



HAL
open science

Algorithmes de géolocalisation à l'intérieur d'un bâtiment en temps différé

Kersane Zoubert-Ousseni

► **To cite this version:**

Kersane Zoubert-Ousseni. Algorithmes de géolocalisation à l'intérieur d'un bâtiment en temps différé. Traitement du signal et de l'image [eess.SP]. Université de Rennes, 2018. Français. NNT : 2018REN1S015 . tel-01838480

HAL Id: tel-01838480

<https://theses.hal.science/tel-01838480>

Submitted on 13 Jul 2018

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

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

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Signal, Image, Vision

École doctorale MATHSTIC

présentée par

Kersane Zoubert-Ousseni

préparée au centre CEA de Grenoble et au centre INRIA de Rennes
UFR Informatique et Électronique (ISTIC)

**Algorithmes de
géolocalisation à
l'intérieur d'un
bâtiment en temps
différé**

**Thèse soutenue à Grenoble
le 10 avril 2018**

devant le jury composé de :

Valérie RENAUDIN

Chercheuse à l'IFSTTAR Nantes / *examinatrice*

François SEPTIER

Maître de conférences à l'IMT Lille Douai /
examineur

François SPIES

Professeur à l'université de Franche-Comté /
rapporteur

Jean-Yves TOURNERET

Professeur à l'ENSEEIHNT / *rapporteur*

François LE GLAND

Chercheur à l'INRIA Rennes / *directeur de thèse*

Christophe VILLIEN

Chercheur au CEA LETI Grenoble / *co-directeur de
thèse*



Remerciements

Quelle aventure que de faire une thèse, et comme toute aventure, elle comporte son lots de personnages qui méritent d'être remerciés à la hauteur de ce qu'ils m'aurent apporté.

Je tiens dans un premier temps à grandement remercier mon encadrant Christophe Villien, pour son suivi assidu, pour ses conseils et recommandations, pour m'avoir permis de garder la motivation à chaque instant, et d'aller au bout avec satisfaction de cette thèse.

Je tiens également à présenter ma gratitude pour mon directeur de thèse François Le Gland pour son encadrement et son suivi depuis Rennes.

Par la suite, un grand merci à l'ensemble des membres du jury qui ont accepté puis pris le temps de juger mes travaux : madame Valérie Renaudin, messieurs François Septier, François Spies et Jean-Yves Tourneret.

Également une pensée pour les permanents avec qui j'ai partagé mon bureau, ou qui m'ont régulièrement conseillé sur le plan scientifique de mes travaux : Benoît Denis, Elisabeth Delevoye et Manuel Pezzin.

Enfin, une grosse pensée pour l'ensemble de mes collègues avec qui j'ai partagé l'ensemble de mes pauses déjeuner et café, ou encore les jeudi foot ou les soirées chez les ou chez les autres : Felipe, Mouhcine, Matthias, Andres, Régis, Matthias, Niels, Edgar, Ibrahima, Tibor, Vita, Victor, Arun, Maxime, Rémy, Élodie, Ludovic ou encore Jimmy.

Évidemment, une très grande gratitude pour ma famille, et particulièrement mes parents, ainsi que mes amis à qui je remercie de sans cesse être présent pour moi.

Merci à ma femme pour son soutien évidemment unique.



Résumé

La géolocalisation *indoor* en temps réel a largement été étudiée ces dernières années, et de nombreuses applications y sont associées. Une estimation en temps différé de la trajectoire présente également un certain intérêt. La géolocalisation *indoor* en temps différé permet par exemple de développer des approches de type *crowdsourcing* qui tirent profit d'un grand nombre d'utilisateurs afin de récolter un grand nombre de mesures : la connaissance du trajet d'un utilisateur muni d'un *smartphone* permet par exemple d'alimenter une carte de fréquentation du bâtiment. Estimer la trajectoire de cet utilisateur ne nécessite pas de traitement en temps réel et peut s'effectuer en temps différé ce qui offre deux avantages. D'abord, l'approche temps réel estime une position courante uniquement avec les mesures présentes et passées, alors que l'approche temps différé permet d'avoir accès à l'ensemble des mesures et permet d'obtenir une trajectoire estimée plus régulière et plus précise qu'en temps réel. Par ailleurs, cette estimation peut se faire sur un serveur et n'a pas besoin d'être portée par un *smartphone* comme c'est le cas en temps réel, ce qui permet d'utiliser une puissance de calcul et un volume mémoire plus importants. L'objet de ces travaux de thèse est de proposer une estimation de la trajectoire d'un individu se déplaçant avec un *smartphone* recevant des mesures de puissance wifi ou bluetooth (RSS) et enregistrant des mesures inertielles (IMU).

En premier lieu, sans la connaissance de la position des murs de la carte, un modèle paramétrique est proposé, basé sur un modèle de propagation d'onde adaptatif pour les mesures RSS ainsi que sur une modélisation par morceaux de la trajectoire inertielle, issue des mesures IMU. Les résultats obtenus en temps différé ont une moyenne d'erreur de 6.2m contre 12.5m en temps réel. En second lieu, l'information des contraintes de déplacement induites par la présence des murs du bâtiment est ajoutée et permet d'affiner l'estimation de la trajectoire avec une technique particulière, comme il est couramment utilisé dans la littérature. Cette seconde approche a permis de développer un lisseur particulière ainsi qu'un estimateur du maximum *a posteriori* par l'algorithme de Viterbi. D'autres heuristiques numériques ont été présentées. Une première heuristique ajuste le modèle d'état de l'utilisateur, qui est initialement uniquement basé sur les mesures IMU, à partir du modèle paramétrique développé sans les murs. Une seconde heuristique met en oeuvre plusieurs réalisations d'un filtre particulière et définit deux scores basés sur les mesures RSS et sur la continuité de la trajectoire. Les scores permettent de sélectionner la meilleure réalisation du filtre. Un algorithme global, regroupant l'ensemble de ces approches permet d'obtenir une erreur moyenne de 3.6m contre 5.8m en temps réel. Enfin, un modèle d'apprentissage statistique basé sur des forêts aléatoires a permis de distinguer les trajectoires qui ont été correctement estimées en fonction d'un faible nombre de variables, en prévision d'une application au *crowdsourcing*.

Mots clés : Localisation *indoor*, estimation en temps différé, modèle paramétrique, lissage particulière, maximum *a posteriori*

Abstract

Real time indoor geolocalization has recently been widely studied, and has many applications. Off-line (post-processing) trajectory estimation also presents some interest. Off-line indoor geolocalization makes it possible for instance to develop crowdsourcing approaches that take advantage of a large number of users to collect a large number of measurements: knowing the trajectory of a smartphone user makes it possible for instance to feed an attendance map. Estimating this trajectory does not need to be performed in real-time and can be performed off-line, two main benefits. Firstly, the real-time approach estimates a current position using present and past measurements only, when the off-line approach has access to the whole measurements, and makes it possible to obtain an estimated trajectory that is smoother and more accurate than with a real-time approach. Secondly, this estimation can be done on a server and does not need to be implemented in the smartphone as it is the case in the real-time approach, with the consequence that more computing power and size memory are available. The objective of this PhD is to provide an off-line estimation of the trajectory of a smartphone user receiving signal strength (RSS) of wifi or bluetooth measurements and collecting inertial measurements (IMU).

In the beginning, without the floorplan of the building, a parametric model is proposed, based on an adaptive pathloss model for RSS measurements and on a piecewise parametrization for the inertial trajectory, obtained with IMU measurements. Results are an average error of 6.2m for the off-line estimation against 12.5m for the real-time estimation. Then, information on displacement constraints induced by the walls is considered, that makes it possible to adjust the estimated trajectory by using a particle technique as often done in the state-of-the-art. With this second approach we developed a particle smoother and a maximum a posteriori estimator using the Viterbi algorithm. Other numerical heuristics have been introduced. A first heuristic makes use of the parametric model developed without the floorplan to adjust the state model of the user which was originally based on IMU alone. A second heuristic proposes to perform several realization of a particle filter and to define two score functions based on RSS and on the continuity of the estimated trajectory. The scores are then used to select the best realization of the particle filter as the estimated trajectory. A global algorithm, which uses all of the aforementioned approaches, leads to an error of 3.6m against 5.8m in real-time. Lastly, a statistical machine learning model produced with random forests makes it possible to distinguish the correct estimated trajectories by only using few variables to be used in a crowdsourcing framework.

Key words: Indoor localization, post-processing estimation, parametric model, particle smoothing, maximum a posteriori



Table des matières

Table des matières	iii
Liste des figures	vii
Liste des tableaux	xi
1 Introduction et contexte général	1
1.1 Enjeux de la géolocalisation <i>indoor</i> en temps différé	2
1.1.1 Un historique personnel de déplacement <i>indoor</i>	3
1.1.2 Le <i>crowdsourcing</i> comme application de l'estimation en temps différé	3
1.2 Application au <i>smartphone</i>	5
1.2.1 Localisation inertielle	6
1.2.2 Mesures de puissance	8
1.2.3 Baromètre et mesures de pression pour la détection du changement d'étage	8
1.2.4 Mesure du GPS	8
1.2.5 Autres sources d'information	9
1.2.6 Définition de la problématique d'estimation en temps différé	9
1.3 Cadre théorique de l'estimation	9
1.3.1 Modèle de régression paramétrique	10
1.3.2 Théorie du filtrage et du lissage	11
1.3.3 Métriques d'évaluation	13
1.4 Plan de thèse	14
1.5 Références	15
2 Modèle de régression paramétrique à partir des mesures RSS et IMU	17
2.1 État de l'art	18
2.1.1 Méthode de localisation par les mesures RSS	18
2.1.2 État de l'art de la géolocalisation par mesure inertielle	21
2.1.3 Fusion des mesures RSS avec l'approche inertielle	25
2.1.4 Synthèse de l'état de l'art	27
2.2 Fusion en temps réel RSS et IMU	27
2.2.1 Modèle d'état	27
2.2.2 Modèle d'observation	29
2.2.3 Méthode d'estimation en temps réel	30
2.3 Fusion paramétrique des mesures RSS et IMU	31
2.3.1 Paramétrisations et définition des modèles	31
2.3.2 Estimateur du maximum <i>a posteriori</i>	39
2.3.3 Résolution numérique de l'optimisation	42
2.4 Résultats	45

2.4.1	Application numérique	45
2.4.2	Illustration sur une trajectoire sans rupture	45
2.4.3	Illustration sur une trajectoire avec des ruptures	47
2.4.4	Estimation en temps différé de la trajectoire réelle	48
2.4.5	Résultats sur la base de données	49
2.5	Conclusion	50
2.6	Références	51
3	Fusion des mesures avec une carte	55
3.1	État de l'art sur la prise en compte des murs d'un bâtiment	56
3.1.1	Contexte de la problématique de <i>map-matching</i>	56
3.1.2	État de l'art des méthodes de <i>map-matching</i> à l'intérieur d'un bâtiment en temps réel	57
3.1.3	État de l'art des méthodes <i>map-matching</i> en temps différé à l'intérieur d'un bâtiment	61
3.1.4	Synthèse de l'état de l'art	65
3.2	Filtre particulaire pour la localisation <i>indoor</i>	65
3.2.1	Théorie du filtre particulaire	65
3.2.2	Algorithme utilisé pour l'estimation en temps réel	68
3.3	Formalisation de l'estimation en temps différé	72
3.3.1	Multi-modalité et problème de continuité	72
3.3.2	Application du lissage sur les <i>clusters</i>	74
3.4	Méthodes de résolution de la problématique du chemin le plus probable	77
3.4.1	Algorithme de Viterbi	77
3.4.2	Approche Bayésienne	78
3.5	Résultats	80
3.5.1	Illustration sur une trajectoire avec des discontinuités	80
3.5.2	Résultats sur la base de données	83
3.5.3	Conclusion	84
3.6	Références	85
4	Heuristiques d'amélioration en temps différé d'un filtre particulaire	87
4.1	Méthode de répétition d'un filtre particulaire	88
4.1.1	Présentation de l'approche	88
4.1.2	Caractérisation non-supervisée de la trajectoire	89
4.1.3	Augmentation de la variabilité du filtre	94
4.1.4	Résultats	96
4.1.5	Synthèse	101
4.2	Couplage de la fusion paramétrique des mesures RSS et IMU avec un filtre particulaire	102
4.2.1	Expression de la trajectoire WIB	102
4.2.2	Algorithme WIB + PF	102
4.2.3	Résultats	105
4.2.4	Synthèse	109
4.3	Méthode de sélection des trajectoires	110
4.3.1	Présentation de l'approche	110
4.3.2	Formalisation de la problématique	111
4.3.3	Méthodes d'apprentissage statistique supervisé	112
4.3.4	Résultats	115
4.3.5	Synthèse	118

4.4	Références	119
5	Résultats généraux	121
5.1	Protocole d'évaluation des méthodes	122
5.1.1	État de l'art	122
5.1.2	Description des campagnes de mesure	123
5.1.3	Post-traitement des mesures et outil de benchmark	128
5.2	Couplage des méthodes	129
5.2.1	Description du couplage des méthodes	129
5.2.2	Application à une longue trajectoire	130
5.2.3	Application à la base de données	136
5.2.4	Conclusion	141
5.3	Références	142
	Conclusion générale	143
	Annexe	I
A	Développements mathématiques	I
A.1	Preuve par récurrence de la formule 2.24	I
A.1.1	Initialisation de la récurrence :	I
A.1.2	Hérédité de la récurrence :	I
A.2	Évaluation des matrices jacobienes	III
A.2.1	Évaluation de la matrice jacobienne \mathbf{J}_z	III
A.2.2	Évaluation de la matrice jacobienne \mathbf{J}_v	IV
A.3	Calcul du score de continuité s_{cont}	V
B	Distribution des variables du modèle d'apprentissage	VII

Liste des figures

1.1	Schéma de la mise en oeuvre du <i>crowdsourcing</i> appliqué à la géolocalisation <i>indoor</i>	4
1.2	Illustration des capteurs et antennes présents dans un téléphone	5
1.3	Schéma de la paramétrisation de la trajectoire inertielle	7
1.4	Schéma de l'estimation de la trajectoire inertielle	8
2.1	Exemple d'une carte de puissance wifi tiré de KLEPAL et collab. [2007]	19
2.2	Illustration de l'estimation par les k-plus proches voisins pour l'approche de <i>fingerprinting</i> déterministe tirée de KHODAYARI et collab. [2010]	20
2.3	Illustration de l'estimation de la trilatération tirée de HIGHTOWER et collab. [2000]	21
2.4	Représentation des mesures a_z en fonction du temps pour détecter les pas d'un individu tiré de SUN et collab. [2009]	22
2.5	Représentation du pendule inversé tirée de JAHN et collab. [2010]	23
2.6	Illustration de la différence entre le repère de déplacement d'un individu et le repère du mobile	24
2.7	Illustration d'un changement de position du téléphone	25
2.8	Illustration d'une approche de fusion des mesures IMU et RSS par une méthode de filtrage selon WILK et collab. [2015]	27
2.9	Trajectoire inertielle contre Trajectoire réelle sur une courte trajectoire	28
2.10	Illustration de l'estimation en temps réel via une fusion des mesures IMU et RSS selon l'algorithme temps réel	30
2.11	Trajectoire inertielle contre Trajectoire réelle sur une longue trajectoire	32
2.12	Trajectoire inertielle séparée en différents segments	33
2.13	Histogramme normalisé sur les paramètres	35
2.14	Représentation des mesures comparées au modèle	36
2.15	Histogramme du biais du modèle	37
2.16	Différence entre le modèle classique et le modèle centré	38
2.17	Représentation de la fonction de coût associée au modèle globale en fonction de β et ϵ	41
2.18	Représentation de la fonction de coût associée au modèle centré sans les mesures de proximité en fonction de β et ϵ	41
2.19	Représentation de la fonction de coût associée au modèle centré avec les mesures de proximité en fonction de β et ϵ	42
2.20	Illustration de la symétrie du modèle d'observation	44
2.21	Présentation de la trajectoire de référence pour l'exemple 1	46
2.22	Exemple de l'estimation RSS + IMU en temps différé sans rupture	47
2.23	Présentation de la trajectoire de référence pour l'exemple 2	48
2.24	Exemple de l'estimation RSS + IMU en temps différé	49

