Fusion de données multi-capteurs à l'aide d'un réseau bayésien pour l'estimation d'état d'un véhicule

Cherif Smaili

UFR mathématiques et informatique

Fusion de données multi-capteurs à l'aide d'un réseau bayésien pour l'estimation d'état d'un véhicule

THÈSE

présentée et soutenue publiquement le : 7 MAI 2010

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Cherif Smaili

Composition du jury

Rapporteurs : Philippe Leray, Professeur des Universités, Nantes.
Roland Chapuis, Professeur LASMEA, Clermont-Ferrand.

Examinateurs : Abder Koukam, Professeur, Belfort-Montbéliard.
Anne Boyer, Professeur des Universités, Nancy2.
Philippe Bonnifait, Professeur UTC, Compiègne.
Francois Charpillet, Directeur de recherche, INRIA Nancy.
Maan El Badaoui El Najjar, Maître de conférence, Université Lille1.

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Je dédie cette thèse à mes chers parents et à toute ma famille

Remerciements

Je remercie ILLAHI, sans son aide je ne serais pas là.

Je tiens à remercier les rapporteurs M. Roland Chapuis et M. Philippe Leray d'avoir accepté de rapporter mes travaux de thèse. Mes remerciements vont également aux autres membres du Jury : Mme Anne Boyer, M. Philippe Bonnifait et M. Abder Koukam.
Un remerciement particulier à M. Phillipe Bonnifait et le laboratoire HeuDiaSyc pour avoir mis à notre disposition les données réelles utilisées dans le cadre de cette thèse. Un grand merci à Anne Boyer pour sa patience et sa disponibilité dont elle a su faire preuve pour mon DRT et qui me fait l'honneur de participer à mon jury de thèse.

Avec une grande reconnaissance, je remercie mon directeur de thèse François Charpillet. Son unique optimisme m'a entièrement transformé, soutenu et réconforté durant les moments les plus difficiles de ces années. Un point très important est qu'en travaillant avec François on ne sent en aucun moment, qu'il est le Big Boss de l'équipe MAIA. En un mot : un grand merci à toi François.

Je voudrais remercier mon co-encadrant Maan El Badaoui El Najjar. Je le remercie pour sa disponibilité, sa rigueur et notamment son savoir faire. Son enthousiasme, sa positivité, sa détermination et sa patience m'ont donné le goût de la recherche. Bien que les 450 kilomètres qui nous séparent (Lille-Nancy), je me sentais toutefois aussi près de Lille que de Nancy. Les multiples voyages et les longues conversations téléphoniques ont atténué cette longue distance. Au-delà de ses qualités scientifiques, ce sont surtout ses qualités humaines que je voudrais souligner. Le proverbe italien dit : trouver un ami, trouver un trésor. En un mot : merci à toi mon ami Maan.

Je dois un grand merci à Cédric Rose pour nos longues discussions, sa disponibilité malgré ses projets et surtout sa connaissance encyclopédique sur les réseaux Bayésiens. Ses qualités humaines sont à souligner. Merci Cédric ou comme j'aime bien t'appelé : Murphy de France et bon courage pour ta vie professionnelle.

Je tiens à exprimer mes sincères remerciements à mon frère Kamel Smaili qui m'a encouragé de près et de loin tout au long de mon DRT et ma thèse et qui est la cause de ma présence en France.
Avant d'oublier, un grand merci pour mes amis d'enfance et camarades de classe qui n'ont jamais hésité à me téléphoner pour prendre de mes nouvelles.
Un grand merci à mes deux coéquipiers de karaté Frank et François qui ont pu lire cette thèse une dernière fois. Je garderai un formidable souvenir de nos séanes du Lundi et du Mercredi.
Enfin et surtout, je ne saurais terminer ces quelques lignes sans une pensée pour mes parents,

sans eux je n'aurais jamais pu arriver à ce stade. Sans oublier mes frères et soeur : Kamel, Youcef, Lila, Samir et Fouad et ma future femme.

Table des matières

Table des figures

Liste des tableaux

Introduction

1 Contexte du travail

La voiture reste un moyen de transport incontournable dans la vie de tous les jours. Grâce à ses caractéristiques séduisantes (forme, couleur, confort, vitesse, facilité d'usage...), elle constitue un véritable objet de désir et suscite un fort engouement du grand public qui voit en elle un moyen de transport indispensable.

Malheureusement, dans les grands centres urbains la voiture pose de nombreux problèmes tant en termes de pollution atmosphérique, sonore, visuelle, que d'encombrement de l'espace. C'est pourquoi à travers le monde de nombreux programmes de recherches visent à mettre au point de nouveaux modes de transport publics, pour compléter ou interconnecter les systèmes traditionnels plus lourds comme le metro, le tramway ou encore le trolleybus.

Cette thèse s'inscrit dans cette problématique au sein du projet CRISTAL (Cellule de Recherche Industrielle en Systèmes de Transports Automatisés Légers) du pôle de compétitivité Alsace Franche-Comté "véhicule du futur". Ce projet réunit des partenaires industriels (LOHR, TRANSITEC, GEA, VULOG, et Technomad) ainsi que des partenaires universitaires (LASMEA, UTBM et INRIA). Il a pour objectif le développement d'un nouveau système de transport public « bi-mode » individuel (mode libre-service) ou collectif (mode navette). En quelques mots, l'idée est de mettre en place au sein des centres urbains des véhicules en libre service qui soient disponibles dans un ensemble de stations réparties dans une zone géographique restreinte.

Pour être efficace, un tel système de transport doit disposer d'une infrastructure de gestion de flotte qui permette à la fois de localiser à tout moment chaque véhicule et d'assurer une adéquation entre l'offre de véhicules et la demande. En ce qui concerne ce second point, Il faut être capable de convoyer dans les zones à forte demande les véhicules qui sont répartis dans des zones à plus faible demande. Pour cela une solution est de collecter les véhicules grâce à un système de ramassage. Nous imaginons qu'un pilote pourrait conduire un véhicule de tête, les véhicules collectés suivraient en mode automatique et en file indienne ce véhicule pour former un train sans accroche matérielle. La navigation en convoi du point de vue des véhicules suiveurs constitue un cas particulier et plus simple de conduite automatisée. Il s'agit pour ces véhicules de suivre un chemin de référence (celui du véhicule de tête) en respectant un écart de sécurité entre les véhicules. Afin de réaliser cette tâche, chaque véhicule du convoi doit être en mesure de se localiser de manière absolue par rapport au chemin de référence et d'une manière relative entre eux.

La localisation constitue donc une brique technologique essentielle du projet CRISTAL. Elle intervient a deux niveaux : la gestion de flotte et l'accroche immatérielle. Cette thèse a pour

objectif de contribuer à ce domaine sous l'angle de l'intelligence artificielle et en particulier, comme nous le verrons, dans le cadre des réseaux bayésiens dynamiques.

2 Problématique scientifique de la localisation

Localiser de manière précise un véhicule ou un robot semble être aujourd'hui une question facile alors que de plus en plus d'automobilistes utilisent quotidiennement un GPS. Pourtant, il s'agit d'une tâche difficile dès lors que l'on cherche à se localiser de manière précise, sûre, intègre et continue. En effet, le positionnement GPS est souvent entaché d'erreurs soit parce que des satellites sont masqués, ce qui est fréquent dans les centres urbains, soit parce que de nombreuses réflexions perturbent l'estimation de la position (effets de multitrajets des ondes des signaux GPS)... La précision peut être grandement améliorée si on utilise des informations cartographiques qui permettent par exemple de contraindre les positions possibles aux seuls segments correspondants à des voies de circulation autorisées [El Najjar, 2003], [Jabbour, 2007]. Cependant, les coordonnées des segments de route sur une carte numérique sont également entachées d'erreurs. Le réseau routier des bases de données cartographiques n'est pas toujours en parfait accord avec la réalité. Il peut contenir des tronçons de routes qui n'existent plus, ou bien au contraire de nouveaux tronçons peuvent ne pas être encore référencés dans la base de données routières. De même, les cartes ne contiennent pas tous les détails, certains sont même approximés grossièrement. Par exemple, un rond-point peut être représenté par un point sur la carte.

Pour pallier ces difficultés, d'autres sources d'informations peuvent être utilisées pour affiner la localisation. Il est envisageable par exemple, d'utiliser des capteurs gyroscopiques qui donneront une information sur le cap du véhicule ou encore un télémètre laser qui donnera une information de distance par rapport à un obstacle dont on connaît la position (par exemple un bâtiment, ou autre...).

La question de la localisation qui est posée dans cette thèse est abordée comme un problème de fusion de sources d'informations de natures variées qu'elles soient symboliques comme les informations cartographiques 2D ou numériques comme les informations délivrées par des capteurs (GPS, Télémètre, Gyroscope,...). Ce problème de la fusion multisources relève d'une question plus générale qui est l'estimation des variables cachées (la position et le cap du véhicule,...) d'un système dynamique (le véhicule, un train de véhicules) étant donné un historique d'observations (les capteurs). La difficulté tient à ce que les modèles dont on dispose ne sont pas toujours conformes à la réalité. Par ailleurs, les variables mesurées pour les raisons énoncées ci-dessus ne sont pas précises. Elles peuvent même être manquantes par intermittence ou non cohérentes temporellement. C'est pourquoi, nous avons choisi dans cette thèse une approche fondée sur la modélisation probabiliste en raison de sa robustesse à l'incertitude tant des modèles que des données mesurées.

Parmi les approches probabilistes, nous avons choisi le cadre des réseaux bayésiens parfois appelés aussi modèles graphiques. Les probabilités permettent à ces modèles de prendre en compte l'aspect incertain présent dans les applications réelles. La partie graphique offre un outil intuitif et attractif dans de nombreux domaines d'applications où les utilisateurs ont besoin de "comprendre" le modèle qu'ils utilisent.

3 Contributions

Dans un premier temps, nous nous sommes intéressés à la fusion de données pour la localisation de véhicules routiers. Le filtrage de Kalman et ses variantes ont été largement étudiés dans la communauté automaticienne pour aborder cette problématique. En intelligence artificielle d'autres outils comme les réseaux bayésiens dynamiques (DBN pour *Dynamique Bayesian Networks*) permettent de construire des filtres de Kalman de façon très générique sous forme de modèles graphiques [Murphy, 2002]. Un intérêt des DBN que nous avons montré dans cette thèse est la simplicité avec laquelle on peut étendre les filtres de Kalman. Nous avons ainsi proposé différentes extensions permettant de gérer des hypothèses multiples ou des informations de nature mixtes symboliques et numériques [Smaili *et al.*, 2008b].

Nous nous sommes également intéressé à la modélisation de systèmes dynamiques non-linéaires et c'est notre seconde contribution. Ce point est particulièrement important car la plupart des modèles réels sont non-linéaires et en particulier les modèles représentant la cinématique ou la dynamique de robots. Le problème de la non-linéarité a été traité dans la littérature sous différents aspects. Une première approche consiste à linéariser localement le système autour de l'estimée courante et d'y appliquer un filtre de Kalman classique. C'est ce qu'on appelle le filtrage de Kalman étendu. Les filtres particulaires constituent une autre approche. Ces filtres proposent de représenter la loi conditionnelle de l'état par un nombre fini de masses pondérées.
Nous avons étudié de manière critique ces deux approches et nous avons proposé deux contributions pour traiter le problème de la non-linéarité dans les réseaux bayésiens dynamiques. La première, inspirée des filtres de Kalman étendus, effectue une linéarisation autour de l'estimée courante. La seconde est une approche par linéarisation exacte. Elle consiste à rechercher une transformation exacte (c'est-à-dire une transformation d'état et de commande inversible) permettant de réécrire le système non linéaire dans un autre espace d'état dans lequel le système considéré est linéaire (sous certaines conditions données par [Murray, 1993]). L'approche développée est connue sous le nom de *modèle chaîné*.

La troisième contribution de ce travail de thèse, s'inscrit dans le domaine de la modélisation et la localisation d'un train dont chaque véhicule suit un chemin de référence donné par le véhicule de tête tout en respectant un écart prédéfini entre les véhicules. Le modèle bayésien que nous proposons s'obtient tout simplement par construction à partir d'une part du modèle développé pour un véhicule seul et d'autre part des liens indiquant les interactions entre véhicules [Smaili *et al.*, 2008a].

Les données manipulées dans cette thèse, pour valider les approches développées, sont des données réelles. Dans une première partie (la localisation d'un véhicule sur une carte), nous utilisons les données réelles d'un essai effectué à Compiègne en France avec le véhicule expérimental STRADA de l'HeuDiaSyc. Ces données ont été mises à notre disposition par Philippe Bonnifait. Dans la seconde partie (la modélisation et la localisation d'un convoi de véhicules), nous utilisons les données réelles d'un essai effectué sur la place Stanislas à Nancy avec le CyCab de l'équipe MAIA du LORIA. Ces données ont été utilisées pour valider la fusion de données pour la localisation du véhicule de tête, les données capteurs des véhicules suiveurs ont été simulées.

4 Organisation du manuscrit

Ce manuscrit est organisé en quatre chapitres. Le chapitre 1 dresse un état de l'art des capteurs et méthodes permettant de traiter le problème de la localisation de véhicules routiers. Les méthodes de localisation sont réparties en deux grandes parties : la localisation sur une carte ou ce qu'on appelle en anglais le "map-matching" ou "road-matching" et la localisation sans utilisation de carte. Les approches les plus utilisées pour la localisation d'un véhicule sur une carte sont exposées et comparées afin de révéler les avantages et les inconvénients de chacune d'elles.

Le chapitre 2 se focalise sur les modèles graphiques et plus particulièrement sur les réseaux bayésiens. Cet outil théorique est utilisé dans les travaux de cette thèse dans le but de modéliser l'incertitude, fusionner les données issues de capteurs hétérogènes et finalement calculer l'estimation de l'état d'un véhicule. Une synthèse sur les réseaux bayésiens à variables discrètes et continues est donnée, ainsi qu'un bref aperçu sur les réseaux hybrides. Une grande part de ce chapitre est consacrée à l'un des algorithmes d'inférence les plus utilisé. Cet algorithme est connu sous le nom de JLO. Dans cette approche, l'inférence est exacte et repose sur une transformation du réseau initial en un arbre de jonction.

Le chapitre 3 détaille notre approche de la localisation fondée sur les réseaux bayésiens. Le problème de la localisation d'un véhicule sur une carte est traité dans ce chapitre comme l'estimation de la position d'un robot étant données les mesures bruitées fournies par les capteurs. Les différents capteurs embarqués sur le véhicule fournissent des informations incertaines et incomplètes. La complémentarité et la redondance de ces informations sont alors deux facteurs essentiels permettant d'accéder à une information globale plus fiable et plus complète. Ainsi, la fusion de données par le réseau bayésien est utilisée dans le but d'exploiter au mieux les avantages de chacune des sources d'informations, tout en essayant de pallier leurs limitations individuelles.
Le réseau bayésien n'est pas seulement utilisé pour la fusion de données mais aussi dans le but de fournir des estimations de plus en plus précises. Ainsi, en présence de plusieurs segments (à l'approche d'une intersection, deux routes rapprochées ou dans un carrefour), le réseau bayésien gère tous les segments candidats jusqu'à ce que la situation devienne non ambiguë. Cette façon d'agir donne plus de confiance sur la localisation par rapport aux méthodes de sélection du segment le plus probable.
La seconde partie de ce chapitre concerne la modélisation d'un convoi de véhicules par un réseau bayésien. Le système de localisation de chaque véhicule du convoi peut être vu comme une extension de la localisation mono véhicule. Le réseau bayésien utilisé pour la localisation d'un véhicule sur une carte (map-matching) est dupliqué pour l'ensemble du convoi, en ajoutant des inter-connexions représentant les interactions inter véhicules.

La fin de ce chapitre est consacrée à la présentation d'une méthode de linéarisation fondée sur les modèles chaînés. L'approche par linéarisation exacte, consiste à rechercher une transformation exacte permettant de réécrire le système non linéaire comme un système linéaire de façon a pouvoir exploiter l'ensemble des outils de l'automatique linéaire pour construire et régler les lois de commandes.

Le chapitre 4 est dédié à la présentation des expérimentations et des résultats obtenus sur des données réelles. Ce chapitre est composé de trois parties :

- La première partie présente et commente les expériences sur la localisation d'un véhicule sur une carte. Les données utilisées sont enregistrées avec un véhicule expérimental de type STRADA. Le parcours d'essais sur Compiègne a été choisi de telle sorte qu'il contienne des situations problématiques pour le système de localisation. Ce parcours présente en effet des situations générant des ambiguïtés sur la position du véhicule (routes rapprochées, intersections et carrefours).
- La seconde partie de ce chapitre est dédiée à la localisation d'un convoi de véhicules. Les données utilisées pour le véhicule de tête sont enregistrées avec un véhicule expérimental de type Cycab conçu par la société Robosoft. Le parcours d'essais est effectué sur la place Stanislas à Nancy. Le Cycab est équipé d'un GPS centimétrique de type GPS-RTK et d'un gyroscope optique. Les autres véhicules du convoi sont simulés. Chaque véhicule est censé être équipé d'un GPS métrique (simulé par l'addition de bruit gaussien aux positions GPS-RTK) ressemblant à ceux trouvés dans nos voitures et d'un télémètre pour estimer la distance avec le véhicule prédécesseur.
- La troisième partie de ce chapitre concerne l'évaluation des modèles chaînés. Les résultats sont comparés avec ceux obtenus par un filtre de Kalman étendu.

Enfin, une conclusion générale et des perspectives finalisent le document.

Chapitre 1

La localisation d'un véhicule terrestre

1.1 Introduction

Localiser un objet, une personne, une voiture ou un robot consiste à déterminer sa position dans un repère donné. Nous nous intéressons dans cette thèse à la localisation d'un objet ou d'un groupe d'objets mobiles en fusionnant différentes sources d'informations issues de capteurs proprioceptifs ou extéroceptifs ou encore issues d'une base de données cartographique. Bien que la localisation apparaisse simple dans sa formulation, elle a nécessité de nombreuses années de recherche. Parmi les facteurs qui rendent cette tâche difficile, il y a l'incertitude des sources d'informations (voir la description des capteurs dans 1.2).

La localisation est un domaine de recherche vaste. Les solutions envisagées varient selon :

- le type de capteurs utilisés
- l'environnement dans lequel évolue l'objet à localiser
- les caractéristiques dynamiques ou cinématiques de l'objet à localiser
- le repère de référence dans lequel on veut localiser l'objet
- ...

De ce fait, on peut distinguer deux grands axes de recherche. La localisation en environnement clos ou "indoor localization" et la localisation en environnement extérieur ou "outdoor localization".
On peut même classifier les recherches selon le type de capteurs utilisés et les méthodes mises en œuvres. Ainsi, on distingue les méthodes de localisation relative (utilisation de capteurs proprioceptifs), localisation absolue (se rapporte à l'utilisation de capteurs extéroceptifs) et finalement la localisation hybride correspondant à l'utilisation conjointe des deux types de capteurs. Notons que tout au long de ce document :

- nous désignons par le mot robot mobile à la fois un véhicule et un robot.
- nous considérons le cas de la localisation dans les milieux extérieurs.
- nous utilisons à la fois les capteurs proprioceptifs et extéroceptifs (localisation hybride).

Après cette présentation de la problématique, donnons dans ce qui suit une description des capteurs les plus utilisés dans le domaine de la localisation.

1.2 Les capteurs

La question de la perception en robotique fait référence à la capacité d'un robot à recueillir et à traiter les informations parvenant de plusieurs capteurs. Le choix des capteurs dépend évidemment de l'application envisagée.

1.2.1 Le GPS

Le GPS ou *Global Positioning System* est un capteur extéroceptif. Il constitue le système par excellence pour connaître la position d'un véhicule dans un repère global. Le GPS est constitué d'un ensemble de satellites mis en place par le département de la défense des États Unis d'Amérique permettant à un récepteur d'acquérir en temps réel, à la fois sa position, sa vitesse de déplacement et un temps de référence précis. La constellation satellitaire complète est, en principe, constituée de 28 satellites placés sur orbite à 20 200 km d'altitude. Leur répartition sur 6 orbites différentes a été étudiée afin de couvrir l'ensemble de la surface terrestre de façon optimale [El Najjar, 2003].
Le principe de fonctionnement du GPS est le suivant : chaque satellite émet un signal constitué de deux codes pseudo aléatoires à savoir le C/A-code (code civil), et le P-code (code précis), plus un code d'informations contenant tous les renseignements concernant l'état des satellites (position, paramètres d'horloges,...).
Le calcul de la position repose sur le principe de la triangulation (voir la figure 1.1). Le récepteur calcule le temps mis par l'onde émise par le satellite. La vitesse de propagation du signal étant connue, le récepteur détermine une sphère sur laquelle est située sa position (distance = temps × vitesse). Avec un deuxième satellite, l'intersection des deux sphères forment un cercle. Puis avec un troisième satellite, un ou deux points. Cependant, le récepteur n'a pas l'heure exacte, le calcul de la position comporte donc une inconnue de temps qui ne peut être résolue que par la donnée d'un quatrième satellite. Autrement dit, avec trois satellites, le récepteur déduit sa position relative par rapport à ces satellites, mais il ne sait pas où ils sont. Il ne peut donc savoir où il se situe lui-même.
En pratique, le récepteur utilise au moins 4 satellites, car le problème réel comporte en plus de ces 4 inconnues, plusieurs corrections. Évidemment, plus le nombre de satellites utilisé est grand, meilleure est la précision.

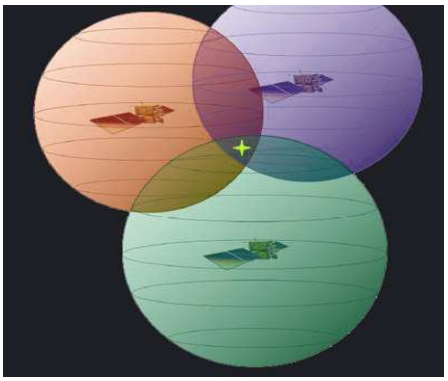


FIG. 1.1 – Principe de la triangulation utilisé par le GPS pour l'estimation de la position d'un véhicule.

En termes de précision, la localisation ainsi obtenue n'est pas très précise en ce qui concerne les GPS grand public (précision métrique). Pour obtenir des résultats plus satisfaisants, on a recours à une méthode différentielle. Cette méthode est connue sous le nom GPS différentiel ou DGPS. Un GPS classique est monté dans un local à terre. La position de son antenne est parfaitement connue à quelques centimètres près (voir la figure 1.2). Ce GPS écoute en permanence tous les satellites visibles dans sa zone. Il analyse le signal de chacun et détermine le retard variable provoqué entre autre par la traversée des basses couches de l'atmosphère. De cette manière on peut obtenir une précision meilleure.

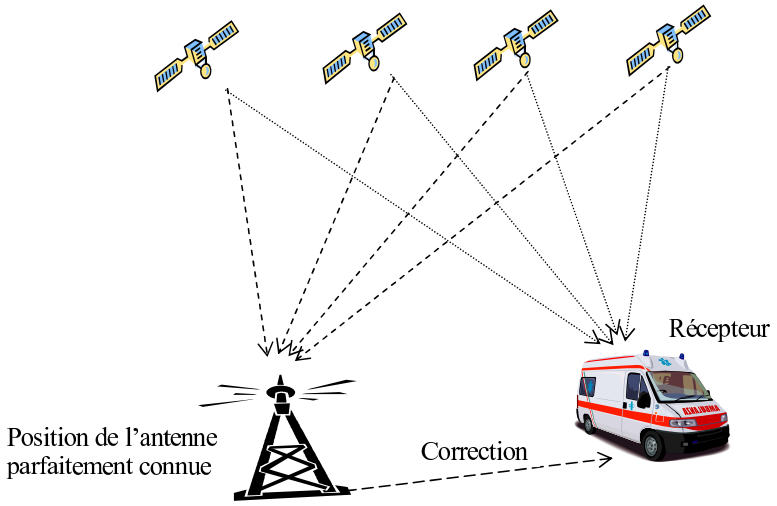


FIG. 1.2 – Calcul de position en mode différentiel DGPS.

1.2.2 Les codeurs incrémentaux

L'odométrie est une technique permettant d'estimer la pose d'un véhicule en mouvement. Cette technique repose sur la mesure individuelle des déplacements des roues pour reconstituer le mouvement global du robot. En partant d'une position initiale connue et en intégrant les déplacements mesurés, on peut ainsi calculer à chaque instant la position courante du véhicule. Entre deux instants d'échantillonnage k et $k+1$, on mesure les distances élémentaires parcourues, notées δ_{d_k} et δ_{g_k} des roues arrières droite et gauche. Dans ce cas, les deux roues n'ont pas forcément le même rayon car il s'agit de pneus qui peuvent être gonflés différemment. La distance séparant les deux points de contacts des roues avec le sol, sera notée e. Le centre de l'essieu M de la figure 1.3 est le centre du repère R_M lié au véhicule que l'on cherche à localiser. Ce point se situe sur la droite reliant les points de contact des roues avec le sol. S'il n'y a pas de glissement, la vitesse du point M est perpendiculaire à l'axe reliant les deux roues, et portée par l'axe (M, x_r). Ainsi au déplacement du point M, la translation élémentaire Δ_k et la rotation élémentaire ω_k du point M sont données par :

$$\begin{cases} \triangle_k = \frac{\delta_{d_k}+\delta_{g_k}}{2} \\ \omega_k = \frac{\delta_{d_k}-\delta_{g_k}}{2e} \end{cases} \tag{1.1}$$

Si on divise ces mesures par la période d'échantillonnage T_e (T_e tend vers zéro), on obtient les vitesses linéaires et angulaires du véhicule.
Rappelons que le but de l'odométrie est de calculer la position du véhicule par rapport à une

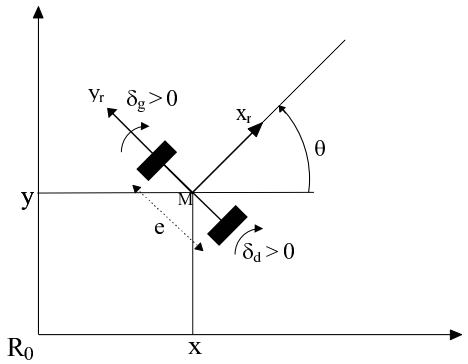


FIG. 1.3 – Modélisation d'un véhicule en 2D

position ou repère connu. Ce qui est équivalent à calculer de manière récurrente la position du robot à l'instant $k+1$ en fonction de la position à l'instant k et des déplacements élémentaires mesurés. Notons M_k et M_{k+1} deux positions successives du véhicule. Sous l'hypothèse d'un mouvement circulaire, on a (voir la figure 1.4) :

$$\triangle = \rho\omega \tag{1.2}$$

où Δ est la distance parcourue par le véhicule le long de l'arc M_kM_{k+1}, ρ représente le rayon de courbure et ω est la rotation élémentaire. Sur la même figure 1.4 on peut faire l'approximation suivante :

$$\|M_kH\| = \|HM_{k+1}\| \approx \rho\sin(\omega/2) \tag{1.3}$$

De l'équation 1.2 on a $\rho = \Delta/\omega$ et par conséquent :

$$\|M_kM_{k+1}\| \approx 2\rho\sin(\omega/2) = 2\frac{\triangle}{\omega}\sin(\omega/2) = \triangle\frac{\sin(\omega/2)}{\omega/2} \tag{1.4}$$

Les variations Δx et Δy sont définies par le vecteur $\overrightarrow{M_kM_{k+1}}$ dont l'angle avec l'horizontale est donné par :

$$(\overrightarrow{M_kM_{k+1}}, \overrightarrow{X_0}) = \theta_k + \omega/2 \tag{1.5}$$

Ainsi on obtient [El Najjar, 2003] :

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + \frac{sin(\omega_k/2)}{\omega_k/2}\Delta_k\cos(\theta_k + \omega_k/2) \\ y_{k+1} = y_k + \frac{sin(\omega_k/2)}{\omega_k/2}\Delta_k\sin(\theta_k + \omega_k/2) \\ \theta_{k+1} = \theta_k + \omega_k \end{cases} \tag{1.6}$$

Si de plus ω_k tend vers zéro, ce qui suppose que la période d'échantillonnage est suffisamment petite par rapport aux dynamiques du véhicule, alors le modèle odométrique donné par l'équation 1.6 peut s'écrire plus simplement comme suit :

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + \Delta_k\cos(\theta_k + \omega_k/2) \\ y_{k+1} = y_k + \Delta_k\sin(\theta_k + \omega_k/2) \\ \theta_{k+1} = \theta_k + \omega_k \end{cases} \tag{1.7}$$

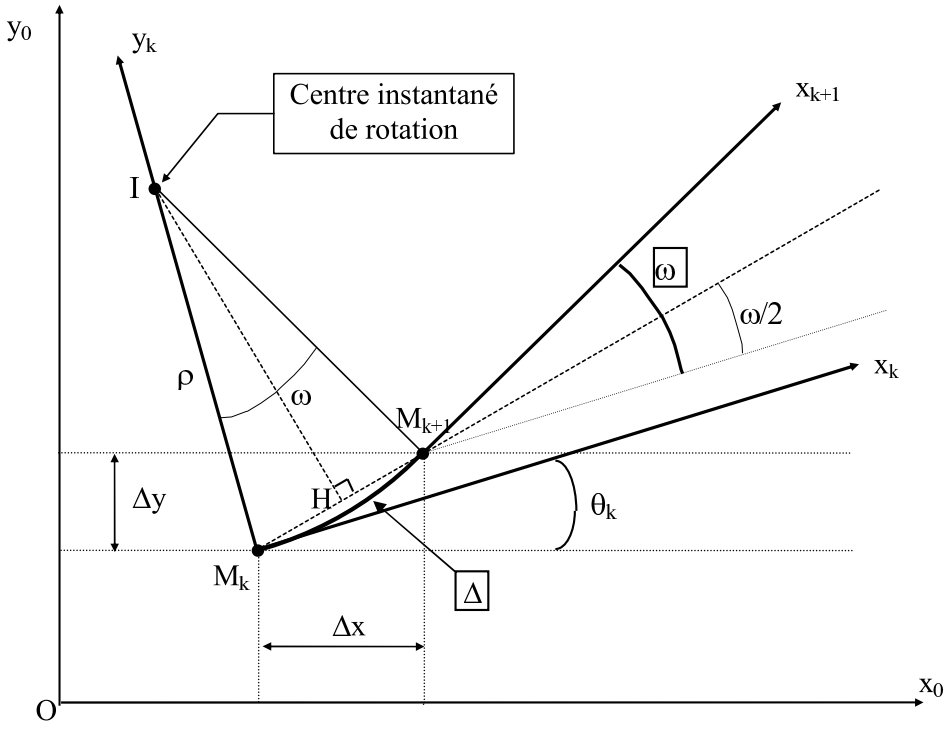


FIG. 1.4 – Schéma de déplacement entre deux instants d'échantillonnage

Ce dernier modèle odométrique est dit modèle odométrique différentiel et il sera utilisé tout au long de cette thèse pour localiser un ou plusieurs véhicules.
Le problème majeur avec les systèmes odométriques, est qu'ils déterminent la position du véhicule par rapport à un point de référence. Cela signifie que l'erreur de positionnement absolu s'accumule proportionnellement à la distance parcourue. Pour pallier la divergence de ce type de capteur, [Bonnifait, 2005] propose d'utiliser une technique odométrique utilisant les 4 roues. Les résultats de l'utilisation de ce modèle illustrent clairement le bénéfice de l'odométrie à 4 roues dont la dérive est moindre par rapport au modèle à deux roues.

1.2.3 Les télémètres

La télémétrie est une technique permettant de calculer ou de mesurer la distance d'un objet par utilisation d'éléments optiques, acoustiques ou radioélectriques (exemple : un télémètre laser). L'appareil permettant de mesurer ces distances est appelé télémètre (ce capteur appartient à la classe des capteurs extéroceptif).
Les capteurs infrarouges par exemple, sont constitués d'un ensemble émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible. Ces capteurs sont surtout utilisés pour la détection d'obstacles. Il faut noter que ce type de capteurs est sensible aux conditions extérieures, notamment à la lumière ambiante, aux surfaces sur lesquelles se réfléchissent les infrarouges et à la température.
Les capteurs ultrason sont un autre type de capteur dédié à la mesure de distance. Ces cap-

teurs utilisent les vibrations sonores dont les fréquences ne sont pas perceptibles par l'oreille humaine. Les ultrasons émis, se propagent dans l'air et sont réfléchis lorsqu'ils heurtent un corps solide. La distance entre la source et la cible peut être déterminée en mesurant le temps séparant l'émission des ultrasons du retour de l'écho.
Le télémètre laser constitue un troisième type de capteur de distance (voir la figure 1.5, source de cette photo [Bayle, 2006]). Ce capteur est de nos jour le moyen le plus utilisé en robotique pour obtenir des mesures précises de distance. Le principe de fonctionnement des télémètres laser est le suivant. Une impulsion est émise par une diode laser et simultanément, une horloge est enclenchée. Cette impulsion lumineuse est réfléchie par le premier obstacle rencontré sur son chemin. L'impulsion lumineuse, renvoyée, arrive sur un récepteur qui déclenche l'arrêt de l'horloge. A partir de ces informations, la distance séparant le télémètre et l'obstacle est donnée. L'angle de balayage varie entre 0 et 180 degrés sur les produits commercialisés les plus courants. La portée d'un tel capteur peut atteindre les 30 mètres mais elle dépend également de la réflectivité des milieux rencontrés [Bayle, 2006].

FIG. 1.5 – La famille des télémètres laser Sick.

1.2.4 Les gyroscopes

Les gyroscopes sont des capteurs proprioceptifs qui permettent de mesurer la vitesse angulaire et l'orientation d'un mobile. Il existe plusieurs types de gyroscopes : mécaniques et optiques pour les plus connus, mais aussi à structures vibrantes, capacitifs,...
Les gyroscopes optiques (voir la figure 1.6), utilisent deux faisceaux lasers émis depuis une même source, pour parcourir des chemins identiques, l'un dans le sens des aiguilles d'une montre, l'autre en sens opposé. Lors de la mise en rotation du gyroscope il existe une différence de marche des deux rayons et des interférences apparaissent. On peut alors déduire la vitesse de rotation du système. Pour plus de détails voir le support de cours de [Bayle, 2006].

1.2.5 Les caméras

Une autre possibilité pour localiser un véhicule, consiste à se tourner vers les techniques de vision [Royer, 2006]. Dans [Thuilot *et al.*, 2006] par exemple, les auteurs construisent la carte 3D de l'environnement à partir d'un enregistrement vidéo de référence, puis lors des opérations de navigation, le véhicule est localisé en mettant en correspondance l'enregistrement vidéo courant et la carte 3D établie précédemment.
L'utilisation de la vision dans le domaine de la localisation est bénéfique cependant, comme tout capteur, elle possède des limitations. Les caméras (voir l'exemple donné figure 1.7) sont souvent sensibles à l'intensité de la lumière (la vision de nuit n'est pas la même qu'en plein jour). D'un autre côté les caméras doivent être calibrées.

FIG. 1.6 – Exemple d'un gyroscope de type optique.

FIG. 1.7 – Exemple d'une caméra utilisée pour la localisation.

1.2.6 Autres sources d'informations pour la localisation

La précision d'une position estimée donnée par un GPS ou par l'odométrie peut être améliorée si on utilise des informations cartographiques 2D ou 3D [Cindy, 2008]. Elles permettent en particulier de contraindre les positions possibles aux seuls segments correspondants à des voies de circulation autorisées. Les cartes numériques offrent une description géométrique du réseau routier (voir la figure 1.8). Cependant, les coordonnées des segments de route sur une carte numérique sont généralement entachées d'erreurs. Le réseau routier de la base de données n'est pas toujours en parfait accord avec la réalité. Il peut contenir des linéaires qui n'existent plus réellement ou bien de nouveaux tronçons ne sont pas encore dans la base, ou encore l'information altimétrique est absente.
Pour déterminer la route sur laquelle un véhicule évolue, on utilise le plus souvent, un Système d'Information Géographique (SIG) qui gère la base de données routière. Grâce à une requête géoréférencée, le (SIG) présélectionne les segments de routes autour de la position estimée dans un rayon choisi. Le résultat est un nombre de segments qui est le plus souvent assez élevé (dans une zone urbaine, le nombre de segments dans un cercle de rayon de 50 m autour d'une position estimée est supérieur à 20).

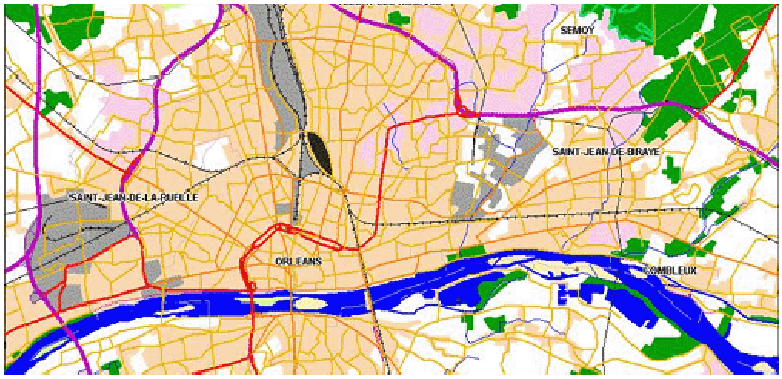


FIG. 1.8 – Exemple d'une carte topographique donnée par Navteq sur la région d'Orléans.

1.3 La fusion de données

1.3.1 Introduction

La fusion de données vise l'association, la combinaison et l'intégration de multiples sources de données de nature différentes (symboliques, numériques,...) représentant des connaissances et des informations diverses dans le but de fournir une information globale plus fiable et plus complète. Les données fusionnées reflètent non seulement l'information générée par chaque source de données, mais encore l'information qui n'aurait pu être fournie par aucune des sources prises séparément.
Le terme de fusion de données s'est étendu à de plus vastes domaines de recherche notamment dans les applications de diagnostic médical, la robotique, la compréhension automatique de documents scientifiques... Tous ces domaines ont en commun le fait de devoir manipuler de grandes quantités de données de natures et de types variés, afin d'obtenir une information de meilleure qualité.

La fusion de plusieurs documents scientifiques par exemple, permettra d'obtenir un document composite et décrivant avec précision la structure et les relations qui existent entre les documents analysés. Le résultat sera une information facile à utiliser pour faire, par exemple, une recherche thématique dans un grand ensemble de documents scientifiques. La qualité de l'information obtenue tient ici à la facilité qu'elle apportera au processus de recherche de documents [Bellot, 2002].
Fusionner l'odométrie et la cartographie permet par exemple de remédier au problème d'intermittence souvent rencontrée dans l'utilisation d'un GPS dans les centres villes. De ce fait, la fusion de données devient particulièrement nécessaire autant d'un point de vue fondamental que pour des applications pratiques.

Dans la suite de ces paragraphes, nous donnons une présentation non exhaustive des techniques de fusion de données parmi les plus courantes. Cette présentation à l'avantage de clarifier la fusion multicapteurs dans le domaine de la localisation.

1.3.2 Approches classiques pour la fusion de données

Un grand nombre de techniques de fusion de donnée ont été décrites dans la littérature. Parmi les plus utilisées on peut citer : le filtre de Kalman, les modèles de Markov cachés, l'estimation bayésienne, la théorie de Dempster-Shafer, la logique floue,... Dans les paragraphes qui suivent, nous donnons une brève description de certaines de ces techniques [Haton *et al.*, 1998].

1.3.2.1 Les modèles de Markov cachés

Les modèles de Markov cachés, ou HMM pour *Hidden Markov Models*, sont des modèles statistiques de données séquentielles, qui ont été largement utilisés dans le domaine de l'intelligence artificielle, en particulier dans le domaine de la reconnaissance de la parole [Rabiner, 1989].
L'hypothèse Markovienne est à la base de nombreux modèles stochastiques dont les HMM. Cette hypothèse suppose que l'état d'un système peut être totalement déterminé à condition de connaître l'état dans lequel il était à l'instant précédant et les observations faîtes sur le système à l'instant courant.
Ces HMM sont particulièrement adaptés à la modélisation de processus stochastiques. Cependant, la gestion de multiples capteurs n'est possible qu'avec un prétraitement de l'ensemble des données issues des capteurs pour les transformer en une observation unique, qui sera l'observation utilisée à chaque pas de temps par le HMM [Bellot, 2002]. Il est difficile aussi dans ce cadre de prendre en compte des connaissances sur la dynamique du système que l'on cherche à localiser.

1.3.2.2 Les modèles graphiques probabilistes

Les modèles graphiques probabilistes sont classiquement définis comme étant un mariage entre la théorie des probabilités et la théorie des graphes. Les probabilités permettent à ces modèles de prendre en compte l'aspect incertain présent dans les applications réelles. La représentation graphique des modèles offre un cadre intuitif et attractif dans de nombreux domaines d'applications où les utilisateurs ont besoin de "comprendre" le modèle qu'ils utilisent.
Les modèles graphiques sont aussi connus sous le nom de réseaux de croyance, réseaux d'indépendance probabiliste ou encore réseaux bayésiens [Pearl, 2001]. Ils forment un cadre très

efficace pour la fusion de données. Ils permettent de modéliser un problème simplement et d'utiliser des données en provenance de multiples sources, quelles soient qualitatives ou quantitatives. Ces données servent à mettre à jour la connaissance et les croyances que l'on a sur le problème. Ces modèles seront largement abordés dans le chapitre 2 qui est entièrement consacré au raisonnement causal probabiliste et plus particulièrement aux réseaux bayésiens. Notons que d'autres techniques de fusions de données seront présentées dans la partie localisation d'un véhicule sur une carte routière (voir section 1.4.2).

1.3.2.3 Filtre de Kalman

Le filtre de Kalman dont on doit le nom à Rudolf Emil Kalman est un estimateur récursif. Cela signifie que pour estimer l'état courant du système : $X_k \in \Re^n$, seul l'état précédent : X_{k-1} et les mesures actuelles : $Y_k \in \Re^m$ sont nécessaires (et éventuellement l'entrée du filtre $U_{k-1} \in \Re^l$).
Étant donné le système linéaire définit comme suit :

$$X_k = AX_{k-1} + BU_{k-1} + W_k \tag{1.8}$$

$$Y_k = H_k X_k + V_k \tag{1.9}$$

On suppose que l'on ne peut pas observer directement le système donné par 1.8, mais on dispose de l'observation Y_k donnée par l'équation 1.9 qui est la somme d'un signal $H_k X_k$, et d'un bruit d'observation V_k. Les variables aléatoires W_k et V_k représentent respectivement le bruit du modèle et le bruit des observations. Ces deux variables sont supposées indépendantes et suivent la loi normale :

$$p(W) \sim N(0, Q)$$

$$p(V) \sim N(0, R)$$

Dans la pratique, les covariances Q et R changent au cours du temps, cependant nous supposons qu'elles restent constantes. Les coefficients des équations 1.8 et 1.9 sont définis comme suit : $A \in \Re^{n \times n}$, $B \in \Re^{n \times l}$ et $H \in \Re^{m \times n}$. On suppose également que :

- la condition initiale X_0 est gaussienne de moyenne $\bar{X_0}$ et de covariance Q_0^X,
- les bruits W_k et V_k et X_0 sont mutuellement indépendants.

Étant donné l'observation à l'instant k : $Y_{0:k} = (Y_0, Y_1, ..., Y_k)$, l'objectif du filtre de Kalman est d'estimer le vecteur aléatoire X_k à partir de $Y_{0:k}$ de façon optimale et récursive. Si on adopte le critère du minimum de variance[1], il s'agit de calculer la loi conditionnelle du vecteur aléatoire X_k sachant $Y_{0:k}$. Comme le cadre est gaussien, il suffit de calculer la moyenne donnée par :

$$p(X_k|Y_{0:k}) \sim N(\hat{X}_k, P_k)$$

avec

$$\hat{X}_k = E[X_k|Y_{0:k}]$$

[1]Soit $\psi(.)$ un estimateur de X sachant Y. Naturellement $\psi = \psi(Y)$ n'est pas égal à X : une mesure de l'écart entre l'estimateur et la varie valeur est fournie par la variance de l'erreur d'estimation (ou erreur quadratique moyenne) $E[|X - \psi(Y)|^2]$. L'estimateur du minimum de variance de X sachant Y est un estimateur $\hat{X}(.)$ tel que $E[|X - \hat{X}(Y)|^2] \leq E[|X - \psi(Y)|^2]$

et la matrice de covariance donnée par :

$$P_k = E[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T | Y_{0:k}]$$

On pose également :

$$p(X_k|Y_{0:k-1}) = N(\hat{X}_k^-, P_k^-)$$

avec

$$\hat{X}_k^- = E[X_k|Y_{0:k-1}]$$

et

$$P_k^- = E[(X_k - \hat{X}_k^-)(X_k - \hat{X}_k^-)^T | Y_{0:k-1}]$$

Les matrices P_k et P_k^- ne dépendent pas des observations Y_k, c'est-à-dire [Le Gland, 2005] :

$$P_k = E[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T | Y_{0:k}] = E[(X_k - \hat{X}_k)(X_k - \hat{X}_k)^T]$$

$$P_k^- = E[(X_k - \hat{X}_k^-)(X_k - \hat{X}_k^-)^T | Y_{0:k-1}] = E[(X_k - \hat{X}_k^-)(X_k - \hat{X}_k^-)^T]$$

Supposons maintenant connue la loi conditionnelle du vecteur $p(X_{k-1}|Y_{0:k-1})$. Pour calculer la loi conditionnelle du vecteur $p(X_k|Y_{0:k})$, on procède en deux temps :

1. *Prédiction* : on calcule la loi conditionnelle du vecteur $p(X_k|Y_{0:k-1})$ en utilisant l'équation 1.8,
2. *Correction* : on corrige la prédiction en tenant compte de la nouvelle observation Y_k donnée par l'équation 1.9.

La question qui se pose maintenant est de savoir ce qu'apporte la nouvelle observation Y_k par rapport aux observations passées $Y_{0:k-1}$? On pose :

$$e_k = Y_k - E[Y_k|Y_{0:k-1}] \tag{1.10}$$

D'après l'équation 1.9 :

$$e_k = Y_k - (H_k E[X_k|Y_{0:k-1}] + E[V_k|Y_{0:k-1}]) = Y_k - H_k \hat{X}_k^-$$

compte tenu du fait que V_k et $Y_{0:k-1}$ sont indépendants.
Le processus e_k est un processus gaussien à valeurs dans $\Re^m$, appelé *processus d'innovation.* Les preuves de toutes ces équations peuvent être trouvées par exemple dans [Le Gland, 2005] et [Compillo, 2004]. Ainsi le filtre de Kalman peut être donné par l'algorithme 1.

1.3.2.4 Filtre de Kalman étendu

Étant donné le système non linéaire définit comme suit :

$$X_k = f_k(X_{k-1}, U_{k-1}, W_k) \tag{1.11}$$

$$Y_k = h_k(X_k, V_k) \tag{1.12}$$

où X_k, Y_k, W_k et V_k sont des variables définies de la même manière que dans le filtre de Kalman classique. On suppose que les fonctions f_k et h_k sont dérivables.

Algorithme 1 Filtre de Kalman

Initialisation

$\hat{X}_0^- = \bar{X}_0 = E[X_0]$

$P_0^- = Q_0^X = cov(X_0)$

Prédiction

$\hat{X}_k^- = A\hat{X}_{k-1} + BU_{k-1}$

$P_k^- = AP_{k-1}A^T + Q$

Correction

$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R]^{-1}$ où K_k est appelé le *gain* de Kalman

$\hat{X}_k = \hat{X}_k^- + K_k(Y_k - H_k\hat{X}_k^-))$

$P_k = (I - K_k H_k)P_k^-$

Le système donné par les équations 1.11 et 1.12, ne peut pas se résoudre explicitement comme dans le cas linéaire/gaussien. L'idée du filtre de Kalman étendu est de linéariser les fonctions f_k eth_k autour de l'estimée courante et d'appliquer la technique du filtre de Kalman classique. Ainsi, on linéarise f_k autour de $(\hat{X}_{k-1}, \hat{U}_{k-1})$, et l'équation d'état s'écrit :

$$X_k = f_{k-1}(\hat{X}_{k-1}, \hat{U}_{k-1}) + \frac{\partial f}{\partial x}|_{(\hat{X}_{k-1}, \hat{U}_{k-1})}(X_{k-1} - \hat{X}_{k-1}) + \frac{\partial f}{\partial U}|_{(\hat{X}_{k-1}, \hat{U}_{k-1})}(U_{k-1} - \hat{U}_{k-1}) + \epsilon_k^1$$

De la même manière, on linéarise h_k autour de $\hat{X}_k^-$, et on obtient :

$$Y_k = h_k(\hat{X}_k^-) + \frac{\partial f}{\partial x}|_{\hat{X}_k^-}(X_k - \hat{X}_k^-) + \epsilon_k^2$$

Notons que ϵ_k^1 et ϵ_k^2 représentent les termes d'ordre supérieur à un. Ils seront négligés par la suite. Le filtre de Kalman étendu peut être donné par l'algorithme 2. D'après [Le Gland, 2005], on peut s'attendre à de bons résultats avec cette technique de filtrage lorsque l'on est proche d'une situation linéaire ou lorsque le rapport signal/bruit est grand[2].

