\boldsymbol{\Omega}}_{k-1|k-1}\end{pmatrix}$ avec
$\bar{\boldsymbol{\Xi}}_{k-1|k-1} = \sum_{j=1}^{2} \mu^{j|2}_{k-1}\big[\boldsymbol{E}^{j|2}\boldsymbol{P}^{(j)}_{k-1|k-1}(\boldsymbol{E}^{j|2})^T + (\boldsymbol{E}^{j|2}\hat{\boldsymbol{x}}^{(j)}_{k-1|k-1} - \bar{\boldsymbol{\xi}}_{k-1|k-1})(\diamond)^T\big]$,
$\bar{\boldsymbol{\Sigma}}_{k-1|k-1} = \boldsymbol{E}\boldsymbol{P}^{(2)}_{k-1|k-1}\boldsymbol{D}^T$,
$\bar{\boldsymbol{\Sigma}}^T_{k-1|k-1} = \boldsymbol{D}\boldsymbol{P}^{(2)}_{k-1|k-1}\boldsymbol{E}^T$ et
$\bar{\boldsymbol{\Omega}}_{k-1|k-1} = \boldsymbol{D}\boldsymbol{P}^{(2)}_{k-1|k-1}\boldsymbol{D}^T$
7: EKFs associés aux modèles de dynamique et de mesure pour $m_k = i$:
a: Calculer les moments de $p(\boldsymbol{x}_k|m_k = i, \boldsymbol{z}_{1:k-1}) \sim \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}^{(i)}_{k|k-1}, \boldsymbol{P}^{(i)}_{k|k-1})$ où
$\hat{\boldsymbol{x}}^{(i)}_{k|k-1} = f^{(i)}(\bar{\boldsymbol{x}}^{(i)}_{k-1|k-1})$ et
$\boldsymbol{P}^{(i)}_{k|k-1} = \boldsymbol{F}^{(i)}\bar{\boldsymbol{P}}_{k-1|k-1}(\boldsymbol{F}^{(i)})^T + \boldsymbol{Q}^{(i)}$, $\boldsymbol{F}^{(i)} = \big[\frac{\partial f^{(i)}(\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}^T_{k-1}}\big]_{\boldsymbol{x_{k-1}} = \bar{\boldsymbol{x}}^{(i)}_{k-1|k-1}}$.
b: Calculer les moments de $p(\boldsymbol{z}_k|m_k = i, \boldsymbol{z}_{1:k-1}) \sim \mathcal{N}(\boldsymbol{z}_k; \hat{\boldsymbol{z}}^{(i)}_{k|k-1}, \boldsymbol{S}^{(i)}_{k|k-1})$ où
$\hat{\boldsymbol{z}}^{(i)}_{k|k-1} = h^{(i)}(\hat{\boldsymbol{x}}^{(i)}_{k|k-1})$ et
$\boldsymbol{S}^{(i)}_{k|k-1} = \boldsymbol{H}^{(i)}\boldsymbol{P}_{k|k-1}(\boldsymbol{H}^{(i)})^T + \boldsymbol{R}^{(i)}$, $\boldsymbol{H}^{(i)} = \big[\frac{\partial h^{(i)}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}^T_k}\big]_{\boldsymbol{x_k} = \hat{\boldsymbol{x}}^{(i)}_{k|k-1}}$.
c: Enfin calculer $\boldsymbol{K}^{(i)} = \boldsymbol{P}^{(i)}_{k|k-1}(\boldsymbol{H}^{(i)})^T(\boldsymbol{S}^{(i)}_{k|k-1})^{-1}$;
$\hat{\boldsymbol{x}}^{(i)}_{k|k} = \hat{\boldsymbol{x}}^{(i)}_{k|k-1} + \boldsymbol{K}^i(\boldsymbol{z}_k - \hat{\boldsymbol{z}}^{(i)}_{k|k-1})$; $\boldsymbol{P}^{(i)}_{k|k} = \boldsymbol{P}^{(i)}_{k|k-1} - \boldsymbol{K}^{(i)}\boldsymbol{H}^i\boldsymbol{P}^{(i)}_{k|k-1}$
8: **pour** $i \in \{1,2\}$ **faire** Calculer $L^{(i)} = \frac{1}{\sqrt{(2\pi)^2|\boldsymbol{S}^{(i)}_{k|k-1}|}} \exp -\frac{1}{2}(\boldsymbol{z}_k - \hat{\boldsymbol{z}}^{(i)}_{k|k-1})^T(\boldsymbol{S}^{(i)}_{k|k-1})^{-1}(\boldsymbol{z}_k - \hat{\boldsymbol{z}}^{(i)}_{k|k-1})$
9: Calculer la probabilité d'être dans le i-ème modèle: $\mu^{(i)}_k = \frac{\mu^{(i)}_{k|k-1}L^{(i)}}{\sum_{l=1}^{2}\mu^{(l)}_{k|k-1}L^{(l)}}$
10: Pour finir, calculer l'estimé global et sa covariance composante par composante: $\forall p \neq 5, q \neq 5$, $\hat{x}_{k|k,p} = \sum_{i=1}^{2}\mu^{(i)}\hat{x}^{(i)}$ et $P_{k|k}, pq = \sum_{i=1}^{2}\mu^{(i)}\big[P^{(i)}_{k|k,pq} + (\hat{x}^{(i)}_{k|k,p} - \hat{x}_{k|k,p})(\hat{x}^{(i)}_{k|k,q} - \hat{x}_{k|k,q})\big]$. Si $p = 5$ ou $q + 5$ alors $\hat{x}_{k|k,p} = \hat{x}^{(2)}$ et $P_{k|k,pq} = P^{(2)}_{k|k,pq}$

TABLE A.1 – L'algorithme de l'IMM-EKF pour deux modèles de dynamique d'ordre non-homogène.

ANNEXE B
INTÉGRATION D'INFORMATION SÉMANTIQUE DANS LES MODÈLES DE TRANSITION BASÉS SUR LE CVAE

Dans le chapitre 5, la dynamique a priori du véhicule est apprise à l'aide de différentes variations de CVAEs. Pour l'ensemble de ces réseaux de neurones, en plus de l'état de la flotte de véhicule, à chaque instant un vecteur d'information est obtenu depuis un réseau supplémentaire. Ce réseau est composé de plusieurs couches de convolution et de couches entièrement connectées visant à prendre en compte automatiquement la sémantique de l'environnement routier autour de l'égo-véhicule pour l'apprentissage de sa dynamique a priori. L'obtention de cartes sémantiques représentant les différentes zones composant le segment routier avoisinant le véhicule peuvent être obtenues hors-ligne [169] ou en-ligne grâce à des capteurs vidéo et des algorithmes de segmentation [170]. De telles cartes, plus complexes et complètes que celle obtenue par l'approche élémentaire présentées ici, sont également connues sous le nom commercial *HDMaps*.

La figure B.1 illustre le processus utilisé pour générer les codes d'information du segment routier pour un égo-véhicule.