3.1	Présentation de la problématique de <i>map-matching outdoor</i> selon BERNSTEIN et KORNHAUSER [1998]	57
3.2	Comportement des particules lors d'un conflit avec un mur	58
3.3	Représentation des mailles de discrétisation selon EBNER et collab. [2015]	59
3.4	Représentation d'une discrétisation en grille selon GALČÍK et OPIELA [2016]	60
3.5	Représentation d'une approche intermédiaire selon ALAOUI et collab. [2016]	61
3.6	Illustration de la segmentation de la trajectoire selon SPASSOV [2007]	62
3.7	Représentation de la carte topologique pour deux bâtiments selon WANG et collab. [2013]	62
3.8	Illustration de l'algorithme de <i>map matching</i> en environnement indoor selon WANG et collab. [2015]	63
3.9	Présentation d'une approche de lissage particulière, tirée de KLEPAL et collab. [2008]	64
3.10	Illustration d'un nombre de <i>clusters</i> trop faible	71
3.11	Illustration d'un nombre de <i>clusters</i> trop grand	72
3.12	Illustration des problèmes de discontinuité à cause des murs	73
3.13	Illustration des problèmes de discontinuité à cause des mesures RSS	74
3.14	Illustration en dimension 1 d'un GMM associé à deux nuages de particules	75
3.15	Représentation du choix des <i>clusters</i> en temps réel	76
3.16	Problématique de la sélection du meilleur chemin	76
3.17	Représentation de la trajectoire de référence	81
3.18	Estimation en temps réel d'une trajectoire <i>map-matchée</i>	81
3.19	Évolution des <i>clusters</i> estimés en temps réel.	82
3.20	Estimations en temps différé par la méthode de sélection des <i>clusters</i>	83
4.1	Schéma de l'heuristique de répétition du filtre	88
4.2	Illustration des scores combinés sur un cas applicatif concret	93
4.3	Schéma de l'approche de la séparation en sous-trajectoires	94
4.4	Représentation de la trajectoire de référence	97
4.5	Illustration de la méthode de répétition du filtre	98
4.6	Illustration des différents cas possibles pour l'ajustement du modèle d'état	103
4.7	Schéma représentant l'algorithme de couplage entre la trajectoire WIB et le filtre particulière	104
4.8	Parcours de la trajectoire de référence	105
4.9	Représentation de la calibration de la trajectoire inertielle	106
4.10	Estimation en temps réel de la trajectoire estimée par le filtre particulière appliqué à la trajectoire inertielle	107
4.11	Estimation en temps réel de la trajectoire estimée par le filtre particulière appliqué à la trajectoire WIB	108
4.12	Schéma de l'application d'un algorithme d'apprentissage supervisé pour la sélection de trajectoire	110
4.13	Illustration d'un arbre de classification	113
4.14	Représentation de l'importance des variables pour le modèle global	116
4.15	Représentation de l'importance des variables pour le modèle réduit	116
4.16	Représentation du taux de faux positifs en fonction du taux de faux négatifs	118
5.1	Plan des locaux du site 1 à Grenoble	123
5.2	Plan des locaux d'un bâtiment de bureaux	124
5.3	Plan des locaux du site 3 à Grenoble	124
5.4	Plan du site 4 à Grenoble	125

5.5	Interface de l'application permettant l'acquisition des données	127
5.6	Schéma de l'outil du benchmark pour tester les différentes versions compilées	128
5.7	Schéma de l'approche de l'estimation final en temps différé	130
5.8	Estimation de la trajectoire en temps réel	131
5.9	Évolution de l'erreur le long de la trajectoire de l'estimation en temps réel .	131
5.10	Estimation après l'étape 1 de l'approche finale	132
5.11	Évolution de l'erreur le long de la trajectoire de l'estimation après l'étape 1 .	133
5.12	Estimation après l'étape 2 de l'approche finale	133
5.13	Évolution de l'erreur le long de la trajectoire de l'estimation après l'étape 2.	134
5.14	Trajectoire AF	135
5.15	Évolution de l'erreur le long de la trajectoire AF	135
5.16	Représentation des ratios de trajectoires correctes pour le rmse et le %d5 pour chaque approche	140
B.1	Distribution de la durée du parcours selon la qualité d'estimation de la tra- jectoire	VII
B.2	Distribution du nombre de pas selon la qualité d'estimation de la trajectoire	VIII
B.3	Distribution du score de continuité selon la qualité d'estimation de la tra- jectoire	VIII
B.4	Distribution du score RSS selon la qualité d'estimation de la trajectoire . . .	IX
B.5	Distribution de l'erreur rmse entre l'estimation en temps différé et la trajec- toire WIB selon la qualité d'estimation de la trajectoire	IX
B.6	Distribution du ratio %d5 entre l'estimation en temps différé et la trajectoire WIB selon la qualité d'estimation de la trajectoire	X
B.7	Distribution de la distance de Frechet entre l'estimation en temps différé et la trajectoire WIB selon la qualité d'estimation de la trajectoire	X

Liste des tableaux

2.1	Évaluation de la fusion IMU et RSS sur la trajectoire 1	46
2.2	Évaluation de la fusion IMU et RSS sur la trajectoire 1	48
2.3	Résultats de l'estimation en temps différé selon le nombre d'itérations et le type de rupture artificielle sur un ensemble de 130 expériences	50
3.1	Évaluation de la méthode de sélection du chemin sur un cas applicatif ainsi que les coûts de calcul associés	82
3.2	Résultats de l'estimation en temps différé selon la méthode de sélection ainsi que le nombre de <i>clusters</i> sur un ensemble de 130 expériences	83
4.1	Représentation des scores sur différentes estimations d'un cas applicatif concret	93
4.2	Évaluation de la méthode de répétition du filtre	98
4.3	Résultats de la répétition du filtre en temps différé selon le nombre d'itérations, le nombre de particules et la distance de séparation sur un ensemble de 130 expériences	100
4.4	Évaluation de l'ajustement du modèle d'état sur un exemple applicatif . . .	107
4.5	Résultats de l'estimation en temps différé selon le nombre d'itérations et le type de rupture artificielle sur un ensemble de 130 expériences	108
4.6	Comparaison du taux de faux négatifs pour un taux de faux positifs fixé à 10% pour chacune des méthodes	118
5.1	Description des différentes campagnes de mesures	126
5.2	Notation des différents algorithmes	130
5.3	Comparaison des temps de coûts de calcul et des temps de calcul effectifs .	137
5.4	Valeurs de la moyenne du %d5 par site de campagne pour les trajectoires sur un seul étage	138
5.5	Valeurs de la moyenne du rmse en mètre par site de campagne pour les trajectoires sur un seul étage	138
5.6	Valeurs de la moyenne du %d5 par site de campagne pour les trajectoires sur plusieurs étages	139
5.7	Valeurs de la moyenne du rmse en mètre par site de campagne pour les trajectoires sur plusieurs étages	139
5.8	Comparaison de toutes les approches sur un ensemble de 190 expériences sur un ou plusieurs étages	141

Chapitre 1

Introduction et contexte général

Sommaire

1.1 Enjeux de la géolocalisation <i>indoor</i> en temps différé	2
1.1.1 Un historique personnel de déplacement <i>indoor</i>	3
1.1.2 Le <i>crowdsourcing</i> comme application de l'estimation en temps différé	3
1.2 Application au <i>smartphone</i>	5
1.2.1 Localisation inertielle	6
1.2.2 Mesures de puissance	8
1.2.3 Baromètre et mesures de pression pour la détection du changement d'étage	8
1.2.4 Mesure du GPS	8
1.2.5 Autres sources d'information	9
1.2.6 Définition de la problématique d'estimation en temps différé	9
1.3 Cadre théorique de l'estimation	9
1.3.1 Modèle de régression paramétrique	10
1.3.2 Théorie du filtrage et du lissage	11
1.3.3 Métriques d'évaluation	13
1.4 Plan de thèse	14
1.5 Références	15

Aujourd'hui, un individu passe en moyenne plus de 75% de son temps à l'intérieur d'un bâtiment [KLEPEIS et collab. \[2001\]](#). Récemment, de très nombreuses applications liées à un environnement *indoor*, c'est-à-dire dans un bâtiment, ont été développées. À l'extérieur d'un bâtiment, ou dans un environnement *outdoor*, les systèmes de navigation par satellite, tels que le GPS (*Global Positioning System*) proposent une solution de localisation très satisfaisante en raison de la précision sur la position obtenue, d'environ 5m pour un *smartphone* classique, et également de la couverture de ce système. Néanmoins, cette technologie n'est plus applicable dans un environnement *indoor*. En effet, différents obstacles, tels que les plafonds, perturbent les signaux satellitaires [PURICER et KOVAR \[2007\]](#). Ces dernières années, de nouvelles technologies ont donc été développées pour la localisation *indoor*. Selon les cas applicatifs, la localisation *indoor* pourra se faire en temps réel, c'est-à-dire qu'un utilisateur pourra connaître sa position courante à chaque instant. D'autres cas d'usage ne nécessiteront pas de connaître la position en temps réel, et s'attacheront à estimer une trajectoire en temps différé, c'est-à-dire d'obtenir la meilleure estimation de la trajectoire parcourue par un utilisateur après que ce dernier ait parcouru l'ensemble de son trajet. Les travaux de cette thèse s'inscrivent spécifiquement dans le cadre de l'estimation d'une trajectoire en temps différé à partir d'un *smartphone*.

Dans ce premier chapitre introductif, seront dans un premier temps développées différentes applications de l'estimation en temps différé. Dans un second temps, nous précisons plus spécifiquement les mesures, ou données d'entrées, qui permettront de définir le cadre de notre étude. Enfin, une présentation de la théorie de l'estimation permettra d'organiser et de proposer le plan de recherche de nos travaux.

1.1 Enjeux de la géolocalisation *indoor* en temps différé

L'estimation d'une position s'effectue en traitant un ensemble d'informations, notamment issues de différentes mesures qui seront captées durant le trajet d'un individu. Si on s'intéresse à la position d'un individu à un instant t pour l'approche en temps réel, l'ensemble des informations disponibles auront été récoltées entre l'instant initial t_0 où le traitement aura démarré et cet instant t . En effet, les informations du futur, donc ultérieures à l'instant t ne sont pas connues en avance. En revanche, l'estimation en temps différé a accès à toutes les informations disponibles, depuis l'instant initial, jusqu'à l'instant final $t_{fin} > t$. À ce même instant t , on a accès à une quantité d'information plus importante qu'en temps réel et il devient possible d'augmenter la précision pour l'estimation en temps différé comparativement au temps réel. L'approche en temps différé est moins fournie dans la littérature par rapport à l'approche en temps réel. En effet, pour la plupart des cas d'utilisation, il est nécessaire d'avoir accès à une estimation en direct de sa position. Néanmoins, certaines applications n'ont pas besoin d'estimation en temps réel, et profitent donc d'une estimation en temps différé.

1.1.1 Un historique personnel de déplacement *indoor*

Un historique de déplacement correspond aux trajets qui ont été effectués par un utilisateur durant une certaine durée et répertorie les lieux qui ont été visités. Ces systèmes historiques sont déjà accessibles en environnement *outdoor* via *Google* ou encore *Waze* qui est une application mobile de navigation GPS, MOBILE [2008]. Ceci pourrait être également développé dans des environnements *indoor*. Différents cas d'usage sont alors possibles pour un utilisateur souhaitant un historique personnel de déplacement. Par exemple, lors d'une visite d'un grand musée sur plusieurs jours, il peut être intéressant d'avoir une vision assez précise des galeries visitées durant la journée. Un autre exemple concerne l'application *Google Now* GOOGLE [2012] qui propose une assistance intelligente personnalisée. En effet, cette application traite l'historique des déplacements afin d'adapter le service à l'utilisateur, et de proposer des conseils et services appropriés.

1.1.2 Le *crowdsourcing* comme application de l'estimation en temps différé

Une autre utilisation qui ne nécessite pas d'estimation en temps réel concerne les différentes problématiques de *crowdsourcing*, ou données de foule. Le *crowdsourcing* est un principe de production participative qui consiste à utiliser un très grand nombre d'utilisateurs différents qui apportent chacun des éléments ou informations afin de résoudre différentes problématiques. Par exemple, le projet international *OpenStreetMap MAP* [2004], lancé en 2004, a vu un million de contributeurs alimenter la production de cartes géographiques précises et actualisées en environnement *outdoor*. Dans cet exemple, chaque utilisateur pouvait y contribuer en ajoutant diverses informations telles que l'ajout de route résidentielle, la fermeture d'une route pour travaux, l'ajout de nouveaux immeubles ou services mais encore la disponibilité de certains lieux aux personnes en fauteuil roulant.

Comme représenté dans le schéma de la figure 1.1, l'utilisateur pourra parcourir une trajectoire qui sera estimée en temps différé, puis envoyée à un serveur pour l'enregistrer dans sa base de données. Par la suite, un traitement des données permettra de mener différentes analyses à partir de l'ensemble des trajectoires enregistrées. Par ailleurs, l'estimation en temps différé n'a pas besoin d'être embarquée, c'est-à-dire que les opérations ne sont pas nécessairement effectuées directement par le *smartphone*. Ces opérations peuvent être effectuées sur un serveur auquel auront été envoyées les différentes mesures du capteur du téléphone. Ce point présente un avantage important de l'estimation en temps différé comparativement à l'estimation en temps réel. Les contraintes de coût d'opération, de mémoire ou encore de consommation de batterie d'un *smartphone* sont largement réduites.

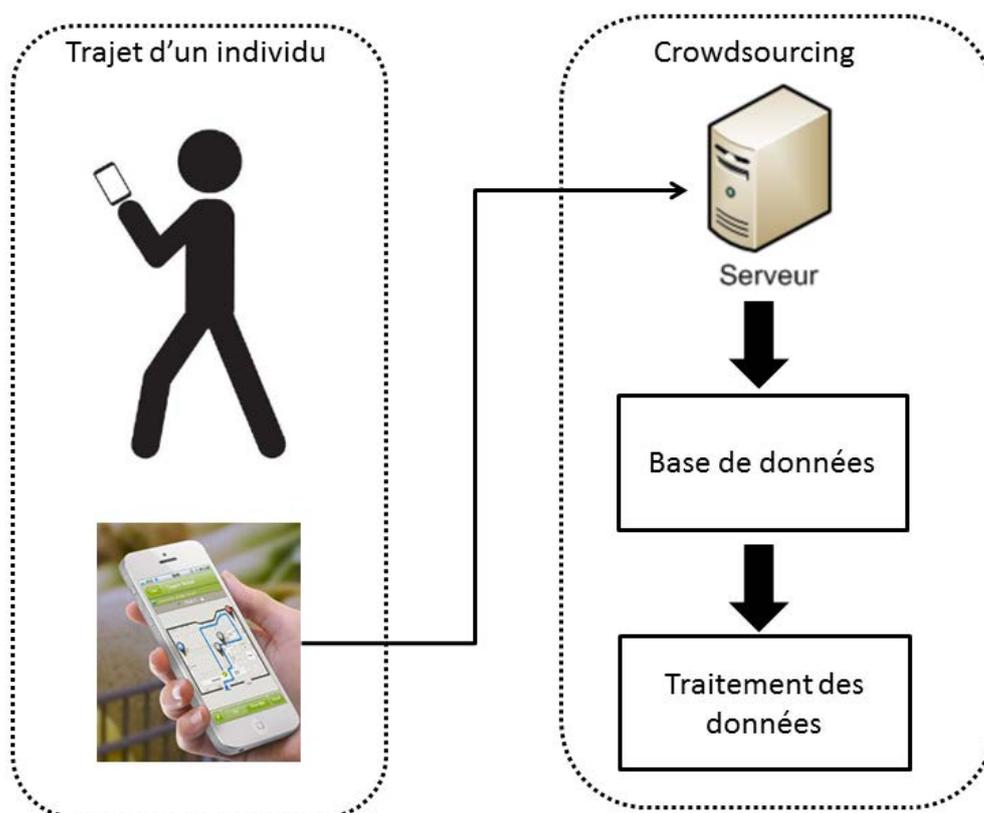


FIGURE 1.1 – Schéma de la mise en oeuvre du *crowdsourcing* appliqué à la géolocalisation *indoor*

A partir d'un grand nombre de trajectoires, il devient possible de développer par le *crowdsourcing* différentes applications et études comme il est décrit dans [GANTI et collab. \[2011\]](#). On peut par exemple mesurer certaines conditions environnementales d'un bâtiment, en considérant des cartes sonores comme ceci a été proposé dans [RANA et collab. \[2010\]](#) en *outdoor*. Également, [STEVENS et D'HONDT \[2010\]](#) discutent de la faisabilité d'intégrer dans les téléphones des capteurs permettant d'évaluer la pollution de l'air. Les campagnes de mesure de pollution deviendraient bien plus simples. Par ailleurs, comme il sera décrit dans les parties 2.1.1.1, le *crowdsourcing* permet également de déterminer des cartes de puissances wifi. Enfin, la mise en oeuvre du *crowdsourcing* à l'intérieur d'un bâtiment permet de développer des analyses sur le déplacement des individus, et de produire des cartes de fréquentation d'un bâtiment, notamment durant les heures de pointe, qui rendraient les déplacements plus commodes par exemple en ajoutant des indications spécifiques aux horaires sensibles. Par ailleurs, ces analyses de fréquentation pourraient également être directement utiles d'un point de vue marketing puisqu'une boutique pourrait avoir l'information de la fréquentation devant son enseigne, voire de son comportement, comme traité dans [ZENG et collab. \[2015\]](#). Si l'utilisation du *crowdsourcing* tend à prendre plus d'importance, comme décrit dans [CHILIPIREA et collab. \[2016\]](#), des questions de confidentialité émergeront. Certaines méthodes de protection des données telles que [YAO \[1982\]](#) ou encore [SWEENEY \[2002\]](#) rendent toujours possible le stockage des données des différents utilisateurs ainsi que leur analyse tout en garantissant le respect de l'intimité des utilisateurs. Par exemple, un bruit de moyenne nulle est ajouté à l'ensemble des données. Ce bruit n'implique pas de biais dans l'analyse des données, notamment lorsque l'on s'intéresse à un calcul de moyenne. Ainsi, la confidentialité des mesures est garantie puisque celles-ci sont aléatoirement perturbées.