1.3.2.5 Filtrage particulaire

Les méthodes fondées sur le filtrage particulaire proposent de représenter la loi conditionnelle de l'état par un nombre fini de masses pondérées. Un ensemble de points appelés particules est généré et chacune de ces particules représente un état probable du système (voir la figure 1.9). Les coefficients de pondération (poids) sur chaque particule sont une mesure du degré de confiance que l'on peut avoir en ces dernières pour représenter effectivement l'état. Les particules évoluent suivant l'équation d'état du système (étape de prédiction) et les poids sont ajustés en fonction des observations (étape de correction).

L'idée de base du filtrage particulaire consiste donc à déterminer des approximations sous la forme :

[2]Notons que lors de la linéarisation (calcul de la Jacobienne), on ignore souvent les dérivées d'ordres supérieurs à un

Algorithme 2 Filtre de Kalman Étendu

Initialisation

$\hat{X}_0 = \bar{X}_0 = E[X_0]$

$R_0 = Q_0 = cov(X_0)$

Prédiction

$\hat{X}_k^- = f(\hat{X}_{k-1}, \hat{U}_{k-1})$

$P_k^- = A_k P_{k-1} A_k^T + B_k Q_{k-1} B_k^T + Q$

Correction

$\hat{X}_k = \hat{X}_k^- + K_k(Y_k - H_k(\hat{X}_k^-))$

$P_k = (I - K_k H_k) P_k^-$

$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R]^{-1}$

avec $A_k = \frac{\partial f}{\partial x}|_{(\hat{X}_{k-1}, \hat{U}_{k-1})}$, $B_k = \frac{\partial f}{\partial u}|_{(\hat{X}_{k-1}, \hat{U}_{k-1})}$ et $H_k = \frac{\partial h}{\partial x}|_{(\hat{X}_k^-, \hat{U}_{k-1})}$

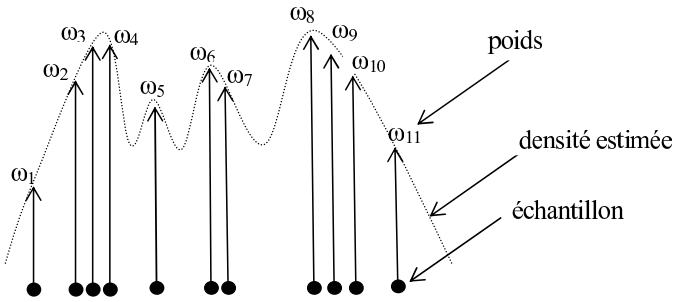


FIG. 1.9 – Approximation d'une densité par un ensemble fini de masses pondérées.

$$p(X_k|Y_{1:k-1}) \approx p^N(X_k|Y_{1:k-1}) = \sum_{i=1}^{N} \omega_{k^-}^i \delta_{\xi_{k^-}^i}(X_k) \tag{1.13}$$

$$p(X_k|Y_{1:k}) \approx p^N(X_k|Y_{1:k}) = \sum_{i=1}^{N} \omega_k^i \delta_{\xi_k^i}(X_k) \tag{1.14}$$

À chaque instant k, il s'agit donc de déterminer un nombre fini de paramètres $(\omega_{k^-}^{i:N}, \xi_{k^-}^{i:N})$ et $(\omega_k^{i:N}, \xi_k^{i:N})$. Afin de calculer ces quantités, le filtrage particulaire procède en deux phases.
Étape de prédiction : supposons connue la distribution de $p(X_{k-1}|Y_{1:k-1})$ donnée par

$$p(X_{k-1}|Y_{1:k-1}) = \sum_{i=1}^{N} \omega_{k-1}^i \delta_{\xi_{k-1}^i}(X_{k-1}) \tag{1.15}$$

On détermine la probabilité conditionnelle $p(X_k|Y_{1:k-1})$ en appliquant la définition suivante :

$$p(X_k|Y_{1:k-1}) = \int_{\Re^n} p(X_k|X_{k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1}$$

En appliquant l'équation de Chapman-Kolmogorov [Compillo, 2004] et [Dahia, 2005], on obtient :

$$p(X_k|Y_{1:k-1}) = \int p(X_k|X_{k-1}, Y_{1:k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1}$$

Dans [Compillo, 2004] et [Dahia, 2005], les auteurs donnent la démonstration de l'équation suivante :

$$p(X_k|X_{k-1}, Y_{1:k-1}) = p(X_k|X_{k-1})$$

et par conséquent la probabilité conditionnelle $p(X_k|Y_{1:k-1})$ s'écrit comme suit :

$$p(X_k|Y_{1:k-1}) = \int p(X_k|X_{k-1}) p(X_{k-1}|Y_{1:k-1}) dX_{k-1}$$

On remplace le terme $p(X_{k-1}|Y_{1:k-1})$ par l'équation donnée par 1.15, et on obtient :

$$\begin{aligned} p(X_k|Y_{1:k-1}) &= \sum_{i=1}^{N} \omega_{k-1}^i \int p(X_k|X_{k-1}) \delta_{\xi_{k-1}^i}(X_{k-1}) dX_{k-1} \\ &= \sum_{i=1}^{N} \omega_{k-1}^i \int p(X_k|X_{k-1}) \delta_{\xi_{k-1}^i}(X_{k-1}) dX_{k-1} \\ &= \sum_{i=1}^{N} \omega_{k-1}^i p(X_k|X_{k-1} = \xi_{k-1}^i) \end{aligned} \tag{1.16}$$

On obtient donc un mélange de lois $p(X_k|X_{k\ 1} = \xi_{k\ 1}^i)$ qui n'est pas sous forme particulaire. Pour obtenir une approximation de type particulaire on peut échantillonner selon cette loi [Compillo, 2004]. Une autre possibilité envisageable, consiste à utiliser :

$$p(X_k|Y_{1:k-1}) = \sum_{i=1}^{N} \omega_{k-1}^i \delta_{\xi_{k^-}^i}(X_k) \tag{1.17}$$

où $\xi_{k^-}^i \sim p(X_k|X_{k-1} = \xi_{k-1}^i)$.

Étape de correction : à l'étape de correction, la loi de densité conditionnelle $p(X_k|Y_{1:k})$ peut être déduite à partir du filtre prédit $p(X_k|Y_{1:k-1})$ à l'étape k selon la formule de correction suivante [Compillo, 2004] :

$$\begin{aligned} p(X_k|Y_{1:k}) &= \frac{p(Y_k|X_k)p(X_k|Y_{1:k-1})}{\int_{\Re^n} p(Y_k|X_k)p(X_k|Y_{1:k-1})dX_k} \\ &= \sum_{i=1}^{N} \frac{\omega_{k^-}^i p(Y_k|X_k)}{\sum_{j=1}^{N} \int \omega_{k^-}^j p(Y_k|X_k)\delta_{\xi_{k^-}^j}(X_k)dX_k} \delta_{\xi_{k^-}^i}(X_k) \\ &= \sum_{i=1}^{N} \frac{\omega_{k^-}^i p(Y_k|X_k = \xi_{k^-}^i)}{\sum_{j=1}^{N} \omega_{k^-}^j p(Y_k|X_k = \xi_{k^-}^j)} \delta_{\xi_{k^-}^i}(X_k) \end{aligned} \tag{1.18}$$

Ainsi, l'approximation de la densité conditionnelle $p(X_k|Y_{1:k})$ est donnée par :

$$p(X_k|Y_{1:k}) \approx \sum_{i=1}^{N} \omega_k^i \delta_{\xi_{k^-}^i}(X_k) \tag{1.19}$$

avec

$$\omega_k^i = \frac{\omega_{k^-}^i p(Y_k|X_k = \xi_{k^-}^i)}{\sum_{j=1}^{N} \omega_{k^-}^j p(Y_k|X_k = \xi_{k^-}^j)}$$

L'algorithme 3 résume l'aspect fonctionnel du filtrage particulaire :

Algorithme 3 Filtrage Particulaire

Prédiction

$p(X_k|Y_{1:k-1}) \approx \sum_{i=1}^{N} \omega_{k^-}^i \delta_{\xi_{k^-}^i}(X_k)$ où

$$\begin{cases} \omega_{k^-}^i = \omega_{k-1}^i \\ \xi_{k^-}^i \sim p(X_k|X_{k-1} = \xi_{k-1}^i) \end{cases}$$

Correction

$p(X_k|Y_{1:k}) \approx \sum_{i=1}^{N} \omega_k^i \delta_{\xi_{k^-}^i}(X_k)$ où

$$\begin{cases} \omega_k^i = \frac{\omega_{k^-}^i p(Y_k|X_k = \xi_{k^-}^i)}{\sum_{j=1}^{N} \omega_{k^-}^j p(Y_k|X_k = \xi_{k^-}^j)} \\ \xi_k^i = \xi_{k^-}^i \end{cases}$$

Le filtre particulaire tel que donné ci-dessus présente un défaut important : les poids des particules ont tendance à diverger de sorte que, après un certain nombre de mesures, la plupart des particules ont un poids négligeable. Ce phénomène est connu sous le nom de dégénérescence des poids. Le système de particules est appauvri et donc ne peut plus représenter correctement

la densité conditionnelle. Afin de remédier à ce problème, plusieurs méthodes ont été proposées dans le but de régulariser les poids. Idéalement les poids doivent tous rester proches de $1/N$, i.e. les particules sont d'égale importance dans l'approximation.
Considérons le critère suivant :

$$N_k^{eff} = \frac{1}{\sum_{i=1}^{N}(\omega_k^i)^2} \in [1, N]$$

qui représente le nombre efficace de particules. Lorsque N_k^{eff} est proche de N alors les particules sont d'égale importance. Il y a dégénérescence des poids lorsque N_k^{eff} est proche de 1 (contribution négligeable de ces poids dans l'approximation).
Un second critère de rééchantillonnage a été proposé par [Pham *et al.*, 2003]. Ce critère est fondé sur le calcul de l'entropie des pondérations. Pour éviter le phénomène de divergence du filtre, une nouvelle étape dite de rééchantillonnage est introduite. L'idée est de dupliquer les particules de poids fort et d'éliminer les particules de poids faible.
Les principales méthodes de sélection de particules comme le résume [Doucet, 1998] sont le tirage multinomial, le rééchantillonnage résiduel, le rééchantillonnage déterministe. L'algorithme de filtrage particulaire est réalisé par l'algorithme 4. Notons que plusieurs optimisations de cet algorithme existent dans la littérature.

Algorithme 4 Filtre Particulaire générique

Initialisation
$\xi^{1:N} \sim p(X_0)$
$\omega^{1:N} \longleftarrow 1/N$

Pour $(k = 1;\ k \leq NB_{iteration})$ **Faire**
$\overset{\sim}{\xi}{}^{i} \sim p(X_k|X_{k-1} = \xi^i)$ pour i=1 :N *échantillonage*
$\overset{\sim}{\omega}{}^{i} \longleftarrow \omega^i p(Y_k|X_k = \overset{\sim}{\xi}{}^{i})$ pour i=1 :N *mise à jour des poids*
$\overset{\sim}{\omega}{}^{i} \longleftarrow \frac{\omega^i}{sum(\overset{\sim}{\omega}{}^{1:N})}$ pour i=1 :N *normalisation des poids*
$N^{eff} \longleftarrow \frac{1}{\sum_{i=1}^{N}(\overset{\sim}{\omega}{}^{i})^2}$
Si $(N^{eff}/N \leq seuil)$ **Alors**
$\xi^{1:N} \longleftarrow rchantillonner(\overset{\sim}{\omega}{}^{1:N}, \overset{\sim}{\xi}{}^{1:N})$
$\omega^{1:N} \longleftarrow 1/N$
Sinon
$\xi^{1:N} \longleftarrow \overset{\sim}{\xi}{}^{1:N}$
$\omega^{1:N} \longleftarrow \overset{\sim}{\omega}{}^{1:N}$
Fin si
Sortie $(\xi^{1:N}, \omega^{1:N})$

Fin Pour

1.4 Méthodes et algorithmes de Localisation

Dans les sections qui suivent, nous présentons les méthodes et les approches couramment utilisées dans le domaine de la localisation. Les méthodes de localisation sans utilisation de cartes sont présentées dans la première partie. Dans la deuxième partie, nous décrivons plus en détails les méthodes et approches de localisation utilisant une carte routière. Cette deuxième partie fait l'objet du travail de recherche présenté dans cette thèse.

1.4.1 Localisation sans carte

Dans le domaine de la robotique mobile, des solutions pour la localisation d'un robot ont déjà été étudiées et présentées. Les méthodes de localisation couramment utilisées sont essentiellement basées sur l'odométrie [El Najjar, 2003], sur des accéléromètres et gyromètres ou ce qu'on appelle (mesures inertielles) [Woodman, 2007], sur le recalage successif d'images, l'utilisation d'amers, de cartes de l'environnement et bien sûr des systèmes de positionnement par satellites.

1.4.1.1 Localisation relative

La localisation relative permet au robot d'utiliser les mesures de ses mouvements propres, données par des capteurs dits proprioceptifs. Le repère de référence dans ce cas, correspond au repère du véhicule dans sa situation initiale. La façon la plus simple d'estimer la position d'un robot est l'utilisation de l'odométrie.

1.4.1.2 Odométrie

L'odométrie est un mode de localisation d'une simplicité remarquable. A l'aide de capteurs proprioceptifs disposés sur les roues du véhicule, on peut mesurer le nombre de tours de chaque roue effectués pendant une durée donnée. Si le temps d'échantillonnage est assez petit on peut même déduire les vitesses linéaires et angulaires. Il est à noter que l'odométrie ne donne qu'une position relative du robot, la position initiale devant être obtenue par d'autres moyens.
Ce système est d'une simplicité remarquable, mais il reste très imparfait. Le calcul de la position du robot est fait en supposant qu'il n'y a pas de glissement et que les paramètres géométriques du robot sont parfaitement connus, notamment le diamètre des roues. Cependant, les diamètres des roues d'un véhicule sont inégaux et varient dans le temps. L'accumulation d'erreurs dues à l'inexactitude des hypothèses formulées justifie l'association de l'odométrie à au moins un autre mode de localisation, ne serait-ce que pour son initialisation.

1.4.1.3 Navigation inertielle

La navigation inertielle [Cindy, 2008], [Woodman, 2007] utilise des accéléromètres et des gyromètres pour mesurer l'accélération et la vitesse angulaire du véhicule. Pour avoir une position en trois dimensions, le capteur inertiel doit comporter trois accéléromètres et trois gyromètres. Les accéléromètres mesurent les accélérations linéaires selon chaque axe, c'est-à-dire la pesanteur, l'accélération du véhicule, l'accélération centrifuge, l'accélération de Coriolis et du bruit. Les gyroscopes mesurent les variations d'orientation du véhicule. On distingue deux types de capteurs inertiels :

1. Les unités de mesures inertielles (Inertial Measurement Unit - IMU) dans lesquelles les données brutes des accéléromètres et des gyromètres sont les seules données de sortie.
2. Les systèmes de navigation inertielle (Inertial Navigation System - INS) dans lesquels ont été ajoutés des algorithmes de navigation permettant de déduire la position et l'orientation à partir des accélérations linéaires et des vitesses angulaires.

Les avantages de tels systèmes inertiels sont les suivants : une bonne précision à court terme, une fréquence d'échantillonnage élevée (entre 100 à 150 Hz), un système autonome qui ne dépend pas de dispositifs extérieurs, l'estimation de nombreux paramètres : position, vitesse, accélération et orientation. Cependant, ce système présente les inconvénients suivants : une mauvaise précision à long terme, un système inertiel a besoin d'être initialisé, la position et la vitesse doivent être calculées à partir de conditions initiales fournies par un capteur extéroceptif, une grande sensibilité à la gravité.

1.4.1.4 La vision

Le recours à la vision par ordinateur a pour objectif de compléter certaines lacunes de la localisation relative. Le principe de l'auto calibration via l'image est de s'appuyer sur les images acquises par une (ou plusieurs) caméra(s) embarquée(s) sur le véhicule. La localisation relative dans ce cas consiste à considérer deux images successives prises par la caméra aux instants t et $t+1$ et à déterminer le déplacement de la caméra [Cindy, 2008].
Une autre méthode de localisation relative consiste à utiliser un laser 2D. Placé sur le véhicule, il balaye horizontalement l'espace. Cette méthode permet d'obtenir un grand nombre de profils télémétriques horizontaux de l'environnement, qui, recalés entre eux, permettent de déduire la position du véhicule. Cette approche a été utilisée dans [Frueh et Zakhor, 2004] et a permis d'obtenir une grande qualité de localisation, mais là encore, à court terme. En effet, les erreurs de recalage s'accumulent (notamment lors des forts changements de direction) et la position estimée du véhicule dérive alors inévitablement [Cindy, 2008].

1.4.1.5 Localisation absolue

Dans le cas de la localisation absolue, le repère de référence est lié à l'environnement sur lequel le robot évolue. Les capteurs utilisés (caméras, télémètres lasers, radars, GPS,...) dans ce type de localisation sont dits extéroceptifs permettent ainsi de percevoir l'environnement dans lequel le robot évolue.
Les méthodes de positionnement absolu peuvent être divisées en plusieurs catégories, basées sur l'utilisation d'amers, de cartes de l'environnement, de cartes routières numériques ou bien par satellites.

1.4.1.6 Localisation en utilisant des amers

La localisation utilisant des amers ou balises repose sur les données externes issues du repérage, par un capteur embarqué, d'amers artificiels ou naturels placés dans l'environnement. La position de ces amers dans l'environnement où évolue le robot est supposée connue avec précision.
Afin de simplifier le problème de localisation, il est souvent admis que la position courante du véhicule est connue approximativement. La qualité de la détection et de la mise en correspondance des amers dépendra alors de la précision de cette estimation initiale et de la position du véhicule.

Plusieurs méthodes de calcul de la pose existent dans la littérature [Cindy, 2008]. La triangulation par exemple, utilise la distance et l'angle de gisement afin de calculer la position et le cap du robot. Contrairement à la méthode de triangulation, la trilatération utilise uniquement des distances (voir la figure 1.10). On voit qu'en connaissant la distance d_1 par rapport au point de référence B_1, on déduit que l'objet cherché se trouve sur le cercle de rayon d_1. En ajoutant d_2, la distance par rapport au point B_2, la position cherchée se réduit à deux points M et M'. Finalement, en ajoutant la distance d_3 par rapport au point B_3, la seule localisation possible de l'objet reste le point M.
Pour plus de détails sur ces méthodes et à la gestion des amers, nous renvoyons le lecteur à [Jabbour, 2007] et [Cindy, 2008].

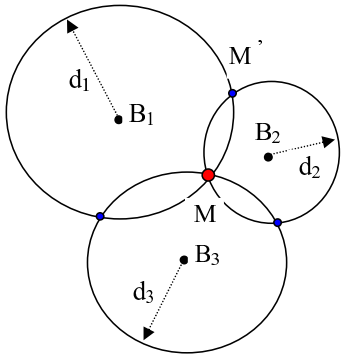


FIG. 1.10 – Principe de la trilatération pour l'estimation de la position.

1.4.1.7 Localisation en utilisant des cartes

Un dernier type de localisation consiste à utiliser les cartes de l'environnement ou les cartes routières numériques. Une carte peut être apprise au préalable par le véhicule dans une phase d'exploration. Elle peut également être fournie par une source extérieure dans une phase d'initialisation ou finalement le robot peut apprendre la carte de l'environnement pendant la navigation.
L'apprentissage de la carte de l'environnement pendant la navigation est connue en anglais sous le nom de SLAM (Simultaneous Localization And Mapping). Ce problème est l'un des principaux défis de la robotique mobile : naviguer à partir d'une position inconnue dans un environnement inconnu en construisant en même temps une représentation de l'environnement et déterminer dans cette représentation la position du robot. Ce sujet a mobilisé de très nombreux chercheurs au cours de la dernière décennie. Nous renvoyons le lecteur à [Thrun, 2003] et au lien suivant : http ://www.cas.kth.se/SLAM/.
La localisation sur une carte routière connue en anglais sous le terme de *map-matching* ou *road-matching* est étudiée en détail dans les paragraphes qui suivent.

1.4.2 Map-Matching : localisation d'un véhicule avec une carte routière

Le problème de la localisation sur une carte (qui fait l'objet d'une grande part de la problématique étudiée dans cette thèse) peut être subdivisé en deux grands problèmes. Le premier concerne la sélection du segment sur lequel le véhicule se situe. Le second consiste à estimer la position du véhicule sur le segment sélectionné. Le terme anglo-saxon pour désigner ces deux problèmes est connu sous le nom de *map-matching* ou *road-matching*. Ce terme est utilisé tout

au long de ce mémoire comme tel ou bien traduit en français par "mise en correspondance d'une estimée sur un segment de route".
La plupart des algorithmes conçus dans ce domaine de recherche utilisent un système de positionnement par satellite (GPS plus odométrie) et une base de données géographique (la carte routière numérique). L'une des hypothèses les plus fréquente dans ce domaine de recherche, consiste à supposer que la position d'un véhicule ne peut se situer que sur un tronçon de route, excluant les zones non référencées dans la base de données routières (les parkings, les parc, les forêts,...).
Les différents algorithmes développés dans le domaine du *road-matching* se fondent essentiellement sur des approches géométriques, topologiques, probabilistes, à base de logique floue,...Les performances de ces approches sont améliorées au fil des ans en raison de l'utilisation de nouvelles cartes plus précises.La qualité de l'estimation donnée par le GPS est un autre facteur d'amélioration. Cependant, ces approches ne sont pas toujours en mesure de donner des résultats satisfaisants particulièrement dans des environnements difficiles et complexes tels que les zones urbaines.
D'après [El Najjar, 2003] et [Quddus, 2006] les différentes approches proposées pour répondre au problème de la localisation sur carte routière peuvent être classées en 3 catégories :

1. Approches géométriques
2. Approches topologiques
3. Approches avancées

Les sections suivantes donnent plus de détails sur ces approches et les performances de chacune d'elles sont ensuite discutées.

1.4.2.1 Approches géométriques

Les algorithmes basés sur l'approche géométrique tiennent compte uniquement de la géométrie du réseau routier et ne considèrent pas la façon dont les segments sont reliés les uns aux autres [Quddus, 2006]. Chaque point estimé est mis en correspondance directe avec le nœud le plus proche. Ce type d'approche est connu sous le nom d'approche géométrique point à point. Bien que cette approche soit à la fois simple et facile à mettre en œuvre, elle a l'inconvenient d'une grande sensibilité à la façon dont le réseau routier (la base de données) est construit. En pratique, les performances d'une telle méthode sont modestes car le point le plus proche ne correspond pas nécessairement au segment sur lequel se situe le véhicule [White *et al.*, 2000]. La figure 1.11 illustre bien ce problème.
Dans cet exemple, les positions données par le système de localisation (GPS) sont indiquées par les chiffres de 1 à 7. La trajectoire réelle du véhicule est représentée par la courbe rouge et les segments correspondants à cette trajectoire sont AB et BD. Cependant les segments de routes sélectionnés par l'approche de type point à point sont AB, BF, FG et BD. Ce qui ne reflète pas la réalité.

Une autre approche de type géométrique dite "point segment" consiste à sélectionner le segment le plus proche au sens de la distance point segment [Quddus, 2006]. Bien que cette approche donne de meilleurs résultats que celle vue précédemment, elle possède ces propres défauts qui la rendent peu utilisable en pratique. La figure 1.12 en est une illustration.
Certes cette approche est bien meilleure que celle présentée précédemment (point à point), cependant comme on le constate sur la figure 1.12, le point le plus proche du segment ne correspond pas nécessairement au segment sur lequel se situe le véhicule (voir par exemple les estimées 3 et 4).

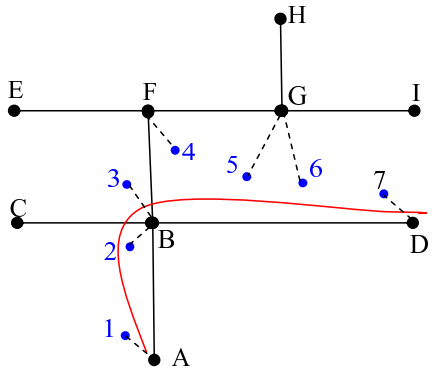


FIG. 1.11 – La localisation de type point à point prend en compte seulement la distance de l'estimée par rapport au nœud le plus proche.

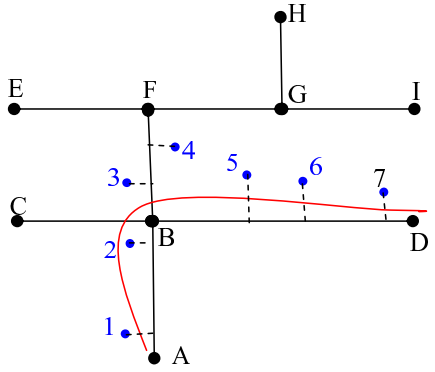


FIG. 1.12 – La localisation de type point segment tient compte seulement de la distance de l'estimée par rapport au segment.

Une autre approche géométrique dite "arc à arc" consiste à mettre en correspondance plusieurs positions estimées consécutives $(p_1, p_2, ..., p_m)$ avec un arc. Un arc est défini dans ce cas, comme étant un ensemble de segments connexes [White *et al.*, 2000]. Dans cette approche on identifie d'abord les nœuds candidats en utilisant l'approche point à point puis on mesure les distances entre les arcs.
Dans [White *et al.*, 2000], l'auteur segmente l'arc à tester en tronçons de même longueur que ceux de la trajectoire suivie par le véhicule. L'arc le plus proche de la courbe formée par les estimées du véhicule est pris comme le chemin (segments) sur lequel le véhicule roule. Cette approche est très sensible au point de départ et donne parfois des résultats inattendus [Quddus, 2006] et [El Najjar, 2003]. D'autres améliorations de cette approche peuvent être trouvées dans [Taylor *et al.*, 2001].

1.4.2.2 Approches topologiques

La performance des algorithmes fondés sur l'approche géométrique peut être améliorée si la topologie du réseau est utilisée. Dans [Meng, 2006], l'auteur utilise la topologie du réseau routier afin de mettre au point un algorithme plus simple. Cet algorithme est fondé sur la corrélation entre la trajectoire du véhicule et les caractéristiques topologiques de la route (les virages, connexion entre les segments,...).
Un certain nombre de tests sont appliqués dans le but d'éliminer les segments ne satisfaisants pas un seuil préalablement prédéfini. L'algorithme est mis en œuvre en utilisant les données du GPS, de l'odométrie et de la cartographie en incluant les informations sur les intersections, les restrictions de virage,... Cet algorithme est discuté dans [Quddus, 2006] car il ne donne pas de bons résultats pour les jonctions de route.

Dans [Greenfeld, 2002], l'auteur utilise l'approche topologique avec les critères suivants :
- La similitude sur l'orientation c'est-à-dire, le degré de parallélisme entre la ligne formée par deux points GPS et les segments de route du réseau,
- La proximité de la position par rapport au segment (approche géométrique point segment),
- La différence d'angle entre le segment de route et l'orientation du véhicule.

[Greenfeld, 2002] propose de donner plus de poids à la similitude sur l'orientation qu'à la proximité. Pour améliorer les performances de cet algorithme, [Quddus, 2006] propose de nouveaux critères plus performants. Il introduit la vitesse du véhicule à la position relative du véhicule par rapport aux segments candidats. Dans ce cas, le cap du véhicule est directement donné par le GPS.
Cette approche regroupe quasiment tous les critères de choix proposés dans la littérature. Cependant, cette même approche est critiquée par l'auteur, car elle reste insuffisante pour traiter des cas comme indiqués par la figure 1.13.
En se référant à l'exemple donné par la figure 1.13, l'algorithme donné ci-dessus identifie le bon segment AB pour les estimations P_1, P_2, P_3. Cependant le reste des estimations (P_4, P_5, P_6, P_7), peuvent être incorrectes si la distance entre P_4, P_5 et les segments BC, BD (et DE, BC pour les estimations P_6, P_7) sont égaux, et que le cap du véhicule est mal estimé.

Tous les exemples donnés jusqu'à présent montrent bien la difficulté de la problématique traitée dans cette thèse. L'approche topologique apporte certaines réponses aux problèmes rencontrés avec les approches géométriques, cependant plusieurs cas de figures restent sans

réponses (voir l'exemple donné par la figure 1.13). Ce qui nous amène à considérer d'autres approches plus performantes.

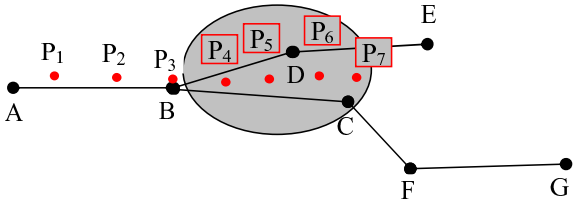


FIG. 1.13 – Un exemple de situation où la sélection du segment sur lequel le véhicule roule est presque impossible. Ce type de situation est souvent rencontré dans la sortie d'un carrefour.

1.4.2.3 Approches avancées du Map-Matching

On désigne par "approches avancées" les algorithmes qui utilisent des outils théoriques probabilistes ou stochastiques tels que les filtres de Kalmam [Krakiwsky *et al.*, 1988], [Tanaka *et al.*, 1990] et [Kim *et al.*, 2000], la théorie de Dempster-Shafer également connue sous le nom de théorie des croyances [El Najjar, 2003], [Yang *et al.*, 2003], la logique floue [Zhao, 1997b], [Syed et Cannon, 2004], et le filtrage particulaire [Gustafsson *et al.*, 2002].

Dans [Yang *et al.*, 2003], les auteurs ont développé un algorithme de map-matching basé sur la théorie de Dempster-Shafer et le filtre de Kalman. Les entrées de l'algorithme (les estimations du GPS) sont filtrées en utilisant un filtre de Kalman. La distance entre la position donnée par le GPS et les segments de route est obtenue en utilisant l'approche géométrique de type point segment. Les poids sont ensuite donnés aux segments sur la base des distances calculées. Par exemple, si la distance est comprise entre 15m et 20m, le poids affecté au segment est de 0,7, et si la distance est comprise entre 5m et 10m, le poids affecté au segment est de 0,9. La théorie de Dempster-Shafer est ainsi appliquée pour obtenir le segment le plus probable. Les auteurs affirment que leur méthode donne 96% de bons résultats sur la base de 1075 estimations. Cependant, ces résultats ne sont pas toujours corrects dans les zones urbaines comme on l'a déjà indiqué dans le cas des méthodes géométriques.

Dans [El Najjar, 2003], l'auteur a mis au point un nouvel algorithme de map-matching capable de supporter les applications en temps réel. Le véhicule est équipé d'un GPS (DGPS) et de l'odométrie. Une approche multi-sources était employée afin de permettre de gérer des informations qui peuvent être imparfaites, mais aussi complexes, hétérogènes, et difficiles à formaliser. Un filtre de Kalman étendu est utilisé pour la fusion des données DGPS et odométriques. Cette estimation ainsi que la fusion des critères de distance, de cap et de vitesse du véhicule sont utilisés par la théorie des croyances pour sélectionner le segment le plus probable. La méthode peut toutefois donner des résultats inexacts dans le cas de routes parallèles, et les jonctions de routes car elle ne tient pas compte de la topologie du réseau. De plus, dans cette étude l'auteur ne donne pas les performances de sa méthode en termes de bonne identification de segments [Quddus, 2006].

Syed et Cannon dans [Syed et Cannon, 2004] décrivent un algorithme de "map-matching" fondé sur la logique floue. L'algorithme fonctionne en deux temps : dans un premier temps le système d'inférence (floue) est utilisé pour sélectionner un ensemble de segments qui sont

à 50m de la position estimée par le GPS (ou de l'estimation donnée par l'odométrie). Un segment est alors identifié en se fondant sur la direction du véhicule par rapport à la direction du segment. La position du véhicule est alors donnée par une projection orthogonale de cette estimation sur le segment.
Dans un second temps, un autre système d'inférence est utilisé pour vérifier si la seconde estimation peut être mise en correspondance avec le segment récemment sélectionné. Les critères utilisés dans ce cas sont la proximité, le cap et la distance parcourue par le véhicule sur le segment. Si l'estimation donnée par le GPS (ou l'odométrie) est incohérente avec la carte, l'algorithme sélectionne un autre segment (étape 1). Comme l'étape 1 nécessite beaucoup de temps (environ 30 secondes), un retard important dans l'estimation peut apparaître.
Un autre inconvénient de cet algorithme est qu'il ne tient compte ni des sources d'erreurs des capteurs, ni des erreurs de la carte. Par conséquent la méthode de localisation du véhicule n'est pas robuste [Quddus, 2006].

Dans [Fu *et al.*, 2004], les auteurs proposent un autre algorithme basé sur la logique floue. Le système d'inférence utilise deux critères pour sélectionner le segment recherché.

1. Le segment le plus proche de la position GPS (méthode géométrique point segment). La distance est qualifiée comme suit : *très petite* (<20m), *petite* (20m-40m), *moyenne* (40m-60m), *grande* (60m-80m) et *très grande* (>80m)
2. Le segment de direction la plus proche du cap estimé du véhicule. La distance est qualifiée comme suit : *très faible* (<5°), *faible* (5°-30°), *moyenne* (30°-45°), *grande* (45°-60°), *très grande* (>60°),

Ce modèle de logique floue est sensible aux mesures du bruit. D'autre part, le cap du véhicule est obtenu à partir du GPS, ce qui n'est pas précis pour des vitesses faibles. D'autre part, l'algorithme ne tient pas compte de l'historique de la trajectoire ce qui mène fort probablement à un mauvais choix de segment.

La méthode proposée par Zhao [Zhao, 1997a] consiste à retenir les segments dont le cap est proche de celui du véhicule en utilisant un simple seuillage. Les segments sont triés en fonction de leur distance à la position estimée. Cette méthode utilise l'approche topologique (par exemple la connexion) vue précédemment afin de réduire le nombre de segments candidats. Le segment en tête de liste est considéré comme étant le segment le plus probable. Pour trouver la position sur le segment sélectionné, Zhao effectue une simple projection sur le segment. Dans cette approche, la sélection de segment échoue dans le cas de routes proches ou bien pour des jonctions de route dont le cap n'est pas précis.

Si on veut gérer ou manier plusieurs segments à la fois (multi-hypothèses) un filtre de Kalman n'est pas adapté. On peut cependant utiliser N filtres en parallèlles (un pour chaque segment candidat). Il s'agit donc d'une méthode très coûteuse en temps de calcul.
Une autre limite des filtres de Kalman est l'hypothèse de distributions de probabilités conditionnelles linéairement gaussiennes à l'instar du bruit. Par conséquent l'inférence exacte n'est pas toujours possible dans le cas où le système est non linéaire et/ou non gaussien [Murphy, 2002]. Cette analyse nous amène à considérer dans ce qui suit les méthodes d'approximations. Les travaux de [Gustafsson *et al.*, 2002] montrent que l'approche particulaire peut être une solution efficace au problème de la localisation sur carte et bien adaptée aux types des données cartographiques manipulées.

Le filtre particulaire tel que présenté ci-dessus s'adapte parfaitement au concept multihypothèses. En fait, N particules sont générées de façon équiprobable sur le réseau routier (sur chaque segment de route se trouve un nuage de point). Lorsque le véhicule se déplace, les particules subissent le même déplacement. Lors de la fusion avec la carte, les particules qui se retrouvent en dehors des routes se voient affecter des probabilités faibles. Les particules de poids faibles sont éliminées et celles de poids forts sont dupliquées [Gustafsson *et al.*, 2002]. Dans ce contexte, la densité de probabilité présente plusieurs maxima, c'est pouquoi on parle de multimodalité. Tant que cette densité de probabilité est multimodale, la situation reste ambiguë.

Exemple de fonctionnement d'un filtre particulaire
Prenons l'exemple donné figure 1.14. L'estimation donnée par le GPS (point rouge) à l'instant $T = 1$, nous confirme que le véhicule roule sur le segment de route numéro 1. La densité de probabilité est unimodale et il n'y a aucune ambiguïté.
Au pas de temps suivant : $T = 2$, la situation devient ambiguë. La densité de probabilité est multimodale car l'incertitude autour du point GPS nous conduit à traiter les segments des routes 1, 2 et 3. De nouvelles particules sont générées sur les trois segments. Chaque particule est affectée d'un poids représentant un état probable. Le même principe est utilisé pour les étapes 3, 4, 5 et 6.

Le filtre particulaire est un outil très puissant permettant de traiter le problème de la non linéarité et de se libérer de l'hypothèse gaussienne. Cependant, comme il faut beaucoup de particules pour obtenir une estimation convergente, les traitements informatiques sont lourds même s'ils sont simples. Avec l'augmentation permanente de la puissance des calculateurs, cet inconvénient s'estompe. La convergence numérique n'est pas toujours garantie et les phénomènes de dégénérescence des particules peuvent apparaître au cours de l'exécution [El Najjar, 2003]. Par conséquent, de nombreuses techniques ont été étudiées pour remédier à ce problème [Dahia, 2005].
Tout utilisateur du filtrage particulaire a comme soucis de répondre à la questions suivante : quel est le nombre minimum de particules à utiliser dans mon application ? Le nombre de particules utilisées est généralement fixé par une procédure d'essais erreurs. En fait, on commence par un très grand nombre de particules et on diminue ce nombre au fur et à mesure jusqu'à trouver le nombre minimal. Notons que ce paramètre (nombre minimal de particules) dépend de l'application elle même, du bruit des capteurs et de l'environnement sur lequel l'application est survenue [Murphy, 2002].

1.5 Conclusion

Dans ce chapitre, plusieurs méthodes de localisation liées à la robotique mobile ont été présentées. Localiser un véhicule paraît être une question facile alors que de plus en plus d'automobilistes utilisent quotidiennement un GPS. Pourtant, il s'agit d'une question difficile dès lors que l'on cherche à se localiser de manière précise et sans interruption de service. Le problème de la localisation est plus compliqué si on suppose un train de véhicules dont seul le premier est piloté, les autres suivant le précédent de manière automatique.
Les estimations données par le GPS sont souvent entachées d'erreurs soit parce que les satellites sont masqués, ce qui est fréquent dans les centres-villes, soit parce que de nombreuses réflexions perturbent l'estimation de la position (effets de multitrajets des ondes des signaux).

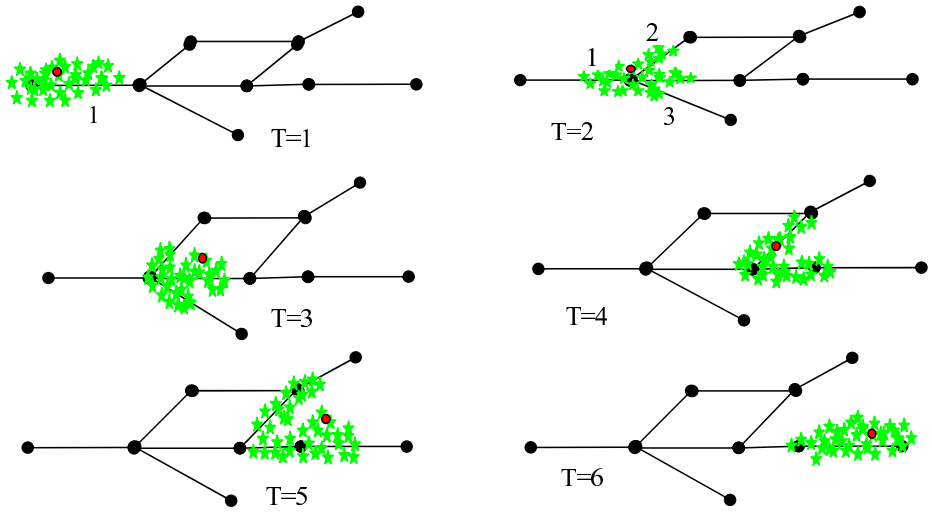


FIG. 1.14 – Évolution du nuage de particules sur le réseau routier. Au fur et à mesure que le véhicule avance, les particules subissent le même déplacement. Lors de la fusion avec la carte, les particules qui se retrouvent en dehors des routes se voient affecter des probabilités faibles. Les particules de poids faibles sont éliminées et celles de poids forts sont dupliquées.

Pour pallier les problèmes liés au GPS, d'autres sources d'informations peuvent être utilisées pour affiner la localisation. On peut par exemple utiliser des capteurs gyroscopiques qui donneront une information sur le cap du véhicule ou encore un télémètre laser qui donnera une information de distance par rapport à un obstacle dont on connaît la position. Les informations cartographiques permettent en particulier de contraindre les positions possibles aux seuls segments correspondant à des voies de circulation autorisées.
Cependant même avec ces divers capteurs et informations, le problème de la localisation n'est pas pour autant résolu vu que ces capteurs présentent eux mêmes des imperfections. Tel capteur est entaché d'erreur, l'autre est sensible à la température, le réseau routier de la base de données n'est pas toujours en parfait accord avec la réalité...
La solution satisfaisant le mieux à tous ces problèmes est la fusion de données. Les données fusionnées reflètent non seulement l'information générée par chaque source (capteur), mais encore l'information qui n'aurait pu être générée par aucune des sources prises séparément.
Plusieurs méthodes de fusion de données dédiées à la localisation sur une carte numérique sont présentées et analysées dans ce chapitre. Chaque méthode possède ses propres avantages et inconvénients.
De notre point de vue, le problème de la fusion multicapteurs avec un modèle cartographique de l'environnement par exemple, relève d'une question plus générale qui est l'estimation des variables cachées (la position, le cap,...) d'un système dynamique (le véhicule, un train de véhicules) étant donné un historique d'observation (les données capteur). La difficulté tient à ce que le système dynamique que constitue un véhicule n'est pas suffisamment bien connu pour être modélisé dans toutes les conditions d'utilisation. Par ailleurs, les variables mesurées pour les raisons énoncées ci-dessus ne sont pas précises. Elles peuvent même être manquantes ou non cohérentes temporairement.

De ce point de vu, nous avons choisi dans cette thèse une approche fondée sur la modélisation probabiliste pour sa robustesse à l'incertitude tant des modèles que des données

mesurées. Ainsi, le chapitre suivant se focalise principalement sur les modèles probabilistes de type réseaux bayésiens. Nous montrons que les réseaux bayésiens est un cadre parfaitement adapté aux problèmes d'incertitude et à la fusion multicapteurs.

Chapitre 2

Les réseaux bayésiens

2.1 Introduction

Si on pose la question suivante : *Pleuvra-t-il demain ?*, il est difficile sinon impossible de répondre par oui ou non. En revanche, en intégrant de nombreux paramètres, on pourra dire qu'il pleuvra "peut-être", "sûrement", "certainement",... c'est-à-dire inconsciemment, on associera une probabilité à l'événement *"il pleuvra demain"*. On peut sans aucun doute annoncer qu'il est quasiment impossible d'affirmer cela à 0 ou à 100%. Ainsi, plutôt que de raisonner sur la véracité ou non d'une proposition, on raisonnera plutôt sur la chance qu'un évènement se réalise. Cette chance se traduira par l'attribution d'une probabilité. Ainsi, pour une proposition A donnée, (exemple A : *il pleuvra demain*), on peut associer une probabilité $p(A)$, telle que $p(A)$ soit comprise entre 0 et 1. Si la proposition A est vraie, alors $p(A) = 1$, et si la proposition A est fausse alors $p(A) = 0$.
Cette notion de probabilité est à la base de nombreux modèles graphiques développés en intelligence artificielle. Parmi les plus utilisés, citons, les HMM (Hidden Markov Model), les réseaux bayésiens, les processus décisionnels de Markov. Nous nous intéressons dans ce qui suit au modèle graphique probabiliste de type réseau bayésien, pour lesquels plusieur terminologies sont utilisées dans la littérature : réseaux de croyance, réseaux probabilistes,...

Dans ce chapitre, nous présentons les notions de base sur les réseaux bayésiens utile à la compréhension ou à la reproduction des résultats présentés dans les chapitres suivants. Pour cela, nous nous sommes largement inspiré des ouvrages et articles suivants [Pearl, 1988], [Castillo *et al.*, 1997], [Cowell *et al.*, 1999] et [Murphy, 2002]. Dans un premier temps, nous abordons les réseaux bayésiens à variables aléatoires discrètes puis les réseaux bayésiens à variables continues. Nous étendons ensuite les notions présentées à la prise en compte de variables discrètes et continues dans le même réseaux. On parle alors de réseaux hybrides [Uri, 2002], [Murphy, 2002], [Cowell *et al.*, 1999]. Ce sont ces réseaux qui ont été à la base des développements de cette thèse. Nous donnerons en fin de ce chapitre les éléments essentiels de cette extension des réseaux bayésiens pour la compréhension de nos travaux.

2.2 Connaissance de base sur la théorie de probabilité

Dans les paragraphes qui suivent, nous présentons le vocabulaire nécessaire ainsi que les notions communément utilisées à la compréhension des réseaux probabilistes.

2.2.1 Distribution de probabilités

Supposons que nous disposions d'un ensemble $X = \{X_1, X_2, ..., X_n\}$ de variables aléatoires discrètes toutes binaires. On note $\{x_1, x_2, ..., x_n\}$ une réalisation ou instanciation de ces variables. La distribution de probabilité jointe *(JPD)*[3] de l'ensemble X est notée :

$$p(x_1, ..., x_n) = p(X_1 = x_1, ..., X_n = x_n). \tag{2.1}$$

et on appelle :

$$p(x_i) = p(X_i = x_i) = \sum_{x_1,...,x_{i-1},x_{i+1},...,x_n} p(x_1, ..., x_i, x_{i+1}, ...x_n) \tag{2.2}$$

la distribution de probabilité marginale *(MPD)*[4] de la $i^{ème}$ variable.
La connaissance d'une occurrence d'un événement peut modifier les probabilités des autres événements. Ainsi, chaque fois que de nouvelles informations deviennent disponibles, notre estimation des probabilités des événements peut évoluer, ce qui nous mène au concept de *probabilité conditionnelle*.

2.2.2 La probabilité conditionnelle

Soient X et Y deux sous-ensembles de variables tel que $p(Y = y) \neq 0$. La distribution de probabilités conditionnelles *(CPD)*[5] de X sachant $(Y = y)$ s'écrit :

$$p(X = x|Y = y) = p(x|y) = \frac{p(x, y)}{p(y)} \tag{2.3}$$

D'après l'équation 2.3, la *(JPD)* de X et Y peut être réécrite comme suit :

$$p(x, y) = p(x|y).p(y) \tag{2.4}$$

Un cas particulier de l'équation 2.3 est obtenu quand X représente une seule variable, alors que Y est un sous-ensemble de variables. Dans ce cas, l'équation 2.3 devient :

$$p(x_i|x_1, ..., x_k) = \frac{p(x_i, x_1, ..., x_k)}{p(x_1, ..., x_k)}; x_i \notin \{x_1, ..., x_k\} \tag{2.5}$$

Cette même formule peut être réécrite en utilisant la définition de la distribution de probabilité marginale *(MPD)* :

$$p(x_i|x_1, ..., x_k) = \frac{p(x_i, x_1, ..., x_k)}{\sum_{x_i} p(x_i, x_1, ..., x_k)} \text{ telle que } x_i \notin \{x_1, ..., x_k\} \tag{2.6}$$

La formule 2.6 illustre la *CPD* de la $i^{ème}$ variable x_i sachant le sous-ensemble $\{x_1, ..., x_k\}$. La somme du dénominateur de la formule 2.6 est donnée pour toutes les valeurs possibles de x_i. Notons que les deux équations données par les formules de la probabilité marginale 2.2 et la probabilité conditionnelle 2.6 s'appliquent toujours si la variable X_i est remplacée par un sous-ensemble de variables.