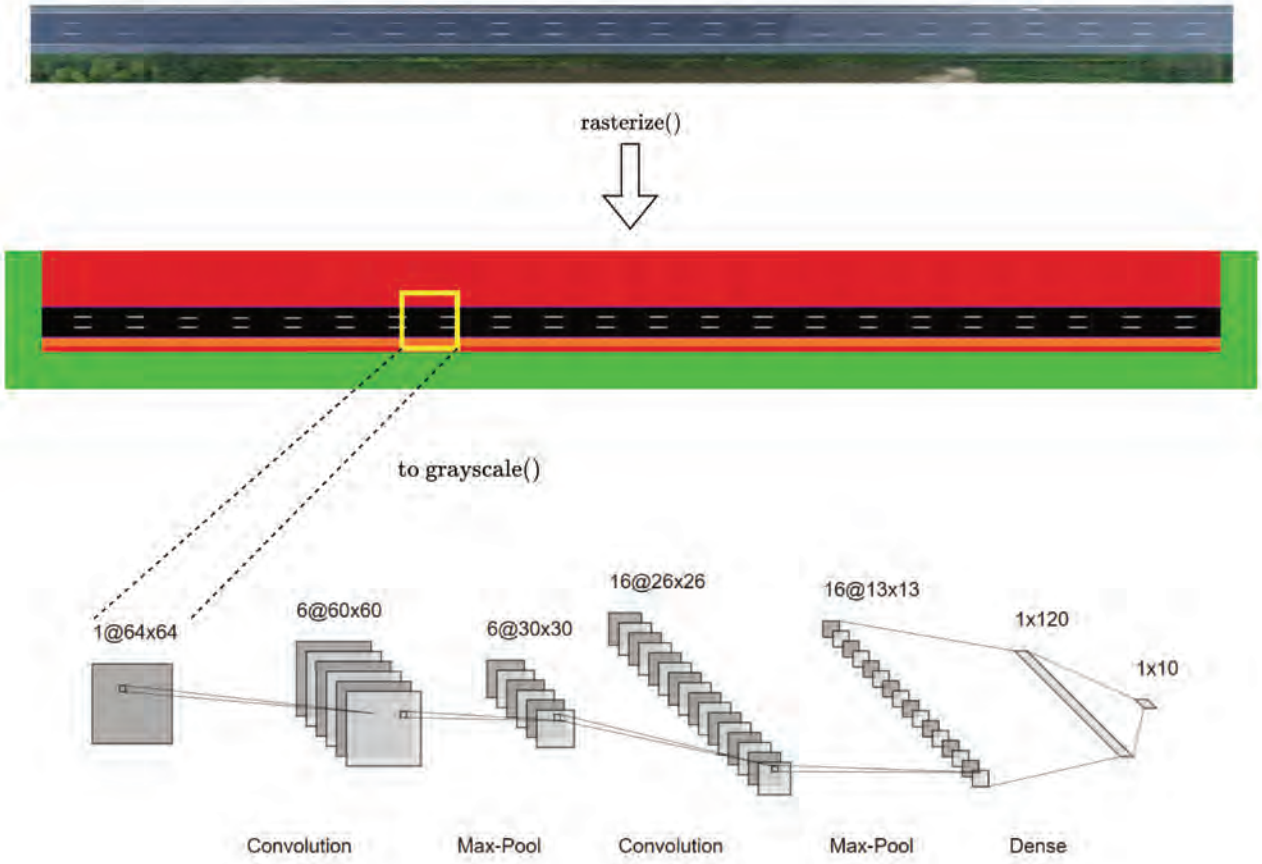


FIGURE B.1 – Processus d'intégration de la sémantique routière autour de l'égo-véhicule. Dans un premier temps, une carte sémantique est construite à partir de la capture aérienne du segment autoroutier sur lequel les véhicules se déplacent. Chaque zone est étiquetée par un opérateur humain. Ici les zones sont : la route en noir, le marquage au sol en blanc, la zone de conduite nominale en rose, la zone d'arrêt d'urgence en orange, les zones non-accessibles en rouge et enfin le vert indique une absence d'étiquette. Lors de l'apprentissage ou de l'inférence, un morceau de l'image d'une taille de 64x64 pixels correspondant à un voisinage d'environ $50m^2$ autour de l'égo-véhicule est extrait de la carte. Ce morceau est ensuite converti en échelle de gris pour ensuite être ingéré par un réseau de neurones dont les différentes couches et opérations sont explicitées sur la figure. En sortie, un vecteur encodant l'information sémantique est obtenu. Celui-ci peut être concaténé au vecteur d'entrée des différents réseaux.

ANNEXE C
MODÈLE UNICYCLE ÉTENDU POUR LA MODÉLISATION CINÉMATIQUE DU VÉHICULE

Le modèle cinématique unicycle étendu peut être utilisé pour représenter l'évolution cinématique d'un véhicule. Ce modèle émule les commandes d'un véhicule standard en utilisant comme variables de commande l'accélération curviligne a et la vitesse de rotation ω. Le modèle cinématique unicycle étendu est décrit par les équations différentielles continues suivantes :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\phi) \\ v\sin(\phi) \\ \omega \\ a \end{bmatrix}, \tag{C.1}$$

où x et y représente la position du véhicule, v sa vitesse curviligne et ϕ son cap. Afin de discrétiser le modèle, on suppose entre deux instants des valeurs constantes pour les variables de commande, ce qui permet d'obtenir un modèle équivalent discret à (C.1) tel que

$$\begin{bmatrix} x^{(t+1)} \\ y^{(t+1)} \\ \phi^{(t+1)} \\ v^{(t+1)} \end{bmatrix} = \begin{bmatrix} x^{(t+1)} \\ y^{(t+1)} \\ \phi^{(t+1)} \\ v^{(t+1)} \end{bmatrix} + \begin{bmatrix} v^{(t)}D_S^{(t)} + \frac{a^{(t)}\sin(\phi^{(t)}+\omega^{(t)}\Delta t)\Delta t}{\omega^{(t)}} + \frac{a^{(t)}}{\omega^{(t)}}D_C^{(t)} \\ -v^{(t)}D_C^{(t)} + \frac{a^{(t)}\cos(\phi^{(t)}+\omega^{(t)}\Delta t)\Delta t}{\omega^{(t)}} + \frac{a^{(t)}}{\omega^{(t)}}D_S^{(t)} \\ \omega^{(t)}\Delta t \\ a^{(t)}\Delta t \end{bmatrix}, \tag{C.2}$$

$$\text{où } D_S^{(t)} = \frac{\sin(\phi^{(t)} + \omega^{(t)}\Delta t) - \sin(\phi^{(t)})}{\omega^{(t)}} \tag{C.3}$$

$$D_C^{(t)} = \frac{\cos(\phi^{(t)} + \omega^{(t)}\Delta t) - \cos(\phi^{(t)})}{\omega^{(t)}}. \tag{C.4}$$

Afin d'éviter les singularités un nouveau modèle est obtenu en évaluant le modèle précédent à la limite $\omega \to 0$.

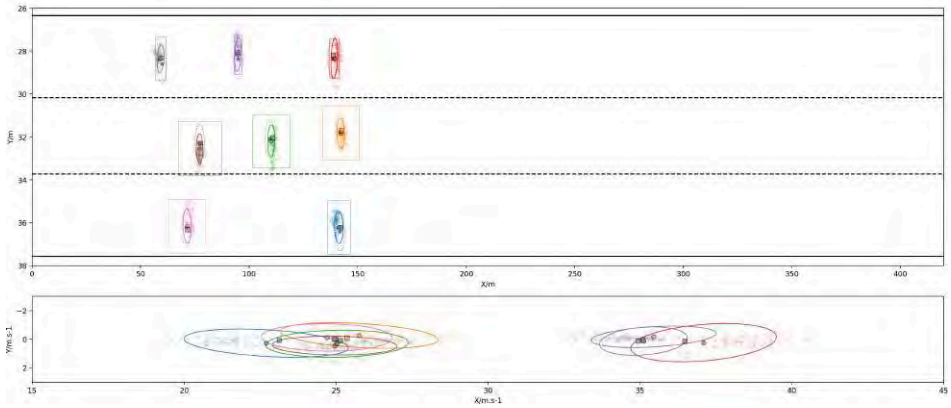
$$\begin{bmatrix} x^{(t+1)} \\ y^{(t+1)} \\ \phi^{(t+1)} \\ v^{(t+1)} \end{bmatrix} = \begin{bmatrix} x^{(t+1)} \\ y^{(t+1)} \\ \phi^{(t+1)} \\ v^{(t+1)} \end{bmatrix} + \begin{bmatrix} v^{(t)}\cos(\phi^{(t)})\Delta t + 0.5a^{(t)}\cos(\phi^{(t)})(\Delta t)^2 \\ v^{(t)}\sin(\phi^{(t)})\Delta t + 0.5a^{(t)}\sin(\phi^{(t)})(\Delta t)^2 \\ 0 \\ a^{(t)}\Delta t \end{bmatrix} \tag{C.5}$$

ANNEXE D
VISUALISATIONS SUPPLÉMENTAIRES DES RÉSULTATS EN ESTIMATION ET PRÉDICTION

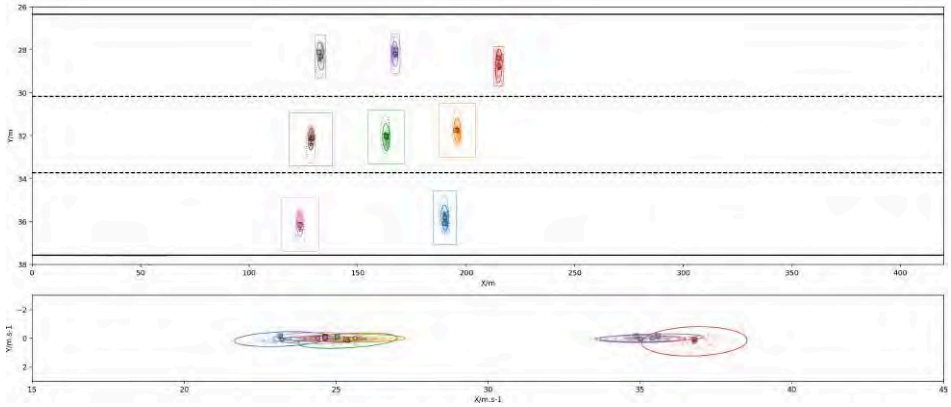
Dans cette section sont présentés des résultats en estimation et prédiction de quatre algorithmes proposés dans cette étude. Ils sont obtenus sur des scénarios différents avec des configurations de bruit de mesure différentes. Comme introduit dans le Chapitre 4 et le Chapitre 5, les résultats sont présentés sous la forme de schémas en vue aérienne. Pour chaque méthode et chaque scénario, trois instants sont illustrés: le début de l'estimation, la fin de l'estimation et la fin de la prédiction. Les quatre méthodes d'estimation et de prédiction présentées sont le MPF w/ GMM avec 100 particules, le VBPF w/ GMM avec 250 particules, le MPF w/ CAB-CVAE avec 100 particules et le MPF w/ CVRAE avec 100 particules.