1.2 Application au *smartphone*

Le cadre de cette thèse est centré sur l'estimation en temps différé à partir d'un appareil mobile, tel qu'un *smartphone*. En effet, l'évolution des téléphones permet de répondre à un nombre croissant d'utilisations. Les *smartphones* deviennent alors des outils de mesure très pratiques parce que très répandus. En effet, selon LIU [2015], en 2015, 61% de la population mondiale possède un *smartphone*. De plus, ces dernières décennies ont vu la prolifération des capteurs systèmes électromécaniques miniaturisés (MEMs) BELL et collab. [2005] dont la taille est parfois inférieure à 1mm ce qui a permis leur intégration dans les *smartphones*. La consommation classique de ces capteurs est de l'ordre du milli-ampère. On décrit dans la figure 1.2 l'ensemble des différents capteurs et antennes généralement présents dans un *smartphone*.

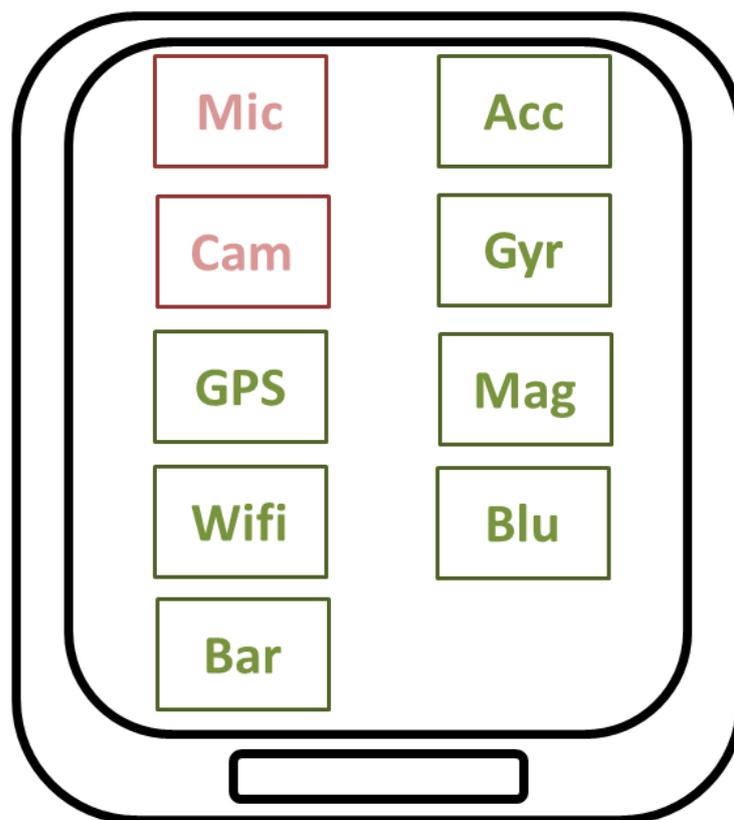


FIGURE 1.2 – Illustration des capteurs et antennes présents dans un téléphone. Cam=caméra, Mic=microphone, Bar=baromètre, Acc = accéléromètre, Gyr=gyroscope, Mag=magnétomètre, Blu=bluetooth. En vert sont représentés les éléments d'intérêt pour nos travaux.

Seuls la caméra et le microphone ne seront pas considérés dans nos travaux. Différents systèmes de navigation traitent des mesures de la caméra, et permettent une précision parfois de l'ordre du cm, mais leurs mises en oeuvre sont fastidieuses. MAUTZ et TILCH [2011] proposent une revue de certaines approches optiques de localisation *in-door*.

1.2.1 Localisation inertielle

1.2.1.1 Description des capteurs MEMs

Accéléromètre

L'accéléromètre est un capteur permettant de mesurer une force, appelée force spécifique, qui est la somme de la pesanteur et de l'accélération propre du mobile. Ces accélérations sont mesurées en $m.s^{-2}$, ou en N ou en G. Dans un téléphone portable, il permet notamment de mesurer l'inclinaison du téléphone et est surtout utilisé pour déterminer si le téléphone est en position portrait ou paysage.

Gyroscope

Le gyroscope 3 axes quantifie les rotations effectuées autour de 3 axes. En effet, il mesure la vitesse angulaire en $rad.s^{-1}$. Le gyroscope repose principalement sur une observation des forces de Coriolis. Une dérive est observable lorsqu'une estimation d'un angle est effectuée à partir de l'intégration des mesures du gyroscope. Différentes sources peuvent justifier cette dérive notamment à cause d'un biais lié au capteur de l'ordre de plusieurs degrés par minute. Il est important d'évaluer ce biais en mesurant les vitesses angulaires lorsque le mobile est à l'arrêt.

Magnétomètre

Le magnétomètre 3 axes permet de mesurer le champs magnétique sur 3 axes. La valeur est donnée en Tesla. Il est notamment utilisé pour mesurer le champs magnétique terrestre et peut servir de boussole en estimant la direction du nord absolu. Le magnétomètre permet donc de déterminer le cap absolu du téléphone. Par ailleurs, en environnement *indoor*, certains éléments tels que des armoires métalliques ou des ascenseurs viennent perturber le champs magnétique. L'estimation du cap absolu est alors sensible à ces perturbations.

1.2.1.2 Présentation d'une trajectoire inertielle

Pour la géolocalisation, les mesures inertielles permettent une géolocalisation autonome, c'est-à-dire totalement portée par le mobile que l'on cherche à positionner. Les mesures inertielles sont les mesures qui font état des mouvements du mobile. Dans le cas spécifique de la localisation *indoor*, les différentes mesures de ces 3 capteurs sont traitées afin de pouvoir estimer une trajectoire inertielle qui sera basée sur le modèle de la marche d'un individu. L'accéléromètre permet de détecter chaque pas effectué par l'individu, ainsi que la longueur de ces pas. Une fusion entre l'accéléromètre, le gyroscope, et de façon optionnelle le magnétomètre, permet d'estimer le cap de déplacement à chaque pas. Dans la figure 1.3 a été représenté le déplacement du mobile après chaque pas par le cercle orange.

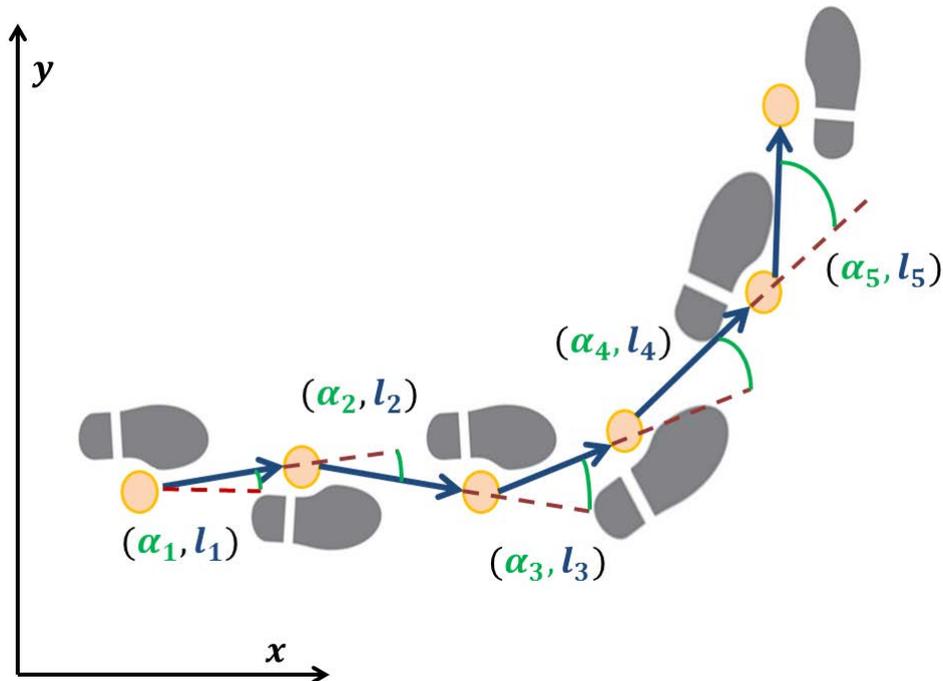


FIGURE 1.3 – Schéma de la paramétrisation de la trajectoire inertielle

Chaque pas effectué correspond à une nouvelle position de la trajectoire inertielle, et ces pas seront alors numérotés et indexés par l'indice temporel k . On note alors la longueur de pas l_k , chaque pas étant représenté par les flèches bleues dans la figure, et l'angle de pas α_k entre l'axe du pas précédent et le nouveau pas, représenté par les angles verts dans la figure 1.3. Si l'information du point de départ d'un individu est disponible ainsi que du déplacement de chacun de ces pas, il devient possible de déterminer son positionnement relatif. Comme représenté dans la figure 1.4, les informations de la détection de pas, ainsi que l'angle et la longueur de chaque pas sont obtenues à partir des différents capteurs inertiels. Les mesures sont reçues à un instant t , aux fréquences d'échantillonnage près. On notera respectivement $\mathbf{a}(t)$, $\omega(t)$ et $\mathbf{b}(t)$ les mesures en 3 dimensions reçues à l'instant t par l'accéléromètre, le gyroscope et la magnétomètre. Un traitement est effectué à partir de ces mesures et permet alors d'obtenir le couple (l_k, α_k) qui caractérise le pas qui a été effectué. Le détail du traitement sera décrit dans la partie 2.1.2 spécifique à l'estimation de la trajectoire inertielle. Par ailleurs, ce traitement des données brutes des capteurs MEMs n'ont pas fait l'objet de recherches supplémentaires durant cette thèse, l'estimation de la trajectoire inertielle reprend des algorithmes déjà développés dans la littérature et mis en oeuvre par le CEA. Ces travaux de thèse se sont alors directement servi des estimations de longueur de pas et d'angle qui vont constituer les nouvelles mesures d'entrée de notre problématique. Comme il sera décrit dans la partie 2.1.2, la trajectoire inertielle ne propose pas une estimation satisfaisante de la trajectoire parcourue par un individu. On utilise alors les mesures de puissances reçues (RSS) des signaux wifi et BLE.

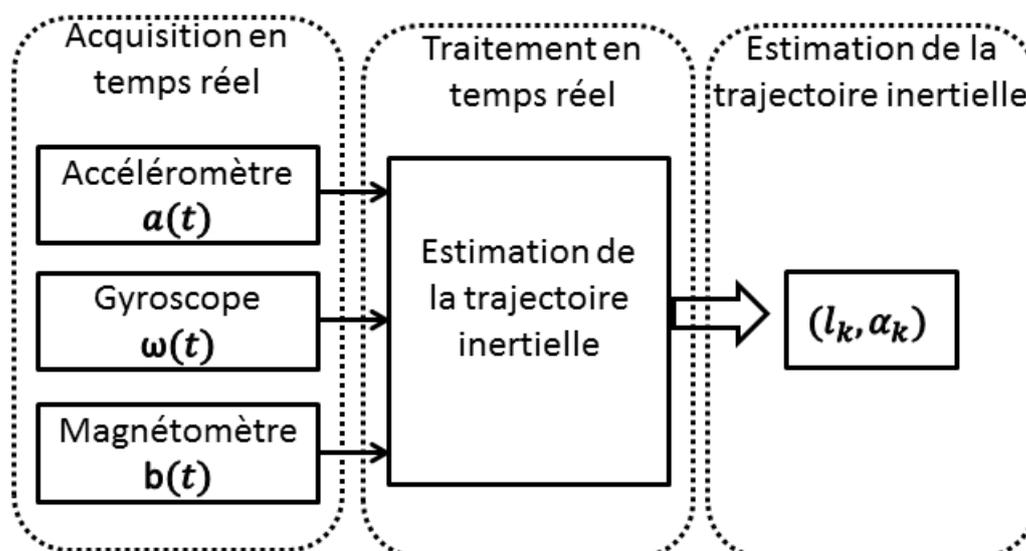


FIGURE 1.4 – Schéma de l'estimation de la trajectoire inertielle

1.2.2 Mesures de puissance

Les mesures RSS permettent de faire un lien entre la puissance reçue par le téléphone et la distance entre le téléphone et une balise d'émission, wifi ou bluetooth. Pour la suite, on considérera systématiquement que la position de chacune des balises est connue. Par ailleurs, on ne fera pas de distinction dans la forme du modèle de propagation entre les mesures wifi et bluetooth qui seront désignées comme étant les mesures RSS. Les mesures wifi ont l'avantage d'être présentes dans de grandes infrastructures publiques, telles que les aéroports, les gares ou encore les centres commerciaux, mais également dans des bâtiments privés tels qu'un bâtiment de bureaux. Les mesures bluetooth sont moins disponibles mais tendent à se populariser, notamment avec les modules BLE (*Bluetooth Low Energy*), qui permettent différentes applications, comme les dispositifs bluetooth permettant de retrouver ses clés lorsqu'elles sont égarées, pour une faible consommation d'énergie. Dans le cadre de la localisation *indoor*, l'utilisation des mesures bluetooth peut se faire relativement aisément en déployant un ensemble de balises dans le bâtiment. Enfin, la plupart des *smartphones* sont équipés d'un récepteur wifi ou bluetooth et rendent l'utilisation de ces mesures très commode.

1.2.3 Baromètre et mesures de pression pour la détection du changement d'étage

La localisation 2.5D consiste à savoir déterminer la position en 2D d'un individu sur le plan d'un étage, et également de savoir définir l'étage sur lequel ce dernier se situe. La détermination de l'étage, ou des changements d'étage, peut être effectuée par exemple en s'intéressant aux mesures de pression par un capteur MEMs présent dans certains appareils portables qui est le baromètre.

1.2.4 Mesure du GPS

La mesure du GPS est utilisée notamment lorsqu'un individu sort du bâtiment pour se déplacer en environnement *outdoor*.

1.2.5 Autres sources d'information

Notre problématique d'estimation sera également basée sur d'autres sources d'information en dehors du *smartphone*.

1.2.5.1 Plan du bâtiment

Le plan du bâtiment correspond à toutes les caractéristiques d'un bâtiment, c'est -à-dire le nombre d'étages, la position des murs, des portes, des cages d'escalier et des ascenseurs. Cette source d'information permet d'ajuster l'estimation de la trajectoire, comme il sera décrit dans la partie 3.1. Si ces plans ne sont pas systématiquement disponibles, en raison de question de propriété privée, dans notre cas, les plans constitueront une donnée d'entrée du problème.

1.2.5.2 Point de départ

Le point de départ de l'utilisateur est considéré comme une donnée connue, de même que l'étage. On peut considérer par exemple que le point de départ correspond à l'entrée d'un bâtiment, cette position pouvant être déterminée par une position GPS qui est suffisamment précise en environnement *outdoor* et qui permettra alors de déterminer l'entrée du bâtiment.

1.2.6 Définition de la problématique d'estimation en temps différé

Ces travaux de thèse considèrent donc le déplacement d'un individu dont on connaît le point de départ. Cet individu se déplace dans un environnement *indoor* dont on connaît les caractéristiques physiques telles que le plan du bâtiment ou encore la position des balises wifi et bluetooth. Cet individu se déplace en tenant un *smartphone* dans sa main, dans une position inconnue, et effectuera un trajet. Au cours de ce trajet, un ensemble de mesures, à partir des capteurs MEMs notamment, sont enregistrées afin d'être traitées en temps différé. On peut définir notre problématique. À partir de cet ensemble d'informations sur l'environnement, et des mesures qui ont été enregistrées en temps réel, comment estimer, en temps différé, le trajet qui a été parcouru par l'individu?

1.3 Cadre théorique de l'estimation

Il est nécessaire de définir un cadre théorique permettant de fusionner, en temps différé, ces différentes sources d'information. La variable d'intérêt de la géolocalisation *indoor* que l'on notera x_k à un instant k , correspond à la position d'un individu, mais également à certains paramètres tels que des biais que l'on décrira dans la partie 2.1.2 associés à la trajectoire inertielle. En temps différé, on cherche à estimer $x_{1:K} = (x_1, \dots, x_K)$. Cette estimation se fait à partir des différentes sources de mesures qui sont les mesures inertielles et que l'on appellera mesures IMU (*Inertial Measurement Unit*), ainsi que des mesures RSS ou encore des mesures du magnétomètre ou du baromètre. Pour la description du cadre théorique, on représentera par $y_{1:K}$ l'ensemble des mesures à chaque instant. On définit également une fonction d'observation $h()$ faisant le lien à chaque instant entre les mesures et le paramètre d'état que l'on cherche à estimer :

$$y_k = h(x_k, \eta_k) \tag{1.1}$$

où η_k correspond à un bruit généralement centré. A partir de ce modèle, on peut ainsi développer deux approches d'estimation : dans un premier temps, un modèle de régression paramétrique puis dans un second temps, une chaîne de Markov cachée et l'estimation d'une loi de lissage. Ces travaux de thèse s'insèrent dans une problématique de temps différé, néanmoins, la théorie de loi de lissage découle de la théorie du filtrage en temps réel, pour cette raison, une description de la théorie du filtrage sera également menée. Enfin, on décrira également l'ensemble des métriques permettant d'évaluer nos estimations.