[3] Joint Probability Distribution
[4] Marginal Probability Distribution
[5] Conditional Probability Distribution

2.2.3 Dépendance et indépendance des variables

Soient X et Y deux sous-ensembles disjoints de l'ensemble des variable aléatoires $\{X_1, ..., X_n\}$. On dit que le sous-ensemble de variables X est indépendant de Y si et seulement si :

$$p(x|y) = p(x) \ \forall x \in X \ et \ \forall \ y \in Y \tag{2.7}$$

Dans le cas contraire, X est dépendant de Y. D'après la formule 2.7, si X est indépendant de Y, alors une connaissance sur Y n'affecte pas la connaissance sur X. De même, si X est indépendant de Y, nous pouvons combiner les équations 2.3 et 2.7 et obtenir :

$$\frac{p(x,y)}{p(y)} = p(x) \tag{2.8}$$

ce qui implique :

$$p(x,y) = p(x)p(y) \tag{2.9}$$

Cette dernière équation indique que si la variable X est indépendante de la variable Y, alors la *(JPD)* de X et Y est égale au produit de leur probabilité marginale.
Une importante propriété de l'indépendance est la symétrie. Ainsi, si la variable X est indépendante de la variable Y, alors Y est indépendante de X. Ceci est dû au fait que :

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(x)p(y)}{p(x)} = p(y). \tag{2.10}$$

En raison de la propriété de symétrie donnée par la formule 2.10, nous disons que X et Y sont indépendantes ou mutuellement indépendantes.
Les concepts de dépendance et indépendance pour deux variables peuvent être étendus au cas où l'on a plusieurs variables. Un ensemble de variables aléatoires $\{X_1, ..., X_m\}$ est indépendant si est seulement si :

$$p(x_1, ..., x_m) = \prod_{i=1}^{m} p(x_i) \tag{2.11}$$

pour toutes les valeurs possibles $x_1, ..., x_m$ de $X_1, ..., X_m$. Dans le cas contraire, ces variables sont dépendantes. En d'autres termes, $X_1, ..., X_m$ sont dites indépendantes si et seulement si leur distribution de probabilité jointe *(JPD)* est égale au produit de leurs distributions de probabilité marginale*(MPD)* (l'équation 2.11 est une forme généralisée de l'équation 2.9). Notons que si $X_1, ..., X_m$ sont dites conditionnellement indépendantes de $Y_1, ..., Y_n$, alors :

$$p(x_1, ..., x_m|y_1, ..., y_n) = \prod_{i=1}^{m} p(x_i|y_1, ..., y_n) \tag{2.12}$$

Définition 1 *(Indépendance conditionnelle) Etant données trois variables aléatoires X,Y et Z. X et Y sont dites indépendantes conditionnellement à Z, et on note $X \perp Y|Z$, si et seulement si pour toutes les valeurs possibles x, y et z prisent par X, Y et Z, on a :*

$$p(X|Y, Z) = p(X|Z) \tag{2.13}$$

Le terme d-separation est utilisé dans ce sens pour désigner deux variables séparées par la connaissance d'une troisièmes. Ces variables sont séparées car lorsque Z est connue, l'information ne circule pas entre X et Y. C'est à dire qu'une nouvelle connaissance de la valeur de Y ne modie en rien les probabilités sur les valeurs de X.

2.2.4 Théorème de Bayes

Le théorème de Bayes est un résultat essentiel en théorie des probabilités. Ce théorème permet de relier les causes aux effets. La formule de Bayes s'écrit :

$$p(x_i|x_1,...,x_k) = \frac{p(x_1,...,x_k|x_i)p(x_i)}{p(x_1,...,x_k)} \tag{2.14}$$

Pour illustrer l'utilisation du théorème de Bayes, supposons qu'un patient peut être dans deux états : en bonne santé représenté par D_m, ou bien avoir une maladie D_i parmi l'ensemble $\{D_1,...,D_{m-1}\}$. Supposons également qu'on a n symptômes représentés par $\{S_1,...,S_n\}$. Étant donné qu'un patient présente un ensemble de symptômes $\{s_1,...,s_k\}$, nous souhaitons calculer la probabilité que le patient ait la maladie d_i. En utilisant le théorème de Bayes donné par la formule 2.14, nous avons :

$$p(d_i|s_1,...,s_k) = \frac{p(s_1,...,s_k|d_i)p(d_i)}{p(s_1,...,s_k)} \tag{2.15}$$

D'après l'équation 2.15 nous pouvons tirer les conclusions suivantes :

- La probabilité $p(d_i|s_1,...,s_k)$ est appelée probabilité *a posteriori* ou conditionnelle, car elle est calculée après qu'on ait pris connaissances des symptômes $S_1 = s_1,...,S_k = s_k$.
- La probabilité $p(s_1,...,s_k|d_i)$ est la vraisemblance qu'un patient souffre de la maladie $D = d_i$ en présentant les symptômes $S_1 = s_1,...,S_k = s_k$.
- La probabilité $p(d_i)$ est appelée probabilité marginale, probabilité *a priori* ou probabilité initiale car elle est définie avant de connaître tous les symptômes.

Ainsi, nous pouvons utiliser le théorème de Bayes pour mettre à jour la distribution de probabilité *a posteriori*, en utilisant la probabilité *a priori* et la vraisemblance.

2.3 Notions de base sur la théorie des graphes

2.3.1 Introduction

Soit un graphe G représenté par le couple $G = (V, E)$, où V est l'ensemble des nœuds du graphe et E est l'ensemble d'arcs représentent la présence ou l'absence de dépendance ou indépendance conditionnelle entre les nœuds de V. En prenant comme exemple la figure 2.1, on peut décrire le couple $G = (V, E)$ par :
$V = \{A, B, C, D, E, F, G, H\}$.
$E = \{A \to C, A \to D, B \to D, B \to E, B \to G, C \to F, D \to F, D \to G, E \to G, E \to H\}$.

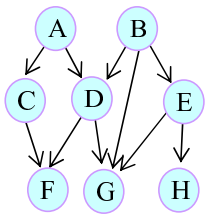


FIG. 2.1 – Exemple d'un graphe à huit noeuds

Définition 2 *(parents et fils) : Étant donné un arc reliant deux nœuds : $X_i \rightarrow X_j$, alors le nœud X_i est appelé parent du nœud X_j et par conséquent, le nœud X_j est appelé nœud fils de X_i. L'ensemble des parents d'un nœud X_i sera noté par la suite par : $pa(X_i)$ ou Π_{X_i} ou simplement Π_i. De même l'ensemble des fils d'un nœud Y_j est représenté par : $ch(Y_j)$, 'ch' pour 'children'.*

En prenant comme exemple la figure 2.1, $pa(G) = \{B, D, E\}$ constitue l'ensemble des parents du nœud G de la figure 2.1 et $ch(B) = \{D, E, G\}$ l'ensemble des fils du nœud B.

Définition 3 *(voisins) : L'ensemble des voisins d'un nœud X_i dans un graphe orienté est représenté par l'union des parents et des fils de ce même nœud. On notera par la suite les voisins d'un nœud X_i par : $ne(X_i) = pa(X_i) \cup ch(X_i)$, 'ne' pour 'neighbors'.*

En prenant comme exemple la figure 2.1, $ne(D) = \{A, B, F, G\}$ constitue l'ensemble des voisins du nœud D de la figure 2.1.

Définition 4 *(graphe complet) : On dit qu'un graphe non orienté est complet s'il existe un lien entre chaque paire de nœuds (c.f. figure 2.2 (a)). Ainsi, le sous-ensemble le plus petit (plus de deux nœuds) qui vérifie cette définition est un graphe de trois nœuds tous reliés entre eux.*

Par exemple, le graphe de la figure 2.2(b) ne contient aucun sous-ensemble complet de trois nœuds ou plus. En revanche, le graphe de la figure 2.2(c) contient deux sous-ensembles complets de taille trois : $\{A, B, C\}$ et $\{D, E, F\}$.

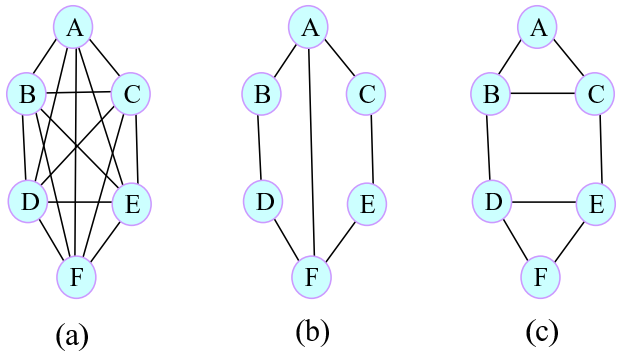


FIG. 2.2 – (a) Exemple d'un graphe complet (b) graphe sans sous-ensemble complet (c) graphe avec deux sous-ensembles complets.

Définition 5 *(moralisation et famille) : La moralisation d'un graphe orienté consiste à relier chaque paire de nœuds ayant un nœud fils commun. Le graphe obtenu est un graphe non orienté dont les arcs sont transformés en arêtes. L'ensemble constitué du nœud et de ses parents est appelé famille.*

Le graphe moral de la figure 2.3(a) est donné par la figure 2.3(b) où les arêtes entre les nœuds B, C et D, E sont rajoutées. Ainsi le nœud D et ses parent $pa(D) = \{B, C\}$ constituent une famille (de même pour le nœud F et ses parent $pa(F) = \{D, E\}$.

Définition 6 *(circuit et cycle) : Un cycle dans un graphe non orienté est un chemin fermé dans lequel le nœud de départ et le nœud d'arrivée sont identiques. On parlera d'un circuit si le graphe est orienté.*

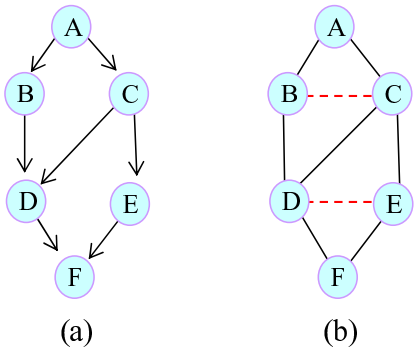


FIG. 2.3 – (a)Exemple d'un graphe orienté (b) graphe moral correspondant

Le chemin $\{A, B, C, D, E, A\}$ du graphe de la figure 2.4(a) représente l'unique circuit de ce graphe. Par contre, le graphe de la figure 2.4(b) comporte plusieurs cycles : $\{A, B, C, D, E, A\}$, $\{A, E, G, D, C, B, A\}$, $\{F, G, E, F\}$,...

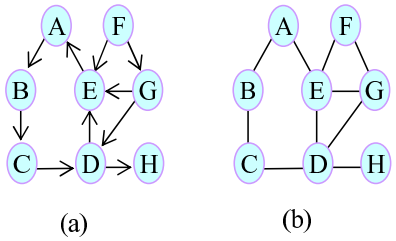


FIG. 2.4 – (a)Exemple d'un graphe orienté avec un seul circuit : $\{A, B, C, D, E, A\}$ (b) exemple d'un graphe non orienté avec plusieurs cycles : $\{A, B, C, D, E, A\}$, $\{A, E, G, D, C, B, A\}$, $\{F, G, E, F\}$,...

2.3.2 Graphes triangulés

Les graphes triangulés constituent une représentation graphique d'un type de modèles probabilistes connu sous le nom de *modèles décomposables* [Castillo *et al.*, 1997]. Les graphes triangulés sont désignés également sous le nom de circuits rigides [Dirac, 1961] ou "chordal graphs" [Gavril, 1974]. Dans ce qui suit, je présente les graphes triangulés et je donne les algorithmes nécessaires à la triangulation d'un graphe. Avant cela, donnons quelques définitions.

Définition 7 *(arc briseur) : un arc briseur ou « Chord » est une arête reliant deux nœuds non consécutifs dans un cycle. Cette arête brise le cycle et décompose celle-ci en deux petits cycles (cf. figure 2.5(b), (c), (d)).*

Définition 8 *(graphe triangulé) : Un graphe non orienté est dit triangulé si et seulement si pour chaque cycle de longueur quatre ou plus il existe un arc briseur (cf. figure 2.5(b), (c), (d)).*.

D'après les définitions données ci-dessus, un graphe peut être triangulé en brisant les cycles de longueur 4 ou plus en y ajoutant des arêtes. Ce processus est appelé *triangulation*.
Il y a plusieurs façons de trianguler un graphe mais il est préférable d'ajouter le moins de

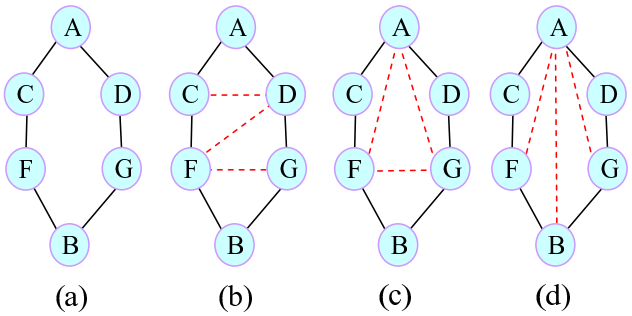


FIG. 2.5 – Plusieurs arcs briseurs pour le même graphe.

liens possibles pour préserver la structure du graphe original autant que possible. Ainsi, le graphe présenté sur la figure 2.5(a) admet plusieurs graphes triangulés (figures 2.5(b), 2.5(c) et 2.5(d)). Dans la littérature, on peut trouver plusieurs algorithmes de triangulation [Huang et Darwiche, 1994], [Kjærulff, 1990] et [Kjærulff, 1992]. Cependant, trouver un algorithme de triangulation optimal[6] est un problème NP-difficile [Kjærulff, 1992], [Yannakakis, 1981].

2.3.3 Algorithme de triangulation

L'algorithme *(MCS) (Maximum Cardinality Search)* décrit ci-dessous, transforme un graphe en un graphe triangulé. Cet algorithme est de complexité linéaire qui dépend du nombre de nœuds et de liens du réseau [Tarjan et Yannakakis, 1984]. Pour présenter l'algorithme *MCS*, donnons quelques définitions.

Définition 9 *(numérotation) : soit un ensemble de nœuds* $V = \{X_1, X_2..., X_n\}$. *Une numérotation* ∇ *est une bijection* $\nabla : \{1, ..., n\} \rightarrow \{X_1, X_2, ...X_n\}$ *qui affecte un entier à chaque nœud. Ainsi,* $\nabla(i)$ *désigne le* $i^{ème}$ *nœud dans la numérotation.*

Définition 10 *(numérotation parfaite) : Étant donné un graphe non orienté* $G = (V, E)$, *la numérotation* ∇ *est dite parfaite pour les nœuds du graphe* G, *si le sous graphe de nœuds :*

$$ne(\nabla(i)) \cap \{\nabla(1), ..., \nabla(i-1)\}$$

est complet pour $i = 2, ..., n$. *Rappelons que* $ne(\nabla(i))$ *représente l'ensemble des voisins du nœud numéro* $\nabla(i)$. *L'algorithme 5 produit pour chaque nœud du graphe* G *une numérotation parfaite.*

Exemple : La figure 2.6(a) montre une numérotation parfaite des nœuds du graphe : $\nabla(1) = A, \nabla(2) = B, \nabla(3) = C, ...$ Vérifions si les conditions de la numérotation parfaite données par l'algorithme ci-dessus sont satisfaites :

- pour $i = 2$, $ne(\nabla(2)) \cap \{\nabla(1)\} = ne(B) \cap \{A\} = \{A, C, D, E\} \cap \{A\} = \{A\}$, qui est un ensemble trivialement complet.
- pour $i = 3$, $ne(\nabla(3)) \cap \{\nabla(1), \nabla(2)\} = \{A, B, E, F\} \cap \{A, B\} = \{A, B\}$, est complet, puisque le lien $A - B$ existe dans le graphe.
- pour $i = 4$, $ne(\nabla(4)) \cap \{\nabla(1), \nabla(2), \nabla(3)\} = \{B, C, D, I\} \cap \{A, B, C\} = \{B, C\}$, est aussi complet.

[6] Selon le nombre de liens ajoutés.

Algorithme 5 Numérotation parfaite

Données : Un graphe non orienté $G = (V, E)$ et un nœud initial X_j (pris au hasard).
Résultats : Une numérotation parfaite ∇ des nœuds du graphe G.

Initialisation : $X_j \leftarrow \nabla(1), NUM = \{X_j\}$.

Pour $(i = 2;\ i \leq n)$ **Faire**
Choisir un nœud $X_k \in \{V - NUM\}$ tel que : $|ne(X_k) \cap NUM|$ est de cardinalité maximale.
$X_k \leftarrow \nabla(i)$.
Ajouter X_k à NUM.
Fin Pour

On peut vérifier que : $ne(\nabla(i)) \cap \{\nabla(1), ..., \nabla(i-1)\}$ est complet pour $i = 5, ..., 9,$. Ainsi, ∇ est une numérotation parfaite du graphe 2.6(a).
Notons qu'une numérotation parfaite n'est pas nécessairement unique (l'algorithme 5 dépend d'un nœud initial, si ce dernier change, on aura une autre numérotation). Le graphe 2.6(b) donne une autre numérotation parfaite pour le même graphe. Notons également qu'il existe des graphes sans numérotation parfaite [Cowell *et al.*, 1999].

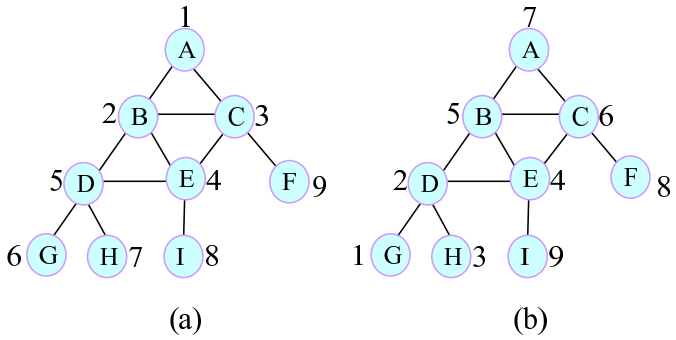


FIG. 2.6 – Deux numérotations parfaites pour le même graphe.

L'algorithme *MCS (Maximum Cardinality Search)* qui transforme un graphe en un graphe triangulé est fondé sur un théorème qui établit le lien entre la triangulation et la numérotation parfaite [Fouhy, 2003]. Le lecteur trouvera la preuve de ce théorème dans [Fulkerson et Gross, 1965].

Théorème 1 *Un graphe non orienté $G = (V, E)$ admet une numérotation parfaite si et seulement si il est triangulé (par conséquent, les graphes avec des cycles de longueur supérieur à 3 n'ont pas de numérotation parfaite).*

A partir de ce théorème, l'algorithme qui transforme un graphe en un graphe triangulé est donné par l'algorithme 6. Cet algorithme utilise la définition de l'arc briseur pour éliminer les cycles de longueur supérieur à 3.

Exemple : Étant donné le graphe de la figure 2.7(a). Appliquons l'algorithme 6 pour transformer ce graphe en un graphe triangulé. Le résultat du déroulement de l'algorithme 6

Algorithme 6 Triangulation d'un graphe non orienté G

Données : Un graphe non orienté à n nœud : $G = (V, E)$ et un nœud initial X_j (pris au hasard).

Résultats : Graphe triangulé $G^t = (V, E \cup E')$.

Initialisation : $E' = \emptyset$.

1. Soit $i = 1$, affecter i au noeud initial X_j ($X_j \leftarrow \nabla(i)$) et $NUM = \{X_j\}$.
2. $i \leftarrow i + 1$.
3. Choisir un nœud $X_k \in V$ tel que $\{ne(X_k) \cap NUM\}$ est maximum.
4. Donner au nœud X_k le numéro $\nabla(i)$.
5. Si $ne(X_k) \cap \{\nabla(1), ..., \nabla(i-1)\}$ est non complet, ajouter à E' les liens nécessaires pour rendre cet ensemble complet et aller à 1, autrement, aller à l'étape 6.
6. Si $i \neq n$ alors aller à l'étape 2.

est donné par le tableau 2.1. La figure 2.7(d) représente le graphe triangulé correspondant au graphe de la figure 2.7(a). Les figures 2.7(b) et 2.7(c) représentent les étapes intermédiaires du déroulement de l'algorithme.

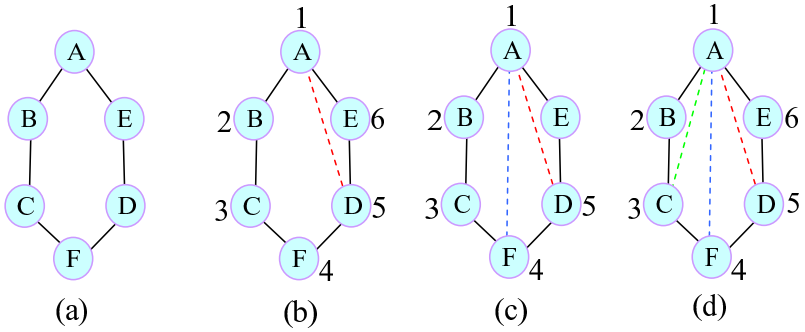


FIG. 2.7 – (a) graphe initial (b) premier arc ajouté pour briser le cycle $\{A, B, C, F, D, E, A\}$ (c) second arc ajouté pour briser le cycle $\{A, B, C, F, D, A\}$ (d) troisième arc ajouté pour briser le cycle $\{A, B, C, F, A\}$

i	$\nabla(noeud)$	ne(noeud)∩ NUM	NUM	E'
–	–	–	∅	∅
1	A	∅	A	∅
2	B	A	A,B	∅
3	C	B	A,B,C	∅
4	F	C	A,B,C,F	∅
5	D	F	A,B,C,F,D	∅
6	E	A,D	∅	A–D
1	A	∅	A	A–D
2	B	A	A,B	A–D
3	C	B	A,B,C	A–D
4	F	C	A,B,C,F	A–D
5	D	A,F	∅	A–D, A–F
1	A	∅	A	A–D, A–F
2	B	A	A,B	A–D, A–F
3	C	B	A,B,C	A–D, A–F
4	F	A,C	∅	A–D, A–F, A–C

TAB. 2.1 – Application de l'algorithme de triangulation (algorithme 6) sur le graphe représenté figure 2.7(a)

2.3.4 Identification des cliques

Définition 11 *(clique) : Une clique dans un graphe non orienté G est un sous graphe de G complet et maximal [Castillo* et al.*, 1997]. Il est complet si chaque paire de nœuds distincts est reliée par une arête. Il est maximal s'il n'est pas contenu dans un plus grand sous graphe complet.*

Exemple : La figure 2.8(a) représente un graphe non orienté qui contient sept cliques. Les sept cliques sont décrites dans la table 2.2. Si nous ajoutons sur le même graphe un ensemble de liens, le graphe contiendra alors un autre ensemble de cliques. En ajoutant des liens entre les paires de nœuds $\{(A,D),(B,E),(B,D)\}$ sur le graphe de la figure 2.8(a), les cliques C_1, C_2, C_3 et C_7 de l'exemple précédent ne respectent plus la définition de la clique et par conséquent, le graphe de la figure 2.8(b) contient les nouvelles cliques décrites dans la table 2.2.

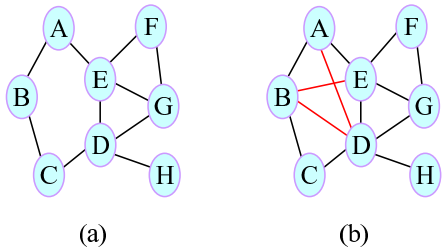


FIG. 2.8 – Exemple de deux graphes qui portent les mêmes noeuds et diffèrent par leur nombre de liens.

	clique (figure 2.8(a))	clique (figure 2.8(b))
C_1	$\{A, B\}$	$\{A, B, D, E\}$
C_2	$\{B, C\}$	$\{B, C, D\}$
C_3	$\{C, D\}$	$\{D, H\}$
C_4	$\{D, H\}$	$\{D, E, G\}$
C_5	$\{D, E, G\}$	$\{E, F, G\}$
C_6	$\{E, F, G\}$	
C_7	$\{A, E\}$	

TAB. 2.2 – Identification des cliques de la figure 2.8(a) et 2.8(b)

2.3.5 Algorithme d'identification des cliques

L'idée derrière cet algorithme est de regrouper un ensemble de variables dans une même clique. Le regroupement des variables ne se fait pas aléatoirement puisque les étapes de moralisation et de triangulation sont déjà faites au préalable. Donc chaque nœud fils et ses parents forment une famille. Cette famille appartient au moins à une clique. L'algorithme 7 identifie les cliques d'un graphe G à n nœuds.

Algorithme 7 Identification des cliques d'un graphe G à n nœuds

Données : Un graphe triangulé $G^t = (V, E \cup E')$.
Résultats : Ensemble de cliques C non optimal.

Initialisation : $i = 1,\ C = \emptyset$

tant que $(i \leq n)$ **Faire**
Chercher les voisins du nœud numéro i
Former une clique C_k pour chaque nœud i et ses voisins numérotés antérieurement et l'ajouter à C
Ordonner (ordre croissant) les cliques selon le nœud numéroté le plus élevé dans chaque clique
$i \leftarrow i + 1$
Fin tant que

L'algorithme 7 utilisé comme tel n'est pas optimal (redondance des cliques), dans la prochaine étape un algorithme d'optimisation de ces cliques est utilisé. L'algorithme d'optimisation des cliques (voir l'algorithme 8) opère comme suit : si une clique C_{i+1} de rang $i + 1$ contient toutes les variables de la clique C_i, alors la clique C_i peut être supprimée.

2.3.6 Chaîne de cliques

Une importante propriété des graphes triangulés est connue sous le terme de *running intersection*. Cette propriété donne aux cliques du graphe un certain ordre. Un ordre de cliques $(C_1, ..., C_m)$ d'un graphe non orienté satisfait la propriété *running intersection* si $C_i \cap (C_1 \cup ... \cup C_{i-1})$ est contenu dans au moins une des cliques $\{C_1, ..., C_{i-1}\}$, pour tout $i = 1, ..., m$ [Castillo *et al.*, 1997]. Une séquence de cliques $\{C_1, ..., C_{i-1}\}$ qui satisfait cette propriété est appelée *chaîne de cliques*. Certains graphes n'ont aucune chaîne de cliques alors que d'autres

Algorithme 8 Optimisation des cliques du graphe G à n nœuds

Données : Ensemble C de m cliques non optimal.
Résultats : Ensemble de cliques C optimal.

Initialisation : $i = 1$

tant que $(i \leq m)$ **Faire**
Identifier une clique $C_{i+1} \in C$ qui contient toutes les variables de la clique C_i
Supprimer la clique C_i
$i \leftarrow i + 1$
Fin tant que

graphes ont plus d'une chaîne. Le théorème énoncé ci-dessous caractérise les graphes avec au moins une chaîne de cliques.

Théorème 2 *Un graphe non orienté possède une chaîne de cliques si et seulement si il est triangulé.*

2.3.7 Graphe et arbre de jonction

Après avoir obtenu un ensemble de cliques optimal et ordonné, l'étape suivante consiste à relier ces cliques entre elles. C'est ce qu'on appelle par la suite, graphe ou arbre de jonction.

Définition 12 *(graphe de jonction) Étant donné un graphe $G = (X, L)$ et un ensemble de cliques $C = \{C_1, ..., C_m\}$, associé à X, tel que $X = C_1 \cup ... \cup C_m$, alors le graphe représenté par la paire $\widetilde{G} = (C, \widetilde{L})$ associée au graphe G est appelé graphe de jonction si $\widetilde{L}$ contient seulement des liens entre les cliques contenant des noeuds communs, c'est-à-dire, $(C_i, C_j) \in \widetilde{L} \Rightarrow C_i \cap C_j \neq \emptyset$ (voir l'exemple de la figure 2.9(a)).*

Notons que le graphe de jonction associé à un graphe donné est unique et d'après la définition donnée ci-dessus, le graphe de jonction est fortement relié même pour les graphes ayant un nombre restreint de nœuds [Castillo *et al.*, 1997]. Par conséquent, l'introduction des structures plus simples (par exemple sous forme d'arbre) et qui conserve la connexion des cliques rend les calculs plus rapides.

Définition 13 *(arbre de jonction) Un graphe de cliques associé à un graphe non orienté G est dit arbre de jonction s'il est sous forme d'arbre et si chaque ensemble de nœuds $\{X\}$ appartenant à deux cliques C_i, C_j ($\{X\} = C_i \cap C_j$), appartient également à chaque clique dans le chemin reliant les cliques C_i, C_j (voir l'exemple de la figure 2.9(b)).*

Le théorème donné par [Jensen, 1988] établit le lien entre l'arbre de jonction et la triangulation.

Théorème 3 *Un graphe non orienté admet un arbre de jonction si et seulement si il est triangulé.*

L'algorithme 9 génère l'arbre de jonction associé à un graphe triangulé en organisant les cliques de telle sorte qu'elles satisfassent la propriété de *chaîne de cliques*.

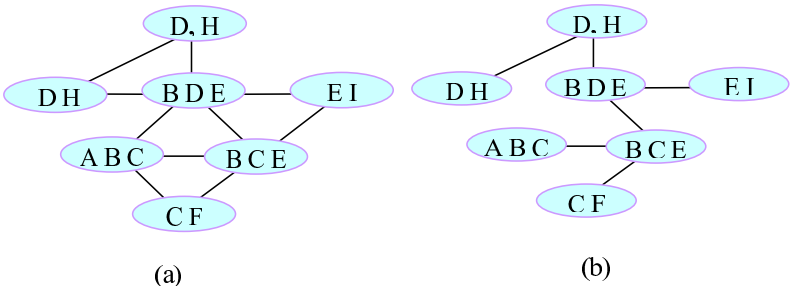


FIG. 2.9 – (a) Graphe de jonction (b) Arbre de jonction.

Algorithme 9 Arbre de jonction associé au graphe G à n nœuds

Données : Un graphe G triangulé.
Résultats : Arbre de jonction de m cliques associé à G.

Initialisation : pour une numérotation des cliques de i=1,...,m.
i=m

Organisation des cliques sous forme de *chaîne* de cliques
tant que $(1 \leq i)$ **Faire**
Pour chaque clique $C_i \in C$, choisir une clique C_k parmi $\{C_1, ..., C_{i-1}\}$ telle que le cardinal $|C_i \cap C_k|$ est maximal.
Ajouter le lien entre C_i et C_k
$i \leftarrow i - 1$
Fin tant que
Définir l'ensemble des séparateurs comme suit :
$S_i = C_i \cap C_{i+1}$.

2.4 Les modèles graphiques probabilistes

2.4.1 Introduction

Supposons que nous disposions d'un ensemble $U = \{X_1, ..., X_n\}$ de variables aléatoires discrètes. L'ensemble U peut être représenté graphiquement par un ensemble de nœuds et d'arcs noté $G = (V, E)$[7] dans lequel :

- V : représente l'ensemble des nœuds du graphe tel qu'il existe une bijection entre V et U ($U \leftrightarrow V$). Chaque nœud de G est donc associé à une et une seule variable de U et on note $V = \{X_1, X_2, ..., X_n\}$.
- E : désigne l'ensemble des arcs de G i.e. $E = \{e(i, j)\}$ où i et j désignent les nœuds X_i et X_j respectivement. Les arcs de la forme e(i,i) ne sont pas autorisés dans E.

Dans la littérature, on trouve deux grands types de graphes probabilistes : les graphes orientés *(DPIN)*[8] (voir la figure 2.10(a)) et les graphes non orientés *(UPIN)*[9] (voir la figure 2.10(b)). Les modèles à base de graphes non orientés sont souvent appelés champs de Markov aléatoires (Markov Random Fields)[Castillo *et al.*, 1997]. Les graphes dits *(DPIN)* sont souvent connus sous le nom de réseaux bayésiens (Bayes Networks), réseaux de croyances (Belief Networks) ou réseaux causaux (Causal Networks). Ces derniers *(DPIN)* sont devenus un outil qui a pour applications privilégiées : le diagnostic (panne, médical, bugs logiciels,...), la classification,... Citons quelques unes de ces applications :

- diagnostic dans le réseau téléphonique [Leray et Gallinari, 1998],
- aide au diagnostic de pannes des imprimantes [Skaanning *et al.*, 1999], et
- aide à déterminer le risque de rechute après transplantation médullaire [Suermondt et Arnylon, 1989].

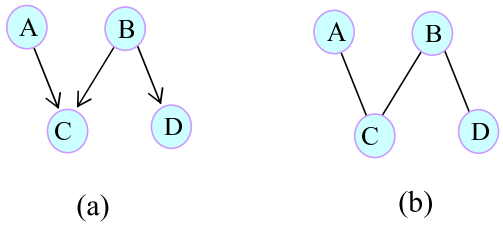


FIG. 2.10 – Exemple d'un graphe (a) orienté (b) non orienté

Définition 14 *(réseau bayésien) : Un réseau bayésien est défini par un ensemble de variables aléatoires (discrètes et/ou continues), par une structure exprimée sous la forme d'un graphe orienté sans circuit (DAG)*[10] *qui indique les relations de dépendance entre ces variables et par un ensemble de probabilités conditionnelles locales qui lie la valeur d'un noeud à celle de ses parents.*
Par la suite, on représentera un réseau bayésien par $G = (V, E)$ où V est l'ensemble des variables du réseau et E l'ensemble des liens entre les variables.

[7] On utilise ces deux lettres pour désigner les Vertices :V et Edges :E
[8] Directed Probabilistic Independence Networks
[9] Undirected Probabilistic Independence Networks
[10] Directed Acyclic Graph

2.4.2 Représentation de la distribution de probabilités par un réseau bayésien

Soit un ensemble U de variables aléatoires toutes binaires, tel que $U = \{X_1, ..., X_n\}$. La probabilité jointe $p(U) = p(X_1 = x_1, ..., X_n = x_n)$ de cet ensemble nécessite un tableau comprenant 2^n entrées, une taille particulièrement grande. En revanche, si nous savons que certaines variables ne dépendent en fait que d'un certain nombre spécifique de variables, alors nous pouvons restreindre le nombre d'entrées et par conséquent la taille mémoire et le temps de traitement. D'après la règle de probabilité conditionnelle donnée par l'équation 2.3 et celle de la distribution de probabilité jointe donnée par 2.4, on a :

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

et par conséquent

$$p(x,y) = p(x|y).p(y)$$

Une généralisation utile de la règle de produit donnée par la formule 2.4 est appelée *la règle de chaîne*. Si nous avons n événements $X_1, ..., X_n$, la probabilité jointe de $U = \{X_1, X_2, ..., X_n\}$ peut être écrite sous la forme de produits des n probabilités conditionnelles. Ce produit peut être trouvé par application répétée de l'équation 2.4, dans n'importe quel ordre [Pearl, 1988] :

$$p(X_1, X_2, ..., X_n) = p(X_n|X_{n-1}, ..., X_2, X_1)...p(X_3|X_2, X_1)p(X_2|X_1)p(X_1) \tag{2.16}$$

Supposons maintenant que les probabilités conditionnelles de certaines variables X_j ne soient pas dépendantes de tous les prédécesseurs de X_j (i.e. $X_1, X_2, ..., X_{j-1}$) mais seulement de ceux qui ont une influence directe sur X_j. Nous appellerons cet ensemble restreint de variables les *parents* de la variable X_j que l'on note : $pa(X_j)$. Alors, nous pouvons écrire : $p(X_j|X_1, ..., X_{j-1}) = p(X_j|pa(X_j))$. Ainsi, la formule 2.16 s'écrira :

$$p(X_1, X_2, ..., X_n) = \prod_j p(X_j|pa(X_j)) \tag{2.17}$$

Cette formule permet de simplifier significativement les calculs de probabilité. Donc, au lieu de spécifier la probabilité de X_j conditionnellement à toutes les réalisations de ses prédécesseurs $X_{j-1}, ..., X_1$, nous ne spécifierons que celles qui sont conditionnées par les éléments de $pa(X_j)$. La définition des parents pose donc la base théorique de la notion de relation modale entre des connaissances dans un réseau bayésien. En effet, il est possible de représenter toute sorte de modalités entre les variables (causale, temporelle, hiérarchique,...).
En général, une seule modalité est utilisée dans un même réseau et la plupart du temps, il s'agit de la dépendance ou indépendance conditionnelle. Cependant, dans la plupart des applications médicales, il s'agit de la causalité [Smaili *et al.*, 2005b]. Ceci permet de représenter l'influence directe d'une variable sur une autre. S'il existe un arc orienté allant d'une variable (nœud) A vers une variable (nœud) B alors A est une cause possible de B, ou encore A a une influence causale directe sur B.
Notons que dans tout ce document, l'influence d'un événement, d'un fait, ou d'une variable sur une autre est représentée par une flèche reliant les deux variables. Ainsi, étant donné un graphe G et une distribution de probabilité p. La décomposition donnée par la formule 2.17 est une condition nécessaire pour que le graphe G soit un réseau bayésien de distribution p, et que p admette une décomposition sous forme d'un produit représentée par le graphe G [Pearl, 2001].

L'exemple représenté figure 2.11 (tiré de [Murphy, 2001]) présente un réseau bayésien simple contenant quatre variables :

- (A) : Le ciel est nuageux (Cloudy).
- (B) : La pluie tombe (Rain).
- (C) : Un arroseur est en marche (Sprinkler).
- (D) : L'herbe est mouillée (Wet Grass).

La distribution de probabilité jointe du réseau représenté par la figure 2.11 est donnée par :

$$p(A, B, C, D) = p(A).p(B|A).p(C|A).p(D|B, C)$$

De ce fait, l'absence d'un arc allant de $(A : Cloudy)$ vers $(D : WetGrass)$ signifie que le ciel nuageux n'a pas une influence directe sur l'état de l'herbe. Autrement dit, le fait que l'herbe soit mouillée est conditionné par le fait que la pluie tombe (B) et/ou que l'arroseur (C) soit en marche. Ce qui rend inutile la connaissance sur la variable "ciel nuageux".

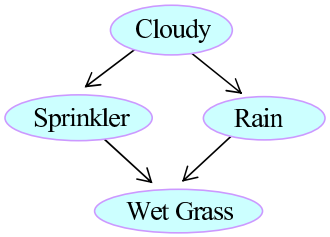


FIG. 2.11 – Exemple d'un réseau bayésien à quatre variables.

2.4.2.1 Factorisation de la *JPD* dans un arbre de jonction

Nous avons vu que l'inférence est coûteuse en terme de temps de calcul si on utilise une *JPD* construite sur l'ensemble des variables. Ce problème est considérablement simplifié si la *JPD* peut être factorisée comme produit de distributions de probabilités conditionnelles *CPD*. La question qui se pose maintenant est quelle est la classe de *JPD* qui peut être représentée par un graphe ? D'après [Castillo *et al.*, 1997], chaque graphe a une factorisation équivalente à une *JPD*. Cependant, toutes les *JPDs* ne peuvent être représentées par un graphe.

Définition 15 *Soient $C_1, ..., C_m$ des sous-ensembles de variables de $X = \{x_1, ..., x_n\}$. Si la JPD de $x_1, ..., x_n$ peut être écrite sous forme de produit de m fonctions non négatives $\Psi_i (i = 1, ..., m)$ comme suit :*

$$P(x_1, ..., x_n) = \prod_{i=1}^{m} \Psi_i(c_i) \tag{2.18}$$

où c_i est une réalisation de C_i, alors la formule 2.18 est une factorisation *de la distribution de probabilité jointe (JPD). La fonction Ψ_i est appelée facteur* potentiel *ou* potentiel *de la JPD [Castillo* et al.*, 1997] [Cowell* et al.*, 1999].*

Définition 16 *(factorisation de la JPD associée à un graphe orienté sans circuit (DAG)) : On dit qu'une distribution de probabilité jointe JPD admet une factorisation selon un graphe (DAG), si la JPD peut être exprimée par :*

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} p(x_i|\pi_i) \tag{2.19}$$

où $p(x_i|\pi_i)$ est la distribution de probabilité conditionnelle CPD de la variable x_i sachant ses parents π_i.

2.4.3 Exemple de transformation d'un graphe en un arbre de jonction

Après avoir donné toutes les étapes de transformation d'un graphe en un arbre de jonction, donnons à présent un exemple où l'on peut appliquer pas à pas tous les algorithmes vus jusqu'à présent.
Étant donné le graphe représenté figure 2.12(a), nous construisons l'arbre de jonction équivalent par les étapes suivantes :

1. Moralisation :
   Relier chaque paire de nœuds de même nœud fils. Le graphe moralisé G^m associé au graphe de la figure 2.12(a) est obtenu en reliant les nœuds C, E et E, G ayant le même nœud fils D et F respectivement. Le graphe moralisé G^m est donné par la figure 2.12(b).

2. Triangulation :
   Ajouter sélectivement des arcs au graphe moralisé G^m pour former un graphe triangulé G^t. En utilisant l'algorithme de triangulation 6, on obtient le graphe triangulé donné par la figure 2.12(c). Notons qu'on a rajouté le lien B, H afin d'éliminer le cycle A, B, C, E, H, A et le second lien H, C pour éliminer le cycle B, C, E, H, B.

3. Identification des cliques :
   En utilisant l'algorithme d'identification des cliques (voir l'algorithme 7) puis celui d'optimisation donné par l'algorithme 8, on obtient l'ensemble des cliques présentées dans les tableaux 2.3 et 2.4.

i	1	2	3	4	5	6	7	8
Clique C_i	A	A,B	A,B,H	B,H,C	C,H,E	C,E,D	H,G,E	E,G,F

TAB. 2.3 – Ensemble des cliques non optimal de la figure 2.12(c).

i	1	2	3	4	5	6
Clique C_i	A,B,H	B,H,C	C,H,E	C,E,D	H,G,E	E,G,F

TAB. 2.4 – L'ensemble des cliques optimal de la figure 2.12(c).

   Notons que l'ordre des cliques donné par le tableau 2.4 satisfait la propriété de "running intersection", c'est à dire : $C_1 \cap C_2 = \{B, H\} \subset C_1$, $C_3 \cap (C_1 \cup C_2) = \{C, H\} \subset C_2$, et ainsi de suite pour C_4, C_5 et $C6$.

4. Construction de l'arbre de jonction :
   Afin d'obtenir l'interconnexion des cliques, utilisons l'algorithme 9. Cet algorithme utilise la propriété de "running intersection" pour construire l'arbre de jonction associé au

graphe triangulé. L'arbre de jonction associé au réseau initial est donné figure 2.12(f) (entre chaque cliques adjacentes C_i, C_j on définit un séparateur $S_i = C_i \cap C_j$). Le graphe associé au réseau initial est donné figure 2.12(e).

5. Factorisation de la *JPD* :
   La distribution de probabilités jointe (*JPD*) associée au graphe 2.12(a) est donnée par :
   $X = \{A, B, C, D, E, F, G, H\}$
   $P(X) = p(A)p(B|A)p(C|B)p(D|C, E)p(E|H)p(F|E, G)p(G|H)p(H|A)$

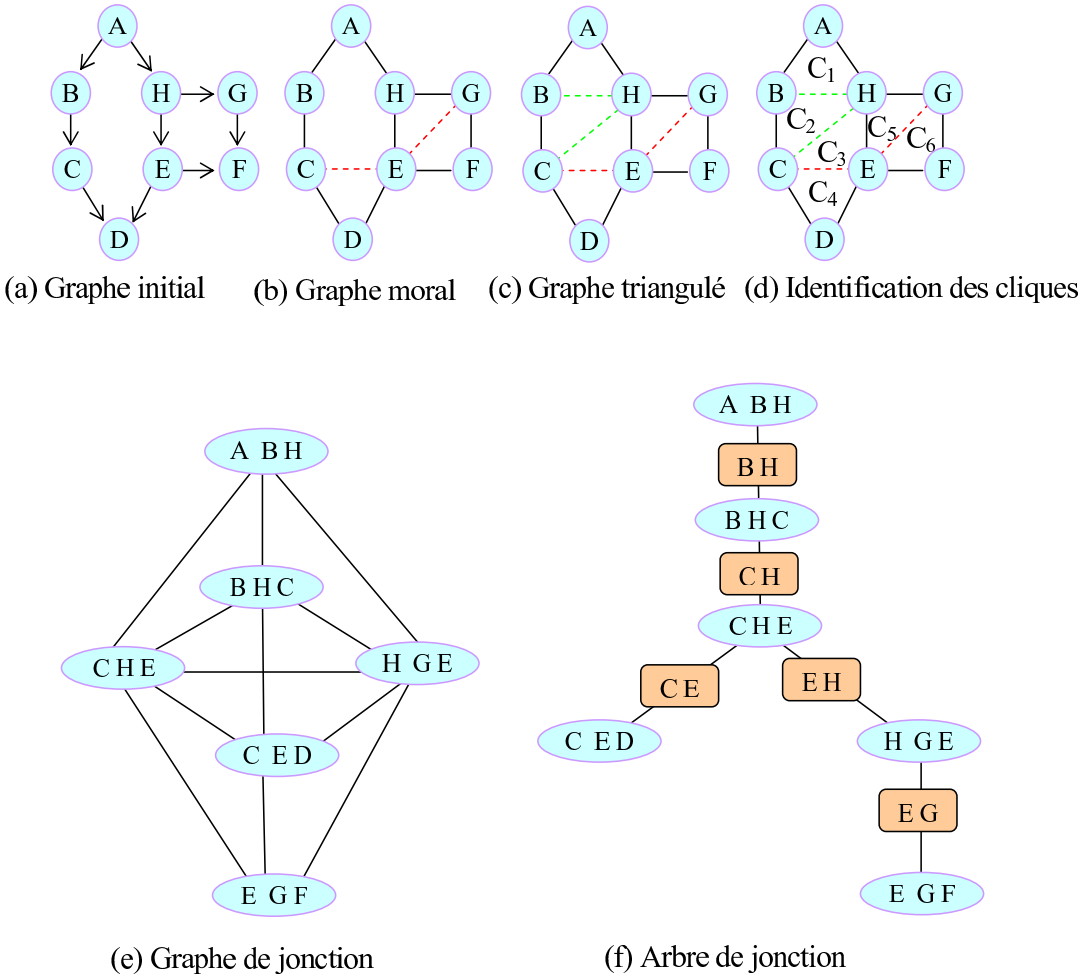


FIG. 2.12 – Étapes de transformation d'un graphe orienté en un arbre de jonction.

2.5 Moteur d'inférence

2.5.1 Introduction

Le but d'utiliser un réseau bayésien est d'inférer de nouveaux faits quand on observe une nouvelle information. Dans le domaine du diagnostic médical par exemple, la tâche d'un tel système consiste à obtenir un diagnostic pour un patient qui présente quelques symptômes (évidences). Le mécanisme qui extrait les conclusions dans les réseaux bayésiens est appelé

inférence.
Nous présentons dans ce qui suit un algorithme d'inférence exacte appelé *JLO* pour *Jensen, Lauritzen, Olesen* [Castillo *et al.*, 1997], les auteurs de cet algorithme. Cet algorithme utilise l'arbre de jonction afin d'effectuer la propagation des messages par de simples passages locaux à travers l'arbre de jonction.
L'algorithme *JLO* s'applique aux réseaux dont les variables sont discrètes [Lauritzen et Spiegehalter, 1988] et [Jensen *et al.*, 1990]. Des extensions pour des variables continues ou mixtes (discrètes et continues) ont été décrites dans [Lauritzen et Wermuth, 1989] et [Cowell *et al.*, 1999].

2.5.2 Phase d'initialisation de l'arbre de jonction

Avant qu'un arbre de jonction puisse être utilisé, il doit d'abord être initialisé pour fournir une représentation locale de chaque variable. Par la suite, l'évidence (observation) peut être entrée dans l'arbre de jonction initialisé, et le calcul local est effectué afin de calculer les probabilités marginales.
L'initialisation de l'arbre de jonction prend en compte les spécifications numériques des probabilités définies sur notre graphe initial (probabilités conditionnelles) et permet de les transformer pour obtenir la forme générale de la distribution jointe $P(x_1, ..., x_n)$ quand elle est représentée par un arbre de jonction. D'après l'équation 2.18, la *JPD* peut être factorisée sous forme d'un produit de fonctions non négatives appelées potentielles donnée par :

$$P(X) = \prod_{i=1}^{m} \psi_i(C_i)$$

Considérons à présent l'ensemble des séparateurs associés à chaque couple de cliques adjacentes dans l'arbre de jonction. On donne à chaque séparateur une fonction ψ_S définie de façon équivalente aux fonctions de potentiel ψ_C et qui est initialisé à 1. Étant donnée cette représentation, on peut factoriser la *JPD* de la façon suivante [Cowell *et al.*, 1999].

$$P(X) = \frac{\prod_{C_i \in C} \psi_{C_i}(X_{C_i})}{\prod_{S_j \in S} \psi_{S_j}(X_{S_j})} \tag{2.20}$$

où $\psi_{C_i}(X_{C_i})$ et $\psi_{S_j}(X_{S_j})$ sont les potentiels des cliques $C_i \in C$ et des séparateurs $S_i \in S$ respectivement. L'initialisation de l'arbre de jonction est effectuée par l'algorithme 10.