Pour chaque figure suivante, chaque graphe correspond au quatrième instant du processus d'estimation pour lequel la position latérale (ordonnée) et longitudinale (abscisse) est affichée. Chaque véhicule est représenté par une couleur distincte. Les estimations sont représentées par des marqueurs ronds. Les ellipses de confiance qui leur sont associées devraient entourer les valeurs de la vérité terrain (marqueurs carrés) avec une probabilité de 99%. Les croix de couleur représentent les particules des filtres. Les positions mesurées et le bruit associé sont représentés par des étoiles et des ellipses en pointillés. Les voies sont droites et délimitées par des lignes pointillées.

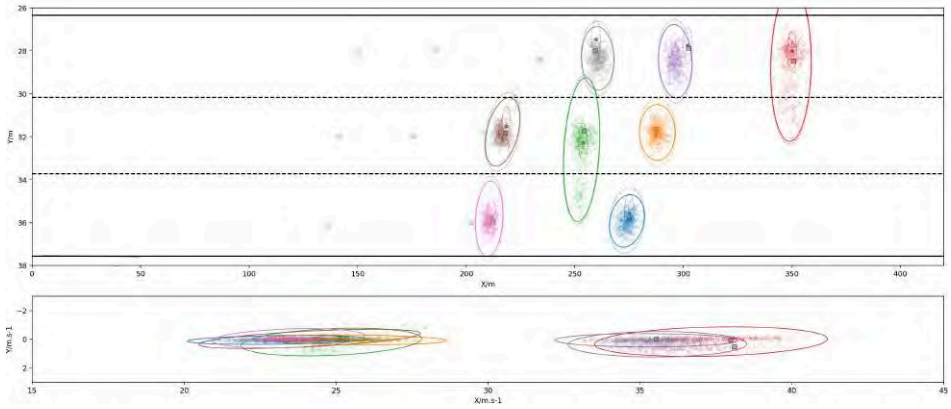
D.1 MPF w/ GMM avec 100 particules



(a)

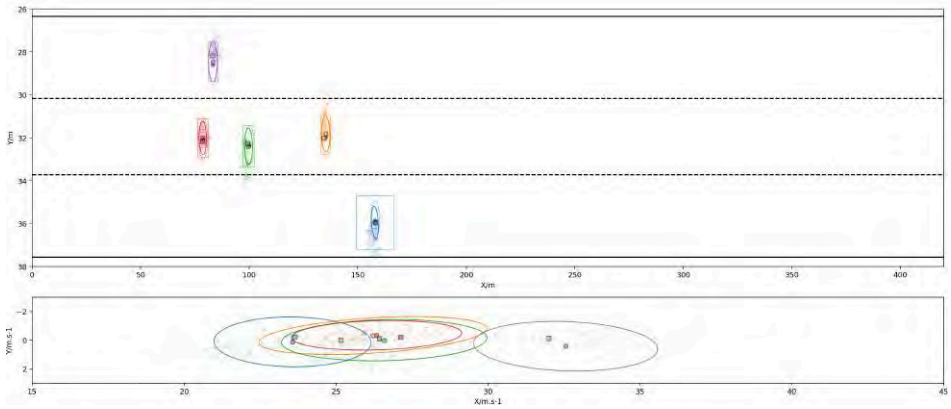


(b)

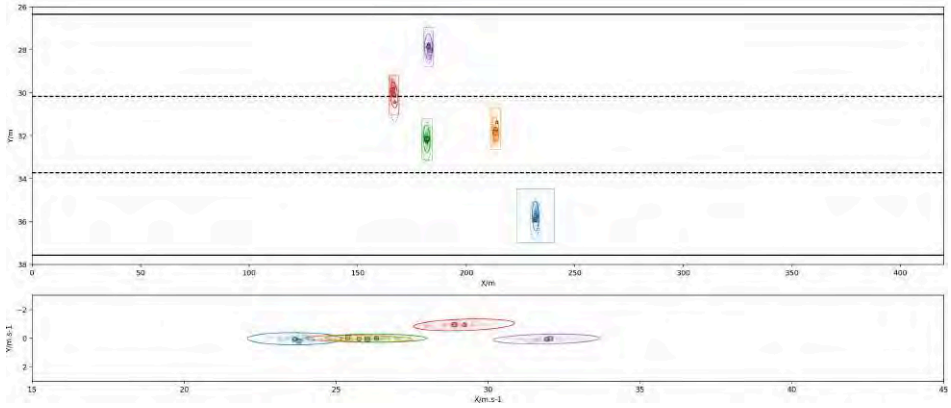


(c)

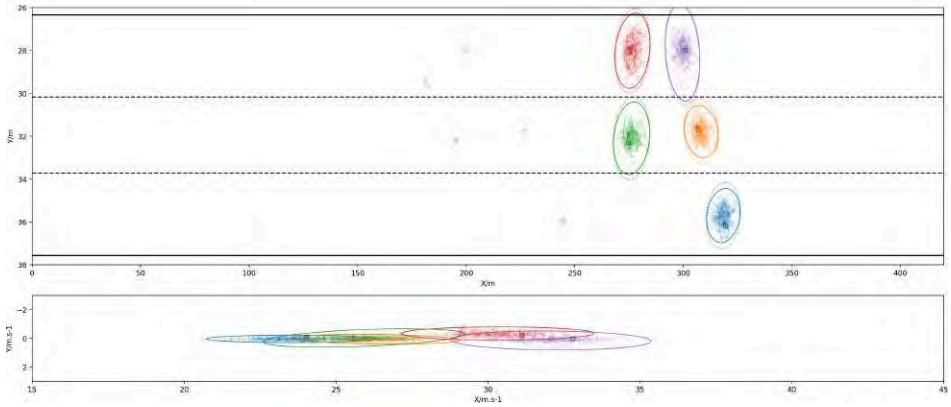
FIGURE D.1 – Scénario 1 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). L'état de la flotte de véhicules avançant en ligne droite est correctement estimé et prédit.



(a)

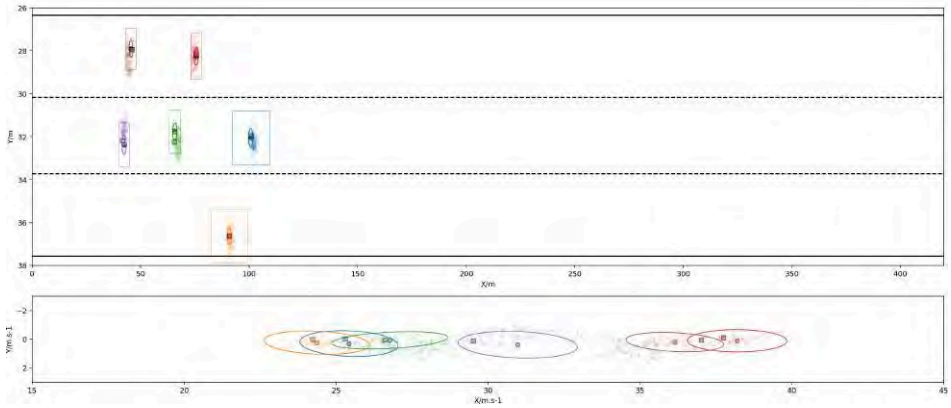


(b)

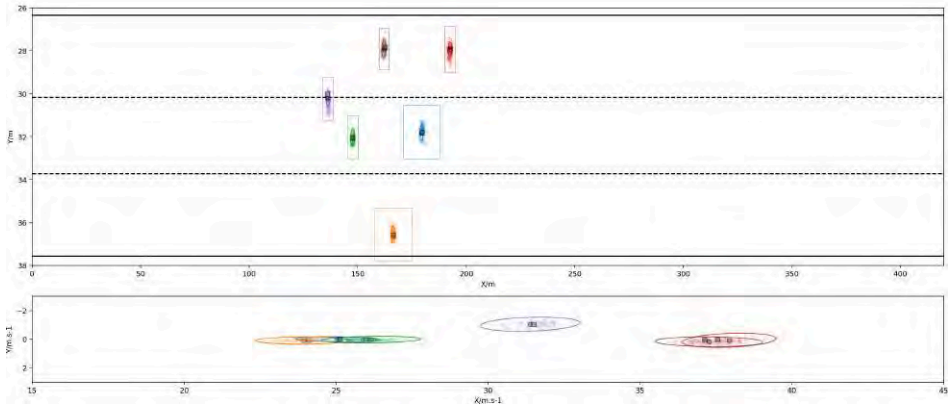


(c)

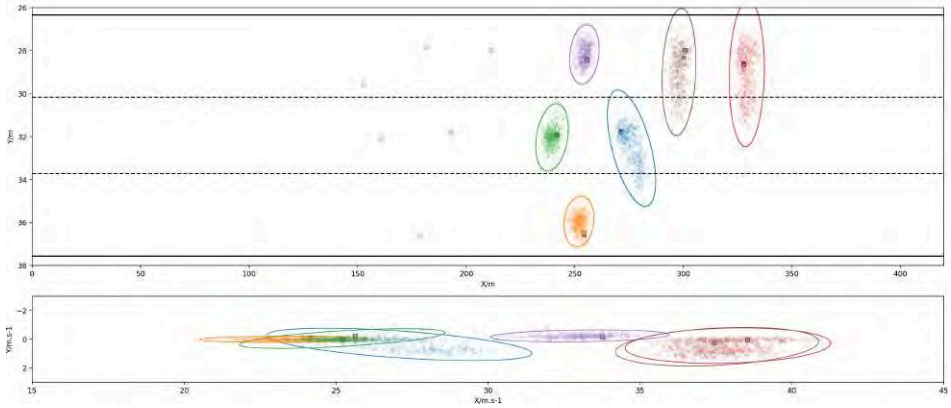
FIGURE D.2 – Scénario 7 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). Après une estimation correcte de la flotte, la prédiction des véhicules rouge et violet effectuant un dépassement est satisfaisante.