1.3.1 Modèle de régression paramétrique

Un modèle de régression paramétrique définit une famille de loi $(P_\theta)_{\theta \in \Theta}$ où θ est un paramètre à estimer de dimension finie. L'objet de ce modèle, dans notre cas, étant de fusionner les différentes sources de mesures afin d'estimer l'état x_k , on décompose dans un premier temps le vecteur de mesure $y_{1:K}$ en deux vecteurs de mesures différents, $(u_{1:K})$ et $(v_{1:K})$ en notant :

$$y_k = (u_k, v_k) \quad (1.2)$$

On définit alors la fonction paramétrique suivante :

$$x_k = T_\theta(v_k) \quad (1.3)$$

En considérant un bruit additif η_k , ainsi qu'un modèle d'observation $g()$ ne dépendant alors que des mesures u_k , on obtient alors :

$$u_k = g(T_\theta(v_k)) + \eta_k \quad (1.4)$$

Pour cette approche, la fonction $h_\theta() = g(T_\theta())$ permet de définir un lien entre les mesures u_k , les mesures v_k et le paramètre θ à partir d'une composition de fonction. Le modèle paramétrique étant défini, il s'agit ensuite de proposer un estimateur du paramètre de recherche θ à partir de l'ensemble des mesures $(u_{1:K}, v_{1:K})$. Toutes les mesures étant utilisées, cette approche correspond bien à une approche en temps différé. Si on considère une estimation $\hat{\theta}$, on peut proposer une estimation de la position du mobile à chaque instant k :

$$\hat{x}_k = T_{\hat{\theta}}(v_k) \quad (1.5)$$

Dans la suite seront uniquement présentés l'estimateur du maximum de vraisemblance et l'estimateur du maximum *a posteriori* qui restent les estimateurs utilisés les plus courants, notamment pour des problèmes de modélisation paramétriques n'étant pas trop complexes.

1.3.1.1 Estimateur du maximum de vraisemblance

Les différentes erreurs de mesure $\eta_{1:K}$ sont considérées comme étant indépendantes. La vraisemblance du modèle correspond au produit des vraisemblances. Si chacune de ces erreurs suit une loi de densité p_k , alors on définit, une fonction de densité $f(u_k, v_k|\theta)$:

$$f(u_k, v_k|\theta) = p_k(u_k - g[T_\theta(v_k)]) \quad (1.6)$$

et la vraisemblance du modèle est alors donnée par :

$$L(\theta|u_{1:K}, v_{1:K}) = \prod_{k=1}^K f(u_k, v_k|\theta) \quad (1.7)$$

On peut définir un premier estimateur qui est le maximum de vraisemblance que l'on notera θ_{MLE} pour *Maximum Likelihood Estimator* et qui résout le problème d'optimisation suivant :

$$\theta_{\text{MLE}} = \underset{\theta \in \Theta}{\operatorname{argmax}} L(\theta | u_{1:K}, v_{1:K}) \quad (1.8)$$

L'estimateur du maximum de vraisemblance présente de nombreuses propriétés statistiques, il est notamment asymptotiquement sans biais et asymptotiquement efficace. Un estimateur efficace signifie qu'il satisfait à l'égalité de Cramer-Rao, c'est-à-dire que parmi les estimateurs non-biaisés, cet estimateur efficace présentera la plus petite variance possible. Cet estimateur est très couramment utilisé, mais présente notamment le défaut de ne pas être numériquement robuste, particulièrement pour des échantillons de petites tailles [ROBERT \[2006\]](#).

1.3.1.2 Estimateur du maximum *a posteriori*

Selon la connaissance du problème, il est possible de définir une loi *a priori* de densité $\pi(\theta)$ sur le paramètre :

$$\theta \sim \pi(\theta) \quad (1.9)$$

Alors, on peut déterminer une loi *a posteriori* $\pi(\theta | u_{1:K}, v_{1:K})$, qui contiendra l'ensemble de l'information disponible sur l'*a priori* et également sur les observations. La formule de Bayes permet d'obtenir une expression de cette loi *a posteriori* :

$$\pi(\theta | u_{1:K}, v_{1:K}) = \frac{f(u_{1:K}, v_{1:K} | \theta) \pi(\theta)}{f(u_{1:K}, v_{1:K})} \quad (1.10)$$

L'estimateur du maximum *a posteriori*, ou estimateur MAP, θ_{MAP} est défini comme étant le mode de cette loi *a posteriori*.

$$\theta_{\text{MAP}} = \underset{\theta \in \Theta}{\operatorname{argmax}} L(\theta | u_{1:K}, v_{1:K}) \pi(\theta) \quad (1.11)$$

Ici, la densité marginale liée aux mesures $f(u_{1:K}, v_{1:K})$ ne dépendant pas de θ , il est possible de s'en affranchir dans la résolution du problème d'optimisation. Cet estimateur présente les mêmes propriétés asymptotiques qu'un maximum de vraisemblance, mais permet d'intégrer une information *a priori* et d'être stable pour des échantillons de taille réduite, [ROBERT \[2006\]](#).

1.3.2 Théorie du filtrage et du lissage

La théorie du filtrage consiste à estimer un état dynamique inconnu, appelé état caché. Dans notre cas, cet état caché correspond à la position d'un individu $x_{1:K}$. Cet état est bien dynamique puisqu'il correspond au déplacement pédestre d'un utilisateur. La théorie du filtrage permet plus précisément de faire le lien en temps réel entre un modèle d'état et un modèle d'observation.

1.3.2.1 Modèle d'état

Le modèle d'état correspond à la connaissance *a priori* sur la dynamique à chaque instant de l'état. On suppose que cet état suit une chaîne de Markov d'ordre 1, c'est-à-dire que la dépendance de l'état x_{k+1} sur les états aux instants passés $x_{1:k}$ est contenue dans l'état x_k .

Ce modèle d'état est caractérisé par une fonction d'état et par une loi de probabilité sur l'erreur η_k représenté dans l'équation 1.12 :

$$x_k = f(x_{k-1}, \eta_k) \quad (1.12)$$

Par ailleurs, il est nécessaire de définir une loi de distribution initiale f_0 , correspondant au point de départ x_0 .

$$x_0 \sim f_0 \quad (1.13)$$

1.3.2.2 Modèle d'observation

Le modèle d'observation décrit la relation entre l'état et les mesures que l'on notera $y_{1:K}$. De plus, on pose l'hypothèse que les erreurs d'observation sont indépendantes conditionnellement à la connaissance de l'état. On peut alors définir un modèle d'observation en notant γ_k l'erreur d'observation à l'instant k , on a :

$$y_k = h(x_k, \gamma_k) \quad (1.14)$$

1.3.2.3 Loi de filtrage

La loi de filtrage correspond à l'approche en temps réel puisque l'inférence sur la position x_k est basée sur les mesures observées jusqu'à l'instant k . La loi de filtrage de cet état caché x_k est alors définie par la loi *a posteriori* $P(x_k | y_{1:k})$. L'enjeu de la théorie du filtrage consiste à pouvoir déterminer cette loi de filtrage qui combine le modèle d'état ainsi que le modèle d'observation. Dans le cas simple où les fonctions $f()$ et $h()$ sont linéaires, et lorsque les bruits η_k et γ_k sont gaussiens, une résolution optimale au sens de l'erreur quadratique moyenne existe, il s'agit du filtre de Kalman [KALMAN et collab. \[1960\]](#) qui caractérise cette loi de filtrage par un processus gaussien qui est totalement défini par son espérance \hat{x}_k et sa matrice de covariance P_k .

$$x_k | y_{1:k} \sim \mathcal{N}(\hat{x}_k, P_k) \quad (1.15)$$

Certains problèmes considèrent des fonctions non-linéaires. Ils existent alors certaines approches basées sur des approximations linéaires, permettant de revenir au cas linéaire et gaussien, telles que le filtre de Kalman étendu [RISTIC et collab. \[2004\]](#). Toutefois, il subsiste des cas d'application bien plus complexes, notamment lorsque les fonctions $f()$ et $h()$ ne peuvent pas être linéarisées. Par exemple, dans le cadre de la navigation *indoor*, il est possible de considérer l'information des murs comme une contrainte au déplacement de l'individu. Cette contrainte ne permet pas de linéariser le problème et nécessitera d'utiliser une méthode d'approximation numérique, comme le filtre particulaire par exemple.

1.3.2.4 Loi de lissage

La loi de lissage pour un état caché x_k est définie par la loi *a posteriori* $P(x_k | y_{1:K})$, où $K > k$ est un instant futur. La loi de lissage formalise l'approche en temps différé, puisqu'il s'agit d'utiliser les mesures reçues après un instant k pour ensuite ajuster l'estimation de x_k . Savoir déterminer cette loi de lissage est complexe lorsque les modèles ne sont ni linéaires ni gaussiens. L'expression de la loi de lissage $P(x_k | y_{1:K})$ est obtenue en marginalisant la loi jointe qui se décompose de la façon suivante, en intégrant selon x_{k+1} :

$$P(x_k | y_{1:K}) = \int P(x_k, x_{k+1} | y_{1:K}) dx_{k+1}$$

une application de la formule de Bayes permet de développer :

$$P(x_k|y_{1:K}) = \int P(x_{k+1}|y_{1:K})P(x_k|x_{k+1}, y_{1:K})dx_{k+1}$$

et par hypothèse, les observations $y_{1:K}$ sont indépendantes conditionnellement aux états $x_{1:K}$, on peut réécrire et en appliquant à nouveau la formule de Bayes, on obtient que :

$$\begin{aligned} P(x_k|y_{1:K}) &= \int P(x_{k+1}|y_{1:K})P(x_k|x_{k+1}, y_{1:k})dx_{k+1} \\ &= P(x_k|y_{1:k}) \int \frac{P(x_{k+1}|y_{1:K})f(x_{k+1}|x_k)}{P(x_{k+1}|y_{1:k})}dx_{k+1} \end{aligned} \quad (1.16)$$

Selon l'équation 1.16, la loi de lissage dépend directement de la loi de filtrage.

1.3.3 Métriques d'évaluation

L'évaluation des méthodes de navigation *indoor* est très sensible au choix de la métrique mais dépend globalement du cas d'application. Par exemple, si l'objet de la localisation est uniquement de déterminer la position finale d'une trajectoire, alors le calcul de la distance entre l'estimation et cette position finale peut suffire. L'erreur peut être également de nature différente, par exemple dans PULKKINEN et VERWIJNEN [2015], l'erreur est binaire et correspond à la détermination correcte ou non de la salle dans laquelle se trouve l'utilisateur.

Dans notre cas, nous sommes intéressés par l'estimation en temps différé de la trajectoire parcourue. On définira deux métriques différentes et complémentaires : la valeur de la rmse (*Root Mean Square Error*) donnée par l'équation 1.17 et le ratio de l'estimation de la trajectoire qui est inférieur à une certaine distance d_m , qui dans notre cas sera égale à $5m$, donnée par l'équation 1.18. Cette erreur sera alors notée %d5. Le choix de la distance de $5m$ correspond à une valeur arbitraire qui correspond à un cas d'utilisation classique. En effet, être localisé à moins de $5m$ à l'intérieur d'un bâtiment correspond à une distance tout à fait raisonnable, par exemple, pour déterminer la salle dans laquelle se trouve l'utilisateur. De plus, comme il sera décrit dans la partie 5.1.2, le protocole qui permet de définir la trajectoire de référence met en oeuvre une interpolation linéaire entre différents points de référence. Cette interpolation implique que la trajectoire de référence ne correspond pas exactement à la trajectoire réellement parcourue, mais l'écart entre ces trajectoires reste raisonnablement faible pour l'évaluation de nos approches. L'existence de cet écart justifie notamment que considérer une distance d_m trop petite n'est plus pertinent. Finalement, si on considère la trajectoire de référence $\mathbf{x}_k = (x_k, y_k)$ que l'on compare avec une trajectoire estimée $\hat{\mathbf{x}}_k = (\hat{x}_k, \hat{y}_k)$ pour tout instant $1 \leq k \leq K$, on définit alors :

$$rmse(\mathbf{x}_{1:K}, \hat{\mathbf{x}}_{1:K}) = \left[\frac{1}{K} \sum_{k=1}^K (x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2 \right]^{1/2} \quad (1.17)$$

et

$$\%d5(\mathbf{x}_{1:K}, \hat{\mathbf{x}}_{1:K}) = \frac{1}{K} \sum_{k=1}^K \mathbb{1}\{d\{\mathbf{x}_k, \hat{\mathbf{x}}_k\} < 5\} \quad (1.18)$$

La valeur du rmse permet d'obtenir une valeur globale de l'erreur de la trajectoire, mais ne permet pas de clairement savoir si l'estimation a divergé, ni pour combien de temps. De même, l'erreur %d5 ne permet pas de faire la distinction entre une erreur de $6m$ ou

une erreur de 20m. Ces deux erreurs sont donc complémentaires et seront utilisées simultanément pour l'évaluation des méthodes. Par ailleurs, ces choix restent simples et tendent uniquement à illustrer le gain obtenu en développant les différentes améliorations en temps différé par rapport à une approche en temps réel.

1.4 Plan de thèse

Dans le chapitre 2, on présentera un état de l'art sur les technologies utilisant des mesures IMU ou RSS. Puis, on développera un modèle de régression paramétrique fusionnant ces deux sources de mesure. Ce chapitre ne tiendra pas compte de la connaissance des plans du bâtiment, la dérive de la trajectoire inertielle sera uniquement corrigée à partir des mesures RSS. L'approche en temps différé permettra dans un premier temps de proposer une paramétrisation par morceaux de la trajectoire inertielle selon un faible nombre de paramètres, mais également de proposer un modèle adaptatif de propagation d'onde pour les mesures RSS. Finalement, l'estimation des paramètres sera obtenue par maximum *a posteriori*.

Le chapitre 3 intégrera ensuite la connaissance des plans du bâtiment. Une revue de la littérature décrira les différentes formalisations permettant d'intégrer l'information de la carte, notamment le filtre particulaire. Un filtre particulaire, dont l'implémentation n'intègre pas ces travaux de thèse, permettra d'estimer une trajectoire en temps réel. Un lissage particulaire sera développé afin de corriger les potentielles discontinuités inhérentes au temps réel. Un maximum *a posteriori* de cette loi de lissage sera également estimé par l'algorithme de Viterbi. Ces deux approches seront appliquées sur des *clusters* estimés en temps réel permettant de synthétiser l'ensemble de l'information des particules simulées en temps réel.

Par la suite, le chapitre 4 développera deux heuristiques numériques en temps différé, pour améliorer l'estimation obtenue par un filtre particulaire en temps réel. Une première heuristique consiste à répéter plusieurs fois un filtre particulaire afin d'estimer différentes estimations et de conserver la meilleure réalisation. Cette section s'attachera à augmenter la variabilité du filtre, mais également à déterminer de façon non-supervisée la meilleure réalisation. Une seconde heuristique combinera la fusion RSS et IMU du chapitre 2 avec un filtre particulaire, afin de l'alimenter avec une trajectoire inertielle de meilleure qualité. Enfin, ce chapitre sera conclu par une stratégie de sélection de trajectoires. En se plaçant dans le cadre du *crowdsourcing*, il est possible de ne pas traiter les estimations de trajectoires jugées de mauvaise qualité. Un modèle d'apprentissage statistique permettant de discriminer les mauvaises estimations sera proposé.

Enfin, un chapitre 5 présentera, dans un premier temps, le protocole expérimental qui aura permis de construire la base de données des trajectoires. Les résultats intermédiaires qui seront présentés dans chacun des chapitres de développement pour juger de la qualité des méthodes seront mesurés sur cette base de données. Les différentes méthodes proposées durant cette thèse seront ensuite comparées entre elles dans ce chapitre de résultat, en terme de performance, mais également en terme de coût de calcul. Pour conclure, ces méthodes seront couplées et permettront de définir une approche finale de localisation *indoor* en temps différé.