Exemple : Essayons de dérouler cet algorithme sur l'arbre de jonction de la figure 2.12(f). D'après cet algorithme, on peut affecter les variables du réseau comme suit :
En tenant compte de l'arbre de jonction de la figure 2.12(f), - $C_1 = \{A, B, H\}$ à cette clique on affecte les variables A, B et H.
- $C_2 = \{B, H, C\}$ à cette clique on affecte seulement la variable C, car les variables B et H sont auparavant affectées à la clique C_1.
- $C_3 = \{C, H, E\}$ à cette clique on affecte seulement la variable E.
- $C_4 = \{C, E, D\}$ à cette clique on affecte seulement la variable D.
- $C_5 = \{H, G, E\}$ à cette clique on affecte seulement la variable G.
- $C_6 = \{E, G, F\}$ à cette clique on affecte seulement la variable F.

Le potentiel de chaque clique est donné par :
- $C_1 = \{A, B, H\}$: $\psi_{C_1}(A, B, H) = p(A)p(B|A)p(H|A)$

- $C_2 = \{B, H, C\} : \psi_{C_2}(B, H, C) = p(C|B)$
- $C_3 = \{C, H, E\} : \psi_{C_3}(C, H, E) = p(E|H)$
- $C_4 = \{C, E, D\} : \psi_{C_4}(C, E, D) = p(D|C, E)$
- $C_5 = \{H, G, E\} : \psi_{C_5}(H, G, E) = p(G|H)$
- $C_6 = \{E, G, F\} : \psi_{C_6}(E, G, F) = p(F|E, G)$

Le potentiel des séparateurs est donné par :
- $S_1 = \{B, H\} : \psi_{S_1}(B, H) = 1$
- $S_2 = \{C, H\} : \psi_{S_2}(C, H) = 1$
- $S_3 = \{C, E\} : \psi_{S_3}(C, E) = 1$
- $S_4 = \{E, H\} : \psi_{S_4}(E, H) = 1$
- $S_5 = \{E, G\} : \psi_{S_5}(E, G) = 1$

Notons qu'en utilisant la formule 2.20, la distribution de probabilité jointe est équivalente à celle donnée par la formule 2.19[11]. Par conséquent, la factorisation de la *JPD* associée à un arbre de jonction est équivalente à celle donnée par un graphe orienté sans circuit *DAG*.
La forme particulière que prend la distribution de probabilités jointe $P(X)$ donnée par l'équation 2.20 est un résultat très important de la théorie des réseaux probabilistes car à partir de cette représentation de la *JPD*, nous pouvons effectuer des calculs consistants de manière complètement locale [Castillo *et al.*, 1997]. Nous discuterons du mécanisme de ces calculs dans le paragraphe suivant.

2.5.3 Calcul local sur l'arbre de jonction

Soient deux cliques adjacentes C_i et C_j dans l'arbre de jonction et soit $S_k = C_i \cap C_j$ le séparateur de ces deux cliques. Le passage d'un flux entre les cliques C_i et C_j correspond à la mise à jour de l'information (le potentiel) de la clique destination suivant l'information (le potentiel) de la clique source (voir la figure 2.13). Ce flux induit une nouvelle représentation de la distribution de probabilités jointe $P(X)$.
Le point important de ce mécanisme est que ces potentiels sont modifiés de manière qu'à chaque instant, la distribution de probabilités jointe donnée par l'équation 2.20 reste une représentation de la distribution de probabilités jointe $P(X)$. On met à jour les potentiels par les formules suivantes [Cowell *et al.*, 1999] [Castillo *et al.*, 1997].

$$\psi^*_{S_k}(X_{S_k}) = \sum_{C_i \setminus S_k} \psi_{C_i}(X_{C_i}) \tag{2.21}$$

où $\psi^*_{S_k}(X_{S_k})$ [12] est la marginalisation sur l'ensemble des variables qui appartiennent à la clique C_i mais pas au séparateur S_k. La clique C_j est mise à jour par :

$$\psi^*_{C_j}(X_{C_j}) = \psi_{C_j}(X_{C_j})\lambda_{S_k}(X_{S_k}) \tag{2.22}$$

où

$$\lambda_{S_k}(X_{S_k}) = \frac{\psi^*_{S_k}(X_{S_k})}{\psi_{S_k}(X_{S_k})} \tag{2.23}$$

[11] Puisque ψ_{S_j} est initialisé à 1, le produit $\prod_{S_j \in S} \psi_{S_j}(X_{S_j}) = 1$, on trouve la factorisation de la *JPD* donnée par la équation 2.19.
[12] Le symbole * représente le nouveau potentiel calculé.

Algorithme 10 Initialisation de l'arbre de jonction associé à un réseau bayésien à n nœuds, m cliques et p séparateurs.

Données : Arbre de jonction.
Résultats : Initialisation de l'arbre de jonction.

Initialisation : $i = 1, k = 2$
tant que $(i \leq n)$ **Faire**
Affecter de façon unique chaque variable x_i à la clique contenant tous ses parents.
$i \leftarrow i + 1$
Fin tant que
Soit A_j $(1 \leq j \leq m)$ l'ensemble des variables affectées à la clique numéro j.
j=1
tant que $(j \leq m)$ **Faire**
Définir le potentiel de chaque clique par le produit des variables affectées à cette clique (si $A_i = \emptyset$ alors $(\psi_i(C_i) = 1)$) :

$$\psi_{C_j}(X_{C_j}) = \prod_{x_l \in A_j} p(x_l|\pi_{x_l})$$

où X_{C_j} représente les variables contenues dans la clique C_j
$j \leftarrow j + 1$
Fin tant que

tant que $(k < p)$ **Faire**
Initialiser la fonction potentielle de chaque séparateur $\psi_{S_k}(X_{S_k})$ à 1
$k \leftarrow k + 1$
Fin tant que

le terme $\lambda_{S_k}(X_{S_k})$ est le facteur de mise à jour du potentiel de la clique cible.

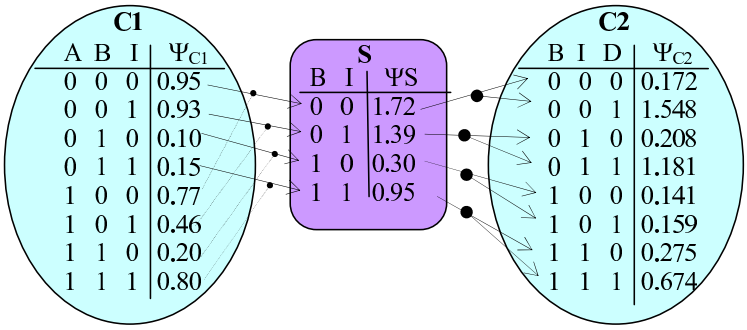


FIG. 2.13 – Passage de messages entre deux cliques voisines $C1$ et $C2$

Pour compléter l'algorithme, un agencement de tels flux peut être défini de manière à ce que toutes les cliques soient mises à jour avec les informations disponibles (toutes les cliques reçoivent une information de chacune de leur voisine). L'agencement le plus simple consiste en un schéma en deux phases où l'on désigne une clique comme étant la racine de l'arbre de jonction. Cette forme de passage des flux est commune à l'algorithme de [Pearl, 1988]. Les deux phases sont décrites comme suit :

- **Phase 1** : consiste à faire remonter les flux le long des arcs des feuilles jusqu'à la clique racine[13] (ce qui est marqué par des flèches pleines sur la figure 2.14).
- **Phase 2** : consiste à faire passer le flux de la racine vers les feuilles de l'arbre dans le chemin inverse (ce qui est marqué par des flèches en pointillées sur la figure 2.14).

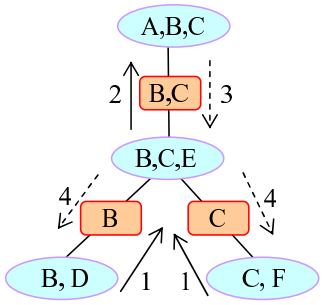


FIG. 2.14 – Propagation des messages entre les cliques du réseau

Le réseau atteint un état d'équilibre et par conséquent, un nouveau passage de messages dans l'arbre de jonction ne modifiera pas les potentiels des cliques.

2.5.4 Introduire une observation dans un arbre de jonction

Introduire une observation (évidence) dans un arbre de jonction, consiste à mettre à jour les distributions de probabilités des variables cachées (exemple : les maladies) selon les observations nouvellement disponibles (exemple : les symptômes). L'effet produit par cette évidence est pris en compte de la manière suivante : étant donné X_h un sous-ensemble de variables dites

[13] la clique (A, B, C) joue le rôle de la racine dans cet exemple.

cachées ou non observées, et soit $v = \{X_i = x_i^*, X_j = x_j^*, ...\}$ le sous-ensemble des variables observées noté par $X \backslash X_h$. Le moteur d'inférence calcul la distribution de probabilités suivante : $p(X_h|v)$. Pour ce faire, nous définissons une fonction indicatrice $g(x_i)$ pour les évidences telle que :

$$g(x_i) = \begin{cases} 1 & \text{si } x_i = x_i^* \\ 0 & \text{sinon} \end{cases}$$

On affecte chaque variable observée X_i à une clique, c'est ce que l'on appelle entrer l'évidence dans la clique. Soit X_C l'ensemble des cliques dans lesquelles l'évidence est connue. La prise en compte de cette observation se traduit par le produit de chaque potentiel $\psi_C(X_C)$ par $g(x_i)$. Dans ce cas, $\psi_C(X_C)$ est remplacée par :

$$\psi_C(X_C) = \psi_C(X_C). \prod_{X_i \in X_C} g(x_i)$$

On peut maintenant propager ces nouvelles évidences à travers l'arbre de jonction, grâce à la fonction de mise à jour décrite précédemment (phase 1, phase 2). Une fois que les flux sont passés le long des feuilles jusqu'à la racine et inversement de la racine vers les feuilles, l'arbre atteint un état d'équilibre. Par conséquent, on obtient une nouvelle représentation des potentiels en fonction de l'évidence. Le potentiel local de chaque clique de l'arbre est donné par l'équation :

$$\psi_{C_i}(X_{C_i}) = p(X_{C_i}, v) \tag{2.24}$$

Finalement, la distribution *a posteriori* des variables cachées d'une clique C_i : $X_h^{C_i}$ sachant l'évidence v est obtenue trivialement par :

$$p(X_h^{C_i}|v) = \frac{p(X_{C_i}, v)}{p(v)} \tag{2.25}$$

où $p(v)$ est un facteur de normalisation obtenu en marginalisant le potentiel de toutes les variables de la clique. Ainsi, le facteur de normalisation est donné par :

$$p(v) = \sum_{X_{C_i}} p(X_{C_i}, v) \tag{2.26}$$

2.6 Réseau bayésien à variables continues

Les réseaux bayésiens introduits jusqu'à présent, utilisent des variables aléatoires qui prennent un nombre de valeurs fini ou dénombrable (l'ensemble de définition est inclus dans N), on parle de variables discrètes. La distribution de probabilités *a priori* ou conditionnelle de chaque variable est représentée par des tables (de même pour le potentiel de chaque clique). Dans cette section, nous étudions le cas où les variables aléatoires pouvent prendre toute valeur réelle (l'ensemble de définition contient un intervalle de $\Re$), on parle de variables continues et par conséquent, de réseau bayésien à variables continues (ou simplement réseau bayésien continu).

2.6.1 Distribution normale

La distribution normale (également appelée loi de Gauss 1855 ou simplement Gaussienne, en l'honneur du grand mathématicien allemand Karl Friederich Gauss 1777-1855) est la loi statistique la plus répandue est la plus utile. Elle représente beaucoup de phénomènes aléatoires. De plus, de nombreuses lois statistiques peuvent être approchées par la loi normale. Dans le cas scalaire, la loi normale d'une variable X est caractérisée par deux paramètres : la moyenne μ et la variance σ^2. La fonction de densité est donnée par :

$$P(X) = \mathcal{N}(X; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{2.27}$$

La notation $\mathcal{N}(X; \mu, \sigma^2)$ représente la variable X par une distribution normale de moyenne μ, est de variance σ^2. Nous utilisons également la notation $\mathcal{N}(\mu, \sigma^2)$ sans noter explicitement la variable aléatoire X.
Dans le cas où la variable $X = x_1, ..., x_n$ est un vecteur, la loi normale de X est caractérisée par deux paramètres : le vecteur moyenne $\vec{\mu}$ et la matrice de covariance Σ. En général, on dit que le vecteur $X = x_1, ..., x_n$ a une distribution normale $\mathcal{N}(\vec{X}; \vec{\mu}, \Sigma)$ où $\vec{\mu}$ est un vecteur de taille n et Σ est une matrice symétrique définie positive de taille $n \times n$ si :

$$P(X) = \mathcal{N}(\vec{X}; \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})) \tag{2.28}$$

Notons que A^T dénote la transposée du vecteur A.

2.6.2 Loi de Gauss linéaire

Dans le cas où la variable $X = x_1, ..., x_n$ est un vecteur, nous pouvons ordonner ces variables de telle sorte qu'on peut les représenter sous forme de réseau bayésien. D'après [Uri, 2002], nous pouvons utiliser la définition de la distribution conditionnelle donnée par l'équation 2.29 pour représenter le vecteur X comme un réseau bayésien.

$$P(x_i|x_1, ..., x_{i-1}) = N(x_i; b_{i,0} + \sum_{j=1}^{i-1} \beta_{i,j} x_j, \sigma_i^2) \tag{2.29}$$

Pour $1 \leq j < i$, nous créons un arc allant de x_j vers x_i si et seulement si $\beta_{i,j} \neq 0$. La distribution de probabilités conditionnelles (CPD) de x_i est appellée (CPD) linéaire est suit la forme donnée par l'équation 2.29 (après avoir éliminer les poids $\beta_{i,j} = 0$). Notons que le cas particulier d'une variable sans parent représente une simple variable gaussienne. Le réseau bayésien dans lequel toutes les distributions de probabilités conditionnelles sont linéaires, est appelé linéaire gaussien ou Linear Gaussian (LG).

2.6.3 Représentation des potentiels dans le cas continu

Dans le cas purement Gaussien (toutes les variables sont de type LG), le potentiel des cliques peut être représenté sous différentes formes : canonique, moment [Lauritzen, 1992], [Murphy, 2002]. Il s'avère que certaines opérations sont plus faciles à exprimer sous forme canoniques et d'autres sont plus faciles à exprimer sous forme moment. D'après [Cowell *et al.*, 1999], [Murphy, 2002], nous pouvons représenter la distribution normale sous la forme :

– **moment** :

$$\phi(x;\ p,\vec{\mu},\Sigma) \stackrel{def}{=} p \times \exp\{-\frac{1}{2}(x-\vec{\mu})^T\Sigma^{-1}(x-\vec{\mu})\} \tag{2.30}$$

où

$$p = (2\pi)^{-n/2}|\Sigma|^{-1/2} \tag{2.31}$$

est la constante de normalisation qui vérifie : $\int_x \phi(x;p,\vec{\mu},\Sigma) = 1$ et n est la dimension de X.

– **canonique** :

$$\phi(x;\ g,\vec{h},K) \stackrel{def}{=} \exp(g + x^T\vec{h} - \frac{1}{2}x^TKx) \tag{2.32}$$

Dans la terminologie des fonctions exponentielles, g, h et K sont appelés les caractéristiques canoniques, et ils sont liés aux caractéristiques de la forme moment par les équations suivantes :

$$K = \Sigma^{-1} \tag{2.33}$$

$$h = \Sigma^{-1}\vec{\mu} \tag{2.34}$$

$$g = \log p - \frac{1}{2}\vec{\mu}^TK\vec{\mu} \tag{2.35}$$

2.6.4 Convertir la Loi de Gauss linéaire sous forme canonique

Dans le cas où la variable aléatoire est un vecteur, la distribution de probabilités conditionnelles de cette variable est donnée par [Murphy, 2001] :

$$\begin{aligned} f(x|z) &= c\ \exp[-\frac{1}{2}((x-\vec{\mu}-\beta^Tz)^T\Sigma^{-1}(x-\vec{\mu}-\beta^Tz)] \\ &= \exp[-\frac{1}{2}(x\ z)\begin{pmatrix} \Sigma^{-1} & -\Sigma^{-1}\beta^T \\ -\beta\Sigma^{-1^T} & \beta\Sigma^{-1}\beta^T \end{pmatrix}\begin{pmatrix} x \\ z \end{pmatrix} + \\ &\quad (x\ z)\begin{pmatrix} \Sigma^{-1}\vec{\mu} \\ -\beta\Sigma^{-1}\vec{\mu} \end{pmatrix} - \frac{1}{2}\vec{\mu}^T\Sigma^{-1}\vec{\mu} + log\ c] \end{aligned}$$

où $c = (2\pi)^{-n/2}|\Sigma|^{-\frac{1}{2}}$.

Nous obtenons par identification avec l'équation 3.31 les caractéristiques canoniques suivant le cas :

– **Vecteur** :

$$g = -\frac{1}{2}\vec{\mu}^T\Sigma^{-1}\vec{\mu} - \frac{n}{2}log(2\pi) - \frac{1}{2}log|\Sigma| \tag{2.36}$$

$$\vec{h} = \begin{pmatrix} \Sigma^{-1}\vec{\mu} \\ -\beta\Sigma^{-1}\vec{\mu} \end{pmatrix} \tag{2.37}$$

$$K = \begin{pmatrix} \Sigma^{-1} & -\Sigma^{-1}\beta^T \\ -\beta\Sigma^{-1^T} & \beta\Sigma^{-1}\beta^T \end{pmatrix} \tag{2.38}$$

– **Scalaire** : dans le cas scalaire, on aura $\Sigma^{-1} = 1/\sigma$, $\vec{\mu} = \mu$, $\beta = b$ et $n = 1$. Ainsi les caractéristiques canoniques g, h et K sont données par :

$$g = \frac{-\mu^2}{2\sigma} - \frac{1}{2} log(2\pi\sigma) \tag{2.39}$$

$$h = \frac{\mu}{\sigma} \begin{pmatrix} 1 \\ -b \end{pmatrix} \tag{2.40}$$

$$K = \frac{1}{\sigma} \begin{pmatrix} 1 & -b^T \\ -b & bb^T \end{pmatrix} \tag{2.41}$$

2.6.5 Opérations sur la forme canonique

Une fois que nous avons la forme canonique de chaque variable, nous pouvons calculer le potentiel initial de chaque clique en multipliant les distributions des probabilités de chaque variable affectée à cette clique.
Nous désignons par la suite la forme canonique par la fonction quadratique $\zeta(X; K, \vec{h}, g)$ (ou simplement par $\zeta(K, \vec{h}, g)$). Par définition, la forme canonique est donnée par la formule 3.31 et il est possible d'effectuer diverses opérations sur cette forme :

– **Initialisation** : Une variable aléatoire représentée par la forme canonique $\zeta(K, \vec{h}, g)$ peut être initialisée par :

$$K = 0,\ \vec{h} = \vec{0},\ g = 0$$

– **Produit** : le produit de deux variables aléatoires écrites sous forme canonique : $\zeta(K_1, \vec{h}_1, g_1)$ et $\zeta(K_2, \vec{h}_2, g_2)$ est définie par :

$$\zeta(K_1, \vec{h}_1, g_1) \times \zeta(K_2, \vec{h}_2, g_2) \overset{def}{=} \zeta(K_1 + K_2,\ \vec{h}_1 + \vec{h}_2,\ g_1 + g_2)$$

– **Division** : la division de deux variables aléatoires représentées par $\zeta(K_1, \vec{h}_1, g_1)$ et $\zeta(K_2, \vec{h}_2, g_2)$ est définie d'une manière semblable à la multiplication en remplaçant l'addition par la soustraction :

$$\frac{\zeta(K_1, \vec{h}_1, g_1)}{\zeta(K_2, \vec{h}_2, g_2)} \overset{def}{=} \begin{cases} \zeta(K_1 - K_2,\ \vec{h}_1 - \vec{h}_2,\ g_1 - g_2) & \text{si } \zeta(K_2,\ \vec{h}_2,\ g_2) \neq 0 \\ 0 & \text{sinon} \end{cases}$$

– **Marginalisation** : étant donné le vecteur suivant :

$$\vec{y} = \begin{pmatrix} \vec{y}_1 \\ \vec{y}_2 \end{pmatrix}$$

avec $\vec{y}_1$ de dimension p et $\vec{y}_2$ de dimension q. Le vecteur $\vec{y}$ est caractérisé par :

$$\vec{h} = \begin{pmatrix} \vec{h}_1 \\ \vec{h}_2 \end{pmatrix}$$

et

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

nous pouvons alors utiliser le lemme 1 suivant pour la marginalisation [Cowell *et al.*, 1999].

Lemme 1 : L'intégrale $\int \phi(\vec{y_1}, \vec{y_2})\, d\vec{y_1} = \phi(\vec{y_2};\ \widetilde{g}, \widetilde{\vec{h}}, \widetilde{K})$ est finie si et seulement si K_{11} est défini positif. Le résultat de cette intégrale est donné par les caractéristiques canoniques suivantes :

$$\widetilde{g} = g + \frac{1}{2}\{p\log(2\pi) - log|K_{11}| + \vec{h}_1^T K_{11}^{-1}\vec{h}_1\} \quad (2.42)$$

$$\widetilde{\vec{h}} = \vec{h}_2 - K_{21}K_{11}^{-1}\vec{h}_1 \quad (2.43)$$

$$\widetilde{K} = K_{22} - K_{21}K_{11}^{-1}K_{12} \quad (2.44)$$

2.6.6 Inférence dans un réseau bayésien à variables continues

L'inférence dans le cas d'un réseau bayésien à variables continues suit les mêmes étapes pour le cas d'un réseau bayésien à variables discrètes. Toutes les étapes de transformation du graphe initial en un arbre de jonction, l'initialisation de cet arbre ainsi que les algorithmes vus dans le cas discret sont applicables dans le cas continu. Notons qu'on utilise le produit et la division de la forme canonique vu ci-dessus pour l'initialisation de l'arbre de jonction ainsi que pour la propagation de l'évidence le long de cette arbre (voir les formules 2.21, 2.22 et 2.23 du cas discret).
A la différence du cas discret, l'évidence $Y = y_A^\star$ est entrée dans toutes les cliques et séparateurs où la variable Y apparaît. Ce qui induit une réduction des dimensions du vecteur $\vec{h}$ et de la matrice K. Ainsi, si $Y = y_A^\star$ est la variable observée et ψ_{C_i} est le potentiel de la clique C_i de caractéristiques canoniques $\zeta(g, h, K)$ données par :

$$\vec{h} = \begin{pmatrix} \vec{h}_1 \\ \vec{h}_A \end{pmatrix}$$

$$K = \begin{pmatrix} K_{11} & K_{1A} \\ K_{A1} & K_{AA} \end{pmatrix}$$

par identification à l'équation 3.31, le nouveau potentiel de la clique C_i, $\psi^*_{C_i}$ est donné par les caractéristiques canoniques suivantes [Cowell *et al.*, 1999] :

$$g^* = g + h_A y_A^\star - \frac{1}{2}K_{AA}(y_A^\star)^2 \quad (2.45)$$

$$h^* = h_1 - y_A^\star K_{A1} \quad (2.46)$$

$$K^* = K_{11} \quad (2.47)$$

Après la propagation de cette évidence le long de l'arbre, le lemme 1 est utilisé afin de calculer la distribution de probabilités des variables cachées sachant cette évidence.

2.7 Réseaux bayésiens hybrides

2.7.1 Introduction

Une autre caractéristique des réseaux bayésiens, est leur capacité à manipuler à la fois dans le même réseau les variables discrètes et continues. Tout au long de ce chapitre, nous avons décrit les réseaux bayésiens à variables discrètes et continues séparément. Cependant, plusieurs applications réelles incluent les deux types de variables dans un même réseau [Cowell *et al.*, 1999], [Uri, 2002], [Smaili *et al.*, 2008b]. Ces modèles sont dits réseaux bayésiens hybrides.

2.7.2 Distribution de probabilités dans le cas hybride

Dans le cas où certains noeuds possèdent des parents discrets, on peut spécifier une gaussienne pour chaque valeur prise par la variable discrète. Dans ce cas, on parle de la distribution de Gauss conditionnelle linéaire ou Conditional Linear Gaussian (CLG) [Cowell *et al.*, 1999]. Notons que dans la représentation (CLG), on autorise les arcs sortants d'un noeud discret vers les noeuds continus ($D \longrightarrow C$). Le cas contraire ($C \longrightarrow D$) n'est possible que dans le cas où la variable continue est observable [Murphy, 2002]. Notons également que si toutes les variables discrètes sont observables, la distribution de probabilités conditionnelles des variables continues sont tous Linear Gaussian (LG). Ainsi, la distribution jointe représentée par la loi de Gauss conditionnelle (CLG) est un mélange de gaussiennes où chaque gaussienne correspond à une instanciation des variables discrètes.

2.7.3 Marginalisation dans le cas hybride

Toute les étapes décrites jusqu'à présent pour les réseau bayésiens à variables discrètes ou continues sont valables pour les réseaux hybrides, à l'exception de la marginalisation qui ne respecte pas la notion de clôture de la (CG) distribution [14]. Considérons le potentiel $\phi(x, y, i, j)$ où x et y sont des scalaires, et i, j sont des variables discrètes binaires. Ainsi, ϕ est un mélange de quatre Gaussiennes (à deux dimensions). Maintenant, si on procède à une marginalisation sur les variables y et j, le résultat sera $\phi(x, i) = \sum_j \int_y \phi(x, y, i, j)$ qui reste toujours un mélange de quatre Gaussiennes (à une dimension). Par conséquent, la marginalisation sur les variables discrètes ne réduit pas nécessairement la taille du potentiel en terme d'entrées.
Si nous multiplions le potentiel ϕ par un autre potentiel $\psi(z, k)$, où z est un scalaire et k est une variable binaire, le résultat sera un mélange de 8 Gaussiennes (à deux dimensions), au lieu de 4 (pour chaque valeur de i et k, ϕ contient deux Gaussiennes). Ainsi, en propageant les messages le long de l'arbre de jonction, les potentiels deviennent des mélanges de Gausiennes avec de plus en plus de composants.
Pour réduire la taille du potentiel (augmentation en exponentielle), on adopte une approximation standard qui consiste à "collapser" les mélanges de Gausiennes en une seule Gausienne (c'est ce qu'on appelle "weak" marginalisation). Pour plus de détails sur cette approximation nous renvoyons le lecteur au fameux livre de Steffen L. Lauritzen [Lauritzen, 1996]. Notons que si les paramètres (moyenne et covariance) des variables continues sont indépendants des variables discrètes [15], la marginalisation sur les variables discrètes ne pose aucun problème.

Une autre manière de résoudre ce problème et de faire appel à la théorie des graphes afin de donner aux graphes d'autres propriétés. Ces propriétés nous permettent à la différence des méthodes approximatives d'accomplir la phase de marginalisation de manière exacte (non approximée) où ce qu'on appelle "strong" marginalisation.

2.7.4 Arbre de jonction avec une racine forte

La non fermeture de la (CG) distribution dans le cas de la marginalisation sur un ensemble de variables discrètes nous oblige à restructurer l'arbre de jonction. Cette restructuration consiste à définir un arbre de jonction avec une racine dite forte. Nous rappelons dans ce qui

[14] La marginalisation par rapport aux variables discrètes d'une (CG) distribution ne donne pas forcément une (CG) distribution, on parle de la non clôture ou la non fermeture de la (CG) distribution [Lauritzen, 1996], [Murphy, 2002]

[15] La variable discrète n'est pas un parent de la variable continue, mais elles partagent la même clique.

suit les principaux résultats, pour plus de détails sur cette théorie, nous renvoyons le lecteur aux références [Lauritzen, 1996], [Cowell *et al.*, 1999].

Définition 17 *(racine forte) : une clique R est dite racine forte, si chaque paire de clique adjacente A et B tel que A est plus proche de R par rapport à B, on a le résultat suivant :*

$$(B \setminus A) \subseteq \Gamma \vee (B \cap A) \subseteq \Delta$$

où Γ et Δ représentent respectivement les variables continues et discrètes.

En d'autres termes, quand un séparateur entre deux cliques voisines n'est pas purement discret, toutes les variables de la clique la plus éloignée de la racine qui ne sont pas dans le séparateur sont continues.

2.7.5 Condition d'existence d'une racine forte

Si un graphe est triangulé et ne contient aucun chemin de la forme $D_i, C_1, C_2, ...C_n, D_j$, (aucun lien entre deux noeuds discrets D_i, D_j passant uniquement par des noeuds continus $C_1, ..., C_n$) alors ce graphe possède au moins une racine forte. Ces graphes sont dits *graphes décomposables* [Lauritzen, 1996].
Considérons le réseau de la figure 2.15(a) (exemple tiré de [Murphy, 1999]). La phase de moralisation ajoute un lien entre S et C (figure 2.15(b)). Ce même lien rend le graphe d'ores et déjà triangulé et nous obtenons les cliques suivantes (S, C, P) et (P, B). Ainsi, l'arbre de jonction est de la forme SCP-P-PB (figure 2.15(c)). Cependant, le graphe de la figure 2.15(b) possède un chemin entre deux variables discrètes passant uniquement par des noeuds continus (S-P-B) et par conséquent l'arbre de jonction ne contient aucune racine forte. Toutefois, si on ajoute un nouveau lien entre S et B (figure 2.15(d)), l'arbre de jonction est de la forme SCP-SP-SPB (figure 2.15(e)). Dans ce cas, SPB possède une racine forte puisqu'elle vérifie la définition d'une racine forte donnée ci-dessus : $B \setminus A = \{S, C, P\} \setminus \{S, P, B\} = \{C\} \subseteq \Gamma$.

Le recours à une racine forte nous assure lors de la propagation des messages (des feuilles jusqu'à la racine) d'utiliser une marginalisation exacte par opposition à la marginalisation approximative [Murphy, 2002]. De ce fait, lorsque nous propageons les messages dans le sens racine feuilles, nous assurons que les potentiels des cliques voisines soient cohérents (les cliques voisines contiennent les mêmes informations sur les variables partagées). Plus de détails sur ce point peut être trouvé dans [Castillo *et al.*, 1997].

Certes l'utilisation d'un arbre de jonction avec une racine forte assure la cohérence des données et l'utilisation d'une marginalisation exacte. Cependant l'inconvénient majeur d'utiliser un arbre de jonction avec une racine forte exige l'ajout de liens supplémentaires afin d'éliminer les liens entre les variables discrètes passant uniquement par des noeuds continus. L'ajout de liens supplémentaires va sûrement augmenter la taille des cliques, et par conséquent la complexité des calculs.
Dans [Murphy, 1999], l'auteur donne la preuve que la taille effective d'une clique est déterminée uniquement par le nombre de noeuds cachés qu'elle contient. Donc l'ajout de liens sur les variables observées n'augmente en aucun cas ni la taille ni la complexité du calcul.

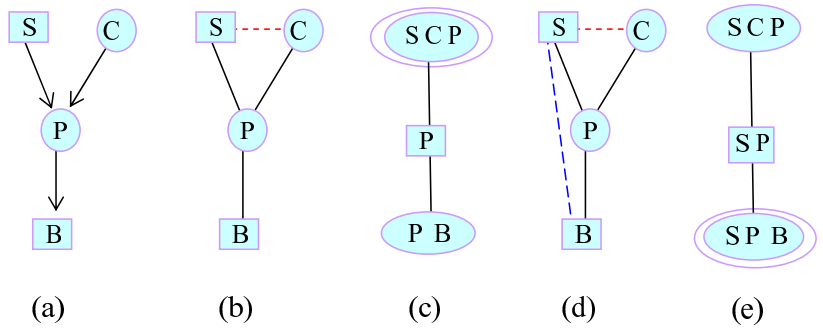


FIG. 2.15 – (a) Exemple d'un réseau bayésien hybride. (b) Étape de moralisation. (c) Arbre de jonction. (d) Ajout d'un nouveau lien pour éliminer le chemin S-P-B. (e) Arbre de jonction avec une racine forte.

2.8 Réseaux bayésiens dynamiques

2.8.1 Introduction

Il est très difficile d'interpréter ou prédire un phénomène sans tenir compte de son évolution dans le temps. Nous nous intéressons dans cette section à la modélisation stochastique de processus dynamiques [Puig, 2003], en se basant sur le formalisme des réseaux bayésiens dynamiques *(DBN)*[16]. Ces réseaux sont une extension des réseaux bayésiens statiques vus jusqu'à présent. Les filtres de Kalman, les modèles de Markov cachés *(HMM)*[17],... peuvent être représenté sous la forme de réseaux bayésiens dynamiques [Murphy, 2002].
Dans de nombreuses applications dans lesquelles on modélise des systèmes dynamiques, on organise les variables $Z_t = (U_t, X_t, Y_t)$ en trois sous-ensembles. U_t représente les variables d'entrées, X_t les variables d'états (non observables) et Y_t les variables observées. On ne s'intéressera qu'aux processus à temps discret, les processus à temps continu ne peuvent être représentés par ce modèle [Murphy, 2002].
On crée un réseau bayésien dynamique, en dupliquant à pas de temps le réseau statique et en reliant ces réseaux par des arcs indiquant les dépendances temporelles. Chaque duplication correspond à l'évolution temporelle du réseau. Dès lors, nous devons définir le modèle de transition, $P(Q_t|Q_{t-1})$, le modèle d'observation, $P(Y_t|Q_t)$, et l'état initial, $P(Q_1)$. Ces distributions peuvent être conditionnées à l'entrée de commande U_t si elle est présente. Dans ce cas, le modèle de transition s'écrit $P(Q_t|Q_{t-1}, U_{t-1})$.
Notons par $B_{\rightarrow}$ le réseau bayésien temporel à deux pas de temps (2TBN). Le modèle de transition et le modèle d'observation sont alors définis dans le (2TBN) comme le produit de la distribution de probabilités conditionnelles :

$$p(Z_t|Z_{t-1}) = \prod_{i=1}^{N} p(Z_t^i|pa(Z_t^i)) \tag{2.48}$$

où Z_t^i est le i^{me} nœud au pas de temps t qui peut être un élément de X_t, Y_t ou U_t et $pa(Z_t^i)$ sont les parents de Z_t^i appartenants ou non au même pas de temps t. Les arcs reliant les nœuds entre les différents pas de temps vont de gauche vers la droite, ce qui est raisonnable d'un point de vue de la causalité. Pour des raisons de simplicité, on suppose que le modèle est

[16] Dynamic Bayesian Networks
[17] Hidden Markov Model

Markovien d'ordre un [18]. On peut représenter l'état initial, $P(Z_1^{1:N})$, par un réseau bayésien à un pas de temps, noté par B_1. Ensemble, B_1 et $B_{\rightarrow}$ représentent le réseau bayésien dynamique. La distribution de probabilité jointe pour T pas de temps, peut être obtenue par "dérouler" le réseau T fois, puis faire le produit de toutes les distributions de probabilités conditionnelles :

$$p(Z_{1:T}^{1:N}) = \prod_{i=1}^{N} p_{B_1}(Z_1^i | pa(Z_t^i)) \times \prod_{t=2}^{T} \prod_{i=1}^{N} p_{B_{\rightarrow}}(Z_t^i | pa(Z_t^i)) \tag{2.49}$$

L'exemple donné par la figure 2.16, montre un (2TBN) et sa version déroulée pour une séquence de longueur $T = 4$. Dans ce cas, $Z_t^1 = Q_t$, $Z_t^2 = Y_t$, $pa(Q_t) = Q_{t-1}$ et $pa(Y_t) = Q_t$, et la distribution de probabilités jointe s'écrit :

$$p(Q_{1:T}, Y_{1:T}) = p(Q_1)p(Y_1|Q_1) \times \prod_{t=2}^{T} p(Q_t|Q_{t-1})p(Y_t|Q_t)$$

S'il existe un arc reliant le même nœud Z^i sur T pas de temps : $Z_1^i \rightarrow Z_2^i \rightarrow \ldots \rightarrow Z_T^i$, ce nœud est dit *persistant*. Les arcs entre deux pas de temps sont vus comme la persistance d'un phénomène au cours du temps, alors que les arcs au sein d'un même pas de temps sont vus comme un effet causal "immédiat" [Murphy, 2002]. Cependant, on peut avoir des arcs reliants des nœuds différents sur T pas de temps :$Z_1^i \rightarrow Z_2^j, Z_2^i \rightarrow Z_3^j, \ldots, Z_{T-1}^i \rightarrow Z_T^j$.

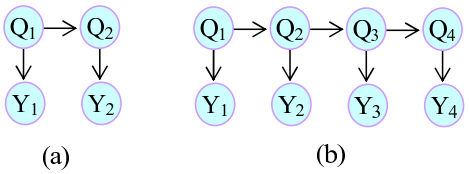


FIG. 2.16 – (a) Exemple d'un réseau (2TBN) (b) Modèle déroulé pour 4 pas de temps (T=4).

2.8.2 Inférence

L'inférence dans un réseau bayésien dynamique, consiste à calculer la distribution de probabilités des variables cachées à un instant donné, connaissant la séquence des observations passées et éventuellement futures. Tout au long de ce document, nous nous focalisons sur le cas du filtrage qui consiste à estimer l'état courant d'après les observations présentes et passées : $p(X_t|y_{1:t})$. La prédiction $p(X_{t+k}|y_{1:t}), k > 0$ et le lissage $p(X_{t-k}|y_{1:t}), k > 0$ ne seront pas utilisés dans ce document.
Si nous définissons le réseau bayésien dynamique par le couple $(B_1, B_{\rightarrow})$, il suffit de dérouler ce réseau pour T pas de temps, et appliquer n'importe quels algorithmes d'inférence statique (voir l'algorithme d'inférence JLO). Cette façon de procéder est connue par l'inférence hors ligne. Une fois l'arbre de jonction construit, il peut être réutilisé pour effectuer l'inférence avec différentes séquences d'observations. Les deux étapes utilisées pour l'initialisation de l'arbre de jonction et pour la propagation d'une évidence sont utilisées de la même façon pour le (2TBN) déroulé.

[18] L'état futur est conditionnellement indépendant de l'état passé sachant l'état courant.

Malgré que cette façon de procéder est simple à mettre en œuvre, la construction de l'arbre de jonction sera rapidement très lourde [19] pour des grandes valeurs de T. L'algorithme de construction de l'arbre de jonction ne tenant pas compte de la structure redondante du réseau, rend cette méthode non optimale en terme de temps d'exécution et d'espace mémoire. Plusieurs solutions ont été proposées pour résoudre ce problème.

L'algorithme de la frontière proposé par [Zweig, 1996], crée un ensemble de nœuds appelé nœuds de la frontière, noté par F. Les nœuds à gauche et à droite de F seront notés par L et R respectivement. A chaque pas de temps T, F doit séparer l'ensemble L de R : $R \perp L|F$. Notons par h_F et e_F les variables cachées et les évidences appartenant à F. e_L et e_R désignent les évidences appartenant à L et R respectivement.
Dans un premier temps (passage de l'information des feuilles vers la racine), nous pouvons calculer la distribution de probabilités jointe $p(F) \stackrel{def}{=} p(h_F, e_F, e_L)$ comme suit. Nous ajoutons un nœud $N \in R$ à la frontière F une fois que tous ses parents se trouvent dans F. On aura $p(e_L, e_F, h_F, N) = p(e_L, e_F, h_F)p(N|e_F, e_F)$ puisque $N \perp e_L|e_F, h_F$. Ainsi, ajouter un nœud à l'ensemble F, consiste à multiplier sa distribution de probabilités conditionnelles par la distribution jointe.
On supprime un nœud N appartenant à F (le déplacer de F vers L) lorsque tous ses enfants appartiennent à la frontière, ce qui revient à faire une marginalisation sur la variable en question (le nœud N). Dans le cas où N est caché, alors $e_{L+N} = e_L$ et $e_{F-N} = e_F$:

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = p(e_L, e_F, h_{F-N})$$

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = \sum_N p(e_L, e_F, N, h_{F-N})$$

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = \sum_N p(e_L, e_F, h_F)$$

car $h_{F-N} \cup \{N\} = h_F$.
Le cas où N est observé est similaire :

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = p(e_{L+N}, e_{F-N}, h_F)$$

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = p(e_L, e_N, e_{F-N}, h_F)$$

$$p(e_{L+N}, e_{F-N}, h_{F-N}) = p(e_L, e_F, h_F)$$

Dans le second passage (de la racine vers les feuilles), nous pouvons calculer la distribution de probabilités jointe $p(F) \stackrel{def}{=} p(e_R|h_F, e_F)$ en ajoutant et supprimant des nœuds dans l'ordre inverse que nous avons utilisé dans le premier passage.

L'algorithme de la frontière décrit ci-dessus utilise à chaque pas de temps, tous les nœuds cachés pour séparer le passé du futur. Dans [Murphy, 2002], l'auteur propose d'utiliser seulement l'ensemble des nœuds possédant une transition sortante vers la tranche de temps suivante. Cet ensemble, appelé *Interface*, d-sépare le passé du futur : $\{V_{1:t-1}, N_t\} \perp V_{t+1:T}|I_t$ où $N_t = V_t \setminus I_t$. Le problème de filtrage se réduit alors au calcul de la distribution de probabilités $p(I_t^{\rightarrow}|y_{1:t})$ à partir de $p(I_{t-1}^{\rightarrow}|y_{1:t-1})$. Cet algorithme nous permet de garder en mémoire uniquement deux pas de temps, qui est essentiel pour le filtrage en ligne.
L'arbre de jonction est construit pour chaque $J_t = I_{t-1} \cup V_t$ (V_t est l'ensemble des nœuds à

[19] Les cliques deviennent de plus en plus grandes, ce qui rend souvent l'inférence exacte impossible

l'étape t) en s'assurant qu'il existe au moins une clique D_t contenant l'interface I_{t-1} et une clique C_t contenant l'interface I_t. Pour cette raison, il suffit d'ajouter un lien entre toutes les variables de I_{t-1} et celles de I_t après l'étape de moralisation. Les arbres de jonctions J_t sont reliés au moyen de leurs interfaces I_t comme le montrent la figure 2.17. Nous pouvons effectuer l'inférence dans chaque arbre séparément, puis transmettre les messages entre eux via les nœuds d'interface.

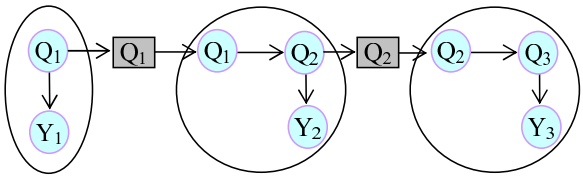


FIG. 2.17 – Application de l'algorithme d'interface sur le réseau de la figure 2.16.

2.8.3 Inférence dans le Switching Kalman Filter

Le Switching Kalman Filter est connu sous plusieurs noms : switching linear dynamical system (LDS), switching state-space model (SSM), jump-Markov model, jump-linear system, conditional dynamic linear model (DLM),... Ce type de modèle est souvent utilisé pour approximer les modèles non linéaires (en supposant que ces modèles sont linéaires par morceaux) [Murphy, 2002]. Le Switching Kalman Filter est représenté par la figure 2.18 sous forme de réseau bayésien hybride où S_t est une variable discrète et X_t et Y_t sont des variables continues représentant respectivement la variable d'état et l'observation. La distribution de probabilités conditionnelles de chaque variable est donnée par :

$$p(X_t = x_t | X_{t-1} = x_{t-1}, S_t = i) \sim N(x_t; A_i x_{t-1}, Q_i)$$

$$p(Y_t = y | X_t = x_t) \sim N(y; C x_t, R)$$

$$p(S_t = j | S_{t-1} = i) = M(i, j)$$

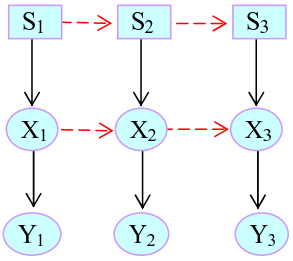


FIG. 2.18 – Modèle de réseau bayésien représentant le Switching Kalman Filter.

Le problème fondamental avec ce type de réseau (Switching Kalman Filter) est que l'inférence est pratiquement infaisable car le nombre de gaussiennes de la distribution de l'état croît de façon exponentielle à chaque pas de temps. Pour voir cela, supposons que la distribution initiale $p(X_1)$ est un mélange de K gaussiennes, une pour chaque valeur de S_1. Lors de la phase de marginalisation sur S_1, chaque gaussienne doit être propagée à travers K équations (une pour chaque valeur de S_2), de telle sorte que $p(X_2)$ est un mélange de K^2 gaussiennes. A l'instant t, la distribution de probabilités $p(X_t|y_{1:t})$ est un mélange de K^t gaussiennes, une

pour chaque combinaison possible de $S_1, ..., S_t$.

Bien que l'inférence est pratiquement infaisable dans le cas des modèles de type Switching Kalman Filter, il existe des cas particuliers où c'est possible. Le cas trivial est lorsque la variable discrète S_k est observable. Dans ce cas, la distribution de probablilités de la variable continue est unimodale (une seule gaussienne). Le second cas ou l'inférence est possible, est lorsque les variables discrètes ne sont pas reliées d'une étape à une autre. Un exemple illustratif est donné dans [Murphy, 2002].

2.8.4 Techniques de réduction du nombre de gaussiennes

Une manière d'atténuer le nombre de gaussiennes qui croît de façon exponentielle dans le temps est de faire appel aux techniques de selections. Les algorithmes dits "pruning algorithms" par exemple réduisent le nombre de gaussiennes par élimination de certaines gaussiennes. Ces algorithmes gardent les N gaussiennes de fortes probabbilités, éliminent les autres et normalisent les probabilités de tels sorte que la somme est égale à 1. Le lecteur peut trouver plus de détails sur ces algorithmes dans [Uri, 2002] et [Bar-Shalom et Fortmann, 1988].
Une autre approche dite "Collapsing" ou General Pseudo-Bayesian algorithms (GPB) consiste à partitionner les gaussiennes en N sous-ensembles. L'algorithme (GPB) limite le nombre de gaussiennes de l'état à N correspondent aux nombres de sous-ensembles. Pour mieux comprendre le fonctionnement de cette approche, donnons un exemple tiré de la thèse de [Uri, 2002]. Supposons que nous avons M gaussiennes à l'étape $t = 1$. M est le nombre de combinaisons possibles de la variable discrète. Après la phase de propagation, nous obtenons M^2 gaussiennes. Ces M^2 gaussiennes sont "collapsées" en M gaussiennes. La figure 2.19 illustre cette approche pour un nombre de gaussiennes égale à 3 ($M = 3$). Pour plus de details sur ces approches, nous renvoyons le lecteur au fameux livre de [Lauritzen, 1996] et à la thèse de [Uri, 2002].
Une autre manière pour faire face au nombre de gaussiennes qui croît de façon exponentielle, est l'utilisation des algorithmes d'inférence approximatifs (exemple les filtres particulaires) qui ont été utilisés avec succès sur ce type de modèle. Ces types d'algorithmes sont traités plus en détails dans [Uri, 2002] et [Murphy, 2002].

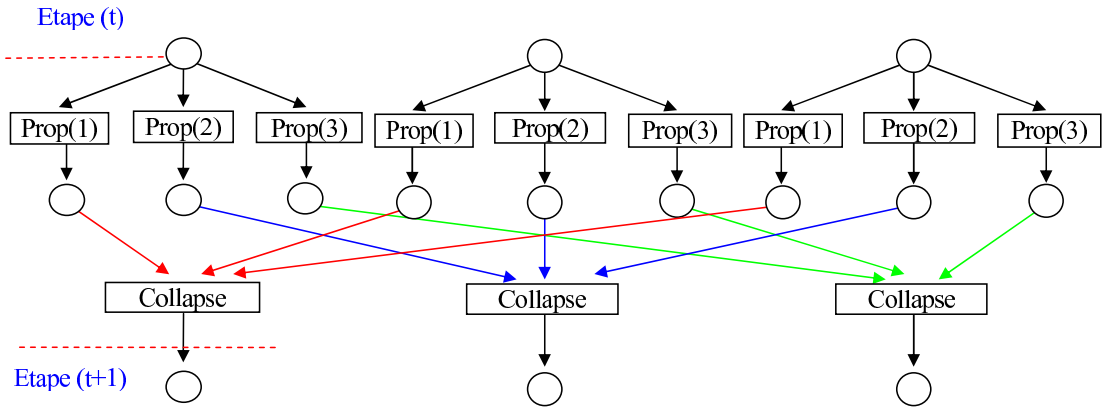


FIG. 2.19 – Exemple de fonctionnement de l'algorithme (GPB) : General Pseudo-Bayesian algorithms pour $M = 3$. Chaque cercle représente une gaussienne (les rectangles avec la notice prop(), représentent la propagation de chaque gaussienne).