(a)

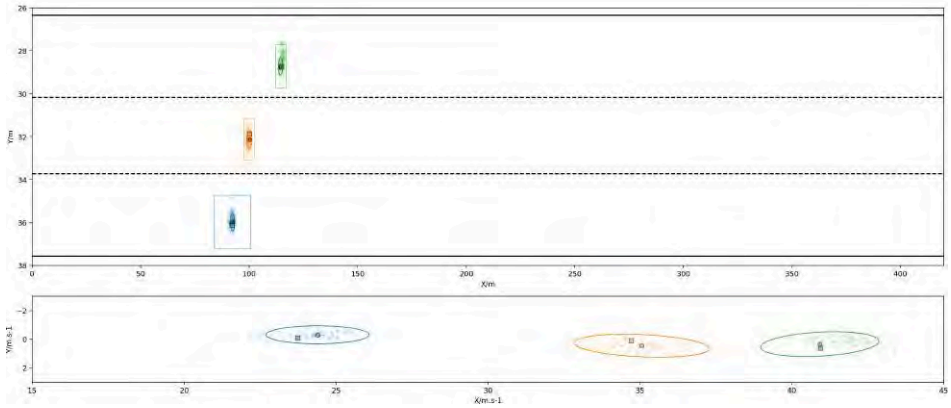


(b)

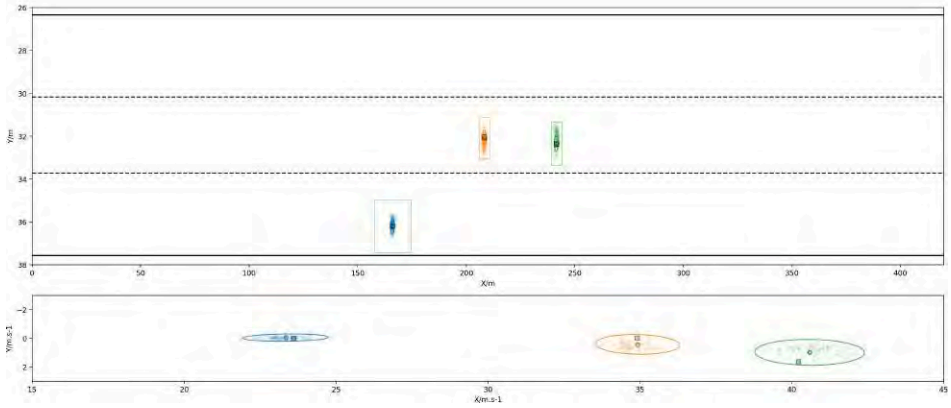


(c)

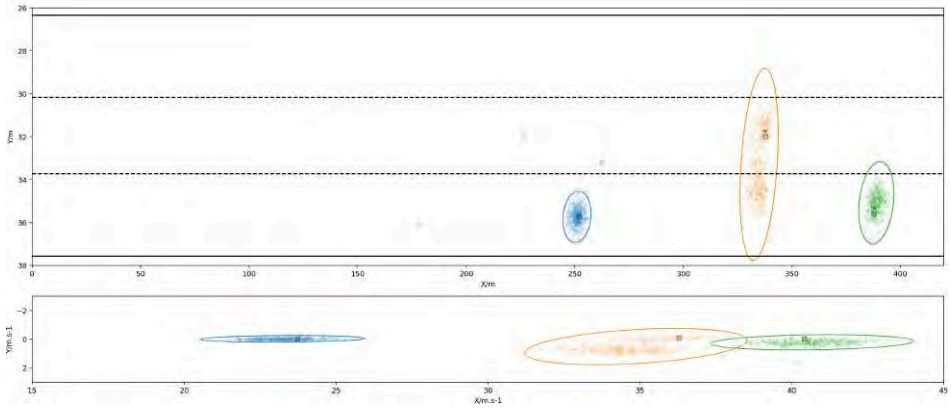
FIGURE D.3 – Scénario 8 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation de la flotte de véhicules est satisfaisante. Lors de ce scénario particulièrement complexe, les distributions prédites sont souvent multimodales.



(a)



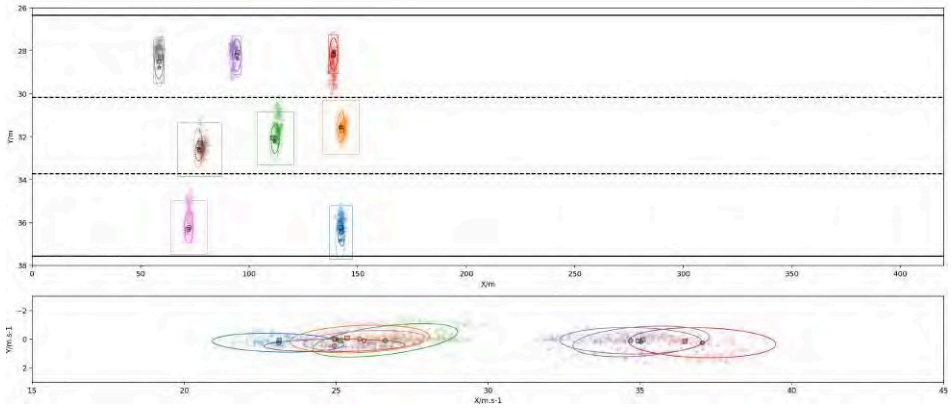
(b)



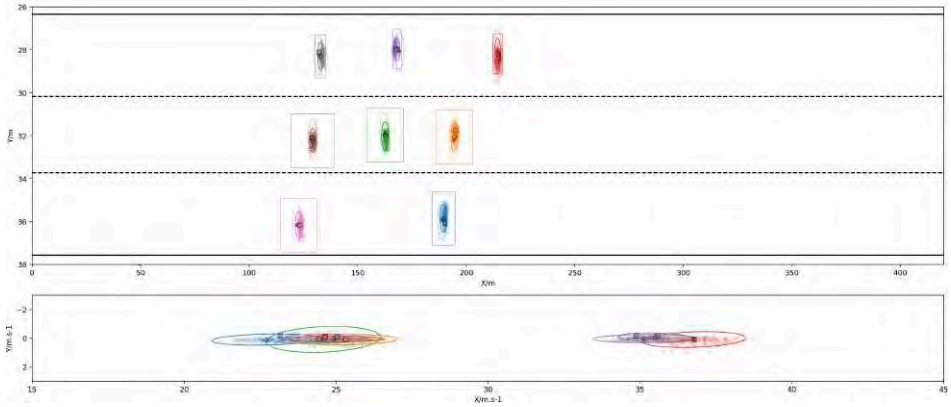
(c)

FIGURE D.4 – Scénario 12 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation et la prédictions sont correctes avec une prédiction multimodale cohérente pour le véhicule orange.

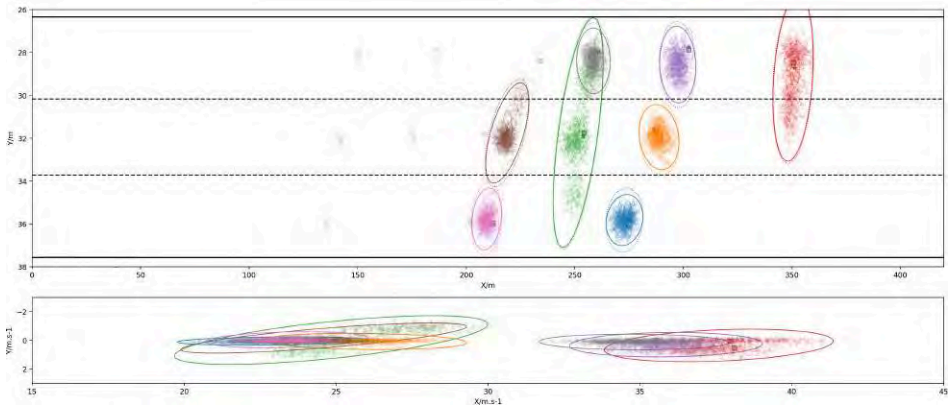
D.2 VBPF w/ GMM avec 250 particules



(a)