1.5 Références

- BELL, D. J., T. J. LU, N. A. FLECK et S. M. SPEARING. 2005, «MEMS actuators and sensors : observations on their performance and selection for purpose», *Journal of Micro-mechanics and Microengineering*, vol. 15, n° 7, p. S153. URL <http://stacks.iop.org/0960-1317/15/i=7/a=022>. 5
- CHILIPIREA, C., A.-C. PETRE et C. DOBRE. 2016, «Big data uses in crowd based systems», dans *Resource Management for Big Data Platforms*, Springer, p. 441–459. 4
- GANTI, R. K., F. YE et H. LEI. 2011, «Mobile crowdsensing : current state and future challenges», *IEEE Communications Magazine*, vol. 49, n° 11. 4
- GOOGLE. 2012, «Google now», <http://www.googlenow.fr/google-now/>. Accessed : 2018-01-05. 3
- KALMAN, R. E. et collab.. 1960, «A new approach to linear filtering and prediction problems», *Journal of basic Engineering*, vol. 82, n° 1, p. 35–45. 12
- KLEPEIS, N. E., W. C. NELSON, W. R. OTT, J. P. ROBINSON, A. M. TSANG, P. SWITZER, J. V. BEHAR, S. C. HERN et W. H. ENGELMANN. 2001, «The national human activity pattern survey (nhaps) : a resource for assessing exposure to environmental pollutants», *Journal of Exposure Science and Environmental Epidemiology*, vol. 11, n° 3, p. 231. 2
- LIU, C. 2015, «Worldwide internet and mobile users : emarketer’s updated estimates for 2015», cahier de recherche, eMarketer. 5
- MAP, O. S. 2004, «Open street map», <http://openstreetmap.fr/>. Accessed : 2017/12/25. 3
- MAUTZ, R. et S. TILCH. 2011, «Survey of optical indoor positioning systems», dans *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, IEEE, p. 1–7. 5
- MOBILE, W. 2008, «Waze», <https://www.waze.com/fr/>. Accessed : 2018-01-05. 3
- PULKKINEN, T. et J. VERWIJNEN. 2015, «Evaluating indoor positioning errors», dans *Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, IEEE, p. 167–169. 13
- PURICER, P. et P. KOVAR. 2007, «Technical limitations of gnss receivers in indoor positioning», dans *Radioelektronika, 2007. 17th International Conference*, IEEE, p. 1–5. 2
- RANA, R. K., C. T. CHOU, S. S. KANHERE, N. BULUSU et W. HU. 2010, «Ear-phone : an end-to-end participatory urban noise mapping system», dans *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ACM, p. 105–116. 4
- RISTIC, B., S. ARULAMPALAM et N. GORDON. 2004, «Beyond the Kalman filter», *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, n° 7, p. 37–38. 12
- ROBERT, C. 2006, *Le choix bayésien : Principes et pratique*, Springer Science & Business Media. 11
- STEVENS, M. et E. D’HONDT. 2010, «Crowdsourcing of pollution data using smartphones», dans *Workshop on Ubiquitous Crowdsourcing*. 4

- SWEENEY, L. 2002, «k-anonymity : A model for protecting privacy», *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, n° 05, p. 557–570. 4
- YAO, A. C. 1982, «Protocols for secure computations», dans *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, IEEE, p. 160–164. 4
- ZENG, Y., P. H. PATHAK et P. MOHAPATRA. 2015, «Analyzing shopper's behavior through wifi signals», dans *Proceedings of the 2nd workshop on Workshop on Physical Analytics*, ACM, p. 13–18. 4

Chapitre 2

Modèle de régression paramétrique à partir des mesures RSS et IMU

Sommaire

2.1 État de l'art	18
2.1.1 Méthode de localisation par les mesures RSS	18
2.1.2 État de l'art de la géolocalisation par mesure inertielle	21
2.1.3 Fusion des mesures RSS avec l'approche inertielle	25
2.1.4 Synthèse de l'état de l'art	27
2.2 Fusion en temps réel RSS et IMU	27
2.2.1 Modèle d'état	27
2.2.2 Modèle d'observation	29
2.2.3 Méthode d'estimation en temps réel	30
2.3 Fusion paramétrique des mesures RSS et IMU	31
2.3.1 Paramétrisations et définition des modèles	31
2.3.2 Estimateur du maximum <i>a posteriori</i>	39
2.3.3 Résolution numérique de l'optimisation	42
2.4 Résultats	45
2.4.1 Application numérique	45
2.4.2 Illustration sur une trajectoire sans rupture	45
2.4.3 Illustration sur une trajectoire avec des ruptures	47
2.4.4 Estimation en temps différé de la trajectoire réelle	48
2.4.5 Résultats sur la base de données	49
2.5 Conclusion	50
2.6 Références	51

2.1 État de l'art

Dans ce chapitre seront traitées les méthodes de géolocalisation à l'intérieur d'un bâtiment qui utilisent des mesures inertielles notées IMU et des mesures RSS. Dans un premier temps, un état de l'art présentera les méthodes d'estimation basées sur les mesures IMU, puis les méthodes utilisant des mesures RSS. Enfin, différentes approches de fusion de ces mesures seront décrites.

2.1.1 Méthode de localisation par les mesures RSS

Les différentes méthodes de géolocalisation utilisant des mesures RSS peuvent être séparées en deux catégories. Celles qui mettent en oeuvre des cartes de puissance wifi, appelées méthodes de *fingerprinting*, et celles qui appliquent un modèle de propagation d'ondes et permettent de calculer la puissance reçue en fonction de la distance entre l'émetteur et le point de mesure. Les méthodes utilisant un modèle de propagation d'ondes sont plus simples à mettre en oeuvre puisqu'elles nécessitent uniquement de connaître la position des balises wifi et bluetooth, ainsi que la puissance d'émission et de définir un modèle de propagation faisant le lien entre les puissances reçues et la distance du mobile à la balise. Néanmoins, ces approches sont largement tributaires de la qualité du modèle qui présente certains défauts dans un environnement *indoor*, qui seront discutés dans la partie 2.1.1.2, tels que les atténuations impliquées par les obstacles. Comparativement, les méthodes de *fingerprinting* estiment la position en utilisant une carte de puissance wifi. Si cette approche permet une meilleure précision, elle nécessite de savoir définir des cartes de puissance wifi, ainsi que de les mettre à jour régulièrement, ce qui est coûteux en ressources humaines et en temps.

2.1.1.1 Géolocalisation par méthodes de *fingerprinting*

La géolocalisation par méthodes de *fingerprinting* a été largement étudiée dans la littérature [HE et CHAN \[2016\]](#). Deux étapes se succèdent pour le *fingerprinting* : une première phase de construction de cartes, puis une seconde phase qui consiste à estimer la position.

Calibration de la carte de puissance

Une carte de puissance se définit à partir d'un certain nombre de points de référence, noté n_{RP} . On note également n_{AP} le nombre de balises ou points d'accès qui émettent les mesures wifi ou bluetooth. Pour chaque point de référence \mathbf{p}_i , pour $i \leq n_{RP}$, on peut associer une position en dimension 2, $\mathbf{pos}_i = (x_i, y_i)$ ainsi qu'un vecteur de puissance RSS associé à chacune des différentes balises d'émission, $\mathbf{r}_i = (r_i^1, r_i^2, \dots, r_i^{n_{act}})$. Ce vecteur de puissance est de dimension $n_{act}^i < n_{AP}$ où n_{act}^i est le nombre de balises pour lesquelles une puissance est reçue à la position (x_i, y_i) .

$$\mathbf{p}_i = [\mathbf{pos}_i \ \mathbf{r}_i] \quad (2.1)$$

Ces points de référence permettent d'associer à différentes positions du plan, les mesures de puissance RSS associées à chaque balise et définissent la carte de puissance que l'on notera RM :

$$RM = \{\mathbf{p}_1, \dots, \mathbf{p}_{n_{RP}}\} \quad (2.2)$$

La réalisation de ces cartes de puissance se trouve être très fastidieuse puisqu'elle nécessite d'explorer chaque point de référence, d'en mesurer la puissance, et d'enregistrer cette information, ce qui peut nécessiter quelques secondes par point de référence, pour assurer une certaine robustesse. La première phase de calibration est donc coûteuse en temps comme en ressources humaines. Par ailleurs, les cartes doivent être mises à jour dès lors que l'environnement évolue, ce qui alourdit encore davantage la mise en oeuvre d'une telle approche. Certaines approches tentent alors de réduire la complexité de la calibration, en optimisant la collecte de données, en considérant une interpolation en utilisant un modèle de propagation d'ondes [KLEPAL et collab. \[2007\]](#), [CHINTALAPUDI et collab. \[2010\]](#) ou encore par l'utilisation d'un graphe dit de Voronoï [LEE et HAN \[2012\]](#). Enfin, la construction de carte peut également être rendue automatique par *crowdsourcing*, [NIU et collab. \[2015\]](#) ou [WANG et collab. \[2016\]](#).

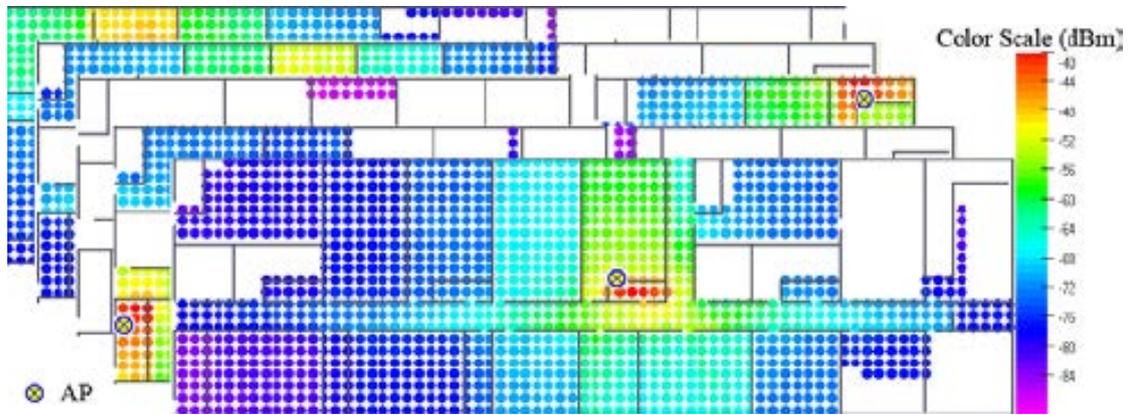


FIGURE 2.1 – Exemple d'une carte de puissance wifi tiré de [KLEPAL et collab. \[2007\]](#)

Estimation de la position par *fingerprinting*

A partir d'une carte calibrée, deux méthodes sont communément mises en oeuvre pour estimer la position d'un individu. Cette estimation peut se faire via une méthode déterministe ou par une méthode probabiliste [DAWES et CHIN \[2011\]](#). La méthode déterministe est basée sur la méthode des k plus proches voisins [BAHL et PADMANABHAN \[2000\]](#). Une mesure de similarité permet de déterminer les k points de référence dans la base de données, dont les mesures sont les plus proches des mesures $\mathbf{r}_{\text{mes}} = (r_{\text{mes}}^1, \dots, r_{\text{mes}}^{n_{\text{AP}}})$.

$$O_{m+1} = \left\{ \underset{i}{\operatorname{argmin}} d[\mathbf{r}_i; \mathbf{r}_{\text{mes}}], \mathbf{p}_i \notin N_m \text{ et } m+1 \leq k \right\} \quad (2.3)$$

où O_k est le k ème plus proche voisin et $N_k = \bigcup_{j=1}^k O_j$ correspond à l'ensemble des k points de la base les plus proches des mesures selon une mesure de similarité $d()$ telle que la distance euclidienne. Cet ensemble peut se construire par récurrence. Ensuite, une moyenne, pondérée ou non, sur la position associée à ces k vecteurs permet d'estimer la position associée aux mesures (figure 2.2).

$$\mathbf{pos}_{\text{est}} = \frac{1}{k} \sum_{\mathbf{p}_i \in N_k} \mathbf{pos}_i \quad (2.4)$$

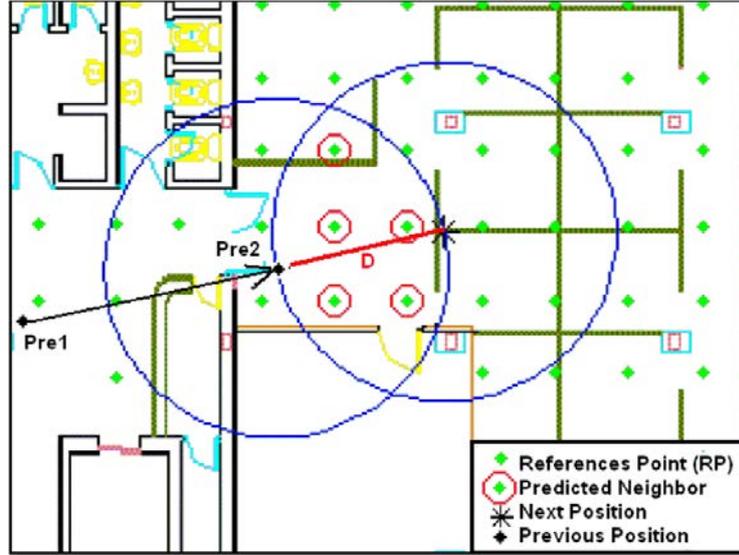


FIGURE 2.2 – Illustration de l'estimation par les k-plus proches voisins pour l'approche de *fingerprinting* déterministe tirée de KHODAYARI et collab. [2010]

L'estimation probabiliste MIROWSKI et collab. [2014] se fait via un maximum de vraisemblance, selon un vecteur de mesure \mathbf{r}_{mes} :

$$\mathbf{pos}_{\text{est}} = \underset{\mathbf{x} \in \mathbb{R}^2}{\operatorname{argmax}} \left[\prod_{i=1}^{n_{\text{AP}}} P(\mathbf{r}_{\text{mes}} | \mathbf{x}) \right] \quad (2.5)$$

où $P(\mathbf{r}_{\text{mes}} | \mathbf{x})$ correspond à la loi des mesures \mathbf{r}_{mes} sachant une position \mathbf{x} . Cette loi peut être estimée par des processus gaussiens, ou encore par l'utilisation d'un réseau bayésien.

2.1.1.2 Localisation via les modèles de propagation RSS

D'autres méthodes de localisation n'utilisent pas de carte de puissance et se basent uniquement sur un modèle de propagation qui fait le lien entre la puissance mesurée par le mobile et la distance entre la balise d'émission et le mobile LUO et collab. [2011]. Toutefois, l'utilisation de ces modèles en espace confiné présente certaines limites. En effet, il n'est pas possible de prédire l'atténuation subie par l'onde de façon précise compte tenu des obstacles, des réflexions multiples etc. L'utilisation unique des modèles de propagation d'ondes est donc discutable HEURTEFEUX et VALOIS [2012]. Néanmoins, en considérant un nombre de points d'accès important, les mesures RSS restent exploitables. Le modèle de propagation d'ondes suivant est communément utilisé FRIIS [1971] :

$$P_{\text{mes}} = P_{tr} + G_t + G_r + 10\alpha_{\text{RSS}} \log \left[\frac{\lambda}{4\pi\sqrt{(x-xb)^2 + (y-yb)^2}} \right] + \eta_z \quad (2.6)$$

où P_{mes} (dBm) est la puissance mesurée, $\begin{pmatrix} xb \\ yb \end{pmatrix}$ correspond à la position connue de la balise d'émission, P_{tr} (dBm) est la puissance de transmission, G_t (dB) et G_r (dB) les gains d'émission et de réception des antennes, λ (m) représente la longueur d'onde, α_{RSS} un facteur sans dimension représentant la perte de propagation et η_z un bruit gaussien de variance σ_z^2 .

De même que pour les méthodes de *fingerprinting*, pour estimer une position, un vecteur de mesures \mathbf{r}_{mes} est associé à un ensemble n_{AP} de balises, dont les positions sont connues. Ce modèle permet d'obtenir, à partir des mesures, une distance du mobile à chaque balise. L'estimation de la position peut alors se faire via un estimateur du maximum de vraisemblance [SUGANO et collab. \[2006\]](#) ou encore via une méthode de trilatération [HIGHTOWER et collab. \[2000\]](#) comme dans la figure 2.3), ou [MAZUELAS et collab. \[2009\]](#) qui effectuent un ajustement en temps réel du modèle sur chaque balise.