2.9 Conclusion

Ce chapitre a donné les principaux aspects concernant les réseaux bayésiens. La notion de factorisation de la distribution de probabilités jointes d'un graphe triangulé est à la base des modèles décomposables. Par la suite, nous avons introduit le théorème de *Jensen* qui fait le lien entre les graphes triangulés et les arbres de jonction. Le graphe initial est transformé en un arbre de jonction où chaque famille de nœuds est regroupée dans une clique, l'arbre ainsi trouvé satisfait la règle de chaînage ("chain rule").
L'arbre de jonction est initialisé pour nous permettre de calculer les distributions des variables cachées. Cette initialisation se fait en deux étapes. On commence par remonter les flux des données à partir des feuilles jusqu'à la racine. La deuxième étape consiste à faire passer les flux de la racine jusqu'aux feuilles. L'arbre atteint ainsi un état d'équilibre.
La propagation d'une nouvelle observation dans l'arbre de jonction utilise le même principe que l'algorithme d'initialisation de cet arbre. La nouvelle observation est passée entre les cliques voisines (de proche en proche) et par conséquent, les potentiels des cliques et des séparateurs sont modifiés.

Les réseaux bayésiens permettent de manipuler non seulement les variables discrètes mais également les variables continues. Dans ce contexte, on a présenté les réseaux bayésiens à variables continues où la distribution de probabilités de chaque variable est supposée linéaire Gaussienne. L'utilisation des variables discrètes et continues dans le même réseau est décrite à la fin de ce chapitre. La seule particularité de ce type de réseau est que la distribution de probabilités de chaque variable est supposée conditionnelle Gaussienne (CG).
La non fermeture de la (CG) distribution (la somme de deux Gaussienne n'est pas une Gaussienne) nous oblige à utiliser des approximations. Une autre manière de contourner ce problème est d'avoir recours une autre fois à la théorie des graphes. L'élimination de certains chemins dans le réseau initial, donne à l'arbre de jonction une nouvelle propriété. On parle alors d'un arbre de jonction fortement décomposable (présence d'une racine forte). Cette propriété donne aux cliques une cohérence de telle sorte qu'à la fin de la phase de propagation des messages, les cliques contiennent les mêmes informations sur les variables partagées.

La modélisation de processus dynamiques par le formalisme des réseaux bayésiens dynamiques est abordée à la fin de ce chapitre. Ces réseaux sont construits en dupliquant à pas de temps le réseau statique et en reliant ces réseaux par des arcs indiquant les dépendances temporelles. Dès lors, nous devons définir le modèle de transition, le modèle d'observation et l'état initial.
L'inférence dans le cas d'un réseau bayésien dynamique suit le même schéma d'inférence pour le réseau bayésien statique. Il suffit de dérouler le réseau pour T pas de temps, et appliquer l'algorithmes d'inférence utiliser sur n'importe quel réseau statique. Cependant, la construction de l'arbre de jonction sera rapidement très lourde pour des grandes valeurs de T, ce qui rend l'inférence exacte impossible dans la plupart des cas. Pour résoudre ce problème, plusieurs approches essayent d'exploiter la redondance du réseau de telle sorte de trouver un ensemble de nœuds qui permet de séparer le future du passé.

Le chapitre suivant présente la méthode développée pour la localisation d'un véhicule ou un train de véhicules sur une carte. Nous avons choisi le cadre des réseaux bayésiens dynamiques, car cette approche probabiliste permet de prendre en compte l'aspect incertain présent dans

les modèles utilisés dans les applications réelles, ainsi que l'incertitude donnée par les capteurs. La fusion multi-capteurs par un réseau bayésien se fait de la même manière qu'un filtre de Kalman. De plus, ce type de réseau nous permet d'utiliser les variables discrètes et continues dans le même réseau. Cette dernière caractéristique est bien utile dans le chapitre suivant car elle nous permet de représenter le problème de "map-matching" par un réseau bayésien dont les segments de routes sont représentés par une variable discrète et la position du véhicule sur le(s) segment(s) par une variable continue.
La partie graphique des réseaux bayésiens offre un outil intuitif et attractif dont la modélisation d'un train de véhicule est réalisée par simple duplication du réseau (servant à la localisation d'un véhicule sur une carte) pour chaque élément du convoi.

Chapitre 3

Approche développée

3.1 Introduction

Le problème de la localisation peut être vu comme l'estimation de la position d'un robot étant donné les mesures bruitées données par un ensemble de capteurs proprioceptifs ou extéroceptifs. La tâche de localisation peut se réaliser par un GPS. Or, comme mentionné auparavant, en milieu urbain, le masquage des signaux des satellites peut parfois être long, en raison de la non visibilité satellitaire et des multitrajets des ondes GNSS [20]. Ainsi, l'utilisation d'autres capteurs est nécessaire si on cherche un positionnement précis, sûr, intègre et sans interruption de service. Cependant, aucun des capteurs utilisés dans le domaine de la localisation qu'il soit extéroceptif ou proprioceptif n'est parfait à lui seul pour la tâche de localisation.

Le problème de la localisation est particulièrement critique dans le cadre de l'accrochage immatériel auquel nous nous sommes intéressé dans cette thèse. Rappelons qu'il s'agit de constituer un train de véhicules dont seul le premier est piloté par un opérateur humain, les véhicules suiveurs étant en mode autopilotage. En effet, une géo-localisation précise d'ordre centimétrique de chacun des véhicules est nécessaire pour les modules de contrôle des véhicules suiveurs (suivi de trajectoire du véhicule de tête et respect d'un écart prédéfini entre les véhicules).
La fusion multicapteurs devient de ce fait, nécessaire et particulièrement importante autant d'un point de vue fondamental que pour les applications pratiques. Les données fusionnées reflètent non seulement l'information générée par chaque capteur, mais encore l'information qui n'aurait pu être inférée par aucun capteur pris séparément. Les modèles graphiques probabilistes et particulièrement les réseaux bayésiens sont parfaitement propices dans ce cadre pour leur robustesse vis-à-vis de l'incertitude tant des modèles que des données mesurées.

Ce chapitre est divisé en deux parties. Dans un premier temps, nous nous intéressons au domaine de la fusion de données par un réseau bayésien pour la localisation d'un robot sur une carte. A la fin de cette partie nous proposons une méthode basée sur les modèles chaînés dans le but de contourner le problème de la non linéarité. Cette approche consiste a rechercher une transformation exacte d'un système non linéaire afin de réécrire ce système comme un système linéaire.

[20] Global Navigation Satellite System.

La seconde partie de ce chapitre concerne la modélisation et la localisation par un réseau bayésien d'un train de véhicules dans le cas où les véhicules suiveurs connaissent le chemin de référence (celui emprunté par le véhicule de tête).

3.2 Position du problème du map-matching

La solution adoptée dans le cadre de cette thèse pour la localisation en monde extérieur est la fusion de trois types d'information : le GPS (le choix du GPS est évidemment justifié par sa large utilisation pour la localisation de véhicules), les codeurs incrémentaux et la base de données cartographique. Naturellement, il nous faut des méthodes qui permettent de manipuler et fusionner toutes ces sources d'informations afin de quantifier la confiance accordée à la précision de l'estimation. L'outil que nous proposons d'utiliser pour la fusion de données et pour l'estimation de la position d'un robot est le réseau bayésien.

3.2.1 Sources d'information utilisées pour la localisation d'un véhicule

3.2.1.1 Estimation donnée par l'odométrie

L'odométrie est une technique permettant d'estimer la pose d'un véhicule en mouvement. En partant d'une position initiale connue et en intégrant les déplacements mesurés par les codeurs incrémentaux, on peut ainsi calculer à chaque instant la position courante du véhicule. La précision donnée par le système odométrique (3.1) est à court terme car ce système dérive au cours du temps. Cette dérive est due à plusieurs facteurs dont le système odométrique ne tient pas compte. Citons par exemple le glissement des roues, l'imperfection des routes (trous, bosses),...Pour cette raison, on a toujours recours à un autre capteur pour corriger l'estimation donnée par ce modèle.

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + \Delta_k \cos(\theta_k + \omega_k/2) \\ y_{k+1} = y_k + \Delta_k \sin(\theta_k + \omega_k/2) \\ \theta_{k+1} = \theta_k + \omega_k \end{cases} \tag{3.1}$$

3.2.1.2 Correction de l'estimation donnée par l'odométrie par un GPS

Les mesures données par le GPS sont relativement simples à manipuler parce que ce sont des points dans un système de coordonnées global connu. Le GPS fournit la position du véhicule sous forme de latitude et longitude ainsi que l'incertitude autour de cette estimée. Cette position est convertie en coordonnées cartésiennes Y_k^{gps}, par une projection dans le même repère du véhicule. La correction de l'estimation donnée par le modèle odométrique est réalisée par le réseau bayésien en utilisant l'équation d'observation suivante :

$$Y_k^{gps} = \begin{bmatrix} x_k^{gps} \\ y_k^{gps} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \theta_k \end{pmatrix} + \beta_k \tag{3.2}$$

où β_k représente le bruit de mesure. A la réception des données GPS, on reçoit également par le biais de la trame *NMEA GST* l'ellipse d'incertitude autour de cette estimée. Cette ellipse peut être représentée par une distribution gaussienne $\beta_k \sim N(\mu, Q)$. La matrice de covariance Q est donnée par :

$$Q = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \tag{3.3}$$

où σ_x^2 et σ_y^2 représentent respectivement les déviations des mesures en longitude et latitude et σ_{xy} est l'autocorrélation.

Le problème de la dérive de l'odométrie est ainsi réglé par l'utilisation du GPS. Cependant, un GPS ne fonctionne en milieu urbain que par intermittence en raison du masquage des satellites. C'est pourquoi le recourt à une troisième source d'information était nécessaire. Depuis que la cartographie est devenu une source d'information assez fiable et manipulable en temps réel, plusieurs personnes pensent qu'il est pertinent de l'utiliser comme une source d'information pour une localisation absolue [El Najjar, 2003].

3.2.1.3 La cartographie

La carte numérique d'un réseau routier est une représentation graphique de l'information spatiale. Cette carte sert d'interface entre le conducteur et la navigation. La précision d'une estimation donnée par le GPS peut être améliorée si on utilise des informations cartographiques qui permettent en particulier de contraindre les positions possibles aux seuls segments correspondants à des voies de circulation autorisées. Cependant, les coordonnées des segments de route sur une carte numérique sont également entachées d'erreurs. La base de données cartographique n'est pas toujours en accord avec la réalité. Elle peut contenir des linéaires qui n'existent plus ou même omettre certains tronçons. Enfin certains détails peuvent être approximés très grossièrement : un rond-point peut être représenté par un point sur la carte.

3.2.1.3.1 Erreurs et incertitude des sources d'informations cartographiques

La première question qui se pose à la création d'une carte est comment représenter les traits ou les caractéristiques du monde réel sur cette carte ? Selon le *National Research Council* [Quddus, 2006], ce processus passe par plusieurs étapes dont le choix de l'échelle de la carte. Une autre étape de construction d'une carte est la représentation des routes comme un ou plusieurs segments. Ainsi, un carrefour peut être représenté comme un ensemble de petit segments ou même parfois par un point si ce dernier est petit (voir la figure 3.1).
L'autre source d'imprécision est liée au système de coordonnées utilisé. La plupart des récepteurs GPS fournissent des données sous forme de latitude et longitude. Cependant, en utilisant la carte on utilise souvent notre propre système de coordonnés. Par conséquent, la conversion est essentielle pour la mise en correspondance entre les deux systèmes. Ce processus réduit également la précision de la carte.

3.2.1.3.2 Modélisation de la zone d'incertitude autour d'un segment

Au paragraphe précédent, on a vu que la carte ne reflète pas vraiment la réalité du terrain. Le véhicule ne roule pas exactement sur le segment représentant la route. Le véhicule se déplace sur une surface 3D alors que la carte représente une vue plane,...Pour toutes ces raisons, nous avons opté pour la construction d'une zone d'incertitude autour de chaque segment [El Najjar, 2003].

Le GPS fournit la position du véhicule sous forme de latitude et longitude ainsi que l'incertitude autour de cette estimée. Par simple souci d'homogénéité avec le GPS, nous représentons par la suite la zone d'incertitude de chaque segment de route par une ellipse. On prend en compte les deux extrémités de ce segment et la largeur de la route. La figure 3.2 montre comment approximer la zone d'incertitude autour du segment par une ellipse. Le repère propre de l'ellipse gaussienne E est attaché au segment et son axe des abscisses est colinéaire avec le segment.

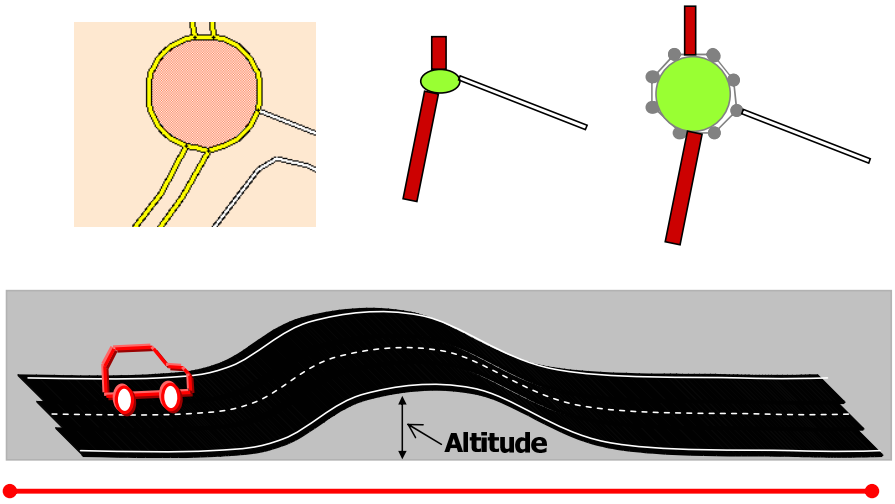


FIG. 3.1 – Un carrefour est représenté soit par un point si son rayon est petit soit par un ensemble de segments si son rayon est grand. En bas de cette figure le déplacement du véhicule se fait sur une surface 3D alors que ce segment de route est représenté par une vue plane

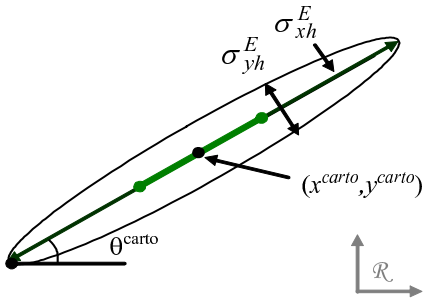


FIG. 3.2 – Approximation de la zone d'incertitude d'un segment de route à l'aide d'une ellipse. Le segment est représenté par son milieu : (x^{carto}, y^{carto}) et son cap θ^{carto}. Les attributs associés à cette approximation sont la longueur et la largeur de route

La construction d'une ou plusieurs observations cartographiques suit le schéma suivant. On projette la position estimée par l'odométrie ou par le GPS sur tous les segments dans un rayon représentant l'ellipse d'incertitude de cette estimation. Chaque projection nous donne une observation ponctuelle $(x^{carto}, y^{carto}, \theta^{carto})$ où $1 \leq i \leq N$ et N représente le nombre de

segments autour de l'estimée [21]. L'équation d'observation cartographique par rapport à l'état du véhicule $X = (x_k, y_k, \theta_k)$ s'écrit :

$$Y_k^{carto} = \begin{bmatrix} x_k^{carto} \\ y_k^{carto} \\ \theta_k^{carto} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \theta_k \end{pmatrix} + \gamma_k \tag{3.4}$$

L'incertitude autour de chaque segment est définie dans l'équation (3.4) par la covariance (γ_k). Cette incertitude peut être représentée par une distribution gaussienne $\gamma_k \sim N(\mu, R_h)$. Dans le repère attaché au segment et dont le centre est (x^{carto}, y^{carto}), nous pouvons quantifier la matrice de covariance R_h de l'erreur cartographique par :

$$R_h = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & 0 \\ \sigma_{xy}^2 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{3.5}$$

où σ_x^2, σ_y^2 et σ_θ^2 représentent respectivement les déviations en longitude, transversale et cap et σ_{xy} l'autocorrélation. Pour plus de détails sur la construction de l'observation cartographique et le calcul de chaque élément de la matrice R_h voir la thèse [El Najjar, 2003].

3.2.2 Modèle d'évolution d'un robot

Considérons un véhicule représenté par la cinématique d'un robot à deux roues donné figure 3.3. Le véhicule est représenté par son origine $M = (x, y)$ du centre de l'essieu arrière. L'angle formé entre la direction du véhicule et l'axe des abscisses représente le sens de mouvement du véhicule noté θ. Dans de parfaites conditions de roulement sans glissement, le modèle cinématique du véhicule donné figure 3.3 est défini par :

$$\dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} v_2 \tag{3.6}$$

où v_1 et v_2 représentent respectivement la vitesse linéaire et angulaire du point M.

Cependant dans la pratique, les vitesses linéaire et angulaire du point M ne sont pas directement calculables à partir des données capteurs, mais déduites des vitesses de rotation des roues droite et gauche ω_L et ω_R. La relation entre les vitesses linéaire et angulaire (v_1, v_2) d'une part et les vitesses de rotation des roues (ω_L, ω_R) d'autre part est défini par :

$$v_1 = \frac{\omega_L + \omega_R}{2} r \tag{3.7}$$

$$v_2 = \frac{\omega_R - \omega_L}{2e} r \tag{3.8}$$

où r est le rayon de la roue et e la demi distance reliant les deux roues .

A partir des équations (3.6), (3.7) et (3.8), l'équation cinématique du système se réécrit comme :

[21] On suppose que toutes les routes sont représentées par au moins un segment dans notre base de données cartographiques c'est-à-dire $N \neq 0$.

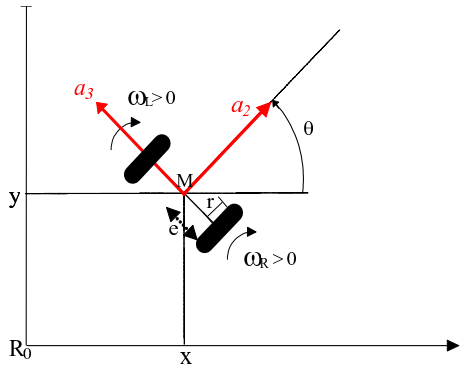


FIG. 3.3 – Modélisation en 2D d'un robot à deux roues.

$$\dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ -\frac{r}{2e} & \frac{r}{2e} \end{pmatrix} \begin{pmatrix} \omega_L \\ \omega_R \end{pmatrix} \tag{3.9}$$

ou plus simplement :

$$\dot{q} = \begin{cases} \dot{x} = \frac{r}{2}(\omega_R + \omega_L)\cos\theta \\ \dot{y} = \frac{r}{2}(\omega_R + \omega_L)\sin\theta \\ \dot{\theta} = \frac{r}{2e}(\omega_R - \omega_L) \end{cases} \tag{3.10}$$

Notons que l'équation (3.10) peut être représentée par le modèle unicycle donné figure 3.4. Le modèle unicycle est le plus simple modèle non holonome approximant un véhicule à une roue unique [LaValle, 1999].

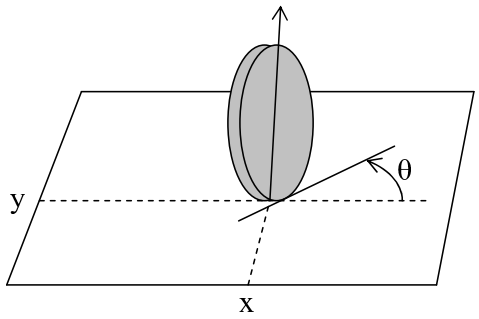


FIG. 3.4 – Représentation graphique des coordonnées d'un modèle unicycle.

3.2.3 Difficulté à manipuler les systèmes non linéaires

L'utilisation de l'inférence exacte dans les filtres de Kalman ou les réseaux bayésiens est principalement possible à cause de la supposition du bruit gaussien additif et de la linéarité du modèle. De ce fait, la mise à jour d'une distribution gaussienne par la règle de Bayes, donne une distribution a posteriori qui est toujours gaussienne. Cependant, l'inférence exacte dans les réseaux bayésiens ou les filtres de Kalman dont la distribution de probabilité conditionnelle des variables cachées n'est pas linéaire gaussienne, n'est pas toujours possible. En particulier,

l'inférence exacte dans les systèmes qui sont non linéaires et/ou non gaussien n'est généralement pas possible [Murphy, 2002].
Pour traiter le problème de la non linéarité d'un modèle, plusieurs approches existent : filtre de Kalman étendu, filtrage particulaire,... Nous proposons dans le paragraphe suivant d'étudier plus en détail le filtre de Kalman pour deux raisons. La première est qu'on veut utiliser la même approche du filtre de Kalman étendu pour rendre le système odomètrique linéaire. La seconde, nous voulons comparer notre approche basée sur le modèle chaîné par rapport au filtre de Kalman étendu. Pour plus de précision sur les autres approches, nous renvoyons le lecteur aux références suivantes [Compillo, 2004], [Doucet, 1998] et [Murphy, 2002].

3.2.3.1 Filtre de Kalman

Un filtre de Kalman est utilisé pour estimer l'état d'un système à partir d'un ensemble d'observations. Plus formellement, pour estimer l'état (X_k) à l'instant k, on dispose de l'information $Y_{0:k} = (Y_0, ..., Y_k)$ et de l'état précédent (X_{k-1}), et le but est d'obtenir le plus d'informations possibles sur (X_k). En terme de probabilité, il s'agit de calculer la loi conditionnelle du vecteur aléatoire (X_k) sachant $Y_{0:k}$: $P(X_k|Y_0, ..., Y_k)$.
Supposons connue la loi conditionnelle du vecteur aléatoire (X_{k-1}) sachant $Y_{0:k-1}$. Le filtre de Kalman procède en deux phases :

- Phase de prédiction : on calcule la loi conditionnelle : $P(X_k|Y_0, ..., Y_{k-1})$, à l'aide de l'équation d'état.
- Phase de correction : on utilise la nouvelle observation Y_k pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

Un filtre de Kalman classique est représenté dans [Murphy, 2002] par un réseau bayésien à variables continues (voir la figure 3.5). L'auteur montre que ce filtre n'est autre qu'un réseau bayésien dynamique. Ce filtre de Kalman suppose que l'état $X_t \in \Re^{N_x}$, la fonction d'observation $Y_t \in \Re^{N_y}$, l'entrée du filtre $U_t \in \Re^{N_u}$ et que les fonctions de transition et d'observation sont linéairement gaussiennes, c'est-à-dire :

$$P(X_t = x_t|X_{t-1} = x_{t-1}, U_t = u) = N(x_t; Ax_{t-1} + Bu + \mu_X, Q)$$

et

$$P(Y_t = y|X_t = x, U_t = u) = N(y; Cx + \mu_Y, R)$$

En d'autre terme, $X_t = AX_{t-1} + BU_t + V_t$ où $V_t \sim N(\mu_X, Q)$ est un bruit gaussien. De même, $Y_t = CX_t + W_t$ où $W_t \sim N(\mu_Y, R)$ est un autre bruit gaussien indépendant de V_t.

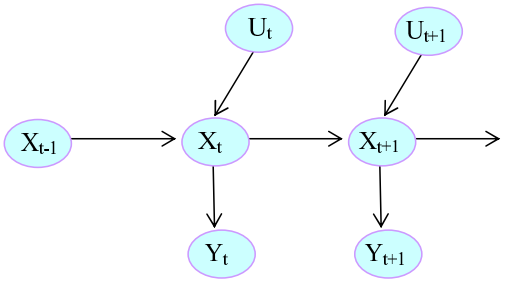


FIG. 3.5 – Représentation d'un filtre de Kalman par un réseau bayésien.

Pour appliquer les outils du filtrage bayésien récursive et typiquement le filtrage de Kalman sur les systèmes non-linéaires, la méthode classique est de linéariser le système autour de l'estimée courante. On parle alors de filtre de Kalman étendu (voir chapitre 1).
Le filtre de Kalman non parfumé ou Unscented Kalman Filter constitue une autre alternative au filtre de Kalman étendu. Ce filtre procède à une approximation de la densité a posteriori par une gaussienne comme dans le cas du filltre de Kalman étendu. Mais plutôt que de faire une approximation des fonctions non linéaires du modèle d'évolution et du modèle de mesure, il réalise une approximation de la densité de probabilité par un ensemble de points pondérés convenablement choisis de façon déterministe. Ces points sont transformés par les fonctions non linéaires d'évolution et de mesure afin d'obtenir une nouvelle densité de probabilité. Cette approximation est appelée la transformation sans parfum (Unscented Transform) [Cindy, 2008], [Dahia, 2005].

Le filtrage de Kalman est couramment utilisé pour la mise en correspondance d'une estimée sur un segment de route ou map-matching. Citons les travaux de [El Najjar, 2003], [Bonnifait, 2005] et [Quddus *et al.*, 2003]. Cependant, si on veut gérer ou manipuler plusieurs segments à la fois, le filtre de Kalman tel qu'on l'a défini précédemment n'est pas adapté à ce type de problème. Bien qu'on puisse utiliser N filtres de Kalman correspondant au nombre de segments candidats. Cette approche a été utilisée dans les travaux de [Jabbour, 2007].
Une autre limitation des filtres de Kalman est liée à leur domaine d'application aux seuls systèmes dynamiques linéaires ou non-linéaires avec un modèle d'erreur gaussien. Ce qui nous mène donc aux méthodes approximatives. Le filtre particulaire décrit dans le chapitre 1 est l'une des méthodes les plus utilisées lorsque le bruit n'est pas gaussien et que le modèle n'est pas linéaire [Compillo, 2004].

Tout au long de cette thèse, nous utilisons le même principe du filtre de Kalman étendu pour représenter les équations non linéaires par un réseau bayésien. La linéarisation du système (voir l'équation 3.1) autour de l'estimée courante et la phase de prédiction sont faites en dehors du réseau bayésien. On initialise le réseau bayésien par le résultat de la phase de prédiction. La phase de correction est faite par le réseau bayésien car les équations d'observations sont linéaires (voir les équations 3.2 et 3.4). Cette dissociation n'apparaît pas si les équations cinématiques sont linéaires, comme c'est le cas des équations d'observations manipulées dans ce chapitre.
Nous proposons dans la suite de ce chapitre, une méthode basée sur les modèles chaînés. C'est ce qui sera appelé par la suite la linéarisation exacte. Cette approche nous permet de contourner le problème de la non linéarité.

3.2.3.2 Introduction aux modèles chaînés

Les modèles chaînés sont souvent utilisés en automatique (non-linéaire) dans le but d'exploiter l'ensemble des outils de l'automatique linéaire pour construire et régler les lois de commandes. Cette méthode n'est pas connue dans le domaine de l'intelligence artificielle, cependant elle est populaire dans le domaine de l'automatique. L'expression générale d'un système non linéaire est donnée par :

$$\dot{X} = f(X) + g(X, U) \tag{3.11}$$

avec X et U les vecteurs d'état et de commande, et f et g deux fonctions non linéaires.
En Automatique non linéaire, deux approches sont principalement utilisées pour concevoir les lois de commandes d'un système [Bom, 2006]. Soit une :

1. fonction de Lyapunov est utilisée,
2. linéarisation exacte du système (3.11) est recherchée.

L'utilisation de la fonction de Lyapunov ne demande aucune transformation du système non linéaire (3.11). La commande de ce système est adressée directement, en s'appuyant sur la théorie de la stabilité de Lyapunov. Pour une bref description de ce théorème voir la thèse de [Bom, 2006], sinon pour plus de détail, nous renvoyons le lecteur à [Vidyasagar, 1993].

3.2.3.3 Linéarisation exacte

La linéarisation exacte consiste à rechercher une transformation exacte (c'est-à-dire une transformation inversible dans un autre espace d'état du vecteur d'état et de commande) permettant de réécrire le système non linéaire (3.11) comme un système linéaire dans un autre espace d'état, de façon a pouvoir exploiter l'ensemble des outils de l'automatique linéaire pour construire et régler les lois de commandes [Bom, 2006]. Une présentation générale des techniques de linéarisation exacte peut être trouvée dans [Isidori, 1995].

L'existence de formes canoniques pour les modèles cinématiques des robots non holonomes est essentielle pour le développement des stratégies de commandes en boucle ouverte et en boucle fermée. La structure canonique la plus utile est connue sous le nom de *modèle chaîné*. L'expression générale d'un système chaîné de dimension n avec 2 entrées de commandes u_1 et u_2 est donnée comme suit [Murray et Sastry, 1993][22] :

$$\begin{cases} \dot{a}_1 = u_1 \\ \dot{a}_2 = u_2 \\ \dot{a}_3 = a_2 u_1 \\ \vdots \\ \dot{a}_n = a_{n-1} u_1 \end{cases} \tag{3.12}$$

ou sous cette forme [Samson, 1995] :

$$\begin{cases} \dot{a}_1 = u_1 \\ \dot{a}_2 = a_3 u_1 \\ \dot{a}_3 = a_4 u_1 \\ \vdots \\ \dot{a}_{n-1} = a_n u_1 \\ \dot{a}_n = u_2 \end{cases} \tag{3.13}$$

Pour révéler la sous-structure linéaire du systeme (3.13), il suffit par exemple de considérer un changement d'échelle des temps. Pour cela, la notation suivante est introduite :

$$a'_i = \frac{d}{da_1} a_i, 1 \leq i \leq n \tag{3.14}$$

[22] Les points au dessus des variables représentent la dérivée

a'_i est la derivée de a_i par rapport à la variable a_1. Si, dans l'expression (3.13) les derivées temporelles sont remplacées par des derivées par rapport à a_1, les $n-1$ dernières lignes du système chaîné se ré-expriment bien comme un système linéaire [Bom, 2006] :

$$\begin{pmatrix} a'_2 \\ a'_3 \\ \vdots \\ a'_{n-1} \\ a'_n \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} a_2 \\ a_3 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u_3 \tag{3.15}$$

où $u_3 = \frac{u_2}{u_1}$ représente une nouvelle commande virtuelle, issue des commandes m_1 et m_2.

Cette nouvelle représentation du système non-linéaire (3.11), permet d'exploiter l'ensemble des outils de l'automatique linéaire pour construire et régler les lois de commandes. Dans notre cas, nous allons utiliser cette approche pour la fusion multicapteurs appliquée à la localisation d'un véhicule. Cette manière de convertir le système non-linéaire sous forme chaînée nous permet d'une part de ne pas avoir recours aux méthodes approximatives telles que les filtres particulaires et d'autre part d'utiliser le filtre de Kalman classique avec toutes ces hypothèses d'utilisation (bruit blanc additif,...).
Les conditions nécessaires et suffisantes pour la conversion d'un système non linéaire avec deux entrées $m = 2$ et n états : (u_1, u_2) et $q = (a_1, a_2, ..., a_n)$ sous forme chaîné sont données dans [Murray, 1993]. En résumé, il faut trouver :

1. Un changement de variable : $a = \Theta(q)$
2. Une transformation d'entrées inversible : $v = \beta(q)u$

Dans [Sordalen, 1993], [Samson, 1995], les auteurs utilisent ces deux conditions pour trouver une représentation sous forme chaînée représentant non seulement un véhicule, mais également la généralisation des systèmes non-holonomes pour un véhicule avec N remorques. Ce même changement de variable est bien adapté à l'étude des commandes des robots mobiles, car il est très souvent possible de choisir la variable u_1, comme la distance parcourue par le robot le long d'un chemin de référence (s) [Bom *et al.*, 2005]. En effet, dans ce cadre, les lois de commandes latérale et longitudinale peuvent être supposées comme étant indépendantes (vrai pour de petites vitesses).

Les preuves et les théorèmes utilisés pour la transformation des modèles non linéaires en modèles chaînés sortent du contexte de cette thèse. Pour ces raisons, nous renvoyons le lecteur au livre de référence sur ce sujet [Isidori, 1995]. Concernant les modèles cinématiques décrivant l'évolution des robots et leurs modèles chaînés correspondants, le lecteur peut se référer aux articles [Murray, 1993] et [Murray et Sastry, 1993].

3.2.3.4 Modèle unicycle et linéarisation exacte

L'expression générale d'un système chaîné de dimension $n = 3$ avec $m = 2$ entrées de commande u_1 et u_2 s'écrit à partir de l'équation 3.12 comme suit :

$$\begin{cases} \dot{a}_1 = u_1 \\ \dot{a}_2 = u_2 \\ \dot{a}_3 = a_2 u_1 \end{cases} \tag{3.16}$$

La non linéarité du modèle unicycle (3.10) est introduite par les fonctions sin() et cos() de l'angle θ qui est l'angle du lacet du véhicule. Le fait de choisir un changement de variable qui nous permet de passer d'un système de coordonnées cartésien (x, y, z) à un système de coordonnées mouvant (a_1, a_2, a_3) ayant comme axe a_2 toujours colinéaire avec la direction du mouvement du véhicule (i.e. l'axe qui tourne avec l'angle du lacet du véhicule) permet d'atténuer sensiblement la non-linéarité du modèle dans le nouveau système de coordonnées. Ainsi, nous effectuons le changement de variable suivant $a = \Theta(q)$ [Fabrizi *et al.*, 1998] :

$$\begin{cases} a_1 = -\theta \\ a_2 = x\cos(\theta) + y\sin(\theta) \\ a_3 = -x\sin(\theta) + y\cos(\theta) \end{cases} \tag{3.17}$$

Les nouvelles variables, a_2 et a_3 sont tout simplement les coordonnées cartésiennes du véhicule (unicycle) évalués dans le nouveau repère attaché au véhicule, de telle sorte que la composante a_2 est alignée avec le cap du véhicule (voir la figure 3.6). Les coordonnées a_2 et a_3 peuvent êtres écrites en fonction de la matrice de rotation donnée par l'équation 3.18. Notons que le changement de variable donné par l'équation 3.17 n'est pas unique dans le sens où la matrice de rotation suit ou non le sens des aiguilles d'une montre. Dans ce cas, on aurait pu prendre le sens contraire aux aiguilles d'une montre, c'est à dire : $a_1 = \theta, a_2 = x\cos(\theta) - y\sin(\theta)$ et $a_3 = x\sin(\theta) + y\cos(\theta)$.

$$\begin{pmatrix} a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.18}$$

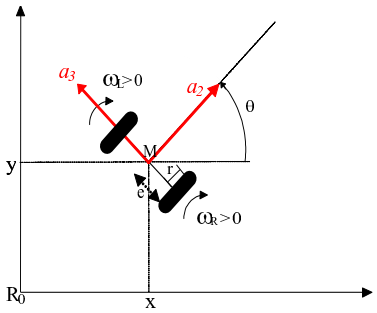


FIG. 3.6 – Les variables a_2 et a_3 représentent le nouveau repère attaché au véhicule.

Le second changement de variable à trouver pour satisfaire les conditions d'existence d'une forme chaînée donné par [Murray, 1993] est la transformation d'entrées inversible : $v = \beta(q)u$. D'après la première équation de (3.17), on a :

$$a_1 = -\theta \Rightarrow \dot{a}_1 = -\dot{\theta} = u_1$$

D'après l'équation (3.6) (troisième équation), nous avons : $\dot{\theta} = v_2$, ce qui nous donne :

$$u_1 = -v_2 \tag{3.19}$$

D'après la seconde équation de (3.17), on obtient :

$$a_2 = x\cos\theta + y\sin\theta$$

$$\Rightarrow \dot{a}_2 = \dot{x}\cos\theta - x\dot{\theta}\sin\theta + \dot{y}\sin\theta + \dot{\theta}y\cos\theta = u_2$$

Remplaçons $\dot{x}$, $\dot{y}$ et $\dot{\theta}$ par l'expression (3.10), nous obtenons :

$$\dot{a}_2 = \frac{r}{2}(\omega_R+\omega_L)\overset{2}{\cos}\theta - x\frac{r}{2e}(\omega_R-\omega_L)\sin\theta + \frac{r}{2}(\omega_R+\omega_L)\overset{2}{\sin}\theta + \frac{r}{2e}(\omega_R-\omega_L)y\cos\theta = u_2$$

$$\Rightarrow u_2 = \underbrace{\frac{r}{2}(\omega_R+\omega_L)}_{v_1}\underbrace{[\overset{2}{\cos}\theta + \overset{2}{\sin}\theta]}_{1} + \underbrace{\frac{r}{2e}(\omega_R-\omega_L)}_{v_2}\underbrace{[-x\sin\theta + y\cos\theta]}_{a_3}$$

$$u_2 = v_1 + a_3 v_2 \tag{3.20}$$

Finalement, le système chaîné se réécrit :

$$\begin{cases} \dot{a}_1 = u_1 \\ \dot{a}_2 = u_2 \\ \dot{a}_3 = a_2 u_1 \end{cases} \tag{3.21}$$

avec :

$$\begin{cases} u_1 = -v_2 \\ u_2 = v_1 + a_3 v_2 \end{cases} \tag{3.22}$$

Supposons maintenant que le véhicule se déplace le long d'une trajectoire sous l'action des entrées $v_1(t)$ et $v_2(t)$. Nous supposons que $v_1(t)$ et $v_2(t)$ prennent des valeurs constantes $v_{1,k}$ et $v_{2,k}$, dans l'intervalle d'échantillonnage $[kT,(k+1)T]$.
En utilisant le changement de variable donné par l'équation (3.22), on peut déduire les valeurs de $v_{1,k}$ et $v_{2,k}$ dans le nouveau système chaîné ($u_{1,k}$ et $u_{2,k}$). En utilisant ce nouveau changement de variable (équation (3.22)), la nouvelle commande $u_1 = -v_2$ est toujours constante dans l'intervalle de temps $[kT,(k+1)T]$, cependant la deuxième commande : $u_2 = v_1 + a_3 v_2$ n'est plus constante dans l'intervalle de temps. Cette commande dépend de v_1, v_2 et de a_3 (le terme a_3 n'est pas constant car il est la solution de l'équation différentielle donnée par (3.21)).

La forme chaînée (3.21) permet le calcul exact de l'intégral. En posant $a^k = a(kT)$, nous obtenons [Fabrizi *et al.*, 1998] :

$$\begin{cases} a_1^{k+1} = a_1^k - Tv_2^k \\ a_2^{k+1} = a_2^k\cos(Tv_2^k) + a_3^k\sin(Tv_2^k) + v_1^k\frac{\sin(Tv_2^k)}{v_2^k} \\ a_3^{k+1} = -a_2^k\sin(Tv_2^k) + a_3^k\cos(Tv_2^k) + v_1^k\frac{\cos(Tv_2^k)-1}{v_2^k} \end{cases} \tag{3.23}$$

Sous la forme linéaire, ce système se réécrit :

$$a^{k+1} = A_k a^k + b^k \tag{3.24}$$

avec

$$A_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(Tv_2^k) & \sin(Tv_2^k) \\ 0 & -\sin(Tv_2^k) & \cos(Tv_2^k) \end{pmatrix} \tag{3.25}$$

et le nouveau vecteur d'entrée :

$$b^k = \begin{pmatrix} -Tv_2^k \\ v_1^k \frac{\sin(Tv_2^k)}{v_2^k} \\ v_1^k \frac{\cos(Tv_2^k)-1}{v_2^k} \end{pmatrix} \tag{3.26}$$

Le système dynamique donné par l'équation (3.24) est linéaire ce qui permet le calcul exacte des coordonnées a_i $(1 \leq i \leq 3)$ à chaque instant d'échantillonnage. L'utilisation de la matrice inverse de l'équation (3.17), permet de revenir dans le repère classique, et par conséquent aux coordonnées x, y et θ.

3.2.3.5 Modèle cinématique bruité

Le système (3.6) ou son équivalent 3.23 ne tient pas compte des perturbations tel que le glissement des roues, la taille différentes des roues,... Par conséquent, le calcul de la position du véhicule connaissant les entrées v_1 et v_2 (ou u_1 et u_2) engendre des erreurs qui s'accumulent au cours du temps (voir le chapitre 1 décrivant l'inconvénient majeur des systèmes odométriques). Un bruit est ajouté dans le modèle chaîné (3.24). Nous obtenons :

$$a^{k+1} = A_k a^k + b^k + w_k \tag{3.27}$$

où $w \sim N(0, Q_k)$. Compte tenu du changement de variable donné par l'équation (3.17), le premier élément de la diagonale de Q_k représente l'incertitude sur le cap du véhicule, tandis que les deux autres éléments décrivent l'incertitude sur la position. Le fait que l'axe a_2 est colinéaire avec la direction du véhicule, suppose que le bruit gaussien représenté par une ellipse dans le repère universel tourne elle aussi avec le même angle θ. Finalement, pour estimer l'état du système à partir des observations partielles, généralement bruitées (GPS, gyroscope, télémètre,...), nous introduisons l'équation d'observation :

$$y_{k+1} = C(a^k) + d_k \tag{3.28}$$

où $d \sim N(0, R_k)$ est un autre bruit gaussien supposé indépendant de w_k.

Pour conclure cette partie, nous avons proposé d'utiliser et tester le passage par la linéarisation exacte d'un modèle d'évolution cinématique et de le mettre sous la forme d'un modèle chaîné pour la fusion de données multi-capteurs. La mise sous forme chaînée que nous proposons d'utiliser dans cette thèse, contourne le problème de la non linéarité du modèle d'évolution utilisé.
La linéarisation exacte revient à réécrire le système d'équations du modèle cinématique dans un autre espace d'état en effectuant un changement de variable approprié. Le choix judicieux du changement de variable et le passage à un espace d'état généré par des vecteurs de bases autres que ceux d'un espace d'état généré par les vecteurs de bases cartésiens, permet la réécriture du système d'équations du modèle d'évolution sous une forme linéaire qui est celle des

modèles chaînés. Nous pensons et nous l'avons démontré et testé avec des données réelles (voir chapitre 4) que les performances des résultats obtenus en passant par la linéarisation exacte sont bien meilleurs par rapport à l'approche donnée par le Kalman étendu.

3.3 Map-matching

Dans la section 2, nous avons présenté une contribution à la fusion de données multi-capteurs. Un apport important de cette contribution est la proposition d'une méthode de linéarisation qui permet de résoudre le problème de fusion de manière exacte et non approchée comme c'est le cas lorsqu'on utilise le filtre de Kalman étendu. Dans la section suivante, nous proposons une approche qui permet d'améliorer la technique de la fusion présentée précédemment en introduisant des informations cartographiques.

3.3.1 Nécessité de l'aspect multi-hypothèses

Le marché des bases de données cartographiques ne cesse de croître et les constructeurs fournissent aux utilisateurs des produits de plus en plus précis et détaillés. Cependant ces détails ne sont pas toujours en faveur des approches et méthodes utilisées pour la mise en correspondance d'une estimée avec le bon segment.
La figure 3.7 montre un bon exemple d'utilisation de deux cartes différentes. L'estimation donnée par le GPS (point rouge) et l'incertitude autour de cette estimée (l'ellipse) nous fait croire que le véhicule peut se trouver sur l'un des trois segments représentés en bleu sur la figure 3.7 (gauche). Sur la même portion de route et avec une carte plus détaillée, le nombre de segments candidats double (segments en rouge) sur la figure 3.7 (droite) [23].

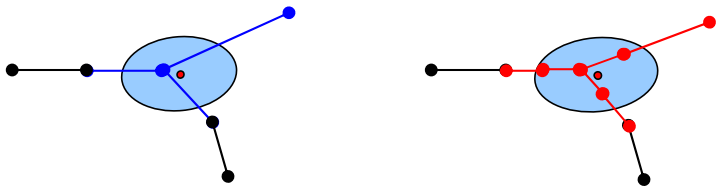


FIG. 3.7 – Le nombre de segments candidats dépend de la précision de la carte et de l'incertitude donnée par l'estimée.

Plusieurs approches de localisation d'un véhicule sur une carte (ou *map-matching*) sont décrites dans le chapitre 1. La plupart des auteurs utilisant ces approches (géométrique, topologique,...) sont d'accord sur le point suivant : leur méthode ou approche ne permet pas de traiter plusieurs candidats à la fois. Ce problème est représenté par la figure 3.8. Sur cette figure, les points rouges représentent l'estimation donnée par le GPS ainsi que l'ellipse d'incertitude. La figure de gauche représente le cas idéal de la mise en correspondance d'une estimée sur un segment. Cependant, le cas le plus fréquent est celui représenté par la figure de droite. Cette ambiguïté est fréquente dans un carrefour ou dans le cas de routes parallèles.
Face à la présence de plusieurs segments candidats, certains selectionnent le segment le plus probable. Dans [Zhao, 1997a] par exemple, l'auteur utilise un seuil pour ne retenir que les segments qui ont un cap proche de celui du véhicule. Un poids pondère chaque segment en fonction de la distance à l'estimée. Finalement le segment de poids le plus fort est retenu.

[23] On suppose qu'on utilise le même GPS dans les deux cas.

Pour trouver la position sur le segment sélectionné, Zhao effectue une simple projection sur le segment. D'autres méthodes de sélection fondées sur la théorie de croyance ou en utilisant la distance de Mahalanobis sont présentées dans [El Najjar, 2003].

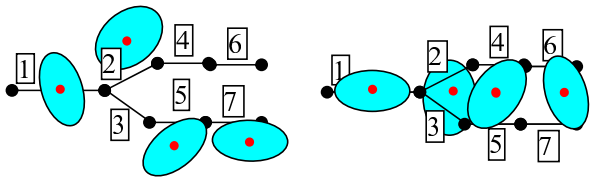


FIG. 3.8 – Les points rouges représentent l'estimation donnée par le GPS ainsi que l'ellipse d'incertitude. La figure de gauche représente le cas idéal de la mise en correspondance d'une estimée sur un segment. Cependant, le cas le plus fréquent est celui représenté par la figure de droite. Cette ambiguïté est fréquente dans un carrefour ou dans le cas de routes parallèles.

3.3.2 Gestion de plusieurs segments

D'un point de vue probabiliste, choisir un segment parmi l'ensemble des candidats se traduit par : *on est sûr à* 100% *que le véhicule est sur ce segment.* Cependant, dans des situations comme celles représentées figure 3.8 de droite, il est difficile sinon impossible de trancher. On se rend compte souvent qu'en choisissant trop tôt une route parmi d'autres (ce qui revient à attribuer une probabilité de 1 à cette route), on a finalement une forte chance de se tromper. La solution à ce type de problème est de garder tous les segments candidats tant que l'ambiguïté n'est pas levée, d'où le multi-hypothèses.

Dans [Gustafsson *et al.*, 2002], les auteurs proposent d'utiliser un filtre particulaire pour la gestion de plusieurs segments. Un nuage de points (N particules) est généré de façon équiprobable sur le réseau routier. Lorsque le véhicule se déplace, les particules subissent le même déplacement. Lors de la fusion avec la carte, les particules qui se retrouvent en dehors des routes se voient affecter des probabilités faibles. Les particules de poids faibles sont éliminées et celles de poids forts sont dupliquées [Gustafsson *et al.*, 2002].
Dans ce contexte, la densité de probabilité présente plusieurs maxima et on parle donc de multimodalité. Tant que cette densité de probabilité est multimodale, la situation est ambiguë. Sur la figure 3.9, l'estimation donnée par le GPS à l'instant $t = 1$, nous confirme que le véhicule roule sur le segment de route numéro 1. La densité de probabilité est unimodale et aucune ambiguïté n'est détectée. Par contre à l'instant $t = 2$, la situation est ambiguë (la densité de probabilité est multimodale) car l'incertitude autour du point GPS nous oblige à traiter les segments de route 1, 2 et 3.

Le filtre particulaire est un outil puissant permettant de traiter le problème de la non linéarité et de se libérer de l'hypothèse gaussienne. Cependant, comme il faut beaucoup de particules pour obtenir une estimation convergente, les traitements informatiques sont lourds même s'ils sont simples. Avec l'augmentation permanente de la puissance des calculateurs, cet inconvénient s'estompe. Les phénomènes de dégénérescence des particules peuvent apparaître et par conséquent, de nombreuses techniques sont étudiées dans ce sens pour remédier à ce problème. Finalement, tout utilisateur du filtrage particulaire a comme souci de répondre à la questions suivante : quel est le nombre minimum de particules à utiliser dans mon application ? Le nombre de particules à utiliser est un choix critique. Ce paramètre dépend de

l'application elle même, du bruit des capteurs et de l'environnement sur lequel l'application fonctionne [Murphy, 2002].

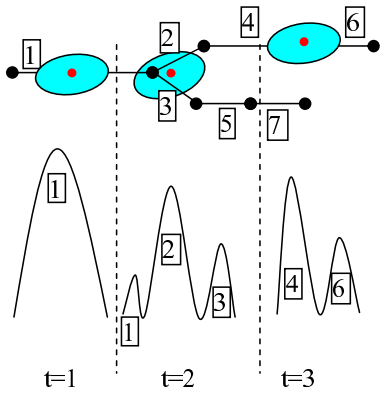


FIG. 3.9 – L'estimation donnée par le GPS ainsi que l'incertitude autour de cette estimation sont représentées par un point rouge et l'ellipse bleue respectivement.