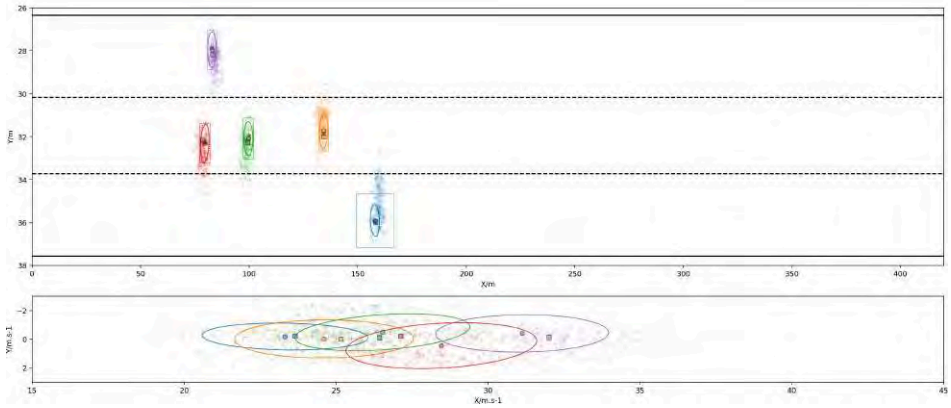


(b)

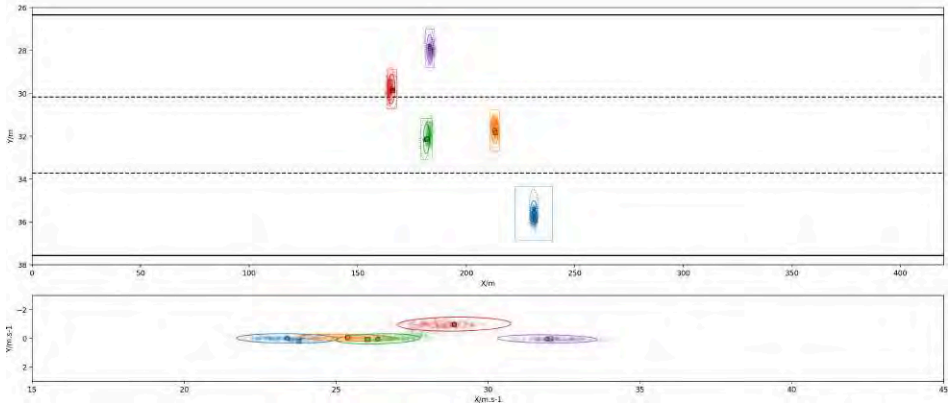


(c)

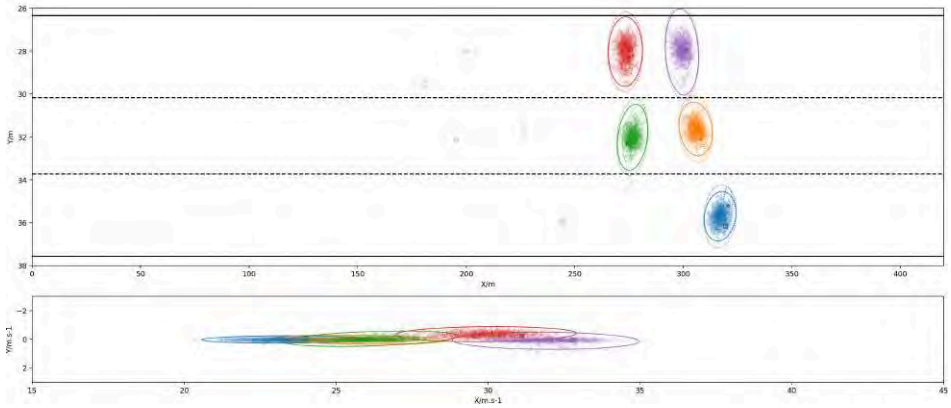
FIGURE D.5 – Scénario 1 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). L'estimation est satisfaisante. Dans l'ensemble, la prédiction de l'état de flotte est correcte. Cependant, la prédiction multimodale pour le véhicule vert ne semble pas correctement considérer une collision possible avec le véhicule gris.



(a)

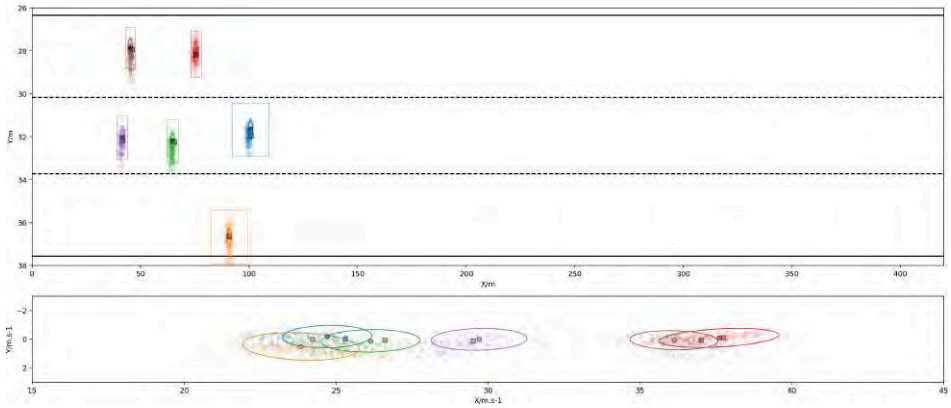


(b)

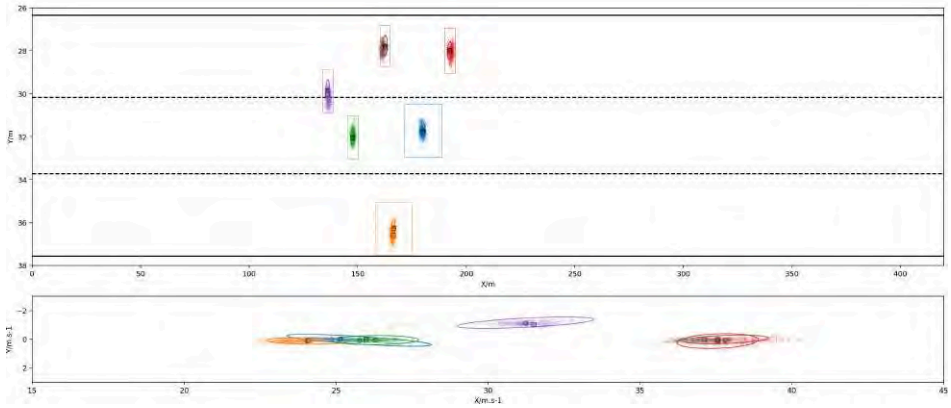


(c)

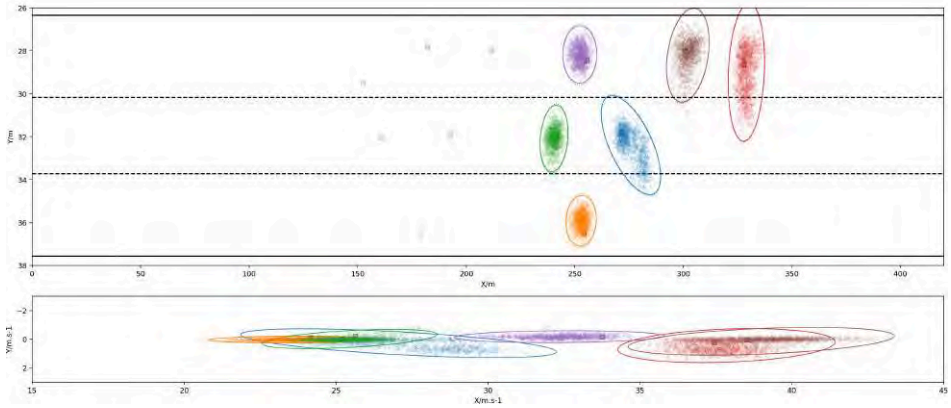
FIGURE D.6 – Scénario 7 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2}\times$ (la taille d'un véhicule standard). La prédiction et l'estimation sont satisfaisantes.



(a)

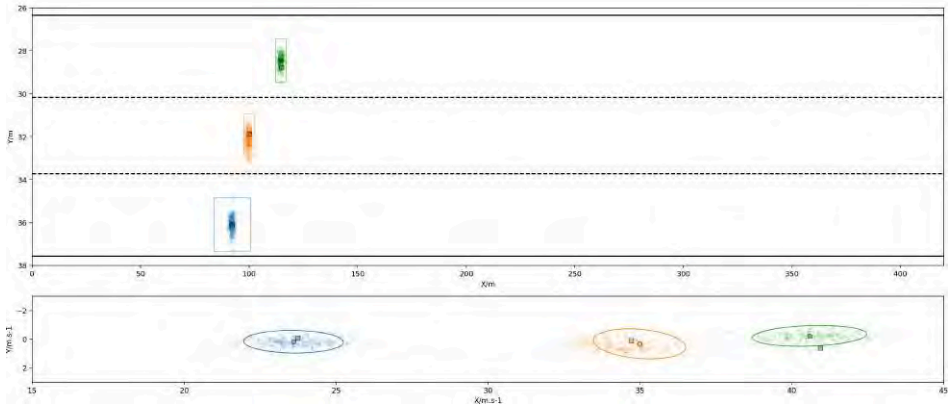


(b)