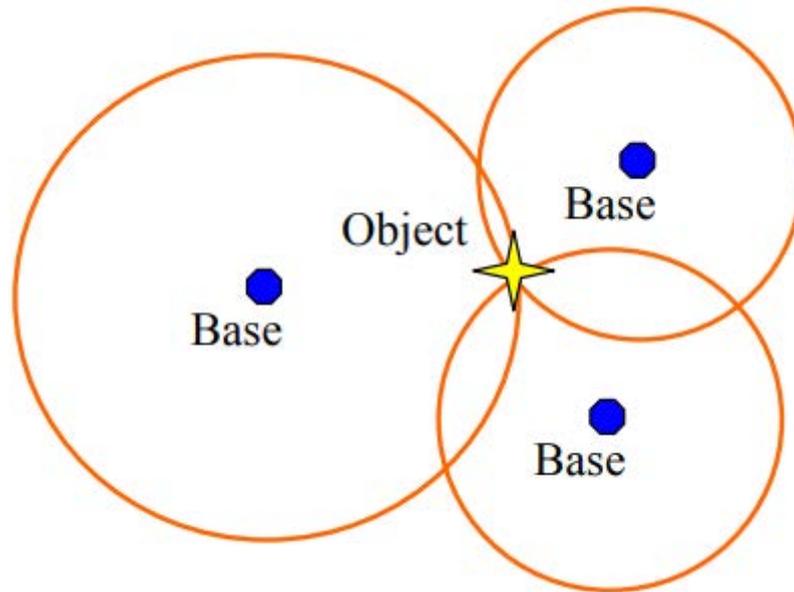


FIGURE 2.3 – Illustration de l'estimation de la trilatération tirée de [HIGHTOWER et collab. \[2000\]](#)

2.1.2 État de l'art de la géolocalisation par mesure inertielle

La navigation inertielle est une technique de navigation autonome qui utilise des mesures d'accélération et de vitesse angulaire afin de déterminer un déplacement, en partant d'une position qui est connue [WOODMAN \[2007\]](#). Au-delà de la navigation à l'intérieur d'un bâtiment, la navigation inertielle trouve beaucoup d'applications dans le domaine de l'aviation ou de la navigation sous-marine notamment. Classiquement, comme décrit dans la partie 1.2.1.1, les différents capteurs utilisés sont l'accéléromètre, le gyroscope ou encore le magnétomètre. Rigoureusement, ces capteurs sont intégrés pour résoudre les équations du mouvement. Toutefois, la qualité de ces capteurs dans les *smartphones* n'est pas suffisamment précise pour estimer la position d'un individu, il est alors nécessaire de baser l'estimation inertielle d'un individu sur un modèle de marche, dont on estime la longueur et l'angle [BEAUREGARD et HAAS \[2006\]](#) comme présenté dans la partie 1.2.1.2. Cette estimation est appelée *Pedestrian Dead Reckoning* (PDR). Une description de l'ensemble des mesures inertielles est reprise dans [MAUTZ \[2012\]](#). On obtient la formalisation suivante :

$$\begin{cases} x_k = x_{k-1} + l_k \cos(\alpha_k) \\ y_k = y_{k-1} + l_k \sin(\alpha_k) \end{cases} \quad (2.7)$$

où k indice chaque instant où un pas a été détecté, (x_k, y_k) est la position en coordonnées cartésiennes du mobile après le pas k , et l_k et α_k représentent la longueur et l'angle de ce pas. La navigation inertielle est alors basée sur 3 étapes successives :

1. Détection de pas

2. Estimation de la longueur de pas
3. Estimation du cap d'orientation

2.1.2.1 Algorithme de détection de pas

La plupart des méthodes de détection de pas sont basées sur l'information que le déplacement pédestre d'un individu suit un motif cyclique que l'on peut retrouver dans différentes mesures, telles que les mesures $\mathbf{a}(t)$ d'un accéléromètre trois-axes :

$$\mathbf{a}(t) = \begin{pmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{pmatrix} \quad (2.8)$$

où t représente l'instant de réception des mesures de l'accéléromètre, typiquement entre 50Hz et 200Hz, et a_x , a_y , et a_z représentent les composantes pour chacun des axes de l'accéléromètre. Parfois, il est avantageux de reconnaître également l'activité humaine, puisque la détection de pas ne sera pas la même si l'individu est en train de marcher ou de courir. [SUN et collab. \[2009\]](#) proposent une classification de l'activité d'un individu en traitant les différentes sorties de l'accéléromètre, en utilisant un réseau de neurones sur des capteurs situés au niveau de la hanche d'un individu. Ensuite, la détection de pas s'effectue sur la composante verticale de l'accéléromètre a_z . Un lissage par noyau d'ondelettes est appliqué sur les données brutes et permet de directement détecter les pas selon la périodicité du signal, comme représenté dans la figure 2.4.

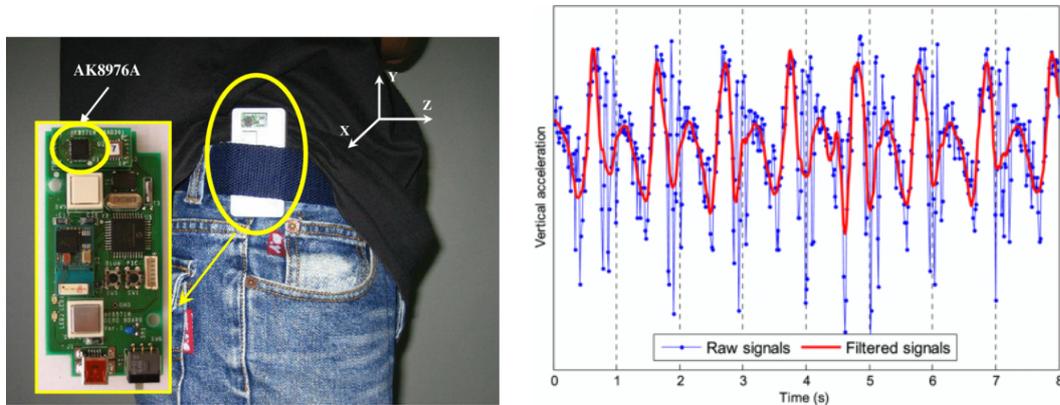


FIGURE 2.4 – Représentation des mesures a_z en fonction du temps pour détecter les pas d'un individu tiré de [SUN et collab. \[2009\]](#)

De même, [SUSI et collab. \[2013\]](#) proposent au préalable une classification par arbre qui permettra de distinguer la position du téléphone selon trois configurations : position fixe, en balance dans la main ou en mouvements irréguliers. Enfin, un algorithme sur la détection de maxima locaux lié à la norme de l'accéléromètre permet d'effectuer la détection de pas. Dans [RACKO et collab. \[2016\]](#), c'est directement la norme de l'accélération qui est évaluée pour détecter chaque pas lorsque celle-ci dépasse un certain seuil.

$$ac(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} - g \quad (2.9)$$

où g représente l'accélération gravitationnelle. Chaque fois qu'un seuil est atteint, un pas est détecté. Dans ce dernier exemple, le téléphone est tenu dans la main. Cette approche

est similaire à celle employée dans notre étude pour l'estimation de la trajectoire inertielle. Dans le cas où les capteurs inertiels sont installés sur le pied d'un individu, grâce à un module *foot mounted*, la détection de pas se fait de façon plus directe par une détection de la fenêtre correspondant à une mesure de vitesse nulle sur les capteurs [SKOG et collab. \[2010\]](#), cet instant d'environ 0.2s correspond au moment où le pied sur lequel est fixé le module est en contact avec le sol.

2.1.2.2 Estimation de la longueur de pas

La longueur de pas est définie comme la longueur parcourue entre l'instant où un individu pose son talon au sol avec un pied et l'instant où le second talon est en contact avec le sol. Cette longueur de pas peut être estimée en utilisant un modèle biomécanique, qui considère une forme simplifiée [JAHN et collab. \[2010\]](#). Les jambes de l'utilisateur sont modélisées par des segments rigides, comme dans la figure 2.5, donc sans genou, de longueur l . La marche de l'individu correspond à un pendule, et l'estimation de la longueur de pas, notée ΔL , est donnée par :

$$\Delta L = K\sqrt{2lh - h^2} \quad (2.10)$$

où h correspond au déplacement vertical du centre de gravité, et K est une constante de calibration. Une estimation empirique a également été proposée dans [WEINBERG \[2002\]](#). La longueur de pas est alors donnée par :

$$\Delta L = K\sqrt[4]{a_{z,max} - a_{z,min}} \quad (2.11)$$

où K correspond à une constante de calibration, et $a_{z,max}$ (respectivement $a_{z,min}$) correspond au maximum (respectivement minimum) de la composante verticale de l'accélération lors d'une foulée. Ces deux modèles proposent une bonne estimation de la longueur de pas lorsque le mobile conserve une position fixe sur l'utilisateur, et lorsque ce dernier se déplace sur une surface plane.

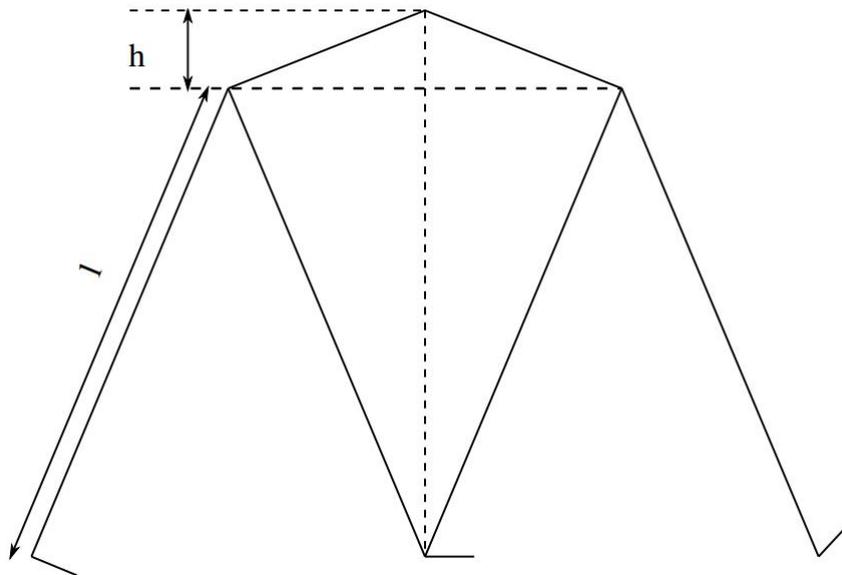


FIGURE 2.5 – Représentation du pendule inversé tirée de [JAHN et collab. \[2010\]](#)

Par ailleurs, un autre modèle empirique est proposé par [RENAUDIN et collab. \[2012\]](#) :

$$\Delta L = c_1 \times f_{pas} + c_2 \times h_{taille} + c_3 \quad (2.12)$$

où f_{pas} correspond à la fréquence du pas, h_{taille} est la taille de l'individu, et c_1 , c_2 et c_3 sont estimés empiriquement par une méthode des moindres carrés. Par la suite, ce modèle de longueur de pas sera celui utilisé dans notre étude car il allie simplicité de calcul, aucun calcul de racine carré n'intervient, mais est également associé à de bonnes performances selon des tests réalisés hors du champ de ces travaux de thèse.

2.1.2.3 Estimation du cap

L'estimation du cap de déplacement d'un individu à partir d'un *smartphone* nécessite de distinguer trois repères représentés dans la figure 2.6 : le repère absolu, représenté en violet, le repère de déplacement de l'utilisateur, représenté en bleu, et le repère du téléphone représenté en rouge. Le cap de la trajectoire est défini dans le repère de la trajectoire, mais les capteurs inertiels sont dans le *smartphone* et mesurent donc des grandeurs physiques dans le repère de ce dernier, et estiment le cap du téléphone par rapport au repère absolu. Comme illustré dans la figure 2.6, selon la position du *smartphone*, les deux repères ne sont pas alignés *a priori*. Il existe donc un décalage, en fonction de la position du téléphone dans la main, entre le cap de la trajectoire et le cap estimé du téléphone, ce décalage définit l'erreur de désalignement.

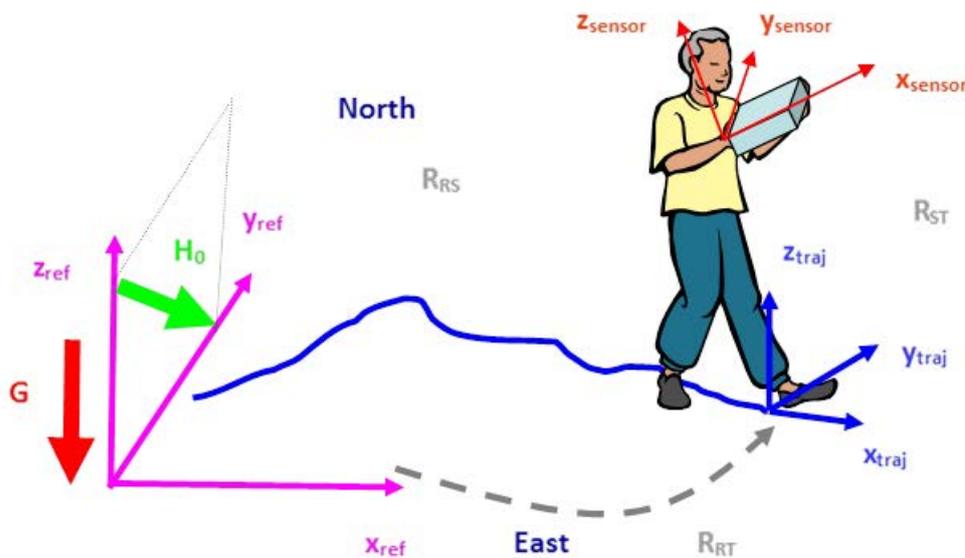


FIGURE 2.6 – Illustration de la différence entre le repère de déplacement d'un individu et le repère du mobile

Estimation de l'attitude du téléphone

La paramétrisation de l'orientation du téléphone dans le repère absolu est effectuée à l'aide d'un quaternion, noté q_t . En raison de la qualité des gyroscopes notamment, il existe un biais pouvant valoir $5^\circ/s$. q_t est ensuite estimé en fusionnant les mesures associées à l'accéléromètre, au gyroscope ou encore au magnétomètre. Cette fusion peut se faire en résolvant un problème d'optimisation paramétrique comme proposé dans [MADGWICK et collab. \[2011\]](#). Classiquement, un modèle de filtrage est mis en oeuvre, et est résolu par un filtre de Kalman étendu (EKF), le problème considéré n'étant pas linéaire. Une revue des différentes paramétrisations pour l'estimation de l'orientation est décrite dans [CRASSIDIS et collab. \[2007\]](#) et distingue diverses adaptations du filtre, le filtre AEKF pour *Additive EKF* ou encore le filtre MEKF, pour *Multiplicative EKF*. Dans notre cas,

le biais β_t sera estimé séparément en tirant profit des instants durant lesquels l'utilisateur ne se déplace pas. La paramétrisation choisie pour l'estimation du cap est l'AEKF.

Estimation du cap absolu

Le cap absolu du téléphone s'estime à l'aide du magnétomètre qui permet de se situer par rapport au nord magnétique. Néanmoins, dans les environnements *indoor*, de nombreuses sources de perturbation du champ magnétique existent telles que des armoires métalliques, ou encore certaines infrastructures électroniques. De plus, comme décrit dans la figure 2.6, le cap absolu du téléphone ne correspond pas au cap absolu de l'utilisateur. La position initiale du téléphone n'étant pas connue, ceci implique une erreur potentielle pouvant atteindre 180° . Par ailleurs, durant le déplacement d'un individu, ce dernier peut effectuer des changements de position du téléphone. On définit un changement de position du téléphone comme étant un mouvement intentionnel de l'appareil provoqué par son utilisateur. Par exemple l'opération de répondre au téléphone, alors qu'il se trouvait dans la poche ou alors de placer le téléphone du mode portrait au mode paysage, comme dans la figure 2.7 impliquant un angle de 90° . Ainsi, savoir détecter un changement de position est important puisque des mouvements du téléphone associés à un changement de position peuvent causer des perturbations dans l'estimation de la trajectoire inertielle, ou impliquer une modification du cap absolu. Dans notre cas, cette détection sera faite selon [VILLIEN \[2014\]](#). Cette méthode base la détection des changements de position sur l'ensemble des rotations effectuées, en dehors des rotations verticales qui vraisemblablement correspondent à un changement de cap de l'utilisateur. La pesanteur est mesurée dans le repère du téléphone. Une rotation non-verticale est alors détectée lorsque la projection de cette pesanteur sur l'axe vertical varie entre deux instants.

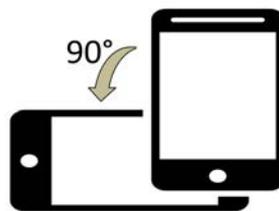


FIGURE 2.7 – Illustration d'un changement de position du téléphone

2.1.3 Fusion des mesures RSS avec l'approche inertielle

Les mesures inertielles sont précises à un horizon de temps court, et peu bruitées mais possèdent la particularité de dériver avec le temps. Les mesures RSS en revanche, bien que n'étant pas très précises ne dérivent pas et permettent une localisation absolue. Ces deux sources de mesure sont complémentaires et peuvent être fusionnées afin d'estimer la position d'intérêt. D'autres sources de mesures sont parfois utilisées avec les mesures inertielles. Par exemple, les mesures inertielles sont fusionnées avec des mesures satellitaires dans un environnement extérieur, lorsque la réception de signaux GPS est possible dans [SADI et KLUKAS \[2013\]](#), ou encore avec des mesures optiques dans [LIU et collab. \[2010\]](#).

2.1.3.1 Fusion des mesures IMU et RSS par méthode de filtrage

Différentes approches appliquent des modèles de filtrage pour proposer une estimation de la position. L'état que l'on cherche à estimer est noté à l'instant k , $\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$, où x_k et y_k représentent l'abscisse et l'ordonnée de la position. Dans le cadre de la localisation *indoor*, le modèle d'état est généralement déterminé par les mesures inertielles. Dans la théorie du filtrage, l'*a priori* sur l'état correspond à la connaissance de sa dynamique. Ici, cette connaissance est donc basée sur les mesures inertielles. On considère alors un modèle PDR, par exemple en deux dimensions comme dans ZHUANG et EL-SHEIMY [2016] qui ne considèrent pas dans cette modélisation de biais sur l'estimation du cap. Ce modèle PDR dépend d'un angle α_{k+1} et d'une longueur de pas l_{k+1} :

$$f(\mathbf{x}_{k+1}|\mathbf{x}_k) : \begin{cases} x_{k+1} &= x_k + (l_{k+1} + \eta_{k+1}) \cos(\alpha_{k+1} + \gamma_{k+1}) \\ y_{k+1} &= y_k + (l_{k+1} + \eta_{k+1}) \sin(\alpha_{k+1} + \gamma_{k+1}) \end{cases} \quad (2.13)$$

où η_k et γ_k correspondent respectivement à l'erreur d'estimation de la longueur et de l'angle du pas de la trajectoire inertielle. Le modèle d'observation représente l'information des mesures RSS

$$z_k = h(\mathbf{x}_k, v_k) \quad (2.14)$$

où v_k représente l'erreur et où la fonction $h()$ qui fait le lien entre les mesures z_k et la position \mathbf{x}_k . Ce peut être paramétré par un modèle de propagation comme proposé dans FINK et collab. [2010] ou encore ZHUANG et EL-SHEIMY [2016], mais peut également dépendre d'une estimation par *fingerprinting* comme dans EVENNOU et MARX [2006] ou encore XIAO et collab. [2011].