Récemment dans [Jabbour *et al.*, 2008], les auteurs proposent d'utiliser le filtre de Kalman pour la gestion de plusieurs segments. A chaque hypothèse (segment candidat), un filtre de Kalman est généré. Par la suite, la fusion de données est utilisée pour estimer la position du véhicule sur chaque segment. Nous estimons que cette approche est bien meilleure que celle basée sur le filtre particulaire. De ce fait, trouver le nombre minimum de particules à utiliser dans le cas du filtre particulaire et éviter le problème de dégénérescence des poids est plus difficile à gérer par rapport à la gestion d'un ensemble de filtre. Dans ce qui suit, nous proposons une approche basée sur les réseaux bayésiens pour traiter le problème de plusieurs segments.

3.3.3 Réseau bayésien et map-matching

La méthode proposée dans cette thèse pour le problème de map-matching (ou la mise en correspondance d'une estimée sur un segment de route) est fondée sur les réseaux bayésiens. Le choix d'utilisation de cet outil est résumé par les points suivants :

- *Réseau bayésien pour traiter l'incertitude* : comme les capteurs utilisés sont généralement imparfaits et fournissent des informations incertaines, l'utilisation de la théorie probabiliste (Bayes) est justifiée.
- *Réseau bayésien généralise les filtres de Kalman* : le réseau bayésien nous offre la possibilité de manipuler les variables discrètes et continues dans le même réseau, ce que ne permet pas de faire un filtre de Kalman classique. Cette caractéristique permet d'englober plusieurs filtres de Kalman dans le même réseau.
- *Réseau bayésien pour la modélisation de la problématique* : la représentation graphique du réseau bayésien est bien adaptée au type de problème étudié dans cette thèse (map-matching). La variable discrète représente les segments sur lesquels le véhicule peut être et la variable continue représente la position du véhicule sur chaque segment. Cette représentation traduit exactement la définition du terme anglo-saxon : map-matching.

- *Réseau bayésien pour le multihypothèse* : le réseau bayésien permet de gérer plusieurs segments à la fois comme le fait le filtre particulaire sans se soucier du nombre de particules à utiliser.
- *Réseau bayésien pour la fusion de données* : Les capteurs utilisés sont amenés à fournir des informations imparfaites (incertaines). Le réseau bayésien permet la fusion de plusieurs capteurs, et ainsi d'accéder à une information globale plus fiable et plus complète. La complémentarité et la redondance des informations sont alors deux facteurs essentiels pour obtenir un tel effet.

3.3.3.1 Sélection et attribution des probabilités aux segmets

Nous utilisons comme boîte noire les travaux de [El Najjar, 2003] pour l'attribution des probabilités des segments. L'auteur utilise à chaque pas de temps, une estimée avec sa matrice de covariance $(\hat{X}, P)$ pour envoyer une requête au système d'information géographique (SIG). Le SIG retourne tous les segments appartenants à un rayon R choisi principalement pour satisfaire des contraintes de temps réel (voir figure 3.10). Un rayon très grand induit des calculs inutiles et un rayon trop petit risque d'être insuffisant pour contenir la route sur laquelle se situe le véhicule.

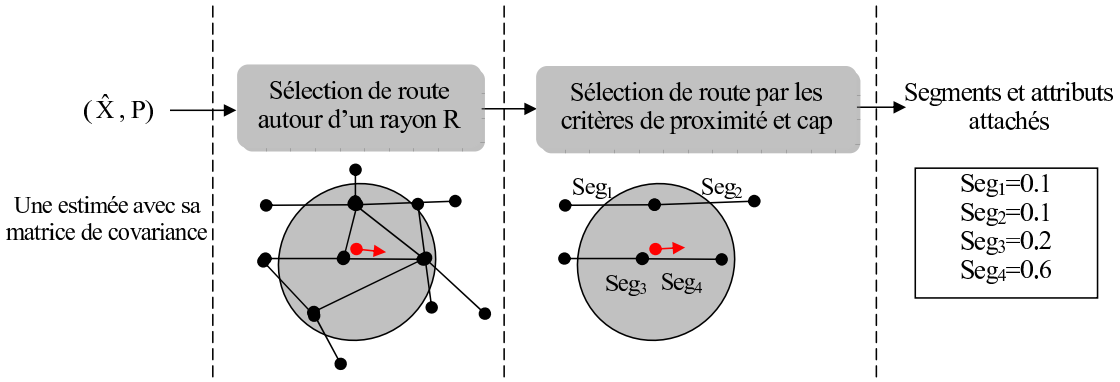


FIG. 3.10 – Extraction des segments autour d'un rayon R et sélection des segments les plus probable en utilisant la distance de Mahalanobis et l'écart entre le cap du véhicule et l'orientation du segment.

Une approche probabiliste basée sur le calcul de la distance de Mahalanobis est utilisée pour attribuer une probabilité à chaque segment. Cette approche appelée méthode de maximum de probabilité effectue une sélection selon les deux critères cités ci-dessus pondérés par les variances : la distance entre l'estimation de la position et l'écart entre le cap et l'orientation des segments. Les segments retournés par le SIG sont classés par distance croissante et on ne garde que les L segments les plus proches dont l'orientation est conforme au cap du véhicule en déplacement. Par exemple, si un segment est plus proche de l'estimée mais sont orientation est perpendiculaire au cap du véhicule, ce segment est éliminé de la liste. A la différence de cette approche qui garde le segment le plus proche [24] pour la construction de l'observation cartographique, nous tenons en compte des L segments pour mettre en valeur le principe de multihypothèse proposé dans cette thèse. Nous favorisons le segment qui vérifie les deux critères cités ci-dessus cependant, en aucun cas nous affectons à ce segment une probabilité

[24] si plusieurs segments ont la même probabilité, l'auteur de cette approche sélectionne l'un de ces segments.

égale à un (sauf si le nombre de segment candidat est égale à un). Cette stratégie renforce bien le concept d'incertitude sur lequel nous travaillons.

3.3.3.2 Modèle de réseau bayésien pour le map-matching

Pour faire valoir le concept de multihypothèse et par conséquent éviter d'ignorer un certain nombre de segments candidats, nous proposons dans ce paragraphe, un modèle de réseau bayésien donné par la figure 3.11.

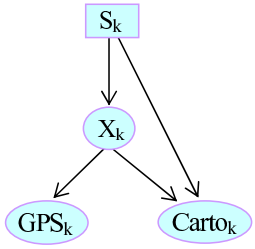


FIG. 3.11 – Modèle de réseau bayésien pour la localisation d'un véhicule sur une carte.

Dans ce modèle, nous utilisons la variable discrète S_k qui représente les segments sur lesquels le véhicule peut être situé : $seg_k^1, seg_k^2, ..., seg_k^n$. Cette variable est initialisée à chaque étape en utilisant les deux critères cités ci-dessus.
La seconde variable utilisée dans le réseau de la figure 3.11 est cachée et de type continue : $X_k = (x_k^i, y_k^i, \theta_k^i)$. La distribution *a posteriori* de cette variable représente la position et le cap du véhicule sur chaque segment candidat de la variable S_k.
Le lien entre la variable discrète S_k et la variable continue $Carto_k$ représente les observations cartographiques (multi-hypothèses). Pour chaque segment candidat de la variable discrète S_k on construit une observation cartographique. Par conséquent, la variable $Carto_k = \{(x_k^{seg_1}, y_k^{seg_1}, \theta_k^{seg_1}), ..., (x_k^{seg_n}, y_k^{seg_n}, \theta_k^{seg_n})\}$.

Le graphe donné par la figure 3.11 nous permet de bien représenter les liens de dépendances conditionnelles entre les variables. Ainsi, la variable d'état $X_k = (x_k, y_k, \theta_k)$ est mise à jour par les observations $Carto_k = \{(x_k^{seg_1}, y_k^{seg_1}, \theta_k^{seg_1}), ..., (x_k^{seg_n}, y_k^{seg_n}, \theta_k^{seg_n})\}$ et $GPS_k = (x_k^{gps}, y_k^{gps})$ si bien sûr les signaux du GPS sont disponibles. L'équation d'observation du GPS est donnée par :

$$GPS_k = \begin{bmatrix} x_k^{gps} \\ y_k^{gps} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \theta_k \end{pmatrix} + \beta_k \tag{3.29}$$

Notons que (x_k, y_k, θ_k) représentent respectivement les coordonnées cartésiennes et le cap du véhicule. A la réception des données GPS, on reçoit également le bruit engendré par ce capteur. La trame *NMEA GST* donne l'ellipse d'incertitude autour de cette estimée. Cette ellipse peut être représentée par une distribution gaussienne $\beta_k \sim N(\mu, Q)$. La matrice de covariance Q est donnée par :

$$Q = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix} \tag{3.30}$$

où σ_x^2 et σ_y^2 représentent respectivement les déviations des mesures en longitude et latitude et σ_{xy} est l'autocorrélation.

Le réseau bayésien calcule la probabilité jointe $P(S_K, X_k|GPS_k, Carto_k)$ qui représente la probabilité que le véhicule se trouve sur chaque segment candidat $S_K = seg_k^1, seg_k^2, ..., seg_k^n$ ainsi que l'estimation de la pose du véhicule (position, cap) sur chaque segment $X_{seg_k^1}, ..., X_{seg_k^n}$.

3.3.3.3 Exemple de spécification numérique des variables

Pour le modèle de réseau donné par la figure 3.11, nous devons spécifier la distribution conditionnelle de chaque variable. Prenons comme exemple la portion de route donnée par la figure 3.12(a). Cette situation d'ambiguïté est fréquente dans les milieux urbains.

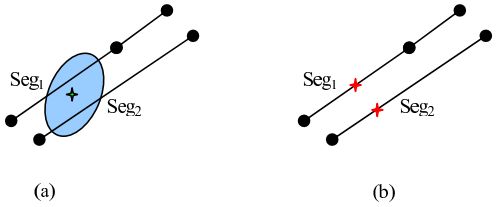


FIG. 3.12 – (a) Exemple de portion de route où la position donnée par les codeurs incrémentaux ou par le GPS ne précise pas si le véhicule roule sur le segment seg_1 ou le seg_2. (b) Multi-hypothèse : exemple de gestion de deux routes par un réseau bayésien.

- *La variable discrète* S_k : cette variable représente les segments sur lesquels le véhicule peut être situé (voir figure 3.12(b)). Dans ce cas, cette variable est donnée comme suit : $S_k = \{seg_1, seg_2\}$. Supposons que le segment seg_1 possède la direction et la position la plus proche de la pose (x, y, θ) du véhicule (estimation donnée par le système odométrique). On affecte à ce segment (par exemple) la probabilité suivante :

$$P(seg_1) = 0.6 \Rightarrow P(seg_2) = 1 - P(seg_1) = 0.4$$

- *La variable continue* X_k : cette variable est initialisée par la projection de la position estimée par le système odomètrique sur chaque segment de la variable S_k. Cette variable est représentée par :

$$P(X_k|Seg_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$$
$$P(X_k|Seg_2) \sim \mathcal{N}(\mu_2, \Sigma_2)$$

  où μ_1, μ_2 représentent la pose du véhicule (x, y, θ) sur chaque segment et Σ_1, Σ_2 représentent les matrices de covariance associées.

- *La variable continue* GPS_k : cette variable représente la première variable observable du réseau. Cette variable donne la position estimée par le GPS. Elle est représentée par :

$$P(GPS_k|X_k) \sim \mathcal{N}(\mu_3, \Sigma_3)$$

  où μ_3 et Σ_3 représentent respectivement la position du véhicule (x, y) et la matrice de covariance associées.

– *La variable continue* $Carto_k$: cette variable représente la seconde variable observable du réseau. La projection de l'estimation donnée par le GPS sur chaque segment de la variable S_k est représentée par :

$$P(Carto_k|Seg_1) \sim \mathcal{N}(\mu_4, \Sigma_4)$$

$$P(Carto_k|Seg_2) \sim \mathcal{N}(\mu_5, \Sigma_5)$$

où μ_4, μ_5 représentent la pose du véhicule (x, y, θ) sur chaque segment et Σ_4, Σ_5 représentent les matrices de covariance associées.

3.3.3.4 Construction de l'arbre de jonction

La figure 3.13 donne l'arbre de jonction correspondant au réseau de la figure 3.11. Rappelons qu'on obtient un arbre de jonction à partir des étapes de moralisation et de triangulation. Le réseau de la figure 3.11(a) est déjà moralisé (les parents d'un même noeuds fils sont reliés entre eux) et au même temps triangulé (aucune boucle de longueur ≥ 4), par conséquent aucun lien n'est donc ajouté.
La clique $\{S_k, X_k, Carto_k\}$ représente la racine de cet arbre. Dans cet exemple, on a une seule possibilité d'affecter les variables aux deux cliques de l'arbre de jonction. Rappelons que pour affecter une variable à une clique (voir chapitre 2), il faut que le(s) parent(s) de cette variable appartiennent à cette même clique ou bien la variable n'a pas de parent. Ainsi, les variables soulignées de l'arbre de jonction donné par la figure 3.13, $\{S_k, X_k, Carto_k\}$ sont affectées à la clique racine et la seule variable affectée à l'autre clique est la variable GPS_k.

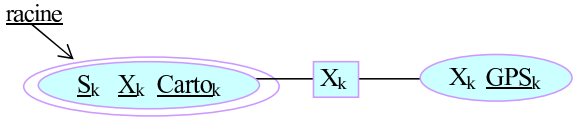


FIG. 3.13 – Arbre de jonction correspondant au réseau bayésien 3.11(a).

3.3.3.5 Initialisation de l'arbre de jonction

Avant qu'un arbre de jonction puisse être utilisé, il doit d'abord être initialisé pour fournir une représentation locale de chaque variable. L'initialisation de l'arbre de jonction prend en compte les spécifications numériques définies sur notre graphe initial. Notons qu'on utilise la forme canonique donnée par l'équation (3.31) pour représenter chaque variable de l'arbre de jonction (voir chapitre 2). Toutes les variables d'une clique seront représentées par le triplet $\phi(g, 0, 0)$ ou $\phi(0, \vec{h}, K)$ suivant si elles sont discrètes ou continues. Les équations calculant les entités g, $\vec{h}$ et K sont données dans le chapitre 2.

$$\phi(x;\ g, \vec{h}, K) \stackrel{def}{=} \exp(g + x^T\vec{h} - \frac{1}{2}x^T K x) \tag{3.31}$$

Le triplet représentant la variable S_k est donné par (on a besoin juste du premier terme g car la variable est discrète) :

$$g_{\{S_k\}}(Seg_k = 0.6) = g_1$$

$$g_{\{S_k\}}(Seg_k = 0.4) = g_2$$

L'affectation de la variable X_k à la même clique racine, donne pour les mêmes valeurs prises par la variable S_k le triplet suivant :

$$g_{\{X_k,S_k\}}(Seg_k = 0.6) = g_3$$
$$\vec{h}_{\{X_k,S_k\}}(Seg_k = 0.6) = \vec{h_3}$$
$$k_{\{X_k,S_k\}}(Seg_k = 0.6) = K_3$$

de façon similaire, on calcule le triplet $\phi(g, \vec{h}, k)$ pour le cas où $Seg_k = 0.4$, nous obtenons :

$$g_{\{X_k,S_k\}}(Seg_k = 0.4) = g_4$$
$$\vec{h}_{\{X_k,S_k\}}(Seg_k = 0.4) = \vec{h_4}$$
$$k_{\{X_k,S_k\}}(Seg_k = 0.4) = K_4$$

3.3.3.6 Mise à jour du réseau par les observations GPS et $Carto$

Les observations (évidences) peuvent être utilisées par l'arbre de jonction initialisé précédemment et le calcul local est effectué afin de calculer les probabilités de chaque variable. L'initialisation des deux variables observables GPS_k et $Carto_k$ suit le même principe utilisé ci-dessus. La variable GPS_k n'a pas de parent de type discret par conséquent, seulement $h_{\{GPS_k,X_k\}}$ et $K_{\{GPS_k,X_k\}}$ sont nécessaires :

$$g_{\{GPS_k,X_k\}} = (0, \vec{h_5}, k_5)$$

La projection de la position donnée par le GPS sur le réseau donne à la variable $Carto$ de nouvelles estimations. Cette variable est représentée par :

$$g_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.6) = g_6$$
$$\vec{h}_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.6) = \vec{h_6}$$
$$k_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.6) = K_6$$

de façon similaire, on calcule le triplet $\phi(g, \vec{h}, k)$ pour le cas où $Seg_k = 0.4$, nous obtenons :

$$g_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.4) = g_7$$
$$\vec{h}_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.4) = \vec{h_7}$$
$$k_{\{Carto_k,S_k,X_k\}}(Seg_k = 0.4) = K_7$$

Rappelons que le séparateur entre les deux cliques du réseau doit être initialisé à 1 afin qu'il permettra de faire passer le flux entre les deux cliques. Le triplet représentant le séparateur est donné par :

$$g_{\{X_k\}}(Seg_k = 0.6) = 0$$
$$\vec{h}_{\{X_k\}}(Seg_k = 0.6) = \vec{0}$$
$$k_{\{X_k\}}(Seg_k = 0.6) = 0$$

de façon similaire, on calcule le triplet $\phi(g, \vec{h}, k)$ pour le cas où $Seg_k = 0.4$, nous obtenons :

$$g_{\{X_k\}}(Seg_k = 0.4) = 0$$

$$\vec{h}_{\{X_k\}}(Seg_k = 0.4) = \vec{0}$$

$$k_{\{X_k\}}(Seg_k = 0.4) = 0$$

L'introduction des observations GPS et $Carto$ dans l'arbre de jonction, consiste à mettre à jour les distributions de probabilités des variables cachées. Le flux de données passe le long des arcs des feuilles jusqu'à la racine puis de la racine vers les feuilles. Chaque clique passe son potentiel à travers le séparateur en utilisant les formules suivantes (voir chapitre 2) :

$$\psi^*_{separateur}(X_k) = \sum_{var \in source \setminus X_k} \psi_{source}(X_{C_i})$$

la clique destination modifie son potentiel par :

$$\psi^*_{destination}(X_{C_j}) = \psi_{destination}(X_{C_j})\lambda_{separateur}(X_k)$$

$$\lambda_{separateur}(X_k) = \frac{\psi^*_{separateur}(X_k)}{\psi_{separateur}(X_k)}$$

le terme $\lambda_{separateur}(X_k)$ est le facteur de mise à jour du potentiel de la clique destination. Rappelons que le produit et la division de deux variables aléatoires écrites sous forme canonique suit les formules suivantes (voir chapitre 2) :

$$\zeta(K_1, \vec{h}_1, g_1) \times \zeta(K_2, \vec{h}_2, g_2) \stackrel{def}{=} \zeta(K_1 + K_2,\ \vec{h}_1 + \vec{h}_2,\ g_1 + g_2)$$

$$\frac{\zeta(K_1, \vec{h}_1, g_1)}{\zeta(K_2, \vec{h}_2, g_2)} \stackrel{def}{=} \begin{cases} \zeta(K_1 - K_2,\ \vec{h}_1 - \vec{h}_2,\ g_1 - g_2) & \text{si } \zeta(K_2,\ \vec{h}_2,\ g_2) \neq 0 \\ 0 & \text{sinon} \end{cases}$$

Une fois que les flux sont passés le long des feuilles jusqu'à la racine et inversement de la racine vers les feuilles, l'arbre atteint un état d'équilibre. Par conséquent, on obtient une nouvelle représentation des potentiels en fonction de l'évidence. Ce qui correspond à corriger la position du véhicule sur chaque segment candidat. Notons qu'on utilise le moteur d'inférence réalisé par Murphy[25] pour faire l'inférence.

3.3.4 Synoptique de la méthode basée sur les réseaux bayésiens

Nous utilisons la fusion des mesures GPS, des mesures des capteurs proprioceptifs et des données de la carte afin de produire une estimation de la pose du véhicule. On s'intéressera en même temps à la recherche de la route (segment de route) sur laquelle le véhicule roule, avec une quantification du degré de certitude, en fonction de la qualité des informations manipulées et de la configuration géométrique du réseau routier autour du véhicule. Le synoptique de notre approche fondé sur les réseaux bayésiens est donné figure 3.14.
Dans un premier temps, supposons que l'on ait une estimation donnée par le système odométrique ou par le GPS. Cette estimation est donnée au Système d'Information Géographique (SIG) afin de sélectionner un ensemble de segments autour de cette estimée. Le résultat de

[25] http ://code.google.com/p/bnt/

cette étape est un ensemble de segments numérotés comme suit : $seg_k^1, seg_k^2, ..., seg_k^n$ où k représente l'instant de sélection. La position sur chaque segment est donnée par la projection de l'estimée sur ces segments. Si la projection orthogonale n'existe pas, alors on prend le point le plus proche du segment en question. Le résultat de la projection produit l'ensemble d'observations cartographiques : $Carto_k = \{(x_k^{seg_1}, y_k^{seg_1}, \theta_k^{seg_1}), ..., (x_k^{seg_n}, y_k^{seg_n}, \theta_k^{seg_n})\}$.
L'introduction des observations $Carto_k$ et GPS_k dans le réseau bayésien permet de mettre à jour les distributions de probabilités. Après inférence, nous obtenons les probabilités de chaque segment candidat sur lequel le véhicule peut être (représenté par la variable S_k) ainsi que la position et le cap du véhicule sur chaque segment (représenté par la variable X_k).

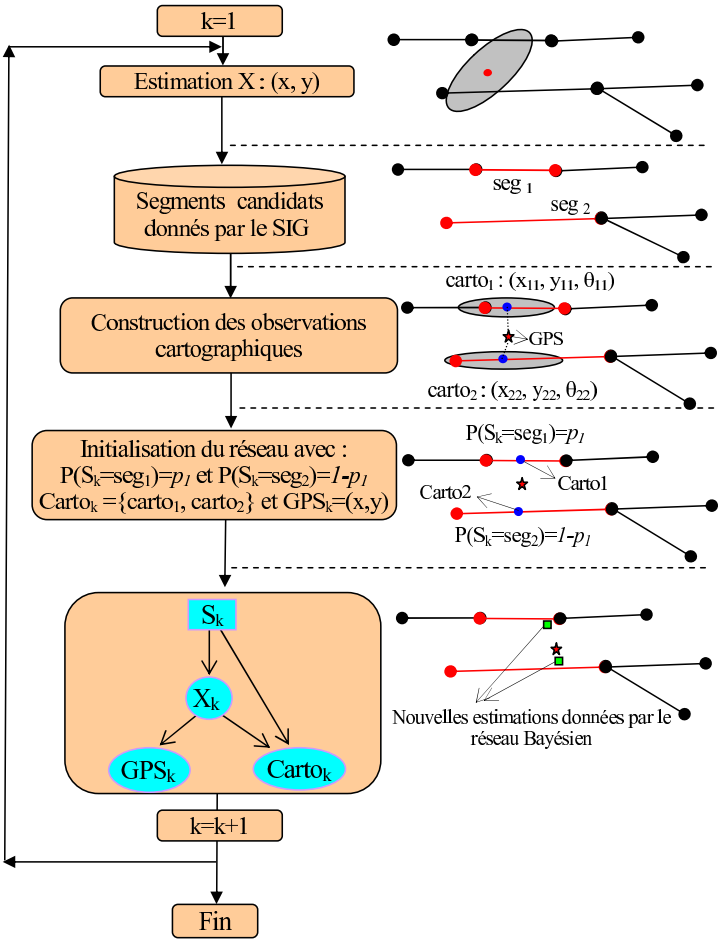


FIG. 3.14 – Synoptique de l'approche basée sur les réseaux bayésiens.

3.3.5 Exemple

L'exemple suivant illustre le fonctionnement de notre méthode. Sur la figure 3.15, le véhicule se déplace à partir du segment de route numéro 1 vers le segment de route numéro 8. Supposons à l'instant $t = 1$ qu'une estimation est donnée par le GPS (ou l'odomètre). A partir de cette position nous sélectionnons tous les segments possibles (cette étape est donnée par le Système d'Information Géographique). Ici, un seul segment est candidat. On projette l'estimation GPS sur le seul segment candidat (segment numéro 1) pour construire une observation cartographique : $Carto_1 = \{(x_1^{seg_1}, y_1^{seg_1}, \theta_1^{seg_1})\}$.

Les observations $Carto_1$ et GPS_1 servent à mettre à jour les distributions de probabilités de la variable cachée X_1. Dans ce cas trivial, on est sûr que le véhicule roule sur le segment de route numéro 1 ($P(S_1 = seg_1|Carto_1, GPS_1) = 1$). La position du véhicule sur le segment seg_1 est donnée par la variable cachée $P(X_1|Carto_1, GPS_1) \sim N(\mu_1, \Sigma_1)$, où $\mu_1 = (x_1, y_1, \theta_1)$ et Σ_1 représente la matrice de covariance (incertitude de cette estimation).
A l'instant $t = 2$, l'incertitude des capteurs (odomètre, carte, GPS) nous oblige à prendre en compte les segments 2 et 3. Ces segments sont utilisés pour générer deux observations cartographiques $Carto_2 = \{(x_2^{seg_2}, y_2^{seg_2}, \theta_2^{seg_2}), (x_2^{seg_3}, y_2^{seg_3}, \theta_2^{seg_3})\}$. La mise à jour du réseau bayésien est faite par les nouvelles observations $Carto_2$ et GPS_2 et nous obtenons deux nouvelles estimations correspondant aux deux segments $P(S_2 = seg_2|Carto_2, GPS_2) = 0.75$ et $P(S_2 = seg_3|Carto_2, GPS_2) = 0.25$. Dans ce cas on a 75% de chance que le véhicule roule sur le segment de route numéro 2 et 25% de chance que le véhicule se trouve sur le segment de route numéro 3. La position du véhicule sur chaque segment est donnée par la probabilité jointe : $P(X_2, S_2|Carto_2, GPS_2)$. Le tableau 3.1 récapitule le processus d'inférence pour $1 \leq t \leq 4$.

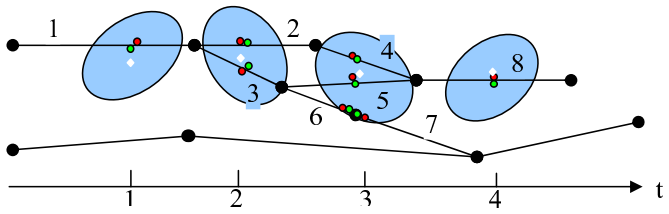


FIG. 3.15 – Exemple de gestion de plusieurs segments en utilisant un réseau bayésien. Les points blancs représentent les estimations données par le GPS. Les ellipses représentent l'incertitude autour de chaque estimée. Les observations cartographiques sont représentées par des points rouges et les estimations données par le réseau bayésien sont données par des points verts.

t	$Carto_t$	$P(S_t\|Carto_t, GPS_t)$	$P(X_t\|Carto_t, GPS_t)$
1	$\{seg_1\}$	$P(S_1 = seg_1\|Carto_1, GPS_1) = 1$	$\mu_1^{seg_1}, \Sigma_1^{seg_1}$
2	$\{seg_2, seg_3\}$	$P(S_2 = seg_2\|Carto_2, GPS_2) = 0.75$	$\mu_2^{seg_2}, \Sigma_2^{seg_2}$
		$P(S_2 = seg_3\|Carto_2, GPS_2) = 0.25$	$\mu_2^{seg_3}, \Sigma_2^{seg_3}$
3	$\{seg_4, seg_5, seg_6, seg_7\}$	$P(S_3 = seg_4\|Carto_3, GPS_3) = 0.55$	$\mu_3^{seg_4}, \Sigma_3^{seg_4}$
		$P(S_3 = seg_5\|Carto_3, GPS_3) = 0.25$	$\mu_3^{seg_5}, \Sigma_3^{seg_5}$
		$P(S_3 = seg_6\|Carto_3, GPS_3) = 0.15$	$\mu_3^{seg_6}, \Sigma_3^{seg_6}$
		$P(S_3 = seg_7\|Carto_3, GPS_3) = 0.05$	$\mu_3^{seg_7}, \Sigma_3^{seg_7}$
4	$\{seg_8\}$	$P(S_4 = seg_8\|Carto_4, GPS_4) = 1$	$\mu_4^{seg_8}, \Sigma_4^{seg_8}$

TAB. 3.1 – Exemple du processus d'inférence donné par le réseau bayésien dans le cas du concept multihypothèse.

3.3.6 Aspect temporel du réseau

Afin de tenir compte de l'évolution temporelle, nous utilisons un réseau bayésien dynamique. Ce réseau est créé en dupliquant le réseau bayésien statique donné par la figure 3.11 et en reliant ces réseaux par des liens d'un pas à l'autre (à chaque fois qu'on a une nouvelle

observation). La figure 3.16 représente le réseau bayésien dynamique sur trois pas de temps pour la localisation d'un véhicule sur une carte. D'après cette figure, nous avons deux nœuds persistants qui sont reliés d'une étape à l'autre. Les nœuds en question sont S_t et X_t. Notons la ressemblance entre le modèle proposé ci-dessus et le Switching Kalman Filter donné dans le chapitre 2.

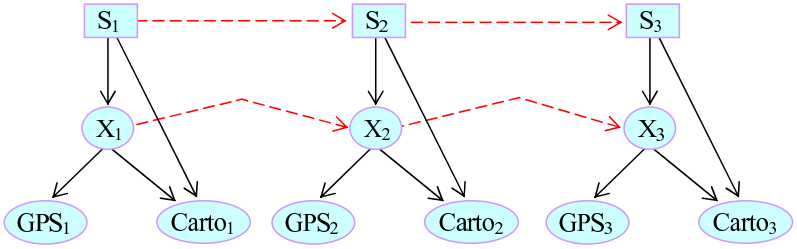
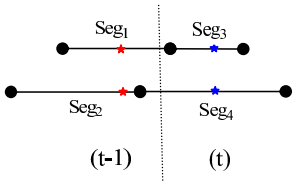


FIG. 3.16 – Réseau bayésien dynamique sur Trois pas de temps ($t = 3$) pour la localisation d'un véhicule sur une carte.

Nous tirons profit des algorithmes de type "pruning algorithms" pour réduire le nombre de gaussiennes dans le cas de notre réseau représenté par la figure 3.16. La distribution le long d'un segment à l'instant t est supposée ne dépendre que de la distribution le long du segment directement connecté à l'instant $t-1$. Les hypothèses sont propagées le long de chaque segment mais pas entre les segments (une seule gaussienne par segments candidat est considérée à chaque instant t). Ce principe n'est pas évident d'après la structure du graphe, mais il est implicite quant à la description des tables de probabilités conditionnelles. Ceci peut se traduire par l'introduction d'une probabilité conditionnelle $P(S_t|S_{t-1})$ égale à la matrice identité. L'exemple donné par la figure 3.17 illustre ce principe.



S_{t-1} \ S_t	Seg_3	Seg_4
Seg_1	1	0
Seg_2	0	1

FIG. 3.17 – Exemple montrant les hypothèses propagées le long de chaque segment $Seg_1 \rightarrow Seg_3$, $Seg_2 \rightarrow Seg_4$ mais pas entre les segments $Seg_1 \rightarrow Seg_4$, $Seg_2 \rightarrow Seg_3$.

L'arbre de jonction est construit pour chaque $J_t = I_{t-1} \cup V_t$:

$$J_1 = V_1 = \{S_1, X_1, gps_1, carto_1\}$$

$$J_2 = I_1 \cup V_2 = \{S_1, X_1\} \cup \{S_2, X_2, gps_2, carto_2\}$$

$$J_3 = I_2 \cup V_3 = \{S_2, X_2\} \cup \{S_3, X_3, gps_3, carto_3\}$$

en s'assurant qu'il existe au moins une clique D_t contenant l'interface I_{t-1} et une clique C_t contenant l'interface I_t. Pour cette raison, il suffit d'ajouter un lien entre toutes les variables de $I_1 = \{S_1, X_1\}$ et celles de I_t après l'étape de moralisation (les liens $S_1 - X_1$ et $S_2 - X_2$ existent déjà dans le réseau statique). Les nœuds de l'interface d-sépare le passé du futur : $\{S_1, X_1, GPS_1, Carto_1, GPS_2, Carto_2\} \perp \{S_3, X_3, GPS_3, Carto_3\}|\{S_2, X_2\}$.
Les arbres de jonctions J_t sont reliés au moyen de leurs interfaces $I_1 = \{S_1, X_1\}$ et $I_2 =$

$\{S_2, X_2\}$ comme le montrent la figure 3.18. Nous pouvons effectuer l'inférence dans chaque arbre séparément, puis transmettre les messages entre eux via les nœuds d'interface.

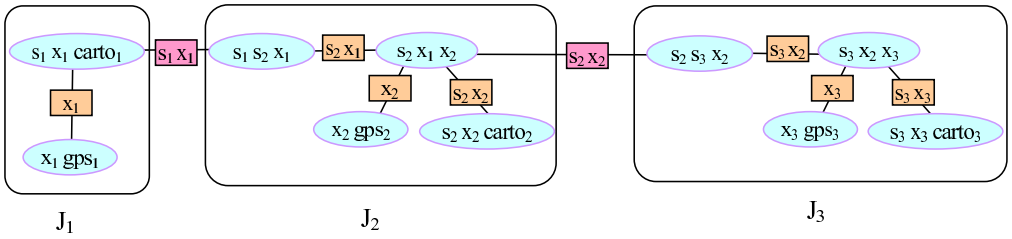


FIG. 3.18 – Schéma montrant comment construire l'arbre de jonction d'un réseau bayésien déroulé. $I_1 = \{S_1, X_1\}$, $I_2 = \{S_2, X_2\}$, $D_2 = \{S_1, S_2, X_1\}$, $C_2 = \{S_2, X_1, X_2\}$, $D_3 = \{S_2, S_3, X_2\}$ et $C_3 = \{S_3, X_2, X_3\}$.

3.4 Convoi de véhicule

3.4.1 Introduction

Dans les dix dernières années, on a vu apparaître un grand nombre de voitures dites intelligentes. Ces voitures apportent à l'usager un moyen de transport de proximité adapté à la ville, baissant les coûts de déplacement, diminuant le stress occasionné par la conduite, réduisant les pollutions sonores et atmosphériques.... Parmi ces voitures intelligentes on trouve les voitures évoluant en convoi.

3.4.2 Problématique étudiée

La tâche de la navigation en convoi consiste à amener un ensemble de véhicules disposé en file d'un point à un autre (suivre un chemin de référence donné par exemple par le véhicule de tête) en respectant un écart prédéfini entre les véhicules (voir la figure 3.19). Afin de réaliser cette tâche, chaque véhicule du convoi doit être en mesure de se localiser par rapport au chemin de référence et par rapport aux autres véhicules.
Dans ce contexte de travail, la plupart des études et expérimentations sur ce thème de recherche sont fondées sur le contrôle d'un convoi. Les auteurs de ces travaux supposent que chaque véhicule est en mesure de se localiser par rapport au chemin de référence par le biais d'un GPS très précis [Thuilot *et al.*, 2002] et [Bom *et al.*, 2005]. Cependant tout au long de cette thèse on a constaté que l'utilisation du GPS pour la localisation n'est pas toujours possible en raison de l'intermittence des signaux données par les satellites : absence de GPS sous un tunnel, estimation erronée si le nombre de satellites n'est pas suffisant... Dans le but de respecter le thème traité dans cette thèse (fusion et localisation), nous nous sommes intéressés au problème de la localisation d'un convoi (train de véhicules). La fusion de données nous permettra de remédier aux problèmes du GPS cités ci-dessus. Notons que lors de la simulation, nous avons utilisé une commande proportionnelle plutôt simpliste.

3.4.3 Localisation absolue et relative

Dans le but de reproduire un chemin de référence donné préalablement à tous les véhicules ou transmis par le véhicule de tête, chaque véhicule doit être en mesure de connaître sa position

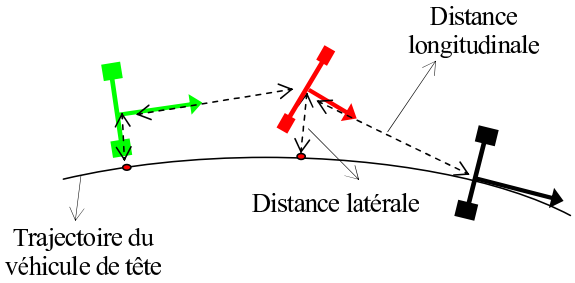


FIG. 3.19 – Représentation d'un convoi constitué d'un véhicule de tête et de deux suiveurs.

par rapport au chemin de référence. Dans ce même contexte, les véhicules constituant le convoi doivent se localiser les uns par rapport aux autres afin de respecter l'écart prédéfini entre les véhicules. La localisation d'un convoi est ainsi divisée en deux types :

1. la localisation absolue, définie comme la localisation de chaque véhicule par rapport au chemin de référence.
2. la localisation relative, définie comme la localisation de chaque véhicule par rapport aux autres.

3.4.3.1 La localisation absolue

La localisation absolue consiste à localiser des véhicules dans un repère global. Des balises magnétiques peuvent être placées dans l'environnement comme cela a été fait dans le projet autoroute automatisée (AHS : Automated Highway System) proposé dans le cadre du projet PATH (California Partners for Advanced Transit and Highways) [Hedrick, 1997]. L'infrastructure dédiée à ce système de localisation fonctionne parfaitement, cependant le coût d'équipement est très élevé ce qui ne permet pas de modifier ou déplacer facilement cette infrastructure. Pour remédier à ce problème, les chercheurs s'orientent vers des systèmes de localisation plus simples à mettre en place.

Les capteurs GPS (Global Positioning System) permettent une localisation avec une précision de l'ordre de quelques mètres. En particulier, les RTK-GPS (Real-Time Kinematic GPS) sont capables de fournir en temps réel une localisation absolue avec une précision centimétrique. Malheureusement, si cette solution est particulièrement séduisante pour la localisation absolue, le fonctionnement n'est pas toujours assuré au milieu urbain comme on l'a vu tout au long de cette thèse. La visibilité des satellites peut être occultée par les grands immeubles ce qui rend cette solution pas toujours disponible. Dans ce cas, la vision peut être une solution alternative aux capteurs GPS.

Dans [Blanc *et al.*, 2005], la localisation du robot mobile est réalisée grâce à une succession d'images clés apprises (phase d'apprentissage). D'une façon presque similaire, [Royer *et al.*, 2004] proposent d'enregistrer des séquences vidéo d'un robot conduit manuellement. A partir de cette séquence, le robot doit être en mesure de reproduire la même trajectoire. Cette trajectoire de référence reconstruite est alors utilisée pour localiser en temps réel le robot mobile. Le coût de ce système de localisation basé sur les capteurs de vision est bien moindre que celui des récepteurs RTK-GPS.

3.4.3.2 La localisation relative

En plus de la localisation absolue, la navigation en convoi impose également que chaque véhicule puisse se localiser par rapport aux autres éléments du convoi afin de maintenir une inter-distance entre les véhicules. Les systèmes de perception tels que les caméras, radars,...sont *a priori* les capteurs de localisation relative par excellence. Cependant, l'inconvénient majeur de ce type de perception et leur champ d'action relativement restreint. Les éléments du convoi les plus en amonts peuvent sortir du champ de perception de ces capteurs ce qui rend ces systèmes non adaptés pour la localisation au sein du convoi. De ce fait, ces capteurs peuvent être parfaitement utilisés par exemple pour la détection d'obstacle ou le(s) véhicule(s) le(s) plus proche(s) [Bom, 2006].

Finalement, la solution la plus efficace pour un convoi de véhicules est d'utiliser la localisation absolue sur chaque véhicule du convoi et par communication, chaque véhicule peut transmettre et recevoir la position des autres véhicules constituant le convoi. *A priori*, cette solution est adoptée par plusieurs chercheurs car elle est cohérente avec la problématique étudiée [Bom, 2006].

3.4.4 Capteurs utilisés sur chaque véhicule du convoi

La flexibilité de l'utilisation des réseaux bayésiens nous permet d'ajouter et de retirer des capteurs d'une façon relativement simple. Chaque capteur est modélisé ou représenté dans le réseau par une variable (une observation) qui sert à mettre à jour l'estimation de la position et du cap : (x, y, θ). Du moment que la trajectoire du véhicule de tête est connu par l'ensemble des voitures du convoi, cette trajectoire doit être de grande précision. Dès lors, le véhicule de tête est équipé d'un système de localisation centimétrique de type GPS-RTK. Ce GPS fournit des mesures avec une fréquence d'échantillonnage de 10Hz, avec une précision de 2cm. Afin de donner au véhicule de tête une bonne estimation de son cap, on alimente ce véhicule par un deuxième capteur : gyroscope.
Dans de nombreux travaux sur un convoi de véhicules, les véhicules suiveurs sont équipés d'un système de positionnement centimétrique GPS-RTK [Bom *et al.*, 2005] et [Martinet *et al.*, 2005] et d'un télémètre. Dans le but de réduire le coût de l'infrastructure du convoi et mettre en valeur le concept de fusion par réseaux bayésiens, les véhicules suiveurs sont équipés d'un GPS de précision métrique. Un télémètre est placé dans chaque voiture pour donner une estimation de la distance séparant le véhicule de celui qui le précède et l'angle par rapport à l'axe des abscisses (voir la figure 3.20). Nous pensons que la fusion de la distance donnée par le télémètre et la position centimétrique donnée par le GPS-RTK, nous permet de réduire l'incertitude autour de l'estimation donnée par le GPS du premier véhicule suiveur. Ainsi chaque véhicule corrige sa position par rapport à son prédécesseur.

3.4.5 Réseau bayésien et train de véhicule

Pour amener un train de véhicules à suivre un chemin de référence et en respectant un écart prédéfini entre les véhicules, nous posons les hypothèses suivantes :

- Le véhicule de tête est conduit par un être humain. Ce véhicule est équipé d'un système de localisation (GPS RTK) centimétrique et d'un gyroscope pour donner son cap.
- Le chemin du véhicule de tête est transmit à l'ensemble des voitures du convoi.

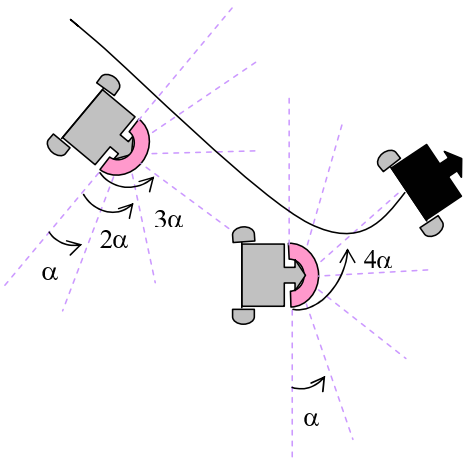


FIG. 3.20 – Le télémètre donne la distance et l'angle par rapport à l'axe des abscisses entre deux véhicules.

- Les véhicules suiveurs sont équipés d'un télémètre pour donner la distance entre les véhicules et l'angle par rapport à l'axe des abscisses. Chaque véhicule suiveur est équipé d'un système de positionnement (GPS) de précision métrique.
- Chaque véhicule du convoi doit maintenir une distance de sécurité (DistS= distance longitudinale) avec le véhicule qui le précède et contrôler l'actionneur de direction pour réduire l'écart latéral (DistL= distance latérale) avec le chemin de référence.
- Le modèle d'évolution de chaque véhicule est donné par l'équation (3.32) (voir chapitre 1, paragraphe 1.2.2 sur les codeurs incrémentaux).

$$X_{k+1} = \begin{cases} x_{k+1} = x_k + d_k \cos(\theta_k + \omega_k/2) \\ y_{k+1} = y_k + d_k \sin(\theta_k + \omega_k/2) \\ \theta_{k+1} = \theta_k + \omega_k \end{cases} \tag{3.32}$$

3.4.5.1 Modèle de réseau pour la localisation d'un train de véhicules

Le modèle de réseau bayésien pour la localisation d'un convoi de véhicules est donné figure 3.21. Le véhicule de tête modélisé par son état X_T est équipé d'un système de localisation centimétrique modélisé par la variable : $GPS = (x_{gps}, y_{gps})$. Un deuxième capteur (gyroscope) est utilisé par le véhicule de tête pour donner son cap est représenté par la variable *Gyro*. La position et le cap du véhicule de tête $X_T = (x_T, y_T, \theta_T)$ seront mis à jour par les deux observations *GPS* et *Gyro*. Notons que la nouvelle équation d'observation (GPS, Gyro) pour le véhicule de tête est donnée par l'équation (3.33).

$$Y_k^T = \begin{bmatrix} x_k^{GPS} \\ y_k^{GPS} \\ \theta_k^{Gyro} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \theta_k \end{pmatrix} + \alpha_k \tag{3.33}$$

où $\alpha_k \sim N(\mu, Q)$. La matrice de covariance Q est donnée par :

$$Q = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{pmatrix} \tag{3.34}$$

où σ_x^2, σ_y^2 et σ_θ^2 représentent respectivement les déviations des mesures en longitude, latitude et cap.

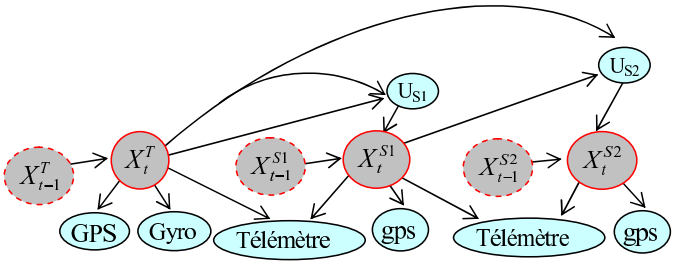


FIG. 3.21 – Modèle de réseau bayésien pour la localisation d'un convoi constitué d'un véhicule de tête et de deux véhicules suiveurs.

Comme la distance donnée par le télémètre nécessite deux positions : (x_1, y_1) et (x_2, y_2), ce capteur est modélisé dans la figure 3.21 entre chaque paire de véhicules. La relation qui lie ces deux positions s'écrit sous la forme suivante :

$$Télémètre \equiv Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.35}$$

De cette manière, connaissant la position centimétrique du véhicule de tête (X_T) et la distance le séparant du premier suiveur (donnée par le télémètre du premier véhicule), on peut réestimer la position du premier suiveur (X_{S1}) et, par conséquent, réduire l'incertitude autour de l'estimation donnée par son GPS. Du moment qu'on a plus de précision sur la position du premier suiveur, cette nouvelle position sert à réestimer la position du second suiveur et ainsi de suite pour chaque véhicule du convoi. La figure 3.22 donne un aperçu de cette approche.

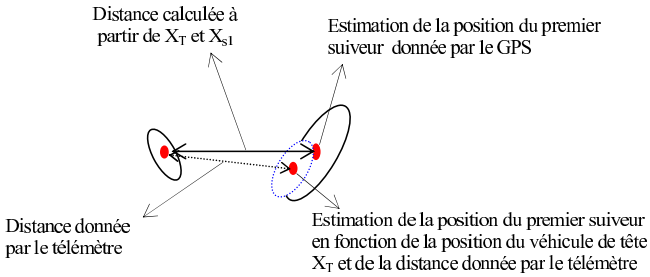


FIG. 3.22 – Ce schéma montre comment réduire l'incertitude donnée par le GPS des véhicules suiveurs en utilisant la distance donnée par leur télémètre et la position du véhicule de tête. Cette correction se fait de proche-en-proche.

Chaque véhicule suiveur est modélisé sur la figure 3.21, par son état X_{S_i}. Chaque véhicule est équipé d'un système de localisation $gps = (x_{gps}, y_{gps})$ et d'un télémètre. La fusion de ces

deux capteurs donne une nouvelle estimation de la position de chaque véhicule. La trajectoire du véhicule de tête doit être transmise à tous les véhicules suiveurs par conséquent, les liens (X_T, U_{S_1}) et (X_T, U_{S_2}) serviront à commander le cap du véhicule pour réduire l'écart latéral (DistL) avec le chemin de référence (ces liens représentent seulement la transmission de la trajectoire du véhicule de tête aux véhicules suiveurs est non pas la probabilité $P(U_{S1}|X_T)$, $P(U_{S1}|X_T)$). Finalement, pour maintenir une distance de sécurité (DistS) chaque véhicule doit connaître la vitesse (ou la position) du véhicule qui précède ce qui est représenté sur le graphe de la figure 3.21 par les liens (X_T, U_{S_1}) et (X_{S_1}, U_{S_2}). Ces liens représentent seulement la transmission de la vitesse ou la position du véhicule prédécesseur au véhicule suiveur est non pas la probabilité $P(U_{S1}|X_T)$, $P(U_{S2}|X_{S1})$.
Notons que le véhicule de tête ne possède pas la variable de commande U, à l'instar des véhicules suiveurs (U_{S_1}, U_{S_2}) car ce véhicule est conduit par un conducteur humain. Notons également, qu'on a retiré l'observation cartographique (voir le cas mono véhicule). La flexibilité des réseaux bayésiens nous permet à tout moment de rajouter d'autres observations (cartographie pour tous les véhicules, gyroscope pour les véhicules suiveurs,...).