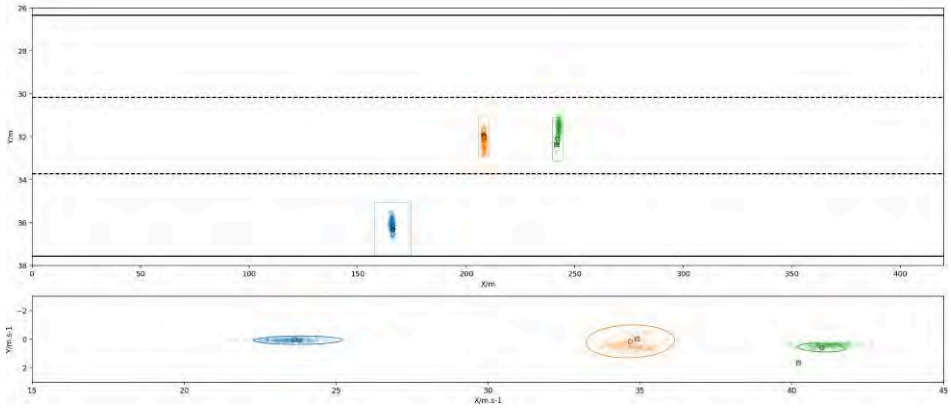


(c)

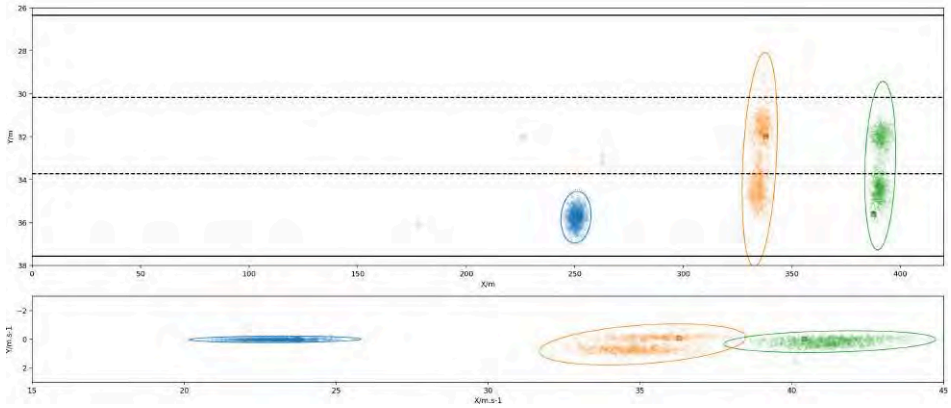
FIGURE D.7 – Scénario 8 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation de la flotte de véhicules est satisfaisante. Lors de ce scénario particulièrement complexe, les distributions prédites sont souvent multimodales.



(a)



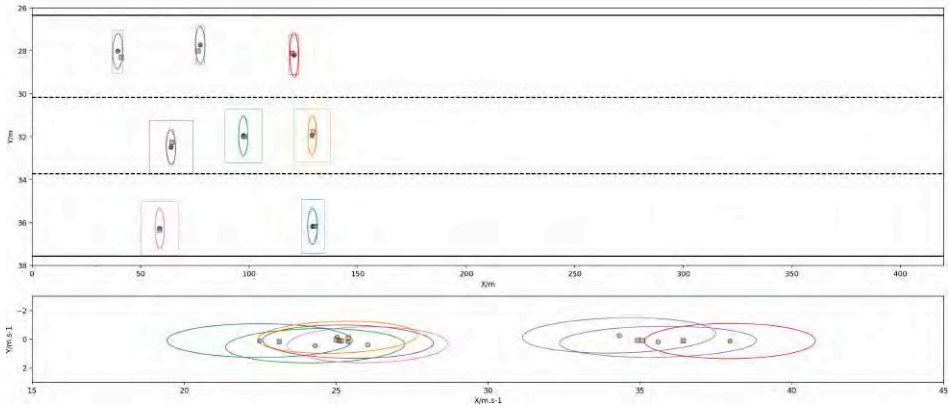
(b)



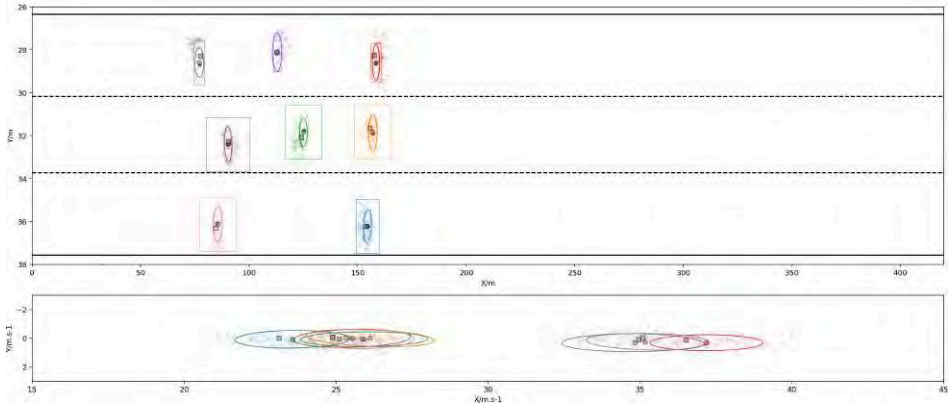
(c)

FIGURE D.8 – Scénario 12 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation et la prédictions sont correctes pour les véhicules bleu et orange avec une prédiction multimodale cohérente pour le véhicule orange. Les résultats pour le vert sont moyen avec la vérité terrain se trouvant dans la queue de la distribution de prédiction.

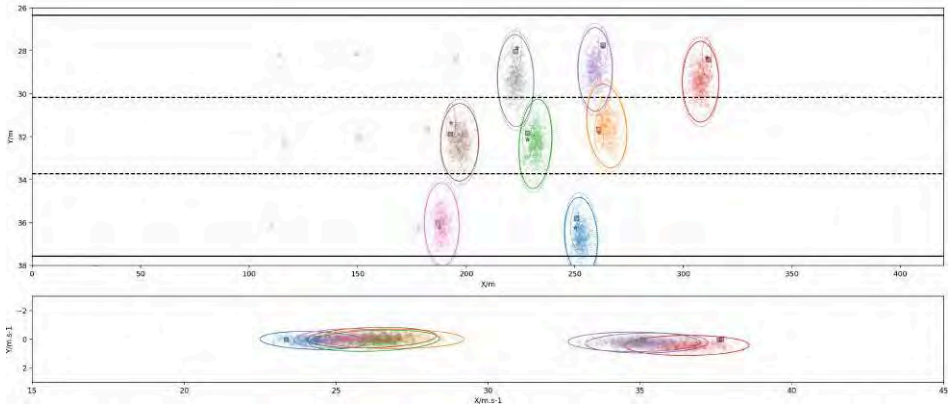
D.3 MPF w/ CAB-CVAE avec 100 particules



(a)

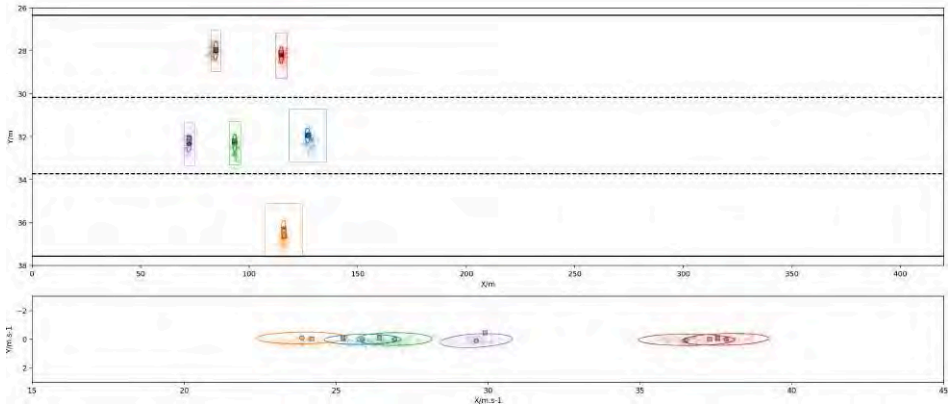


(b)

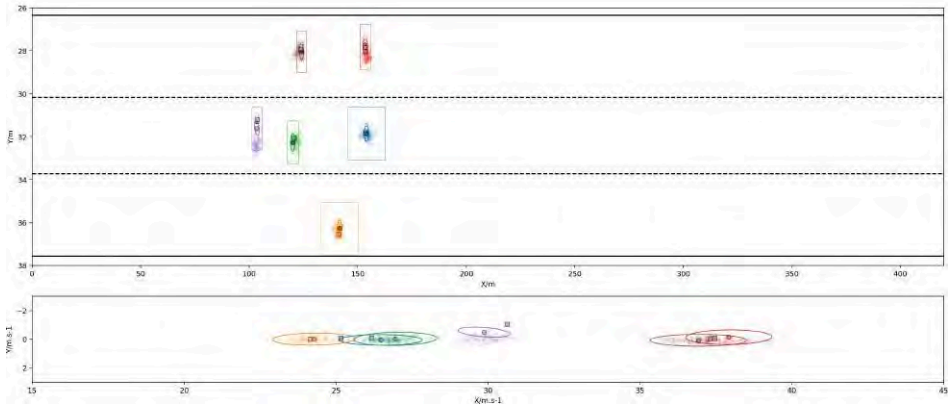


(c)