2.1.3.2 Fusion par moindres carrés récursifs

WILK et collab. [2015] proposent une architecture en temps différé d'estimation de la trajectoire. Cette estimation se fait sur un serveur pour mettre à jour de façon automatique une carte de puissance wifi, mais s'exprime comme une problématique en temps réel. Une position est estimée à chaque instant par une approche de *fingerprinting*, notée \mathbf{f}_k avec une erreur relative $a_{fi}\mathbf{f}_k$, a_{fi} étant un scalaire sans dimension qui quantifie l'erreur relative. Les mesures inertielles définissent un déplacement paramétré par une vitesse v_k et un angle α_k avec des erreurs relatives associées $a_v v_k$ et $a_\alpha \alpha_k$ où a_v et a_α correspondent également à des scalaires sans dimension. A chaque instant, un système linéaire est défini :

$$\mathbf{A} \times \begin{pmatrix} \mathbf{x}_k \\ h_k \end{pmatrix} = \mathbf{B} \quad (2.15)$$

où la matrice \mathbf{A} correspond au déplacement dynamique entre l'instant $k-1$ et k , et \mathbf{B} associe une position \mathbf{x}_k à la position estimée par *fingerprinting* \mathbf{f}_k . Ces matrices dépendent des valeurs \mathbf{f}_k , v_k , α_k et \mathbf{x}_{k-1} , et h_k représente une correction d'angle à appliquer. La position est donnée directement à chaque instant par :

$$\begin{pmatrix} \mathbf{x}_k \\ h_k \end{pmatrix} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (2.16)$$



FIGURE 2.8 – Illustration d’une approche de fusion des mesures IMU et RSS par une méthode de filtrage selon [WILK et collab. \[2015\]](#). La trajectoire de référence est représentée par la trajectoire noire, la trajectoire par *fingerprinting* par la trajectoire bleue, et la trajectoire fusionnée par la trajectoire violette.

2.1.4 Synthèse de l’état de l’art

Les méthodes d’estimation utilisant les mesures RSS ou IMU sont très courantes et continuent d’être perfectionnées. Néanmoins, ces estimations sont très souvent focalisées sur une estimation en temps réel. Des approches en temps différé existent mais concernent des problématiques différentes, telles que le SLAM (*Simultaneous Localization And Mapping*) qui résout la problématique d’estimation, mais dans le même temps la question de la construction de cartes. Par la suite, nous présenterons dans un premier temps l’approche qui a été développée en temps réel qui correspond à une approche de filtrage déjà présente dans la littérature. Dans un second temps, on présentera une modélisation paramétrique par morceaux de la trajectoire inertielle, qui sera ajustée par les mesures inertielles.

2.2 Fusion en temps réel RSS et IMU

La solution de fusion en temps réel appliquée met en oeuvre un modèle de filtrage qui nécessite de définir le modèle d’état, basé sur les mesures inertielles, ainsi que le modèle d’observation, basé sur les mesures RSS. Cette solution en temps réel était déjà disponible et n’a donc pas été implémentée durant ces travaux de thèse. Celle-ci a permis de quantifier le gain d’une approche en temps différé comparativement à une approche en temps réel.

2.2.1 Modèle d’état

Les mesures inertielles permettent de définir le modèle d’état à partir de la trajectoire PDR. Chaque pas effectué par l’individu est détecté, et sa longueur ainsi que son cap sont estimés. On peut décrire les différentes sources d’erreur de l’estimation de cette trajectoire inertielle :

- La détection de pas présente 1 à 5% d'erreur voire plus dans des cas de piétinement qui sont des situations complexes. L'erreur sur le comptage de pas étant relativement faible, cette erreur ne sera pas modélisée. Par ailleurs, tenir compte de cette erreur se révèle être complexe dans la formalisation, pour un gain relativement limité.
- L'erreur sur la longueur de pas peut valoir 5 à 20 % sur la longueur parcourue totale. En effet, un modèle statistique est mis en oeuvre pour estimer la longueur de chaque pas, à partir des mesures de l'accéléromètre. Mais la taille d'un individu ne suffit pas à estimer précisément cette longueur de pas, chaque individu ayant un profil de marche qui lui est propre [LADETTO \[2003\]](#). Les limites du modèle statistique impliquent alors l'existence d'un potentiel biais de l'ordre de 10%.
- Trouver le nord absolu dans un environnement magnétiquement perturbé peut par exemple impliquer une erreur sur le calcul du cap absolu de 90°. A cette erreur s'ajoute l'erreur de désalignement du téléphone, par rapport à la trajectoire de déplacement, pouvant parfois atteindre par exemple 180°. Enfin, une dérive due aux capteurs, notamment le biais du gyroscope, peut causer une erreur de précision de 5 à 20°.

L'erreur sur le modèle de pas, ainsi que l'erreur sur le calcul du cap sont stables pendant un certain temps. Ceci est illustré dans la figure 2.9 qui présente une trajectoire inertielle en fonction de la vraie trajectoire. Sur cet exemple, la trajectoire, d'une longueur de 130m, présente une erreur de cap de 115° et une erreur sur le modèle de pas de 10%. Cette erreur cap est représentée sur la figure par l'écart entre les flèches oranges, représentant le cap de la trajectoire de référence et celui mesuré par la trajectoire inertielle.

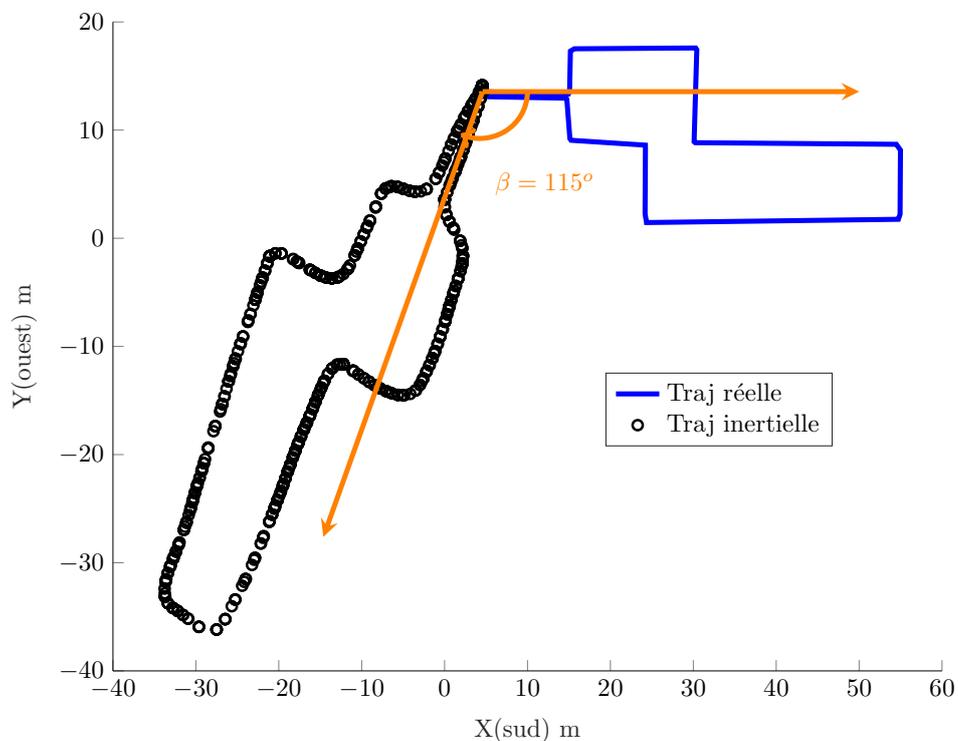


FIGURE 2.9 – Représentation de la trajectoire inertielle en comparaison avec la trajectoire réelle. L'erreur de cap est de 115° et de longueur de pas $1 + \epsilon = 0.9$

On peut traduire ces deux erreurs par un biais sur le cap β et un biais sur la longueur de pas ϵ . Ces biais constituent alors des paramètres à estimer en plus de la position. On définit un vecteur d'état \mathbf{x}_k en dimension 4 :

$$\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \\ \epsilon_k \\ \beta_k \end{pmatrix} \quad (2.17)$$

Ici, ϵ_k et β_k sont donc représentés par des processus stochastiques sujets au modèle d'état suivant :

$$f(\mathbf{x}_{k+1}|\mathbf{x}_k) : \begin{cases} \epsilon_{k+1} = \epsilon_k + \eta_\epsilon \\ \beta_{k+1} = \beta_k + \eta_\beta \\ x_{k+1} = x_k + (1 + \epsilon_{k+1})(l_{k+1} + \eta_l) \cos(\alpha_{k+1} + \beta_{k+1} + \eta_\alpha) \\ y_{k+1} = y_k + (1 + \epsilon_{k+1})(l_{k+1} + \eta_l) \sin(\alpha_{k+1} + \beta_{k+1} + \eta_\alpha) \end{cases} \quad (2.18)$$

où η_ϵ , η_β , η_l et η_α correspondent à des bruits gaussiens centrés de variance respective σ_ϵ^2 , σ_β^2 , σ_l^2 et σ_α^2 . η_ϵ correspond à l'erreur du biais sur la longueur de pas, η_β est l'erreur du biais sur le cap de la trajectoire, η_l est l'erreur de la mesure de la longueur de pas à partir des mesures inertielles et η_α est l'erreur de l'angle du pas également estimée à partir des mesures inertielles.

2.2.2 Modèle d'observation

Le modèle d'observation repose sur l'exploitation de mesures de puissance radio. En effet, comme cela a été vu, la puissance mesurée en un point est fonction de la distance entre l'émetteur et le récepteur. Cette relation de dépendance peut être très complexe selon l'environnement de propagation, et une description spatiale précise des puissances reçues en différents points est généralement établie par des relevés sur le terrain. Il s'agit de l'approche par *fingerprinting*. Cependant, cette méthode est coûteuse à mettre en oeuvre, et nécessite des mises à jour régulières de la carte. Nous choisissons donc un modèle de propagation simple afin de disposer d'une expression analytique de la fonction d'observation. Ce modèle est donné par l'équation 2.6, pour une position donnée $\mathbf{p}^k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$

$$h(\mathbf{p}^k, \mathbf{b}(k)) = P_{tr} + G_t + G_r + 10\alpha_{RSS} \log \left[\frac{\lambda}{4\pi \times d\{\mathbf{p}^k, \mathbf{b}(k)\}} \right] + v_k \quad (2.19)$$

où $\mathbf{b}(k) = \begin{pmatrix} x_b^k \\ y_b^k \end{pmatrix}$ correspond à la position de la balise qui a été activée à l'instant k , P_{tr} (dBm) est la puissance de transmission, G_t (dB) et G_r (dB) les gains d'émission et de réception des antennes, λ (m) représente la longueur d'onde, α_{RSS} un facteur sans dimension représentant la perte de propagation, $d\{\cdot, \cdot\}$ la distance euclidienne et v_k un bruit de mesure en dBm. Différents biais sont à l'origine de certaines erreurs qui rendent ce modèle peu fiable pour un environnement *indoor* :

- La gain d'émission des balises peut être mal connue, environ 1 à 2 dB d'erreur.
- La gain du récepteur, en l'occurrence le téléphone, peut être biaisée, environ 1 à 2 dB d'erreur.

- Un défaut de polarisation entre le téléphone et l'antenne d'émission, environ 10 à 30 dB d'erreur.
- Une position durable du téléphone contre le corps ou dans la poche pourrait entraîner une atténuation supplémentaire, environ 10 à 30 dB d'erreur.

2.2.3 Méthode d'estimation en temps réel

Un algorithme temps réel utilise une approche similaire à un algorithme de filtrage pour fusionner les mesures IMU et RSS. En effet, une taille de fenêtre h est définie, qui correspond par exemple à un temps de 5 secondes. Sur cette fenêtre, à partir d'un instant k , on considère alors un segment de la trajectoire PDR entre les pas k et $k+h$. On estime alors pour chacun de ces segments la position du point de départ (x_k, y_k) ainsi qu'un paramètre de rotation β_k et d'homothétie ϵ_k afin que les mesures du modèles correspondent aux mesures RSS. A l'instant suivant $k+1$, le même procédé est utilisé sur le segment défini entre les instants $k+1$ et $k+h+1$. On définit également un modèle d'état entre l'estimation des paramètres entre les instants k et $k+1$. La figure 2.10 présente l'estimation en temps réel par l'algorithme. Sur cet exemple, on peut noter que cette estimation recalibre la trajectoire inertielle, mais se trouve être limitée. Ceci est dû à la mauvaise qualité du modèle de propagation qui est utilisé pour estimer cette trajectoire.

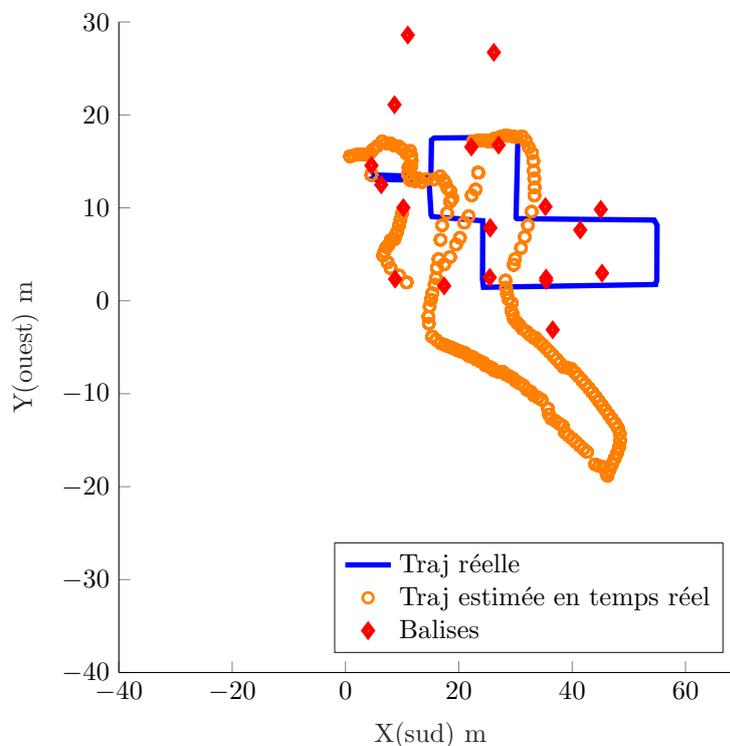


FIGURE 2.10 – Illustration de l'estimation en temps réel via une fusion des mesures IMU et RSS selon l'algorithme temps réel

2.3 Fusion paramétrique des mesures RSS et IMU

2.3.1 Paramétrisations et définition des modèles

Les erreurs associées à la trajectoire inertielle sont caractérisées par des biais et on peut dans un premier temps considérer qu'à une rotation d'angle β ainsi qu'une homothétie de paramètre $1 + \epsilon$ près, la trajectoire inertielle correspond à la trajectoire réelle, comme illustré dans la figure 2.9. Néanmoins, d'autres éléments impliquent que cette modélisation avec une unique rotation et un unique changement d'échelle n'est pas toujours suffisante.

- En effet, à chaque pas, l'angle est mesuré par une fusion entre l'accéléromètre et le gyroscope qui a tendance à dériver avec le temps. Ici, le magnétomètre n'est pas traité car son utilisation permet d'éviter la dérive du cap absolu mais les perturbations magnétiques détériorent le cap sur le court terme, impliquant des déplacements non linéaires. Dans la figure 2.11, qui est une trajectoire d'environ 600m on observe que la trajectoire inertielle dérive au cours du temps.
- De plus, le cap du téléphone peut varier suite à un changement de position du téléphone, comme évoqué dans la partie 2.1.2.3, et impliquer un changement de cap dans l'estimation de la trajectoire inertielle alors que l'individu n'aura pas tourné.

La figure 2.11 représente un exemple basé sur des mesures réelles. On retrouve que le cap initial présente une erreur d'environ 45° , qui est représentés à nouveau par les flèches oranges. Néanmoins, on observe une dérive successive de la trajectoire. En effet, le segment associé au début de la trajectoire et celui associé à la fin devrait être confondus puisqu'il s'agit du même long couloir. On observe une dérive d'environ 20° entre ces deux trajectoires qui correspond à la dérive cumulée, représentée par l'écart entre la flèche orange et la flèche verte. De plus, la trajectoire présente une erreur de 5% sur le modèle de longueur de pas.

2.3.1.1 Paramétrisation d'une trajectoire par morceaux

L'observation de la figure 2.11 démontre que la forme globale de la trajectoire inertielle est correcte, et qu'en appliquant quelques transformations géométriques simples, telles que des rotations ou des homothéties, il est possible d'obtenir une bonne estimation de la trajectoire réellement parcourue par l'individu. L'objet de ce chapitre est de tirer profit des mesures RSS pour estimer les angles de rotation et paramètres d'échelle à appliquer à la trajectoire inertielle. Notre approche repose donc sur une représentation par morceaux de la trajectoire, où chaque segment a pour origine un point dit de rupture. Il s'agira donc d'estimer les paramètres de transformation (ϵ_i, β_i) pour chacun de ces segments.