3.4.5.2 Problématique du modèle proposé

Les équations d'observations manipulées jusqu'à présent (GPS, cartographie, gyroscope) étaient des équations linéaires. Cependant, la seule observation qui ne peut être écrite sous forme linéaire est la distance donnée par le télémètre (voir l'équation (3.35)). En vue de garder les fonctions d'observations linéaires, une transformation est faite sur le réseau bayésien de la figure 3.21.

3.4.5.3 Nouveau modèle pour la localisation d'un train de véhicule

La variable représentant le télémètre dans le réseau de la figure 3.21 est une fonction non linéaire de variables gaussiennes (X_T, X_{S_1}, X_{S_2}). Le problème auquel nous nous sommes affronté est le suivant :

1. On veut réécrire la distance donnée par le télémètre sous une forme linéaire.
2. On veut tirer profit de l'exactitude de l'estimation donnée par le RTK-GPS du véhicule de tête et du télémètre dans le but de donner une estimation plus précise de chaque véhicule suiveur.

Afin de tenir compte de ces deux remarques, nous proposons l'idée suivante. Du moment que chaque véhicule suiveur possède un télémètre pour estimer la distance avec celui qui le précède, on procède comme suit. Le premier véhicule suiveur S_1 estime sa position à partir de la position et la distance le séparant de celui du véhicule de tête. Connaissant la distance D_1 (donnée par son télémètre) et la position du véhicule de tête (x_T, y_T) transmise par ce dernier, on construit une nouvelle observation comme suit (voir la figure 3.23) :

$$\begin{cases} x_{S_1}^{nouv} = x_T - D_1 \cos(\beta_1) \\ y_{S_1}^{nouv} = y_T - D_1 \sin(\beta_1) \end{cases} \tag{3.36}$$

De la même manière, le second véhicule suiveur estime sa position à partir de celui qui le précède (S_1) et de la distance D_2 donnée par son télémètre. Dès lors, chaque véhicule suiveur S_i où $2 \leq i \leq n$ estime sa position comme suit :

$$\begin{cases} x_{S_i}^{nouv} = x_{i-1} - D_i \cos(\beta_i) \\ y_{S_i}^{nouv} = y_{i-1} - D_i \sin(\beta_i) \end{cases} \tag{3.37}$$

L'équation linéaire de l'observation construite à partir de la distance donnée par le télémètre et de la position du prédécesseur s'écrit donc :

$$Y_{S_i}^{nouv} = \begin{bmatrix} x_{S_i}^{nouv} \\ y_{S_i}^{nouv} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \alpha_k \tag{3.38}$$

où α_k est le bruit gaussien produit par le télémètre et x_k, y_k représentent la position du véhicule.

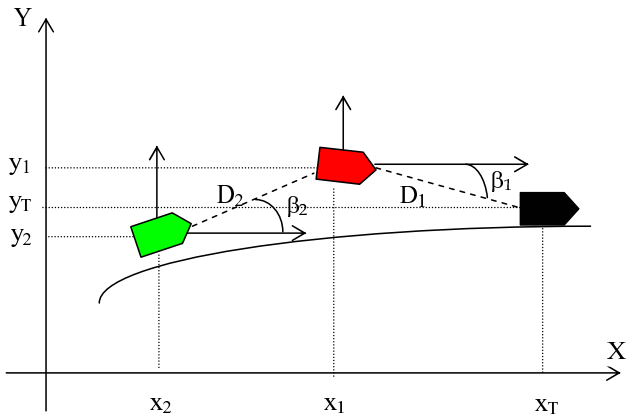


FIG. 3.23 – Nouvelle estimation construite à partir de la position du véhicule prédécesseur et de la distance mesurée par le télémètre.

A partir de cette nouvelle représentation des mesures télémètriques, on peut faire les remarques suivantes :

1. L'équation (3.38) répond bien aux deux remarques citées ci-dessus.
2. Une fois de plus, on voit la flexibilité que nous donne les réseaux bayésiens à manipuler (rajouter, enlever et fusionner) des nouvelles données.

Le nouveau modèle pour la localisation d'un convoi de véhicule est donné figure 3.24. Ce modèle est équivalent à celui donné figure 3.21. La position estimée pour chaque véhicule suiveur (X_{S_i}) est le résultat de la fusion de la nouvelle observation créée Y_i^{nouv} et du GPS.

3.4.6 Commandes proportionnelles

Pour amener un ensemble de véhicules disposé en file à suivre une trajectoire de référence donnée par le véhicule de tête tout en respectant un écart prédéfini entre les véhicules, chaque véhicule du convoi doit connaître sa position et la position (ou vitesse) du véhicule qui le précède. Pour valider nos expériences de localisation, nous utilisons des commandes simples dites commandes proportionnelles. Des commandes plus développées sont utilisables dans les travaux de [Thuilot *et al.*, 2002] et [Bom, 2006].
En se référant à la figure 3.25, chaque véhicule du convoi peut ajuster sa vitesse V_{S_i} connaissant la vitesse (ou position) du véhicule prédécesseur. Ainsi, la commande longitudinale de chaque

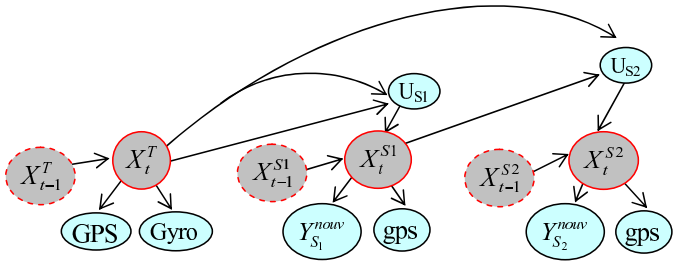


FIG. 3.24 – Nouveau modèle de réseau bayésien pour la localisation d'un convoi constitué d'un véhicule de tête et de deux véhicules suiveurs.

véhicule est donnée par l'équation (3.39). Le terme DR_i représente la distance réelle donnée par le télémètre, et k_1 est un facteur de proportionnalité permettant de spécifier une convergence plus ou moins rapide vers la distance de sécurité DS.

$$V_{S_i} = V_{S_{i-1}} + k_1(DR_i - DS) \tag{3.39}$$

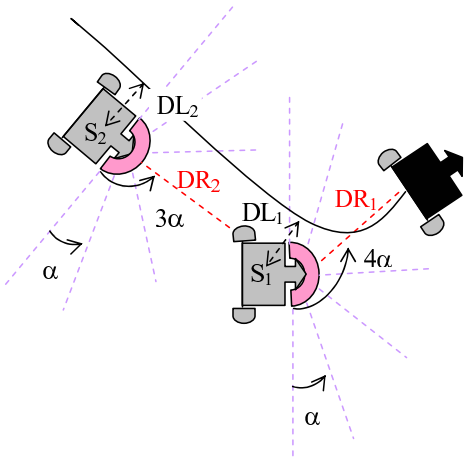


FIG. 3.25 – Un modèle de convoi constitué d'un véhicule de tête et de deux véhicules suiveurs. La distance réelle donnée par le télémètre est représentée par DR_1 et DR_2, et la distance latérale de chaque véhicule par rapport à la trajectoire du véhicule de tête est donnée par DL_1 et DL_2.

En se référant toujours à la figure 3.25, chaque véhicule du convoi doit ajuster son cap de telle sorte qu'il reproduise la trajectoire du véhicule de tête. Ainsi, connaissant le cap du véhicule de tête θ^T à l'instant k (la tangente de la courbe au point perpendiculaire de la position du chaque véhicule) et son propre cap θ^{S_i}, on calcul la différence :

$$\Delta cap = |(\theta^T - \theta^{S_i}|$$

La commande latérale de chaque véhicule est donnée par l'équation (3.40). Le terme DL_i représente la distance latérale de chaque véhicule par rapport à la trajectoire de référence et k_2 est un facteur de proportionnalité permettant de converger plus ou moins rapidement vers la trajectoire de référence indiquée par le véhicule de tête.

$$\theta_{S_i} = \theta_{S_{i-1}} \pm (\Delta cap + k_2.DL_i) \tag{3.40}$$

Notons que l'équation (3.40) ne tient pas compte de l'angle de braquage des roues. Dans ce cas, les véhicules sont considérés (représentés) comme des robots à deux roues. Pour changer leur directions, ces robots tournent sur place pour atteindre le cap voulu.

3.5 Conclusion

Le problème de la localisation de véhicules est traité dans ce chapitre comme étant l'estimation de la position d'un robot étant données les mesures délivrées par ses capteurs. Ces mesures sont imparfaites et entachées d'incertitude. La complémentarité et la redondance des informations sont alors deux facteurs essentiels permettant d'accéder à une information globale plus fiable et plus complète. Ainsi, la fusion de données est utilisée dans le but d'exploiter au mieux les avantages de chacune des sources d'informations, tout en essayant de pallier leurs limitations individuelles.
Après avoir énuméré un certain nombre de méthodes de fusions de données, nous nous sommes orientés vers les réseaux bayésiens. L'utilisation de cet outil est justifiée par les points suivants. Les capteurs utilisés sont généralement imparfaits et fournissent des informations incertaines, l'utilisation de la théorie de Bayes pour la modélisation et la manipulation de cette incertitude est donc justifiée. Le réseau bayésien permet la fusion de plusieurs capteurs de manière la plus intuitive étant donnée sa représentation graphique. D'un autre côté, les réseaux bayésiens généralisent les filtres de Kalman.
Le point le plus important quant à l'utilisation des réseaux bayésiens, est que cet outil nous permet en particulier de gérer des hypothèses multiples décrites tout au long de ce chapitre. Cette approche basée sur les réseaux bayésiens, ne tient pas compte seulement du segment le plus probable comme c'est le cas dans plusieurs travaux basés sur les filtres de Kalman. À l'approche d'une intersection ou sur deux routes rapprochées, plusieurs segments peuvent être candidats, pour cette raison, nous gérons plusieurs hypothèses (segments) jusqu'à ce que la situation devienne non ambiguë.

Tout au long de la première partie de ce chapitre, les systèmes non linéaires sont linéarisés autour de l'estimée courante, afin d'appliquer les techniques du filtre de Kalman linéaire. A la fin de la première partie de ce chapitre, nous avons présenté une méthode basée sur les modèles chaînés pour contourner le problème de la non linéarité. La linéarisation exacte, consiste à rechercher une transformation exacte permettant de réécrire le système non linéaire comme un système linéaire. Le système chaîné de départ à n états et 2 entrées est un système non linéaire, donc non transformable en un système linéaire avec les deux entrées u_1 et u_2. Cependant, si nous choisissons la commande u_1 comme une fonction du temps, éventuellement constante, ce n'est plus une variable de commande (puisque nous fixons sa loi horaire). Dans ce cas le système de départ (non linéaire) ne possède plus qu'une seule entrée de commande et les $n-1$ dernières lignes du système chaîné se ré-expriment bien comme un système linéaire.

Dans la seconde partie de ce chapitre, nous nous sommes posés cette question : pourquoi ne pas étendre le concept de localisation d'un véhicule pour localiser un train de véhicule ? Le train de véhicule est supposé suivre un chemin de référence donné par le véhicule de tête, tout en respectant un écart prédéfini entre les véhicules.

La localisation d'un train de véhicule peut être vue comme une extension de la localisation d'un mono véhicule. Le réseau bayésien utilisé dans la première partie pour la localisation d'un véhicule sur une carte est dupliqué pour l'ensemble du convoi. On a ajouté par la suite des inter-connexions pour finalement représenter le train de véhicules.
Le problème rencontré dans cette partie est comment transformer la fonction non linéaire donnée par le télémètre en fonction linéaire (dans le but de garder toutes les équations d'observations comme étant des équations linéaires) ? La flexibilité des réseaux bayésiens nous a permis de contourner ce problème par la construction d'une nouvelle observation qui est la fusion de la distance donnée par le télémètre et la position du véhicule précédant. La fusion des données du véhicule de tête (position) avec celui du télémètre permet un calcul plus précis de la position du véhicule suiveur S_1. Cette même position servira à estimer la position du second véhicule suiveur S_2 et ainsi de suite pour le reste des véhicules constituant le convoi. De cette manière, l'ellipse d'incertitude autour de l'estimée donnée par le système de localisation utilisé sur chaque véhicule se réduit.
Le chapitre suivant est dédié à la mise en œuvre et à la validation expérimentale des méthodes proposées dans ce chapitre.

Chapitre 4

Résultats et experiences

4.1 Introduction

Ce chapitre présente les tests effectués et les résultats obtenus afin de valider notre approche. Dans la première partie nous présentons les expériences et les résultats pour la localisation d'un véhicule sur une carte. La seconde partie concerne la modélisation et la localisation d'un convoi de véhicule. La dernière partie présente les modèles chaînés.

4.2 Résultats expérimentaux de la localisation d'un véhicule sur une carte

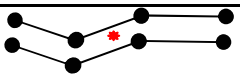
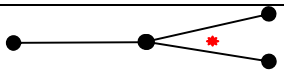
La première partie de ce chapitre, concerne la localisation d'un véhicule sur une carte ou map-matching. Dans cette expérience, nous utilisons les données réelles d'un essai effectué à Compiègne[26] en France avec un véhicule expérimental. Ce véhicule est équipé d'un récepteur GPS. Ce récepteur permet une localisation avec une correction différentielle, par l'utilisation d'un satellite géostationnaire lui envoyant des corrections. La précision annoncée par les constructeurs de ce capteur est métrique.
Avant de se lancer dans les expériences, nous commençons par donner une définition plus précise de ce qu'on appelle les situations d'ambiguïtés lors de la mise en correspondance d'une estimation sur un segment de route.

4.2.1 Situations d'ambiguïtés

Dans la plupart des applications qui nécessitent une localisation sur une carte routière, le cas le plus simple (idéal) pour n'importe quel algorithme de localisation est d'avoir un seul segment. Tout au long de cette thèse nous avons vu que les cartes présentent de plus en plus de détails, et par conséquent, le nombre des segments candidats à traiter est de plus en plus important.
Dans le cas où la méthode de sélection de segment fournit plusieurs segments candidats, il est intéressant de vérifier si ces segments font partie de la même route car dans ce cas, il n'y a pas vraiment d'ambiguïté. Ainsi, nous devons définir les situations d'ambiguïté qui se produisent dans la mise en correspondance d'une estimation sur une carte.

[26] Données mises à disposition par l'équipe véhicule intelligent de l'Heudiasyc de l'Université de Technologie de Compiègne.

Le tableau 4.1 présente les différentes situations recensées. Les trois premières lignes de ce tableau ne correspondent pas à une situation d'ambiguïté. Même en présence de plusieurs segments, chaque segment correspond à une partie de la même route.
A la première ligne du tableau 4.1, le champ "Pas de segment" correspond à une route qui n'est pas encore représentée sur la carte ou bien à une zone qui n'est pas affectée à la circulation (parking, espaces verts,...). Notons que tout au long de ces expériences, nous supposons que le véhicule roule uniquement sur des zones de circulation.
Les situations d'ambiguïté les plus fréquentes sont celles données par les deux dernières lignes du tableau 4.1. Les arcs parallèles et les jonctions de route sont plus fréquents dans les ronds-points et les agglomérations urbaines. Ces deux cas ont une influence directe sur l'ensemble des méthodes et algorithmes de localisation. Notons que ces deux cas d'ambiguïté sont les seuls cas étudiés dans nos expériences.

Pas de segment		Non ambiguë
Un segment		Non ambiguë
Plusieurs segments de la même route		Non ambiguë
Routes parallèles		Ambiguë
Jonction de route		Ambiguë

TAB. 4.1 – Description des situations “ambiguës” et “non ambiguës” dans la mise en correspondance d'une estimation sur une carte routière ou ce qu'on appel map-matching.

4.2.2 Expériences

Le parcours choisi pour tester la validité de notre approche est donné figure 4.1. Ce parcours est choisi en raison de la présence de deux routes parallèles et deux carrefours qui font l'objet de sources d'ambiguïtés (cas idéaux d'ambiguïtés).
Sur la figure 4.2, nous avons tracé les estimations données par le modèle odomètrique. L'inconvénient majeur de l'odométrie comme en l'a vu tout au long du chapitre 1, est sa dérive au cours du temps dû au glissement des roues, à l'imperfection des routes (trous, bosses)... Cette dérive est bien représentée sur cette figure. Pour faire face à cette dérive, on a souvent recours à un autre capteur (typiquement le GPS) pour recaler l'odométrie.

4.2.2.1 Expérience 1 : localisation sans utilisation du GPS

Dans la première expérience (voir figure 4.3), nous avons utilisé les données GPS au début de l'essai (sur le premier carrefour) puis nous avons simulé un masquage pour le reste de l'expérience. En l'absence d'information donnée par le GPS, l'algorithme de localisation devient un algorithme utilisant uniquement l'odométrie et la cartographie. Nous remarquons, que malgré l'absence des données GPS, les estimées données par le réseau bayésien sont plus proches des estimées GPS sauf sur une petite zone due à l'accumulation des erreurs odométriques. L'algorithme se rattrape directement après quelques itérations. Notons que sur cette figure, nous avons donné l'estimation la plus probable pour ne pas encombrer la figure et rendre ce schéma lisible. Les figures 4.4 et 4.6 donnent un aperçu plus détaillé sur des portions de route.

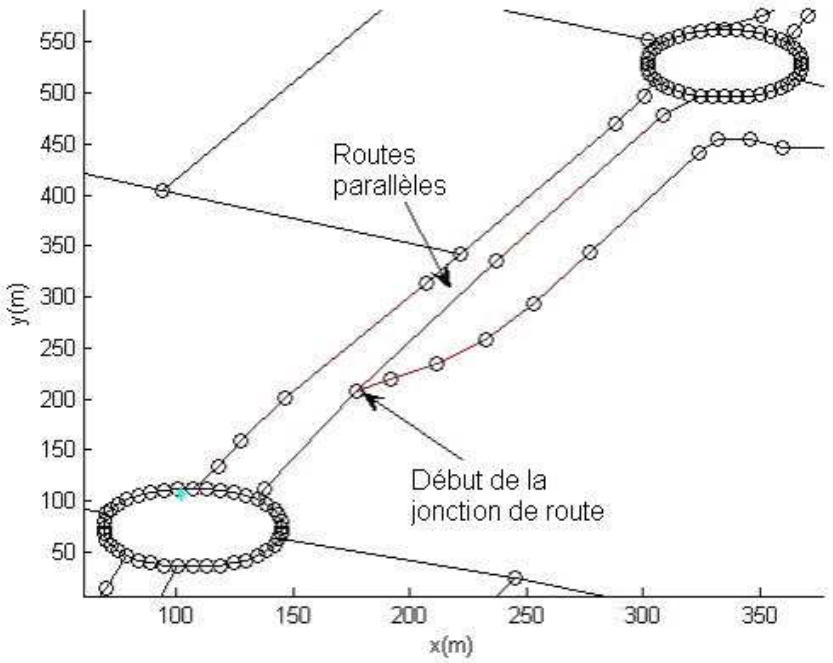


FIG. 4.1 – Vue globale du parcours d'essai. Ce parcours est choisi volontairement pour traiter l'ambiguïté fréquemment rencontrée dans le cas d'une jonction de routes et routes parallèles.

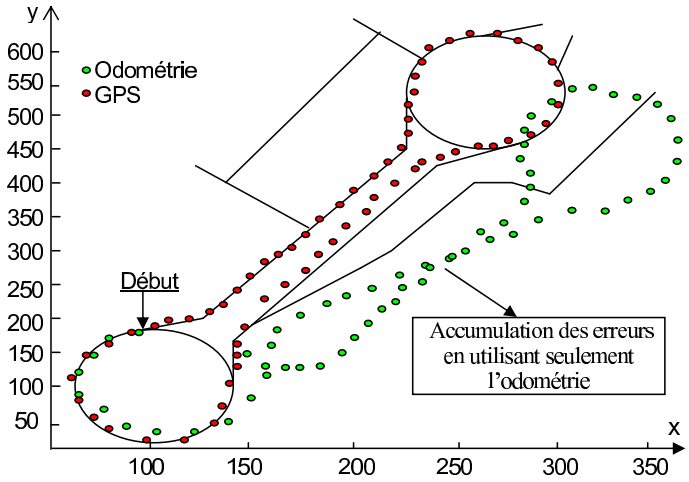


FIG. 4.2 – Ce schéma montre l'accumulation des erreurs dues à l'utilisation de l'odométrie.

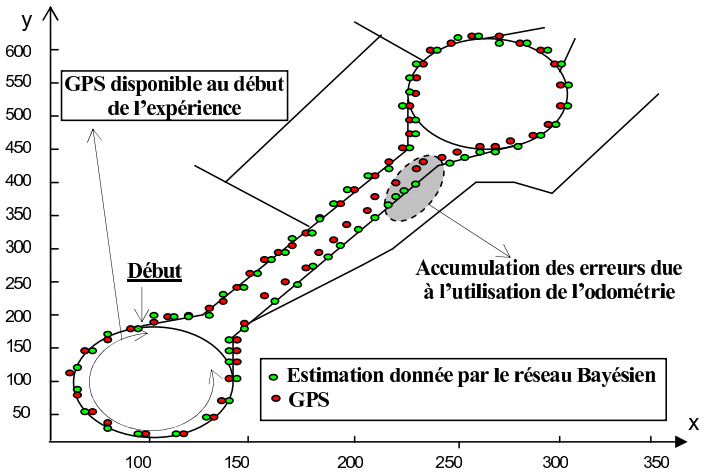


FIG. 4.3 – Utilisation des réseaux bayésiens pour l'estimation des positions du véhicule. La zone entourée montre bien l'accumulation des erreurs due à l'utilisation de l'odométrie.

4.2.2.2 Expérience 2 : situation d'ambiguïté dans le cas de routes parallèles

Dans cette deuxième expérience, nous voulons montrer comment notre réseau bayésien gère les situations d'ambiguïtés. Pour simuler le masquage des données GPS dans les milieux urbains, nous masquons les données GPS par intermittence.
Du moment que les données GPS ne sont pas utilisées, l'incertitude autour de l'estimée donnée par l'odométrie augmente et par conséquent le Système d'Information Géographique (SIG) sélectionne les segments dans un rayon plus grand. De ce fait, un ou plusieurs cas d'ambiguïtés apparaissent.
Dans le cas de la figure 4.4, on est en présence de deux routes parallèles, ce qui est un cas typique d'ambiguïté. Le réseau bayésien gère plusieurs segments sur plusieurs pas de temps jusqu'à la présence des données plus précises du GPS qui rendent l'incertitude autour de l'estimée minime. Notons que dans toutes nos expériences, n'a pas été tenu compte du sens de circulation.
La figure 4.5 donne plus de détails sur la zone d'ambiguïté de la figure 4.4. Le segment le plus probable dans chaque étape est représenté en pointillés. La distribution de probabilités de chaque segment est donnée par le tableau 4.2.

i	Segments	$P(Seg_i)$
1	Seg_1, Seg_2	$P(Seg_1) = 0.20, P(Seg_2) = 0.79$
2	Seg_1, Seg_2	$P(Seg_1) = 0.20, P(Seg_2) = 0.79$
3	Seg_1, Seg_2, Seg_3	$P(Seg_1) = 0.10, P(Seg_2) = 0.71, P(Seg_3) = 0.18$
4	$Seg_1, Seg_2, Seg_3, Seg_4$	$P(Seg_1) = 0.06, P(Seg_2) = 0.64, P(Seg_3) = 0.15, P(Seg_4) = 0.14$

TAB. 4.2 – Détail du processus d'inférence donné par le réseau bayésien pour chaque segment de route : routes parallèles (voir le schéma de la figure 4.5)

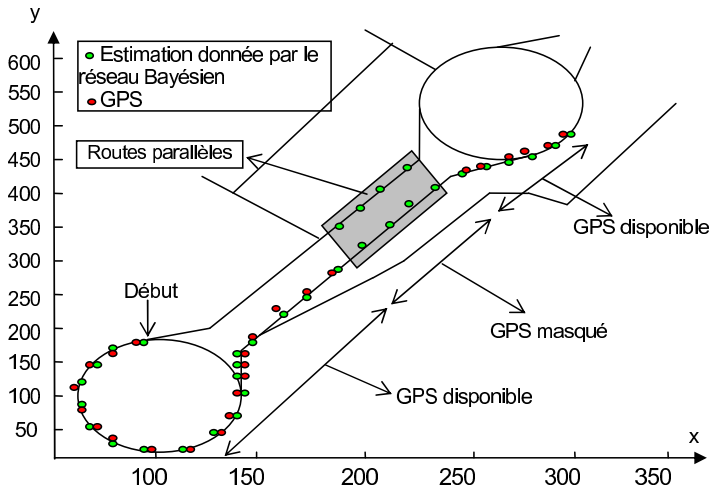


FIG. 4.4 – Gestion de l'ambiguïté dans le cas d'une route parallèle traité par le réseau bayésien.

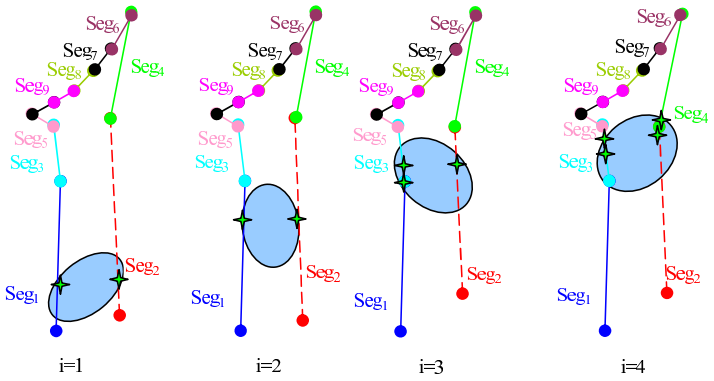


FIG. 4.5 – Détail sur la zone : route parallèle de la figure 4.4. Le segment en pointillé représente le segment le plus probable.

4.2.2.3 Expérience 3 : situation d'ambiguïté dans le cas d'une jonction de route

Dans cette troisième expérience, nous voulons montrer comment notre réseau bayésien gère les situations d'ambiguïtés dans le cas d'une jonction de route souvent rencontrée à la sortie d'un carrefour.
Pour simuler le masquage des données du GPS, nous cessons d'utiliser ces données juste avant la jonction de route (voir la figure 4.6). En absence d'information précise, l'algorithme de localisation n'utilise plus que l'odométrie et la cartographie uniquement. L'incertitude autour de l'estimée augmente, ce qui contraint le réseau bayésien à gérer plusieurs segments sur plusieurs pas de temps jusqu'à la présence des données du GPS qui rendent l'incertitude autour de l'estimée minime et par conséquent le nombre de segments candidats diminue.
La figure 4.7 donne plus de détail sur la zone d'ambiguïté de la figure 4.6. D'après le tableau 4.2, on peut remarquer parmi les différentes hypothèses (segments candidats), le réseau bayésien accorde une plus grande probabilité au bon segment.

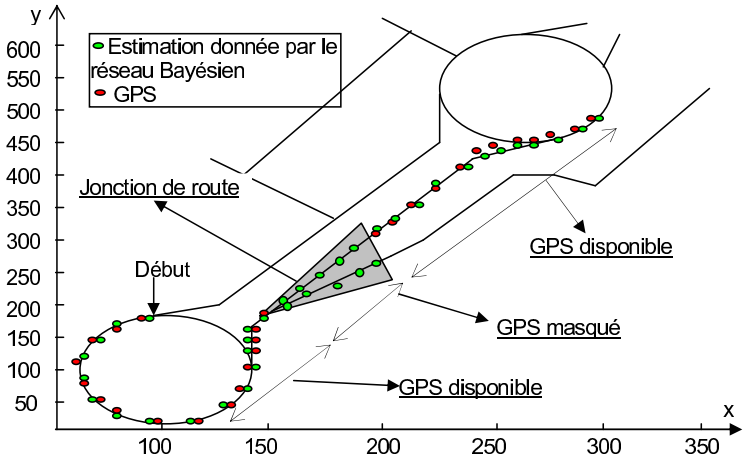


FIG. 4.6 – Gestion de l'ambiguïté dans le cas d'une jonction de route traité par le réseau bayésien.

i	Segments	$P(Seg_i)$
1	$Seg_1, Seg_2, Seg_3, Seg_6$	$P(Seg_1) = 0.08, P(Seg_2) = 0.13, P(Seg_3) = 0.14, P(Seg_6) = 0.65$
2	Seg_2, Seg_3, Seg_6	$P(Seg_2) = 0.07, P(Seg_3) = 0.18, P(Seg_6) = 0.74$
3	Seg_3, Seg_4, Seg_6	$P(Seg_3) = 0.13, P(Seg_4) = 0.15, P(Seg_6) = 0.72$
4	Seg_4, Seg_5, Seg_6	$P(Seg_4) = 0.2, P(Seg_5) = 0.11, P(Seg_6) = 0.69$
5	Seg_5, Seg_6	$P(Seg_5) = 0.28, P(Seg_6) = 0.71$

TAB. 4.3 – Détail du processus d'inférence donné par le réseau bayésien pour chaque segment de route : jonction de route (voir le schéma de la figure 4.7).

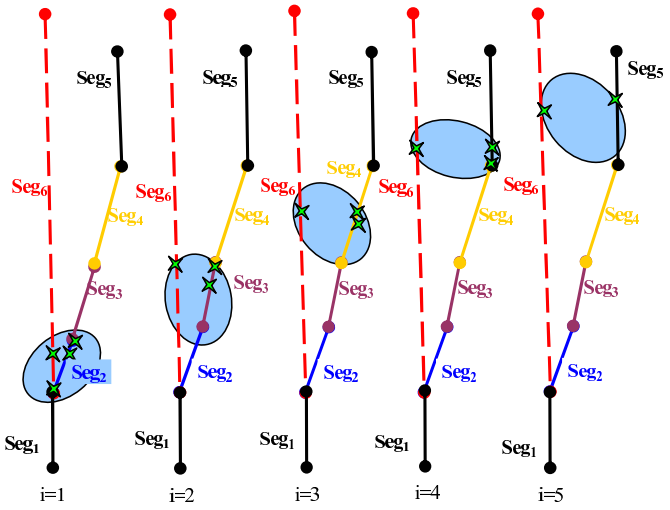


FIG. 4.7 – Détail sur la zone : jonction de route de la figure 4.6. Le segment en pointillé représente le segment le plus probable.

4.3 Résultats expérimentaux de la localisation d'un convoi de véhicule

Cette deuxième partie d'expériences concerne la localisation d'un convoi de véhicules. Dans cette partie, nous utilisons les données réelles d'un essai effectué sur la place Stanislas de Nancy. Avant de présenter les expériences, nous commençons par donner une description détaillée sur le véhicule de tête utilisé.

4.3.1 Description des véhicules

L'Institut National de Recherche en Informatique et en Automatique (INRIA) est a l'origine de la conception de petits véhicules électriques nommés Cycab (voir la figure 4.8). Les Cycab constituent une nouvelle plate-forme de recherche sur les véhicules intelligents. L'utilisation de ce type de véhicule a pour objectif d'offrir une alternative à la voiture d'aujourd'hui. Ces véhicules sont électriques (non polluants), pratiques et doués d'une certaine autonomie de décision.
Ces véhicules sont spécialement conçus pour des zones de forte circulation : centre ville, gare,.... En effet, leurs faibles dimensions (longueur 1,90 m, largeur 1,20 m, poids 300 kg) sont des avantages dans de tels environnements. Le Cycab peut transporter deux personnes à une vitesse théorique maximale de 18 Km/h. Il est équipé d'un moteur électrique de $1Kw$ sur chacune de ces quatre roues motrices. Ces 4 moteurs sont alimentés par 8 batteries permettant une autonomie de 2 heures en plein régime. Ce type de véhicule peut être conduit en mode automatique ou en mode manuel à l'aide de joystick.
Le Cycab est équipé d'un ordinateur dont la fonction est de rassembler l'information reçue par les différents capteurs utilisés et générer les commandes appropriées aux types d'applications. Les commandes sont ensuite transmises à l'ordinateur embarqué dans le Cycab et au moyen d'un bus CAN vers les microcontrôleurs agissant sur les actionneurs du robot. Le même processus peut se faire dans l'autre sens où les odomètres remontent les vitesses de chacune des 4 roues.

FIG. 4.8 – Le Cycab constitue une nouvelle plate-forme de recherche sur les véhicules intelligents.

4.3.2 Le récepteur GPS

Le récepteur GPS utilisé est le GPS Sagitta02 de Thales Navigation (voir la figure 4.9). Le récepteur Sagitta 02 est un récepteur bi fréquence L1/L2. Il est connecté au PC par liaison série RS232. Il a une cadence de 10Hz pour les données brutes et de 20Hz pour les données calculées. Il dispose de plusieurs modes : WAAS/EGNOS, DGPS, RTK,... Le mode RTK (Real Time Kinematic) dont il dispose, lui permet de fournir un positionnement centimétrique, en temps réel avec une couverture radio jusqu'à 40 kilomètres. Dans ce mode, une base fixe est utilisée, elle envoie ses corrections au récepteur mobile par UHF [Cindy, 2008].

FIG. 4.9 – GPS Thales Sagitta02.

4.3.3 Trajectoires des véhicules suiveurs

Le véhicule de tête est conduit manuellement sur le parcours donné figure 4.10. La courbe verte représente la trajectoire acquise par le récepteur GPS Sagitta 02 en mode RTK. Les véhicules suiveurs sont ici simulés et asservis avec les lois de commandes simples décrites dans le chapitre 3 (voir la page 102).
Dans cette expérience, nous supposons que tous les véhicules suiveurs sont initialement sur la trajectoire de référence et n'ont pas nécessairement le même cap que le véhicule de tête. La vitesse réelle du véhicule de tête est donnée figure 4.11.
Chaque véhicule suiveur dispose par WiFi et en temps réel la trajectoire du véhicule de tête et la vitesse du véhicule qui précède. On suppose que le réseau de communication est parfait (sans rupture de communication et avec un temps de latence nul) et les détections télémétriques sont instantanées.

La figure 4.12 représente les positions temporelles du convoi données par le réseau bayésien. Les véhicules suiveurs déterminent leurs positions en utilisant d'une part un GPS (avec une incertitude de 10 mètres) et d'autre part le télémètre (avec une incertitude de 15 centimètres) et la position du véhicule prédécesseur. On peut remarquer que les véhicules suiveurs reproduisent la trajectoire du véhicule de tête. Les figures 4.13, 4.14 et 4.15 donnent respectivement les trajectoires de suiveur 1, suiveur 2 et suiveur 3 camparées à la trajectoire de référence donnée par le GPS RTK.

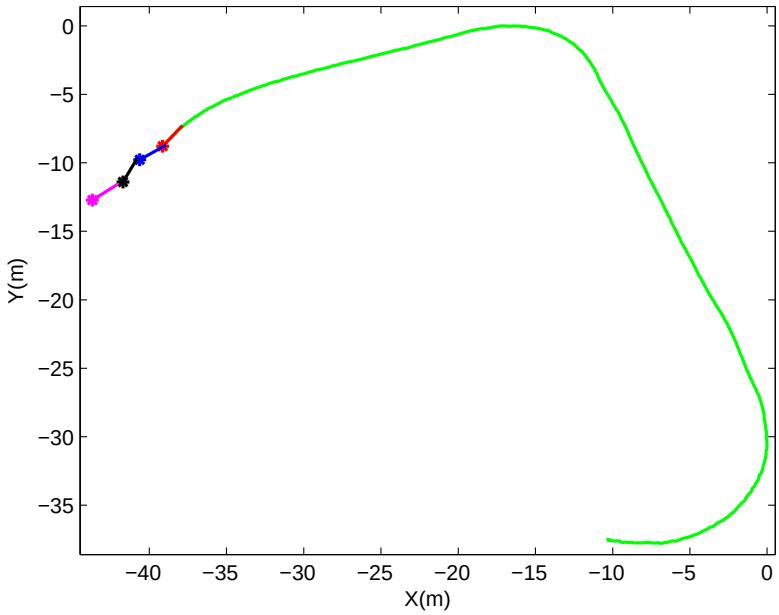


FIG. 4.10 – Trajectoire du véhicule de tête donnée par le récepteur GPS Sagitta 02 en mode RTK et les positions initiales des véhicules suiveurs.

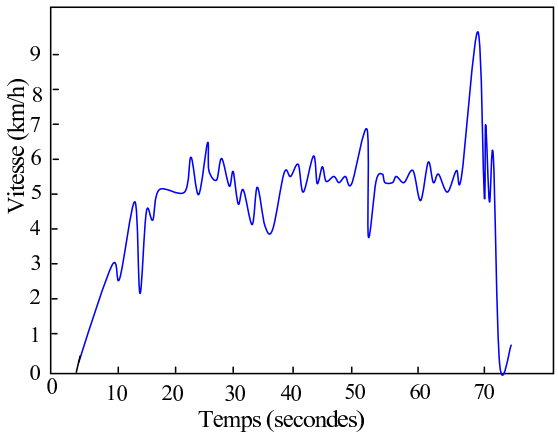


FIG. 4.11 – Vitesse du Cycab (véhicule de tête) pendant le test.

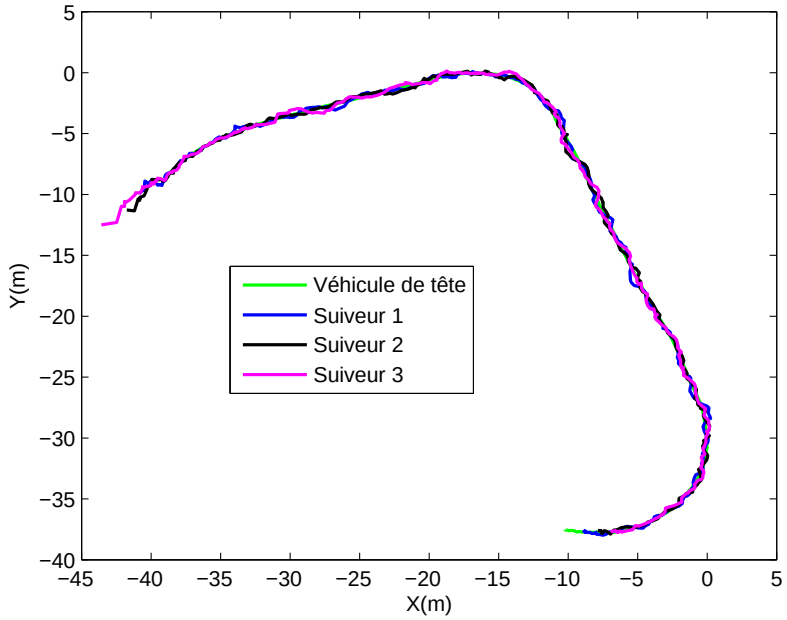


FIG. 4.12 – Trajectoires des véhicules suiveurs données par le réseau bayésien.

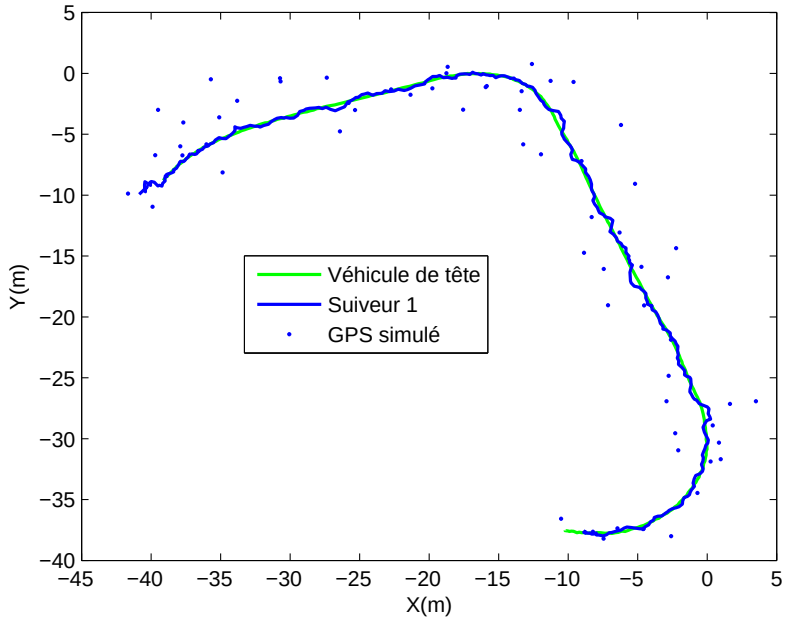


FIG. 4.13 – En bleu la trajectoire du véhicule suiveur 1 donnée par le réseau bayésien comparée à celle du véhicule de tête (en vert).

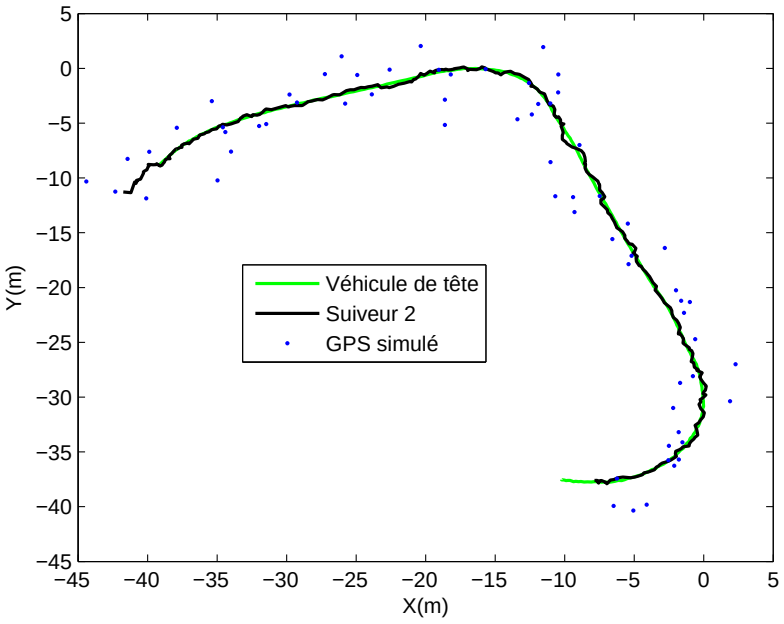


FIG. 4.14 – En noir la trajectoire du véhicule suiveur 2 donnée par le réseau bayésien comparée à celle du véhicule de tête (en vert).

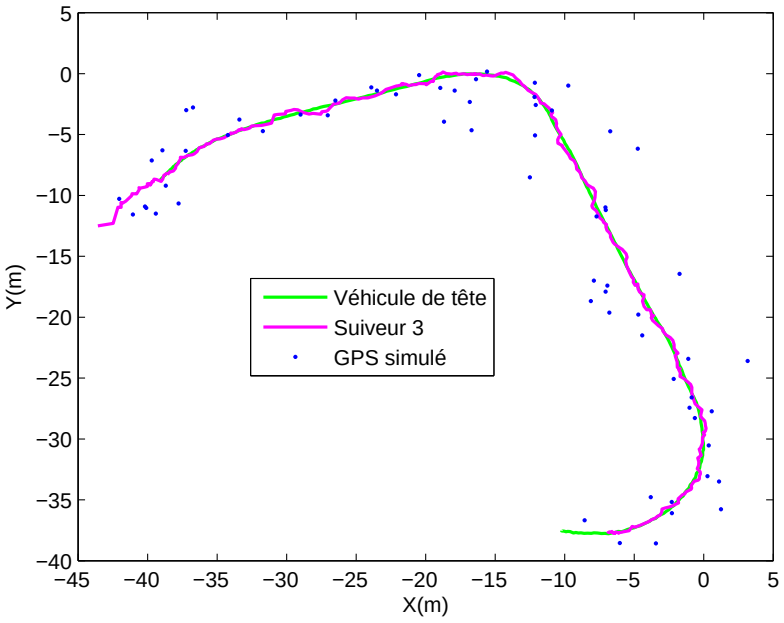


FIG. 4.15 – En magenta la trajectoire du véhicule suiveur 3 donnée par le réseau bayésien comparée à celle du véhicule de tête (en vert).

4.3.4 Inter-distance entre les véhicules

Le deuxième point à prendre en compte dans la formation d'un convoi de véhicules est le fait d'assurer le contrôle en vitesse des membres d'un convoi de façon à respecter une consigne donnée sur l'écart entre les véhicules. Cette consigne peut être exprimée soit comme inter-distance, soit comme un temps relatif entre les véhicules ou finalement comme une consigne hybride mêlant une inter-distance et un temps relatif.
Du moment que les vitesses des véhicules utilisés (Cycab) n'excèdent pas les $18km/h$, une inter-distance est préférée. Dans nos expériences, nous avons choisi de maintenir une distance de sécurité de $1.5m$.
Les figures 4.16, 4.17 et 4.18 donnent en fonction du temps l'inter-distance entre les véhicules. Au départ, cette inter-distance est choisie arbitrairement de tel sorte à prendre en compte tous les cas possibles (maintenir la même vitesse, augmenter la vitesse et réduire la vitesse). L'inter-distance aperçue (représentée en rouge) par chaque véhicule suiveur est comparée à l'interdistance réelle (représentée en vert). On peut remarquer que cette distance de sécurité est généralement respectée.
Notons que la vitesse réelle du véhicule de tête donnée figure 4.11 présente plusieurs oscillations (en aucun cas cette vitesse est constante), ce qui influence directement les courbes de distances données par les figures 4.16, 4.17 et 4.18. D'autre part, la commande longitudinale utilisée (proportionnelle) est rudimentaire.

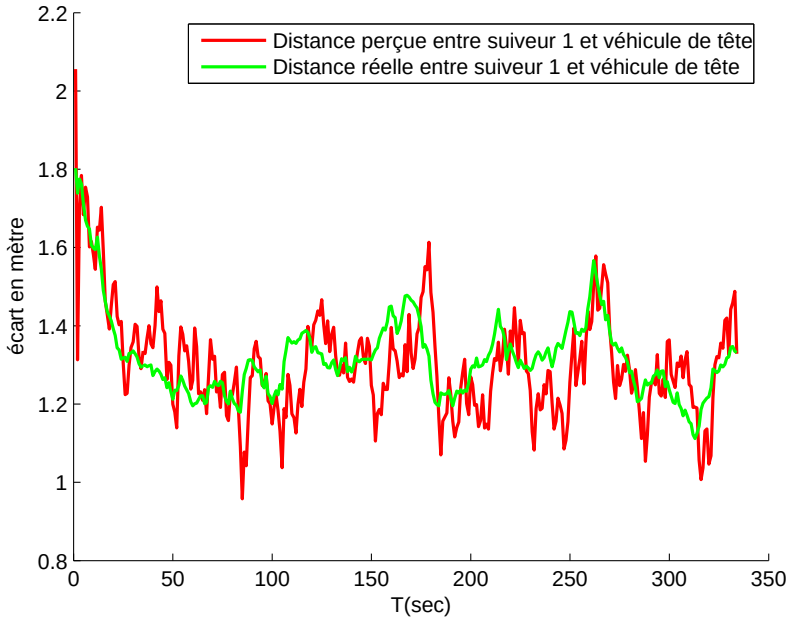


FIG. 4.16 – En rouge l'inter-distance (entre suiveur 1 et le véhicule de tête) aperçue par le suiveur 1. En vert l'inter-distance réelle entre le suiveur 1 et le véhicule de tête.

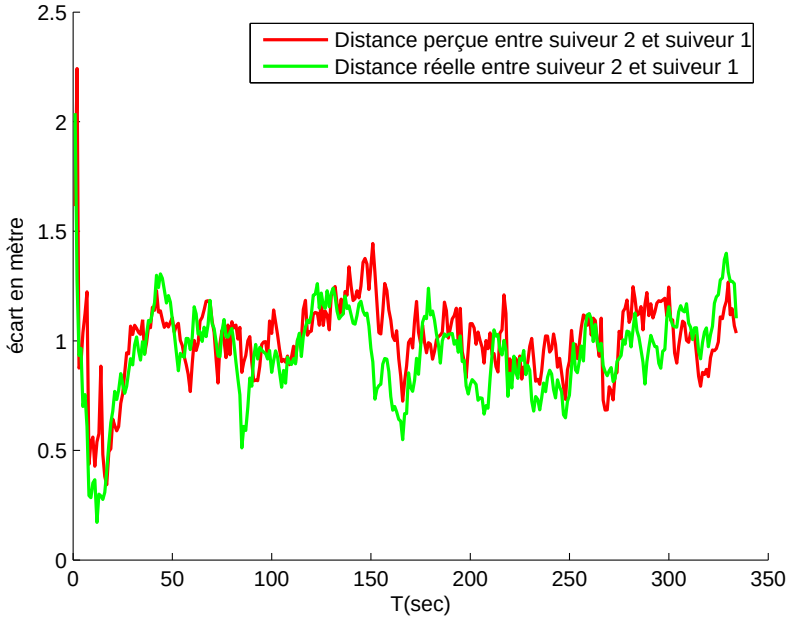


FIG. 4.17 – En rouge l'inter-distance (entre suiveur 2 et suiveur 1) aperçue par le suiveur 2. En vert l'inter-distance réelle entre le suiveur 2 et le suiveur 1.