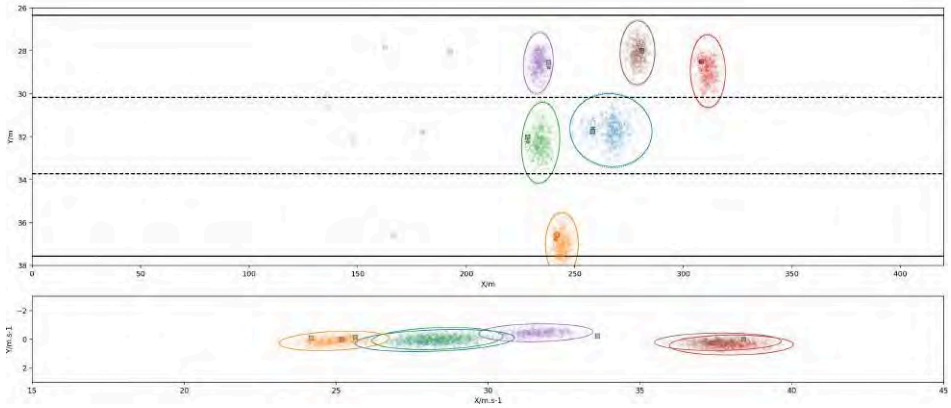
FIGURE D.9 – Scénario 1 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2} \times$ (la taille d'un véhicule standard). L'état de la flotte de véhicules avançant en ligne droite est correctement estimé et prédit.



(a)

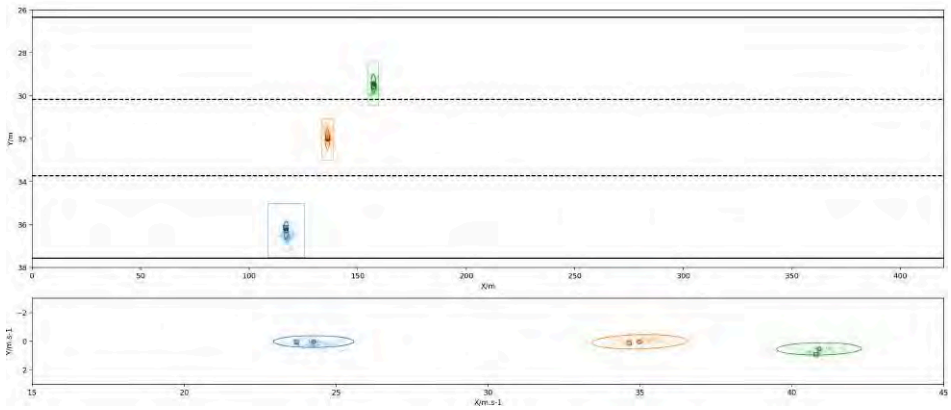


(b)

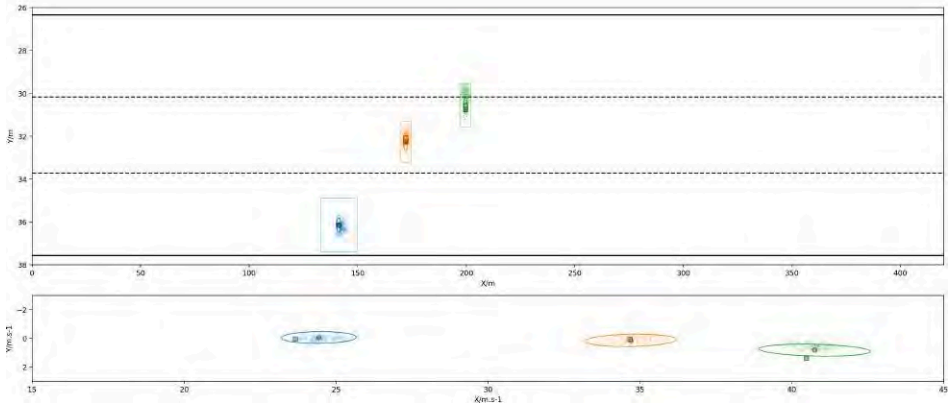


(c)

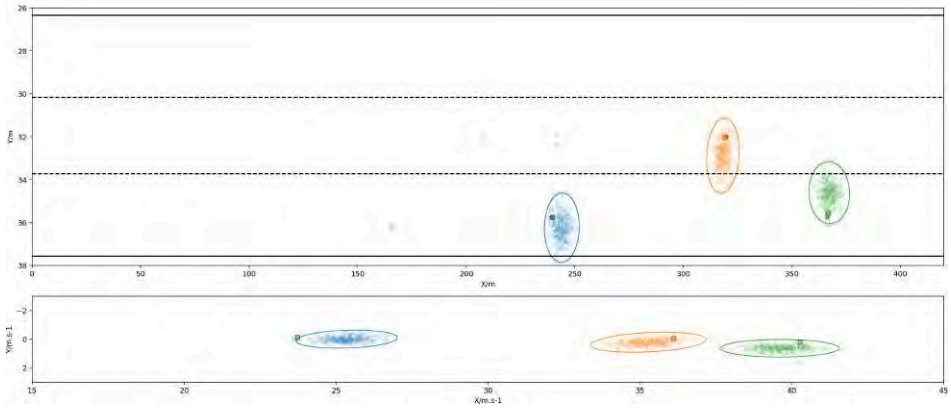
FIGURE D.10 – Scénario 8 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation de la flotte de véhicules est satisfaisante. Cependant, certaines vérités terrains se trouvent dans les queues des distributions prédites. La prédiction de l'état du véhicule bleu est trop conservative.



(a)



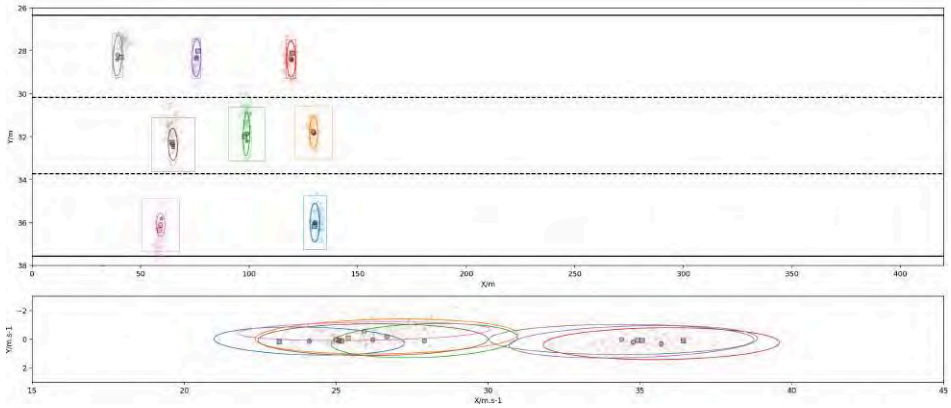
(b)



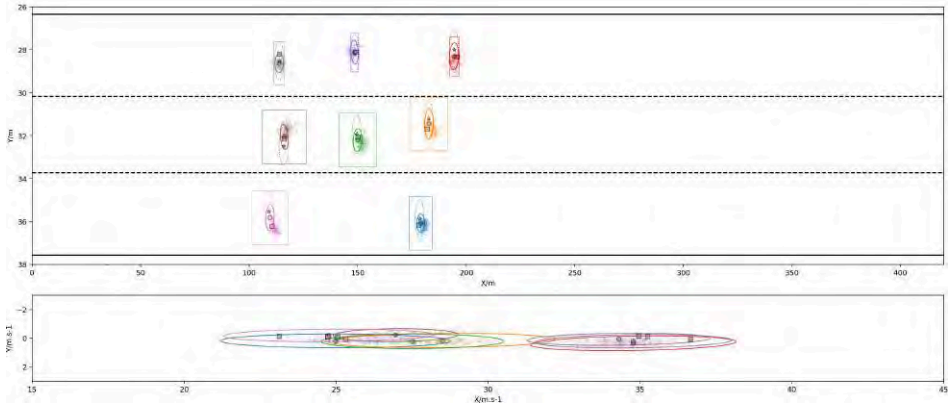
(c)

FIGURE D.11 – Scénario 12 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation et la prédiction sont satisfaisantes pour les trois véhicules.