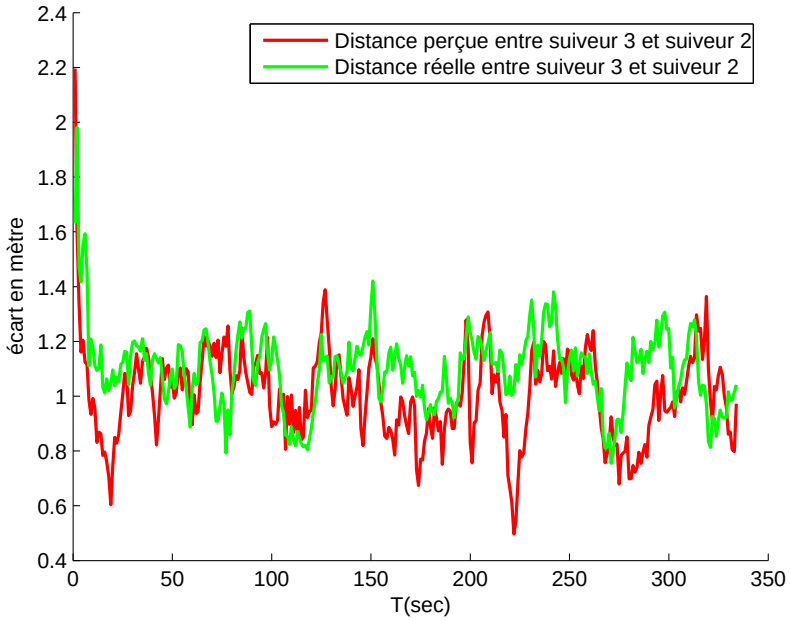


FIG. 4.18 – En rouge l'inter-distance (entre suiveur 3 et suiveur 2) aperçue par le suiveur 3. En vert l'inter-distance réelle entre le suiveur 3 et le suiveur 2.

4.4 Transformation du modèle cinématique d'un véhicule en modèles chaînés

Cette troisième et dernière partie de ce chapitre traite le problème de la non linéarité. Nous proposons d'utiliser les modèles chaînés afin de trouver une transformation exacte du système non linéaire décrivant la cinématique d'un véhicule. Cette transformation nous permet d'éviter les méthodes approximatives. Le filtre de Kalman étendu est principalement utilisé en robotique car la totalité des modèles cinématique sont non linéaires.
Nous proposons dans cette partie d'expériences de comparer et d'évaluer les résultats d'un filtre de Kalman étendu et d'un réseau bayésien décrivant la cinématique d'un véhicule écrite sous forme d'un modèle chaîné.

4.4.1 Localisation d'un véhicule en utilisant le modèle chaîné

Pour évaluer les performances de la méthode proposée (modèle chaîné), nous proposons de comparer notre méthode basée sur les réseaux bayésiens (dont les équations cinématiques sont écrites sous forme de modèle chaîné) par rapport à un filtre de Kalman étendu classique. Rappelons que le modèle cinématique d'une roue s'écrit (voir chapitre 3) :

$$\dot{q} = \begin{cases} \dot{x} = \frac{r}{2}(\omega_R + \omega_L)\cos\theta \\ \dot{y} = \frac{r}{2}(\omega_R + \omega_L)\sin\theta \\ \dot{\theta} = \frac{r}{2e}(\omega_R - \omega_L) \end{cases} \tag{4.1}$$

Rappelons également que le modèle chaîné équivalent au modèle donné par l'équation 4.1 est donné par (voir le chapitre 3) :

$$a^{k+1} = A_k a^k + b^k + w_k \tag{4.2}$$

où w_k représente le bruit du modèle, $w_k \sim N(0, Q_k)$ et la matrice A_k s'écrit :

$$A_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(Tv_2^k) & \sin(Tv_2^k) \\ 0 & -\sin(Tv_2^k) & \cos(Tv_2^k) \end{pmatrix} \tag{4.3}$$

avec le nouveau vecteur d'entrée :

$$b^k = \begin{pmatrix} -Tv_2^k \\ v_1^k \frac{\sin(Tv_2^k)}{v_2^k} \\ v_1^k \frac{\cos(Tv_2^k)-1}{v_2^k} \end{pmatrix} \tag{4.4}$$

Nous utilisons les données réelles d'un essai effectué sur la place Stanislas de Nancy (voir figure 4.10). Le Cycab utilisé dans cette expérience est équipé d'un récepteur GPS Sagitta02 de Thales Navigation. Le mode RTK (Real Time Kinematic) dont il dispose lui permet de fournir un positionnement centimétrique, en temps réel. La trajectoire donnée par le GPS-RTK est utilisée comme référence afin de comparer les résultats du réseau bayésien et du filtre de Kalman. Cette trajectoire est de longueur de 150 mètres et la durée de l'expérience est d'environ 2 minutes. La vitesse du véhicule est donnée figure 4.11. Le temps d'échantillonnage est de une seconde ($T = 1$).

Le deuxième capteur utilisé dans cette expérience est un gyroscope. Ce capteur donne au Cycab une estimation plus précise de son cap. Finalement, les codeurs incrémentaux utilisés sur les roues arrières du Cycab (ω_L et ω_R), servent au calcul des nouvelles entrées du système chaîné :

$$\begin{cases} u_1 = -v_2 \\ u_2 = v_1 + a_3 v_2 \end{cases} \tag{4.5}$$

Notons que les observations $GPS : (x_{gps}, y_{gps})$ et $Gyro : \theta_{gyro}$ doivent être converties dans le nouveau repère (a_1, a_2) lié au véhicule en utilisant le changement de variable donné par l'équation 4.6.

$$\begin{cases} a_1 = -\theta \\ a_2 = x\cos\theta + y\sin\theta \\ a_3 = -x\sin\theta + y\cos\theta \end{cases} \tag{4.6}$$

Cette conversion nous permet d'écrire la nouvelle équation d'observation comme :

$$\widehat{Y}_{k+1} = \begin{pmatrix} \widehat{\theta}_{gyro} \\ \widehat{x}_{gps} \\ \widehat{y}_{gps} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1^k \\ a_2^k \\ a_3^k \end{pmatrix} + d_k \tag{4.7}$$

où $d_k \sim N(0, R_k)$ est un autre bruit gaussien supposé indépendant de w_k. Notons que Q_k et R_k sont des matrices diagonales de dimensions $(3,3)$ fixées tout au long de l'expérience par : $Q_k = diag\{0.01, 0.01, 0.01\}$ et $R_k = \{0.1, 0.1, 0.1\}$.

4.4.2 Représentation d'un filtre de Kalman par un réseau bayésien

Le filtre de Kalman étendu représentant la cinématique du véhicule donnée par l'équation 4.1 (après discrétisation) peut être représenté par le réseau bayésien dynamique donné par la figure 4.19. Ce réseau est complètement constitué de variables continues[27]. La seule variable cachée représentant l'état du véhicule (x, y, θ) est représentée par X_k. Cette variable est mise à jour par les observations *GPS* et *Gyroscope* et par l'entrée du filtre (les codeurs incrémentaux utilisés sur les roues arrières) représentée par la variable U.

Le réseau bayésien représentant la cinématique du véhicule sous forme chaînée (équation 4.2) est donné par la figure 4.20. La variable cachée représentant l'état du véhicule (a_1^k, a_2^k, a_3^k) est représentée par a^k. Rappelons que les nouvelles variables a_2^k et a_3^k sont tout simplement les coordonnées cartésiennes (x, y) du véhicule évalués dans le repère attaché au véhicule et a_1^k est l'angle θ formé entre la direction du véhicule et l'axe des abscisses. La variable d'état a^k est mise à jour par les observations *GPS* et *Gyroscope* (converties dans le nouveau repère (a_1, a_2)) et par la nouvelle entrée du filtre représentée par la variable b^k.

[27] Nous avons représenté le GPS et le gyroscope par deux nœuds distincts juste pour montrer qu'on utilise deux capteurs. Dans la réalité et comme le montre l'équation d'observation 4.7, ces deux capteurs sont représentés par la même équation et par conséquent par un seul nœud.

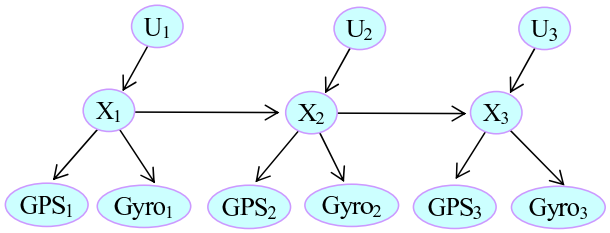


FIG. 4.19 – Représentation de la cinématique du véhicule par un réseau bayésien dynamique.

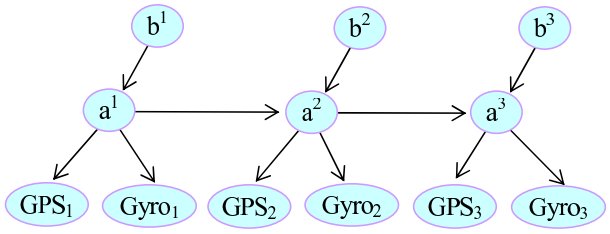


FIG. 4.20 – Représentation sous forme chaînée la cinématique du véhicule par un réseau bayésien dynamique.

4.4.3 Comparaison entre un filtre de Kalman étendu et le modèle chainé

Dans le but de mettre en valeur le modèle chaîné, nous avons tracé les deux trajectoires données par le filtre de Kalman étendu (représenté par la figure 4.19) et le réseau bayésien sous forme chaînée (représenté par la figure 4.20). Les deux trajectoires sont représentées respectivement en rouge et en bleue sur la figure **??**. La trajectoire réelle donnée par le GPS-RTK (centimétrique) qui sert de trajectoire de référence est représentée en vert. Sur cette même figure, nous avons bruité le GPS-RTK de façon à avoir un GPS métrique ressemblant à celui utilisé dans nos voitures. Ces mesures sont représentées sur la figure **??** par des points noirs. La position du véhicule (x, y, θ) est estimée dans le cas du filtre de Kalman étendu et dans le réseau bayésien sous forme chaînée par la fusion des mesures GPS (bruité), les données fournies par les codeurs incrémentaux et le gyroscope.
On remarque que la trajectoire en bleue donnée par le modèle chaîné, se rapproche le mieux de la trajectoire centimétrique donnée par le GPS-RTK (en verte). Cette trajectoire est plus lisse par rapport à la trajectoire donnée par le filtre de Kalman étendu (en rouge).
La trajectoire donnée par le filtre de Kalman étendu présente plusieurs points qui s'éloignent de la trajectoire (des pics) réelle en vert. Nous pensons que ces résultats étaient prévisibles car dans le cas où on linéarise le système autour de l'estimée courante, on ne tient pas compte des dérivées d'ordre supérieur à un (calcul de la Jacobienne), ce qui correspond à une approximation (voir le chapitre 1, la partie décrivant le filtre de Kalman étendu). Par contre, en utilisant le modèle chaîné, on a une transformation exacte du système, ce qui explique la non existence de ces points.

Nous avons calculé la distance entre chaque point estimé par les deux méthodes (filtre de Kalman étendu, réseau bayésien sous forme chaînée) et ceux donnés par le GPS-RTK. La figure **??**, montre bien que les points estimés par le modèle chaîné (courbe en bleue) sont les plus proches de la courbe réelle donnée par le GPS centimétrique. Ce résultat confirme la robustesse

de la transformée exacte donnée par les modèles chaînés par rapport à l'approximation donnée par le filtre de Kalman étendu.

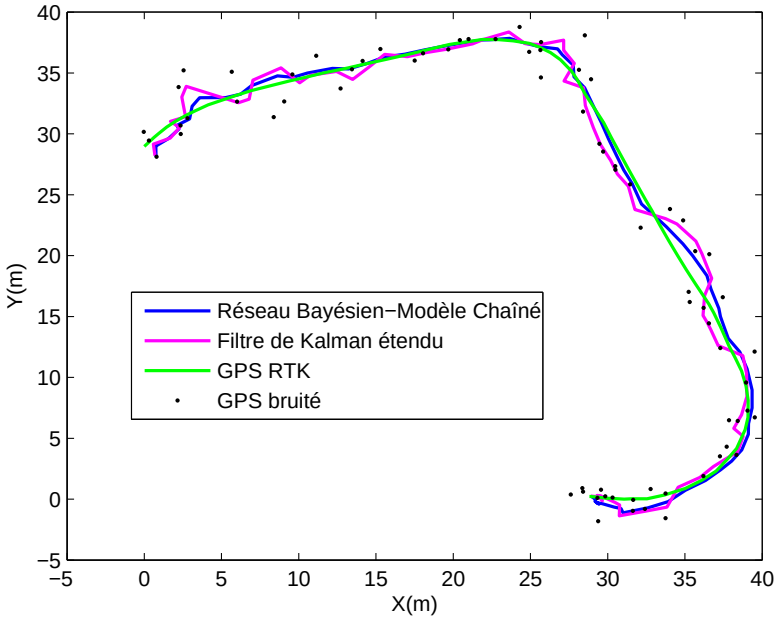


FIG. 4.21 – Trajectoires données par le filtre de Kalman étendu (représenté par un réseau bayésien) et le réseau bayésien sous forme chaînée.

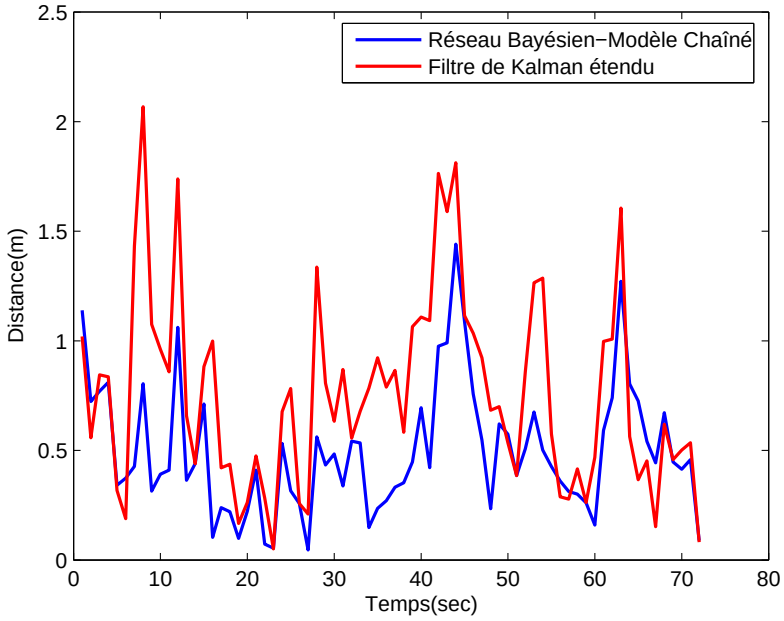


FIG. 4.22 – Distance entre chaque point donné par le GPS-RTK et les estimations données par le filtre de Kalman étendu, réseau bayésien écrit sous forme chaînée.

4.5 Conclusion

Dans ce chapitre, nous avons présenté une méthode basée sur les réseaux bayésiens dédiée à la localisation d'un véhicule sur une carte. Une des caractéristiques les plus intéressantes des réseaux bayésiens est la flexibilité et la facilité déconcertante avec laquelle on peut étendre un modèle. Cette caractéristique intéressante a été largement illustrée notamment lors de l'ajout de capteurs. Nous avons également montré qu'il était facile d'utiliser des informations hybrides (symboliques ou numériques). En particulier, l'ajout d'informations cartographiques ne posent aucun problème.
Il s'est avéré par les expériences menées et présentées dans ce chapitre que la fusion de l'odométrie et de l'observation cartographique peut donner une bonne estimation de la position du véhicule y compris lorsque le GPS est masqué pendant un court laps de temps.
La stratégie présentée par notre approche ne tient pas seulement compte du segment le plus probable. En se rapprochant d'une intersection par exemple ou dans un carrefour, plusieurs segments de routes peuvent être de bons candidats, par conséquent, notre approche gère plusieurs hypothèses (segments) jusqu'à la levée de l'ambiguïté. Cette stratégie est consistante avec l'incertitude des capteurs utilisés ce qui permet d'avoir confiance en cette méthode.

La deuxième partie de ce chapitre était consacrée à la modélisation et à la localisation d'un convoi de véhicules en utilisant les réseaux bayésiens. Le train de véhicules est vu dans ce cas comme étant une extension du réseau servant à la localisation d'un véhicule sur une carte (map-matching). Le réseau bayésien utilisé dans cette partie est dupliqué pour l'ensemble du convoi. En ajoutant des interconnexions entre les blocs du réseau, nous obtenons finalement une représentation du convoi de véhicules.
La fusion des données du véhicule précédant avec celui du télémètre permet à chaque véhicule

suiveur de réduire l'incertitude donnée par son système de localisation (GPS) ou même de se localiser sans utiliser ce système.
Rappelons que le problème d'un convoi de véhicules est complexe, car les commandes longitudinales et latérales nécessitent une bonne précision de la localisation du train de véhicules. Notons que nous nous sommes intéressé plutôt à la localisation qu'à la commande. Cependant, pour valider notre approche nous avons utilisé une simple commande dite proportionnelle.

Le problème de la non linéarité des systèmes a fait l'objet de la dernière partie de ce chapitre. La transformation du système non linéaire d'un véhicule sous forme chaînée permet de représenter l'équation d'un système unicycle[28] sous forme linéaire.
Le filtre de Kalman et sa version étendue sont très utilisés en robotique, principalement pour traiter le problème de la non linéarité[29]. Cependant, la linéarisation du système autour de l'estimée courante peut introduire une large erreur sur la moyenne et la covariance calculées a posteriori et peut même dans d'autres cas faire diverger le filtre.
Les performances des modèles chaînés présentés dans ce chapitre, montrent leurs capacités d'inférence exacte par opposition à l'inférence approximative. Ainsi, la fusion des données (odométriques, GPS et gyroscope) par le réseau bayésien dont les équations sont écrites sous forme chaînée donne de meilleurs résultats par rapport à la fusion donnée par un filtre de Kalman étendu. Cette méthode permet de fournir en continu une position plus exacte.

[28] Nous avons utilisé le modèle unicycle mais un modèle tricycle peut aussi être transformé sous forme chaîné.
[29] La totalité des équations représentant la cinématique des robots sont non linéaire

Conclusion et perspectives

Le travail présenté dans cette thèse se situe dans le cadre de la fusion de données appliquée au problème de la localisation. Il s'est appuyé principalement sur l'utilisation de modèles à base de réseaux bayésiens pour la localisation de véhicules sur une carte en milieu urbain.
Tout d'abord, nous avons présenté un état de l'art des méthodes existantes afin de situer notre approche. A cet effet, nous avons pu diviser le problème de la localisation d'un véhicule en deux grandes approches. Celle purement numérique, fusionne l'information délivrée par les capteurs proprioceptifs et extérioceptifs. La seconde, symbolique et numérique, complète les sources d'informations par une base de donnée cartographique.
Le problème de la localisation sur une carte est connu sous le terme map-matching ou road-matching et consiste à trouver le segment sur lequel le véhicule roule et la position de ce véhicule sur ce même segment.
La localisation sur une carte est un problème assez difficile vu les capteurs utilisés qui sont entachés d'erreurs. La solution envisagée est l'utilisation de la fusion multi-capteurs par un réseau bayésien. Cette fusion permet de fournir une information globale plus fiable et plus complète qui n'aurait pu être fournie par aucune des sources prises séparément.
L'utilisation de cartes de plus en plus précises augmente le nombre de segments de la route. Par conséquent, la sélection du segment le plus probable devient de plus en plus difficile. Pour cette raison, nous avons opté pour une approche plus sûre. Cette approche est basée sur la représentation et la manipulation de plusieurs segments à la fois. Ainsi, au lieu de confirmer et dire que le véhicule est sur un tel segment de route (ce qui est aberrant dans les situations d'ambiguïtés), on donne une confiance à chaque segments. Cette approche a été testée sur des données réelles sur un véhicule classique. Nous restons confiant sur les performances de cette approche.

La deuxième contribution de cette thèse concerne la modélisation et la localisation d'un convoi de véhicules. La représentation graphique donnée par un réseau bayésien est très importante. Cette étape demande une certaine abstraction et beaucoup d'imagination. La question que nous nous sommes posé après l'étape de modélisation et localisation du véhicule sur une carte est : pourquoi ne pas étendre ce même concept (modélisation et localisation) pour un train de véhicules ?
Ainsi, le train de véhicules est considéré dans ce cas comme étant une extension du réseau bayésien servant à la localisation d'un véhicule sur une carte. Le réseau bayésien utilisé dans cette partie est dupliqué pour chaque véhicule du convoi. Les interconnexions entre les blocs du réseau sont ajoutées afin de finaliser la modélisation graphique du convoi.
Afin de ne pas rendre l'infrastructure encore plus coûteuse, nous avons équipé seulement le véhicule de tête par un GPS centimétrique de type GPS-RTK (et éventuellement le dernier véhicule pour résoudre le problème de l'accumulation d'erreurs) et simuler un GPS métrique et un télémètre pour les autres véhicules.

La fusion de données (position ou vitesse du véhicule précédent avec celui du télémètre) permet à chaque véhicule suiveur de réduire l'incertitude donnée par son système de localisation (GPS) ou même de se localiser sans utiliser ce système. Notons que dans cette partie, nous nous sommes intéressé principalement à la modélisation et la localisation du convoi par opposition à la conception de commandes. Pour valider notre approche, nous avons utilisé de simples commandes dites proportionnelles.

La troisième contribution de cette thèse concerne la représentation de la cinématique d'un véhicule sous forme chaînée. L'utilisation de l'inférence exacte dans les réseaux bayésiens ou le filtre de Kalman suppose un modèle de bruit Gaussien additif et la linéarité du modèle. La mise à jour d'une distribution Gaussienne par la règle de Bayes, donne une distribution a posteriori qui est toujours Gaussienne. Cependant, l'inférence exacte dans les réseaux bayésiens dont la distribution de probabilités conditionnelles des variables cachées n'est pas linéaire Gaussienne, n'est pas toujours possible. De ce fait, on peut procéder à une linéarisation autour de l'estimée courante (filtre de Kalman étendu), ou bien on peut appliquer une méthode complètement approximative telle que les filtres particulaires.
Par opposition à la linéarisation autour de l'estimé courante (voir filtre de Kalman étendu), l'approche par linéarisation exacte cherche une transformation exacte (une transformation d'état et de commande inversible) permettant de réécrire le système non linéaire comme un système linéaire, de façon a pouvoir exploiter l'ensemble des outils de l'automatique linéaire. Les performances du modèle chaîné présenté dans ce chapitre sur des données réelles, montre bien la supériorité de cette approche. La fusion des données (odométriques, GPS et gyroscope) par le réseau bayésien dont les équations sont écrites sous forme chaînée donne de meilleurs résultats par rapport à la fusion donnée par un filtre de Kalman étendu. Cette méthode permet donc de fournir en continu une position plus précise.

Comme perspectives de ces travaux, nous envisageons d'appliquer la transformation en modèle chaîné pour la localisation d'un véhicule sur une carte. De la même façon, le train de véhicules sera modélisé par un réseau bayésien dont les équations non linéaires seront transformées en modèles chaînés. Ces modèles nous ouvrent une nouvelle branche de recherche où la confiance donnée pour une estimée sera plus fiable, robuste et précise. Certes, les résultats seront meilleurs, et ils seront comparés aux résultats donnés dans cette thèse.

La seconde perspective concerne la comparaison du filtre de Kalman classique sous forme chaînée par rapport à un filtre de Kalman non parfumé et le filtre particulaire. Connaître avec précision les avantages et les inconvénients des uns par rapport aux autres est un point très important pour la conception de nouveaux algorithmes plus robustes.

L'autre branche dans laquelle il serait intéressant d'investir et sûrement améliorer notre approche concerne le convoi de véhicules. Tout au long de cette thèse nous nous sommes intéressé principalement à la modélisation et la localisation. L'utilisation et la conception de commandes plus développées améliorent sans aucun doute la mise en oeuvre du convoi.

Finalement au niveau applicatif, nous envisageons de mettre dans les véhicules expérimentaux (pour chaque véhicule du convoi) un GPS centimétrique de type GPS-RTK. Nous comparons par la suite nos résultats de localisation basés sur la fusion de la position du véhicule prédécesseur, la distance donnée par le télémètre et un GPS métrique par rapport aux données de ce GPS-RTK.

Journal

- Cherif Smaili, Maan E.El Najjar and François Charpillet.« A road matching method for precise vehicle localization using hybrid Bayesian network ». Jounal of Intelligent Transportation Systems, 12(4) :176-188, October 2008.

Chapitre de livre

- Cherif Smaili, Maan E.El Najjar, François Charpillet and Cédric Rose. « Multi-Sensor Fusion for Mono and Multi-Vehicle Localization using Bayesian Network ». Tools in Artificial Intelligence 2008. ISBN 978-953-7619-03-9.

Conférences internationales

- Cherif Smaili, Cédric Rose and François Charpillet. « Using Dynamic Bayesian Networks for a Decision Support System. Application to the Monitoring of Patients Treated by Hemodialysis ». International Computer System and Information Technology Conference, Algiers June, 2005.
- Cherif Smaili, Cédric Rose and François Charpillet.« A decision support system for the monitoring of patients treated by hemodialysis based on a bayesian network ». 3rd European Medical and Biological Engineering Conference, Prague, Novembre 2005.
- Cédric Rose, Cherif Smaili and François Charpillet. « A Dynamic Bayesian Network for Handling Uncertainty in a Decision Support System Adapted to the Monitoring of Patients Treated by Hemodialysis ». The 17th IEEE International Conference on Tools with Artifical Intelligence, November, 2005 Hong Kong.
- Cherif Smaili, Maan E.El Najjar and François Charpillet. « Multi-Sensor Fusion Method using Dynamic Bayesian Network for Precise Vehicle Localization and Road Matching ». The 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), October 2007 Patrace (Greece).
- Cherif Smaili, Maan E.El Najjar and François Charpillet « Multi-sensor Fusion Method Using Bayesian Network for Precise Multi-vehicle Localization ». The 11th International IEEE Conference on Intelligent Transportation Systems (ITSCI), October 2008 Beijing (China).

Workshops

- Cherif Smaili, Maan E.El Najjar et François Charpillet « Méthode de fusion de données multi-capteurs sûre et intèger : application à la géo-localisation mono et multi-véhicules ». Workshop sur Surveillance, Sûreté et Sécurité des Grands Systèmes (GIS 3SG) Nancy, 3-4 Juin 2009.

Mémoires

- Cherif Smaili, « Utilisation des Réseaux bayésiens dans un Système d'aide à la décision. Application aux patients traités par hémodialyse ». Mémoire pour l'obtention d'un Diplôme de Recherche Technologique de l'université Nancy 2, France, 2005.

Logiciels

- Cédric Rose, Cherif Smaili et François Charpillet. BAYABOX est une boîte à outil générique écrite sous forme d'une bibliothèque Java. Cette boîte d'inférence bayésienne est déposée auprès de l'Agence des Propriétés des Programmes et propose les outils suivants :
  1. Transformation des réseaux bayésiens en un arbre de jonction (moralisation, triangulation, identification des cliques, optimisation de l'arbre,...)
  2. Inférence pour les réseaux bayésiens discrets
  3. Inférence pour les réseaux bayésiens continus
  4. Apprentissage des paramètres pour les réseaux bayésiens discrets.

Bibliographie

[Bar-Shalom et Fortmann, 1988] Y. Bar-Shalom et T. Fortmann. *Tracking and data association.* Academic Press, INC, 1988.

[Bayle, 2006] Bernard Bayle. Robotique mobile. Rapport technique, Ecole Nationale Supérieure de Physique de Strasbourg. Université Louis Pasteur, June 2006.

[Bellot, 2002] David Bellot. *Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine.* PhD thesis, Université Henri Poincaré, Novembre 2002.

[Blanc *et al.*, 2005] G. Blanc, Y. Mezouar, et P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *22nd International Conference on Robotics and Automation (ICRA), Spain*, 2005.

[Bom *et al.*, 2005] J. Bom, F. Thuilot, F. Marmoiton, et P. Martinet. Nonlinear control for urban vehicles platooning, relying upon a unique kinematic gps. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2005.

[Bom, 2006] Jonathan Bom. *Etude et mise en oeuvre d'un convoi de vehicules urbains avec accrochage immateriel.* PhD thesis, Université Blaise Pascal - Clermont II, Juillet 2006.

[Bonnifait, 2005] Philippe Bonnifait. Contribution à la localisation dynamique d'automobiles. application à l'aide à la conduite. HDR, Université de Technologie Compiègne, Decembre 2005.

[Castillo *et al.*, 1997] Enrique Castillo, José Manuel Gutiérrez, et Ali S Hadi. *Expert Systems and Probabilistic Network Models.* Springer-Verlag New York, Inc, 1997.

[Cindy, 2008] Cappelle Cindy. *Localisation de véhicules et détection d'obstacles. Apport d'un modèle virtuel 3D urbain.* PhD thesis, Université des Sciences et Technologies de Lille, decembre 2008.

[Compillo, 2004] Fabien Compillo. Fitrage particulaire. introduction et application à la poursuite de mobile dans un réseau cellulaire. Rapport technique, Ecole Chercheurs en traitement du signal, October 2004.

[Cowell *et al.*, 1999] G.Robert Cowell, A.Philip Dawid, S. Lauritzen, et David Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer-Verlag New York, Inc, 1999.

[Dahia, 2005] Karim Dahia. *Nouvelles méthodes en filtrage particulaire. Application au recalage de navigation inertielle par mesures altimétriques.* PhD thesis, Université Joseph Fourier, janvier 2005.

[Dirac, 1961] G.A. Dirac. On rigid circuit graphs abh. In *Math. Semin. Univ. Hamburg Bd 25 (1961) 71-76*, 1961.

[Doucet, 1998] A. Doucet. On sequential simulation based methods for bayesian filtering. Rapport technique, CUFD/F-INFENG/TR.310, Signal Processing group, Department of Engineering, University of Cambridge, 1998.

[El Najjar, 2003] Maan El badaoui El Najjar. *Localisation dynamique d'un véhicule sur une carte routière numérique pour l'assistance à la conduite.* PhD thesis, Université de Technologie Compiègne, 2003.

[Fabrizi *et al.*, 1998] E. Fabrizi, G. Oriolo, S. Panzieri, et G. Ulivi. A kf-based localization algorithm for nonholonomic mobile robots. In *6th IEEE Mediterranean Conference on Control and Automation*, 1998.

[Fouhy, 2003] John Fouhy. *Computational Experiments on Graph Width Metrics.* PhD thesis, Victoria University of Wellington, 2003.

[Frueh et Zakhor, 2004] C. Frueh et A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision, 60(1) :524*, 2004.

[Fu *et al.*, 2004] M. Fu, Jie Li, et M. Wang. A hybrid map-matching algorithm based on fuzzy comprehensive judgment. In *IEEE Proceedings on Intelligent Transportation Systems, pp. 613-617*, 2004.

[Fulkerson et Gross, 1965] D.R. Fulkerson et O. A. Gross. Incidence matrices and interval graphs. In *Pacific Journal of Mathematics 15(1965), 835-855*, 1965.

[Gavril, 1974] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. In *JCTB 16(1974), pp.47-56*, 1974.

[Greenfeld, 2002] J.S. Greenfeld. Matching gps observations to locations on a digital map. In *In proceedings of the 81st Annual Meeting of the Transportation Research Board, Washington D.C.*, January 2002.

[Gustafsson *et al.*, 2002] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, et P. Nordlund. Particles filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing, Vol. 50, Nr2, pp 425-435*, 2002.

[Haton *et al.*, 1998] J. P. Haton, M. C. Haton, et F. Charpillet. Numeric/symbolic approaches for data and information fusion. *In FUSION*, 1998.

[Hedrick, 1997] K. Hedrick. Longitudinal control development for ivhs fully automated and semi-automated system : Phase iii. Rapport technique, UCB-ITS-PRR-97-20, PATH, Berkeley, CA (USA), 1997.

[Huang et Darwiche, 1994] Cecil Huang et Adnan Darwiche. Inference in belief networks : A procedural guide. Rapport technique, Elsevier Science Center Inc, 1994.

[Isidori, 1995] Alberto Isidori. *Nonlinear control systems.* Springer-Verlag, London, third edition, 1995.

[Jabbour *et al.*, 2008] Maged Jabbour, Philippe Bonnifait, et Véronique Cherfaoui. Map-matching integrity using multihypothesis road-tracking. *Journal of Intelligent Transportation Systems*, 12(4), 2008.

[Jabbour, 2007] Maged Jabbour. *Localisation de véhicules en milieu urbain à l'aide d'un lidar et d'une base de données navigable.* PhD thesis, Université de Technologie de Compiègne, France, Novembre 2007.

[Jensen *et al.*, 1990] F. Jensen, S. Lauritzen, et K. Olsen. Bayesian updating in recursive graphical models by local computations. In *Computational Statisticals Quarterly, 4 :269-282*, 1990.

[Jensen, 1988] F.V. Jensen. Junction tree and decomposable hypergraphs. Rapport technique, JUDEX, Aalborg, Denmark, 1988.

[Jordan *et al.*, 1998] M.I. Jordan, Lawrence K. Saul, et Ghahramani Zoubin. An introduction to variational methods for graphical models. In *Machine Learning 37(2) :183– 233*, 1998.

[Kim *et al.*, 2000] W. Kim, G. Jee, et J. Lee. Efficient use of digital road map in various positioning for its. In *IEEE Symposium on Position Location and Navigation, San Diego, CA.*, 2000.

[Kjærulff, 1990] Uffe Kjærulff. Triangulation of graphs - algorithms giving small total space. Rapport technique, R 90-09, Department of Mathematics and Computer Science. Institute of Electronic Systems. Aalborg University, 1990.

[Kjærulff, 1992] Uffe Kjærulff. Optimal decomposition of probabilistic networks by simlated annealing. In *Statistics and Computing, Vol 2, pp7-17*, 1992.

[Krakiwsky *et al.*, 1988] E.J. Krakiwsky, C.B. Harris, et R.V.C. Wong. A kalman filter for integrating dead reckoning, map matching and gps positioning. In *Proceedings of IEEE Position Location and Navigation Symposium, pp. 3946.*, 1988.

[Lauritzen et Spiegehalter, 1988] S.L. Lauritzen et D.J. Spiegehalter. Local computations with probabilities on graphical structures and their applications to expert systems. In *Journal of Royal Statistical Society B, 50 : 2, pp. 157 - 224*, 1988.

[Lauritzen et Wermuth, 1989] S.L. Lauritzen et N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. In *Annals of Statistics, 17, 31-57*, 1989.

[Lauritzen, 1992] S.L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. In *JASA, 87(420) :1089-1108*, 1992.

[Lauritzen, 1996] S. Lauritzen. *Graphical Models.* OUP, 1996.

[LaValle, 1999] S. M. LaValle. *Planning algorithms. Lien : http ://msl.cs.uiuc.edu/planning/.* Published online, 1999.

[Le Gland, 2005] François Le Gland. Introduction au filtrage en temps discret. filtre de kalman, filtrage particulaire, modèles de markov cachés. Rapport technique, Université de Rennes 1, 2005.

[Leimer, 1993] H.G. Leimer. Optimal decomposition by clique separators. In *Discrete Mathematics Vol 113, pp90-123*, 1993.

[Leray et Gallinari, 1998] Philippe Leray et Patrick Gallinari. Une architecture neuro-bayésienne pour le traitement spatio-temporel d'alarmes. application au diagnostic dans le réseau téléphonique. journées nationales sur les modèles de raisonnement, 1998.

[Martinet *et al.*, 2005] P. Martinet, B. Thuilot, et J. Bom. From autonomous navigation to platooning in urban context. In *Proceedings of the IARP-Workshop on Adaptive and Intelligent Robots : Present and Future, vol 1.1, pp. 1-9*, 2005.

[Meng, 2006] Y. Meng. *Improved Positioning of Land Vehicle in ITS Using Digital Map and Other Accessory Information.* PhD thesis, Department of Land Surveying and Geoinformatics, Hong Kong Polytechnic University., 2006.

[Murphy, 1999] Kevin P. Murphy. A variational approximation for bayesian networks with discrete and continuous latent variables. *UAI*, 1999.

[Murphy, 2001] Kevin P. Murphy. An introduction to graphical models, May 2001.

[Murphy, 2002] Kevin P. Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning.* PhD thesis, University of California Berkeley, 2002.

[Murray et Sastry, 1993] R. M. Murray et S. S. Sastry. Nonholonomic motion planning : Steering using sinusoids. *IEEE Trans. on Automatic Control,*, 38, no.5 :700–716, 1993.

[Murray, 1993] R. M. Murray. Control of nonholonomic systems using chained forms. *Fields Institute communications,*, 1 :219–245, 1993.

[Pearl, 1988] J. Pearl. *Probabilistic reasoning in intelligent systems : Networks of plausible inference.* Morgan Kaufman Publishers, Inc., San Mateo, CA, 2nd edition, 1988.

[Pearl, 2001] J. Pearl. *Causality- Models, reasonig and inference.* Cambridge University Press, 2001.

[Pham *et al.*, 2003] D. T. Pham, K. Dahia, et C. Musso. A kalman-particle kernel filter and its application to terrain navigation. In *Proceeding of the Fusion 2003 Conference, Australia,*, 2003.

[Puig, 2003] Bénédicte Puig. *Modélisation et simulation de processus stochastiques non gaussiens.* PhD thesis, Université Pierre et Marie Curie - Paris VI, 2003.

[Quddus *et al.*, 2003] Mohammed Quddus, Washington Yotto Ochieng, Lin Zhao, et Robert B. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7(3), 2003.

[Quddus, 2006] Mohammed Quddus. *High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications.* PhD thesis, University of London, January 2006.

[Rabiner, 1989] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE, volume 77, pages 257285.*, 1989.

[Rose *et al.*, 2005] Cédric Rose, Cherif Smaili, et François Charpillet. A dynamic bayesian network for handling uncertainty in a decision support system adapted to the monitoring of patients treated by hemodialysis. *The 17th IEEE International Conference on Tools with Artificial Intelligence, Hong Kong*, 2005.

[Royer *et al.*, 2004] E. Royer, Lhuillier, M. Dhome, et T. Chateau. Towards an alternative gps sensor in dense urban environment from visual memory. In *British Machine Vision Conference, volume 1, pages 197, Kingston (England).*, 2004.

[Royer, 2006] Eric Royer. *Cartographie 3D et localisation par vision monoculaire pour la navigation autonome d'un robot mobile.* PhD thesis, Université BLAISE PASCAL - CLERMONT II, Septembre 2006.

[Samson, 1995] Claude Samson. Control of chained systems. application to path following and time-varying point-stabilization of mobile robot. *IEEE Trans. on Automatic Control*, 40, no.1 :64–77, 1995.

[Skaanning *et al.*, 1999] Claus Skaanning, Finn V. Jensen, et Uffe Kjaerulff. Printer troubleshooting using bayesian networks. Rapport technique, Department of Computer Science, Aalborg University, Denmark, 1999.

[Smaili *et al.*, 2005a] Cherif Smaili, Cédric Rose, et François Charpillet. A decision support system for the monitoring of patients treated by hemodialysis based on a bayesian network. *3rd European Medical and Biological Engineering Conference, Prague*, 2005.

[Smaili *et al.*, 2005b] Cherif Smaili, Cédric Rose, et François Charpillet. Using dynamic bayesian networks for a decision support system application to the monitoring of patients treated by hemodialysis. *International Computer Systems and Information Technology Conference, Algiers*, 2005.

[Smaili *et al.*, 2008a] Cherif Smaili, Maan E.EL Najjar, et François Charpillet. Multi-sensor fusion method using bayesian network for precise multi-vehicle localization. In *The 11th International IEEE Conference on Intelligent Transportation Systems (ITSCI), Beijing (China).*, 2008.

[Smaili *et al.*, 2008b] Cherif Smaili, Maan E.EL Najjar, et François Charpillet. A road matching method for precise vehicle localization using hybrid bayesian network. *Journal of Intelligent Transportation Systems*, 12(4), 2008.

[Sordalen, 1993] O. J. Sordalen. Conversion of the kinematics of a car with n trailers into a chained form. *IEEE International Conference on Robotics and Automation*, 1 :382–387, 1993.

[Suermondt et Arnylon, 1989] H. J. Suermondt et M. D. Arnylon. Probabilitic prediction of the outcome of bone-marrow transplantation. In *Kingsland (ed). Proc 13th SCAMC. New-York :IEEE Computer Press 208-12*, 1989.

[Syed et Cannon, 2004] S. Syed et M.E. Cannon. Fuzzy logic-based map-matching algorithm for vehicle navigation system in urban canyons. In *proceedings of the Institute of Navigation (ION) national technical meeting 26-28, California, USA*, 2004.

[Tanaka *et al.*, 1990] J. Tanaka, K. Hirano, T. Itoh, H. Nobuta, et S. Tsunoda. Navigation system with map-matching method. In *Proceeding of the SAE International Congress and Exposition, pp. 4050.*, 1990.

[Tarjan et Yannakakis, 1984] R.E Tarjan et M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclity of hypergraphs and selectively reduce acyclic hypergraphs. In *SIAM Journal of Computing, 13 :566-579*, 1984.

[Taylor *et al.*, 2001] G. Taylor, G. Blewitt, D. Steup, S. Corbett, et A. Car. Road reduction filtering for gps-gis navigation. In *Transactions in GIS, ISSN 1361-1682, 5(3), 193207.*, 2001.

[Thrun, 2003] Sebastian Thrun. *Robotic mapping : a survey.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2003.

[Thuilot *et al.*, 2002] B. Thuilot, C. Cariou, P. Martinet, et M. Berducat. Automatic guidance of a farm tractor relying on a single cp-dgps. In *Autonomous Robots, Vol 13, Number 1, pp. 53-71(19)*, 2002.

[Thuilot *et al.*, 2006] B. Thuilot, E. Royer, F. Marmoiton, P. Martinet, M. Dhome, et M. Lhuillier. Navigation autonome de véhicule urbains par différentes modalités capteurs (rtk-gps et vision monoculaire. In *Journées Démonstrateurs en Automatique, Angers, France*, 2006.

[Uri, 2002] Lerner Uri. *Hybrid bayesian networks for reasoning about complex systems.* PhD thesis, Stanford University, department of computer science, October 2002.

[Vidyasagar, 1993] M. Vidyasagar. *Nonlinear systems analysis.* Prentice-Hall, Englewoods, second edition, 1993.

[White *et al.*, 2000] C. White, D. Bernstein, et A. Kornhauser. Some map-matching algorithms for personal navigation assistants. In *Transportation Research Part, C 8, 91-108*, 2000.

[Woodman, 2007] O. J. Woodman. An introduction to inertial navigation. Rapport technique, UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, 2007.

[Yang *et al.*, 2003] D. Yang, D. Cai, et Y. Yuan. An improved map-matching algorithm used in vehicle navigation system. In *IEEE Proceedings on Intelligent Transportation Systems pp. 1246-1250.*, 2003.

[Yannakakis, 1981] M. Yannakakis. Computing the minimum fill-in is np-complete. In *SIAM J. Alg. Disc. pp. 77–79. 12*, 1981.

[Zhao, 1997a] Yilin Zhao. *Intelligent Transportation Systems : Vehicle Location and Navigation Systems.* Artech House, 1997.

[Zhao, 1997b] Yilin Zhao. *Vehicle Location and Navigation System.* Artech House, Inc., 1997.

[Zweig, 1996] G. Zweig. A forward-backward algorithm for inference in bayesian networks and an empirical comparison with hmms. Master's thesis, Dept. Comp. Sci., U.C. Berkeley, 1996.

Résumé

Cette thèse présente la fusion multi-capteurs par un réseau Bayésien appliqué au problème de localisation d'un véhicule sur une carte. La mise en correspondance d'une estimation sur un segment de route ou Road-matching consiste à trouver le segment sur lequel le véhicule roule et la position de ce véhicule sur ce segment. Plusieurs algorithmes utilisent la fusion des estimations données par l'odométrie et le GPS pour traiter le problème du road-matching. Cependant, une simple combinaison du GPS et de l'odométrie ne permet pas de se localiser de manière précise et sans interruption de service. La précision et la continuité de service peuvent être améliorées si on utilise des informations cartographiques qui permettent en particulier de contraindre les positions possibles aux seuls segments correspondants à des voies de circulation autorisées.
Dans de nombreux cas, lorsqu'un véhicule se trouve devant des situations ambiguës comme les routes parallèles, les jonctions de routes,... plusieurs auteurs cherchent à sélectionner le segment le plus probable. Cette phase est souvent une source d'erreurs. Dans cette thèse nous proposons de traiter tous les segments candidats jusqu'à la levée de l'ambiguïté.

Le problème de la localisation devient encore plus compliqué quand il s'agit d'une localisation multi-véhicules pour une navigation autonome des véhicules suiveurs. Pour un train de véhicules dont seul le premier est piloté par un opérateur humain et dont les véhicules suiveurs sont en mode autopilotage, une géo-localisation précise d'ordre centimétrique de chaque véhicule est plus que nécessaire pour les modules de contrôle pour le suivi de trajectoire du véhicule de tête.
Un train de véhicule peut être vu comme la généralisation du modèle de réseau Bayésien pour la localisation d'un véhicule sur une carte. Nous dupliquons le réseau autant de fois qu'on a de véhicule. Nous rajoutons des liens de connexions entre les véhicules afin de concevoir le train de véhicule.

Le filtre de Kalman et sa version étendue sont très utilisés en robotique, principalement pour traiter le problème de la non linéarité. Cependant, la linéarisation du système autour de l'estimée courante peut introduire des erreurs sur la moyenne et la covariance calculées *a posteriori* et peut même dans d'autres cas faire diverger le filtre.
La transformation du système non linéaire d'un véhicule sous forme chaînée permet de représenter son équation cinématique sous forme linéaire. Par conséquent, cette transformation nous évite de faire appel aux méthodes d'inférence approximatives.

Mots-clés: *Réseau bayésien, GPS, Odométrie, Cartographie, Map-matching, Localisation, Multi-capteurs, Mutihypothèses, Fusion de données, Modèle chaîné, Ambiguïté, Train de véhicule.*

Abstract

This thesis presents a multi-sensor fusion strategy for a novel road-matching based on Bayesian Network. Road-matching is a technique that attempts to locate an estimated vehicle position on road network. Many road-matching algorithms have been developed and widely incorporated into GPS/DR vehicle navigation systems for both commercial and experimental ITS applications. However, simply combining GPS and DR cannot provide an accurate vehicle positioning system. In that case, we use the digital map as an observation to improve the reliability of road-matching.

In many cases, when a vehicle is in front ambiguous situations (close roads, junction roads...), we try to identify the most likely segment. Inevitably, this identification is closely related to the accuracy of sensors and therefore the identification of the most likely segment remains a real problem. To resolve these ambiguous situations, this work presents a multi-sensor fusion and multi-modal estimation realized using Bayesian Network. The strategy presented in this work doe not keep only the most likely segment. When approaching an intersection, several roads can be good candidates for this reason we manage several hypotheses until the situation becomes unambiguous.

Multi-vehicle localisation method presented in this work can be seen like an extension of the method presented above. In the sense that we have duplicate the Bayesian Network used to fuse measurements sensors to localize one vehicle for several ones. Than, we have added vehicles inter-connexion to represent finally the platoon of vehicles in the context of Bayesian Network. This kind of platoon representation for vehicles poses estimations permits additionally to implement control law based on near-to-near approach, which can only be seen as a first step in platoon control design.

The use of Extended Kalman Filter has been extensively used for localizing such a vehicle from multi-sensors data fusion. The non-linearity of the vehicle kinematic model gave raise to several extensions of classical Kalman filters such as Extended Kalman Filter and Unscented Kalman Filter. In order to avoid the linearization process, we propose to take advantage of the possibility of transforming the kinematic system in chained form using an appropriate feedback transformation. With the linear form (chained form), an exact inference computation on the Bayesian network can be performed. Real data are used (ABS sensors, a differential GPS receiver and an accurate digital roadmap) to illustrate the performance of our approach.

Keywords: *Bayesian Network, GPS, Deduced Reckoning, Digital map, Map-matching, Localization, Multi-sensors, Multihypotheses, Data fusion, Chained form, Ambiguity, Platoon.*