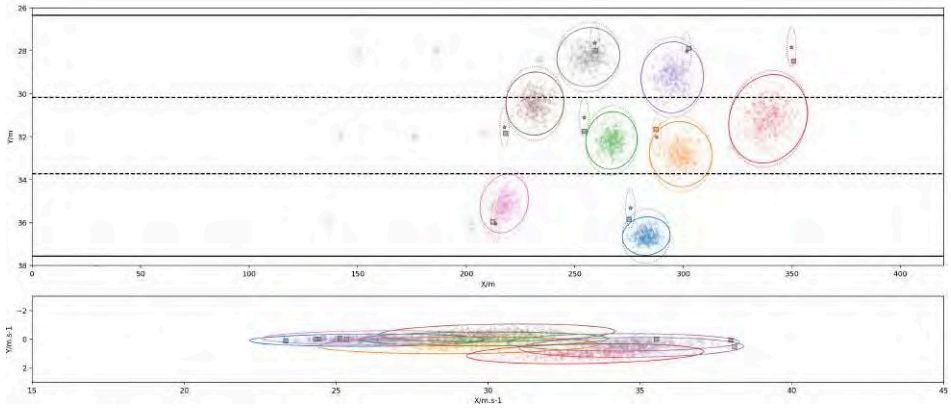
D.4 MPF w/ CVRAE avec 100 particules



(a)

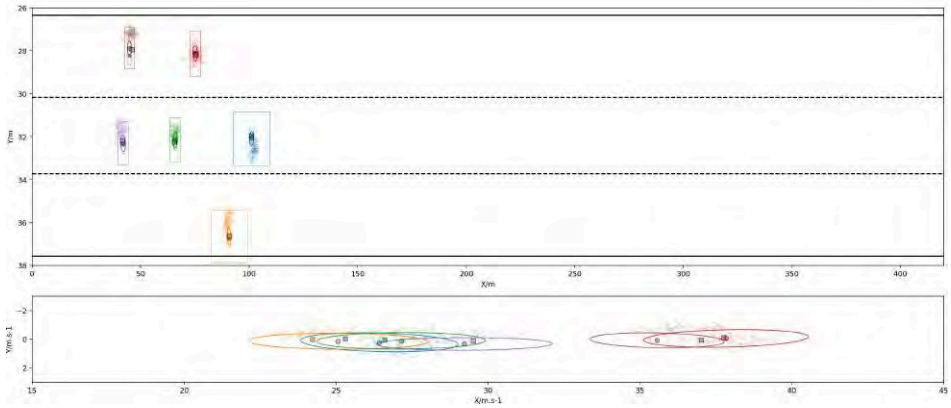


(b)

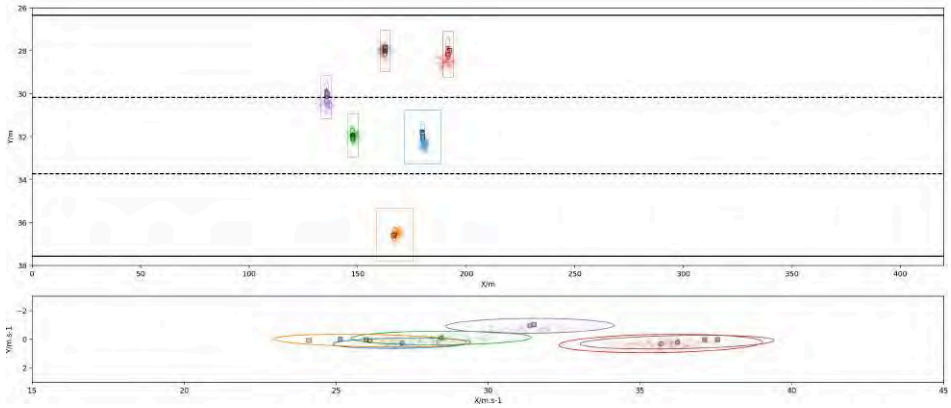


(c)

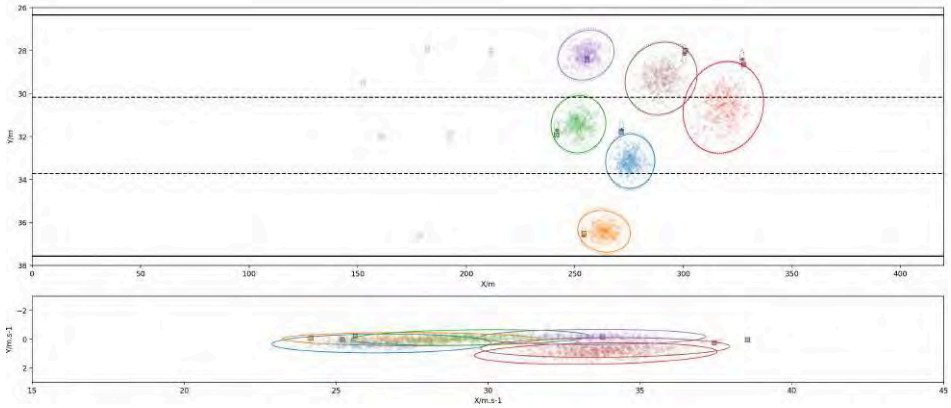
FIGURE D.12 – Scénario 1 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{2} \times$ (la taille d'un véhicule standard). Pour ce scénario relativement simple, bien que l'estimation soit correcte, la prédiction de la flotte de véhicule n'est pas satisfaisante. La vérité terrain est souvent hors des distributions produites.



(a)

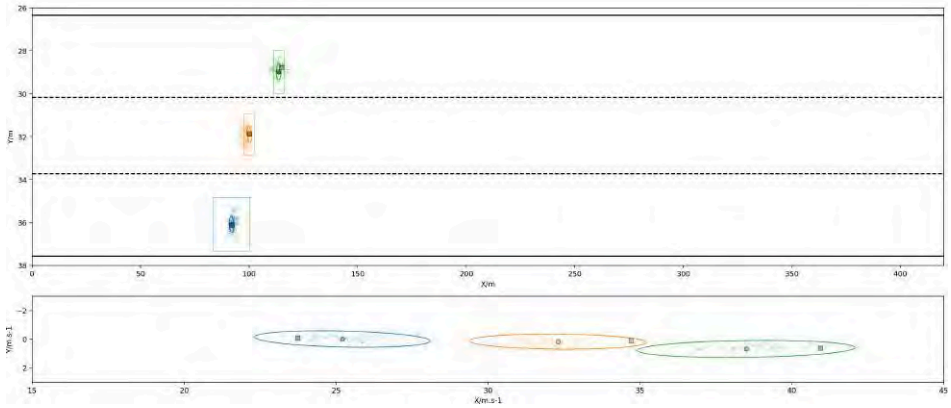


(b)

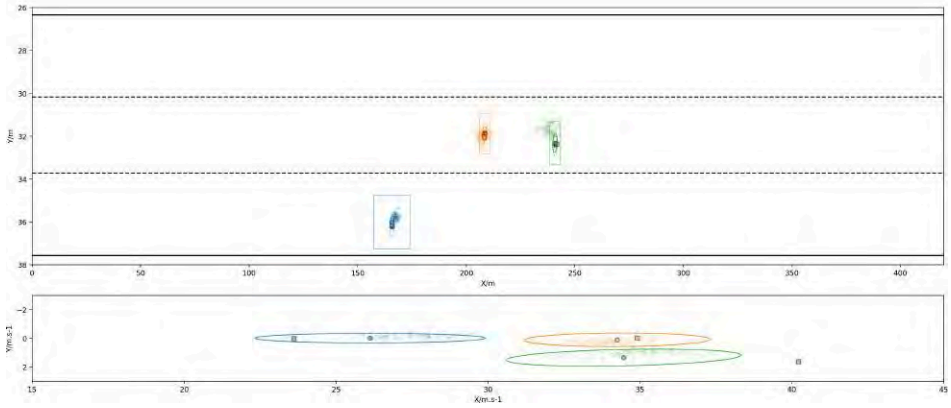


(c)

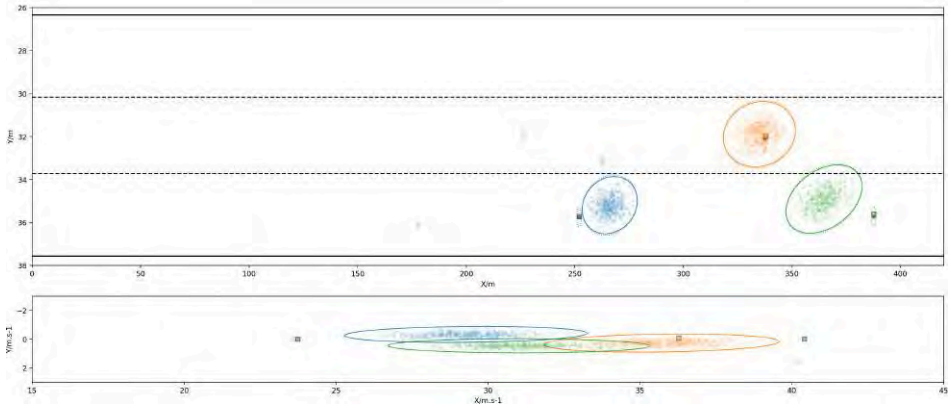
FIGURE D.13 – Scénario 8 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). L'estimation est correcte mais la prédiction de la flotte de véhicule n'est pas satisfaisante. La vérité terrain se trouve souvent dans les queues des distributions produites.



(a)



(b)



(c)

FIGURE D.14 – Scénario 12 – Intervalle de confiance à 99% de probabilité du bruit de mesure = $\pm\frac{1}{4}\times$ (la taille d'un véhicule standard). Ici, l'estimation est passable avec une vitesse mal estimée pour le véhicule vert. La prédiction est mauvaise.