Définition des points de rupture

La figure 2.11 illustre bien qu'il est nécessaire d'appliquer plusieurs rotations à la trajectoire inertielle. En effet, en augmentant les degrés de liberté, il devient possible d'avoir une meilleure estimation de la trajectoire réelle *via* la trajectoire inertielle. Les défauts de modèle de longueur de pas étant relativement stationnaires au sein d'une trajectoire, les deux principales erreurs à compenser sont :

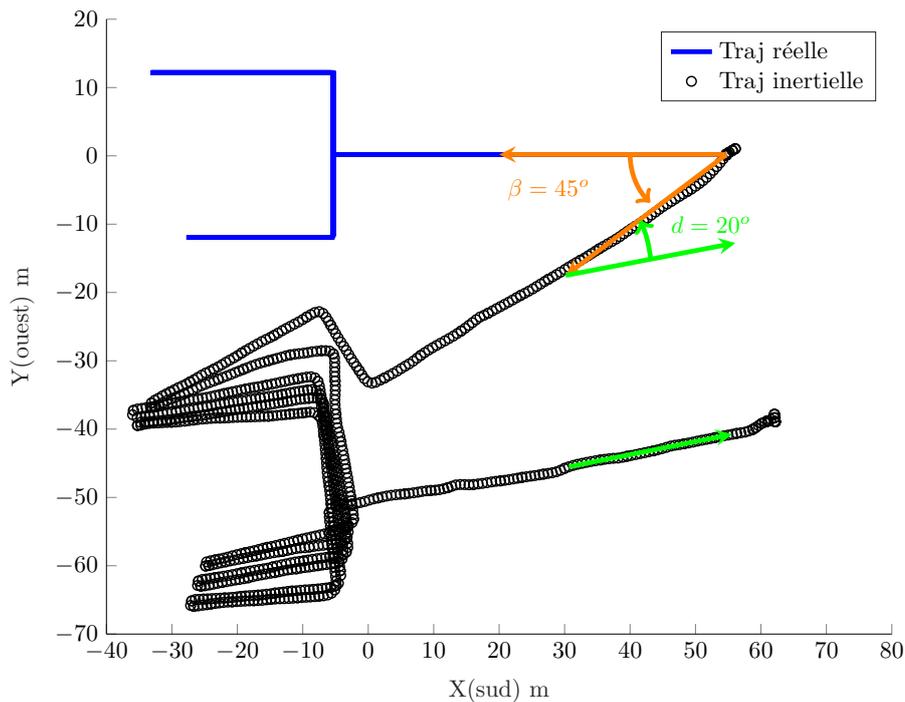


FIGURE 2.11 – Représentation de la trajectoire inertielle en comparaison avec la trajectoire réelle sur une longue trajectoire pour laquelle on observe la dérive du gyroscope. Le schéma explicatif de cette trajectoire est donné dans la figure 2.23

- les changements de position du téléphone qui entraînent une modification de cap brutale, relativement similaire à un virage
- les dérives lentes du cap qui s'accumulent en fonction du temps et qui impliquées par la qualité du gyroscope

Il est nécessaire de savoir détecter lorsque l'individu change la position de son téléphone [VILLIEN \[2014\]](#). La seconde situation concerne la dérive du gyroscope, dont l'effet prend plus d'importance lorsqu'il s'agit de trajectoires sur de longues distances. On choisit d'imposer un point de rupture artificiel, chaque fois qu'une certaine distance d_{rupt} sera parcourue, par exemple $d_{rupt} = 150\text{m}$, ou encore d'imposer un nombre fixe de points de rupture répartis le long de la trajectoire.

Le nombre de ruptures sera noté n_r et la position de chacun de ces points de rupture sera déterminée préalablement. On a choisi de fixer *a priori* le nombre ainsi que la position des points de rupture. La sélection entièrement automatisée des points de rupture suppose de savoir détecter les dérives lentes de cap, ce qui est fastidieux. De plus, on cherche à estimer des angles pour corriger la trajectoire. Imposer des points de rupture sans contrôle, et donc potentiellement à des positions trop rapprochées, peut causer certains problèmes d'identifiabilité. D'ailleurs, cette même raison justifie que la distance d_{rupt} ne saurait être trop faible. Un avantage du traitement en temps différé étant de profiter de la forme globale de la trajectoire inertielle en considérant de longs morceaux de trajectoire, cette approche gagne donc à avoir un faible nombre de paramètres à estimer.

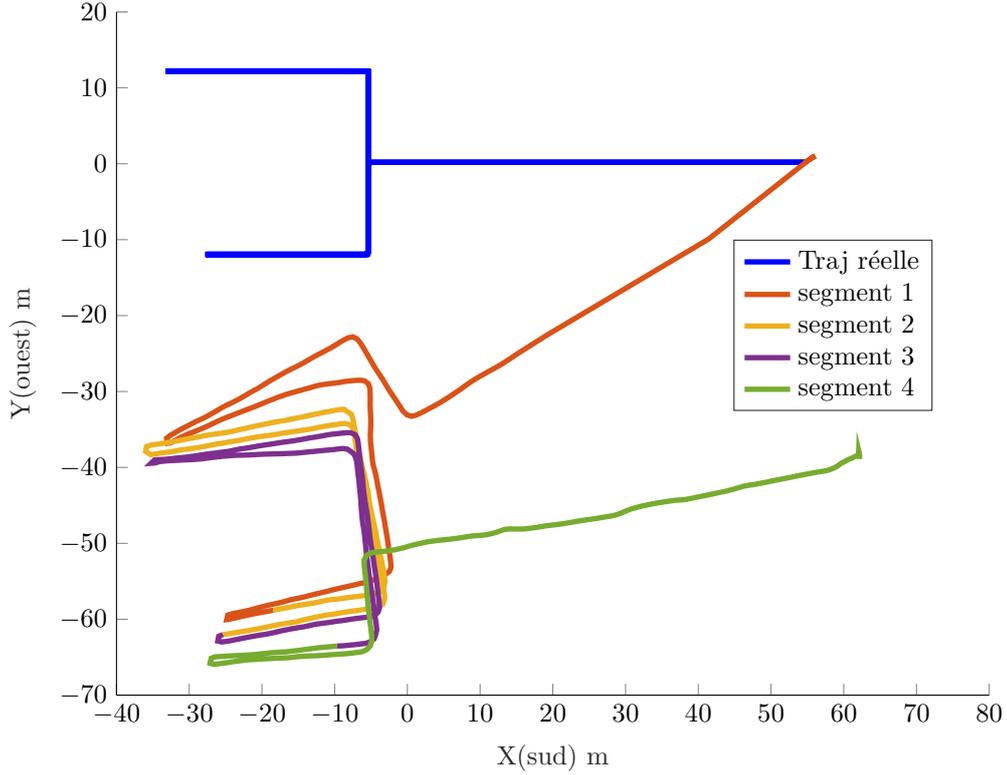


FIGURE 2.12 – Représentation de la trajectoire inertielle avec les différents segments pour $d_{rupt} = 150\text{m}$

Formalisation du modèle par morceaux

k représente l'indice de pas, et le nombre de pas total est noté K . On peut représenter, pour chaque instant $1 \leq k \leq K$, une position inertielle $\mathbf{p}_{\text{pdr}}^k = \begin{pmatrix} x_{\text{pdr}}^k \\ y_{\text{pdr}}^k \end{pmatrix}$ ainsi qu'une position réelle $\mathbf{p}_{\text{reel}}^k = \begin{pmatrix} x_{\text{reel}}^k \\ y_{\text{reel}}^k \end{pmatrix}$ que l'on cherche à estimer. Par ailleurs, on note $\mathbf{P}_{\text{pdr}} = \mathbf{p}_{\text{pdr}}^{1:K}$ (respectivement $\mathbf{P}_{\text{reel}} = \mathbf{p}_{\text{reel}}^{1:K}$) l'ensemble de la trajectoire inertielle (respectivement réelle). Le point de départ de la trajectoire étant connu, on a bien :

$$\mathbf{p}_{\text{pdr}}^1 = \mathbf{p}_{\text{reel}}^1 \quad (2.20)$$

L'origine de la trajectoire est défini comme le premier point de rupture. On note la suite strictement croissante $k_{1:n_r}$, majorée par K , correspondant aux différents indices des points de rupture. La position sur la trajectoire inertielle de la rupture i , pour $1 \leq i \leq n_r$, est donc $\mathbf{p}_{\text{pdr}}^{k_i}$. On pose par définition $k_{n_r+1} = K$. On définit alors, pour $1 \leq i \leq n_r$, un segment \mathbf{S}_i comme étant l'ensemble des points contenus entre le point de rupture k_i et le point de rupture précédent $k_{i+1} - 1$. On a pour $1 \leq i \leq n_r$:

$$\mathbf{S}_i = \bigcup_{j=k_i}^{j=k_{i+1}-1} \mathbf{p}_{\text{pdr}}^j \quad (2.21)$$

L'ensemble des segments $\mathbf{S}_{1:n_r}$ forme bien une partition de la trajectoire inertielle. Pour chaque point du segment \mathbf{S}_i , il s'agit donc d'estimer un angle de rotation $\beta_i \in [-\pi; \pi]$ ainsi

qu'une homothétie de paramètre $(1 + \epsilon_j) \in \mathbb{R}$.

On note l'ensemble des paramètres à estimer θ , un vecteur colonne de taille $2n_r$:

$$\theta^t = ([\epsilon_1 \ \beta_1 \ \dots \ \epsilon_{n_r} \ \beta_{n_r}]) \quad (2.22)$$

et on note

$$\mathbf{P}(\theta) = \mathbf{T}(\theta, \mathbf{P}_{\text{pdr}}) = \mathbf{p}_\theta^{1:K} \quad (2.23)$$

où $\mathbf{T}()$ est la fonction de transformation qui applique les rotations et homothéties de paramètre θ à la trajectoire PDR, $\mathbf{P}(\theta)$ correspond à l'estimation de la trajectoire associée au paramètre θ et \mathbf{p}_θ^k la position corrigée par le paramètre θ à l'instant k . On notera $\mathbf{C}_i(\theta)$ l'image du segment \mathbf{S}_i après application des transformations de paramètre θ à la trajectoire inertielle. On observe que la position de $\mathbf{C}_1(\theta)$ ne dépend que des paramètres (β_1, ϵ_1) . De même, la position de $\mathbf{C}_2(\theta)$ ne dépend que de $(\beta_1, \epsilon_1, \beta_2, \epsilon_2)$ etc. On prouve alors par récurrence sur i que pour tout $1 \leq i \leq n_r$, et pour tout k tel que $\mathbf{p}_{\text{pdr}}^k \in \mathbf{S}_i$, en posant $\beta_0 = 0$, $\epsilon_0 = 0$ et $\mathbf{p}_{\text{pdr}}^{k_0} = \mathbf{p}_{\text{pdr}}^0 = 0$:

$$\mathbf{p}_\theta^k = \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^k - \mathbf{p}_{\text{pdr}}^{k_i}) + \sum_{l=1}^i \left[\prod_{j=1}^l (1 + \epsilon_{j-1}) \mathbf{R}(\beta_{j-1}) \right] (\mathbf{p}_{\text{pdr}}^{k_l} - \mathbf{p}_{\text{pdr}}^{k_{l-1}}) \quad (2.24)$$

où $\mathbf{R}(\beta) = \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix}$ correspond à la matrice de rotation d'angle β . Cette récurrence est développée en annexe A.1.

2.3.1.2 Connaissances *a priori* sur les paramètres

Les transformations appliquées à la trajectoire inertielle ont pour vocation de corriger des erreurs spécifiques, telles que le biais de longueur de pas ou le biais de cap. Concernant le biais de longueur de pas, on sait qu'il peut impacter l'erreur relative d'environ 10%. Cette connaissance *a priori* peut être exploitée afin d'améliorer l'estimation du biais de longueur de pas, par exemple on sait qu'il ne fait pas sens d'obtenir un paramètre d'échelle ϵ_i de l'ordre de 2. Par ailleurs, il existe également un cas pathologique lorsque l'un de ces paramètres d'échelle vaudrait $\epsilon_i = -1$, qui impliquerait alors que tous les segments suivants $[\mathbf{S}_1, \dots, \mathbf{S}_{n_r}]$ seraient nuls et confondus au point de rupture $\mathbf{p}_\theta^{k_i}$. En outre, s'agissant d'estimation d'angles, la valeur modulo 2π constitue une réponse suffisante. Mais pour un nombre important de paramètres, il peut être intéressant de proposer un *a priori* qui favoriserait le segment $[-\pi; \pi]$ afin de permettre une meilleure stabilité de l'optimisation. On peut donc, pour chaque couple de paramètres (ϵ_i, β_i) , supposer que les paramètres sont tirés d'une loi gaussienne :

$$\begin{pmatrix} \epsilon_i \\ \beta_i \end{pmatrix} \sim \mathcal{N} \left[\theta_{\text{prior}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_{\text{prior}} = \begin{pmatrix} \sigma_{\epsilon, \text{prior}}^2 & 0 \\ 0 & \sigma_{\beta, \text{prior}}^2 \end{pmatrix} \right] \quad (2.25)$$

où $\sigma_{\epsilon, \text{prior}} = 0.2$ et $\sigma_{\beta, \text{prior}} = 180^\circ$ par exemple. Ces *a priori* sont bien indépendants puisque les sources des erreurs sont liées à des capteurs indépendants. La figure 2.13 représente l'histogramme de l'erreur globale sur le paramètre d'échelle globale, ainsi que sur l'erreur de cap initial.

Les valeurs des paramètres décrivant la connaissance *a priori* concernant θ n'a pas d'impact significatif sur l'estimation. Son rôle est d'améliorer la stabilité de l'optimisation numérique. Par ailleurs, on considère une indépendance sur l'*a priori* entre chaque

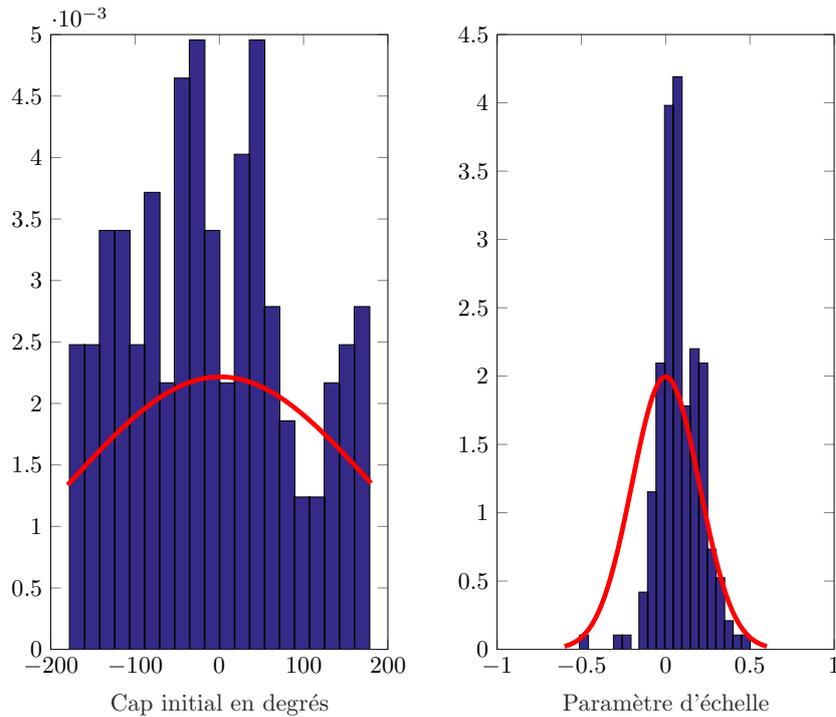


FIGURE 2.13 – Histogramme normalisé sur les paramètres d’angle initiale à gauche, et sur le changement d’échelle à droite. Ces histogrammes ont été mesurés sur un effectif de 190 expériences. En rouge est représentée la distribution *a priori* choisie.

couple de paramètres, c’est-à-dire que (ϵ_i, β_i) et (ϵ_j, β_j) , pour $1 \leq i < j \leq n_r$, sont supposés indépendants. Cette hypothèse est limitée et il aurait été possible d’affiner le modèle de dépendance de chaque couple de paramètres entre eux puisqu’il existe une dépendance hiérarchique entre les différents couples de paramètres pour des raisons physiques. En effet, un utilisateur ne change pas sa façon de marcher d’un segment à l’autre. C’est-à-dire que l’estimation du couple (ϵ_j, β_j) dépend de (ϵ_i, β_i) puisque les transformations appliquées à un segment sont également appliquées aux segments suivants. En pratique, cette hiérarchisation est déjà prise en compte lors de l’estimation des paramètres. De plus, l’étude du modèle de corrélation entre les paramètres peut se révéler fastidieuse pour un gain non significatif sur l’estimation.

2.3.1.3 Analyse et ajustement du modèle de propagation d’ondes

Un modèle de propagation d’ondes a été présenté dans la partie 2.1.1, permettant d’obtenir une relation analytique entre la distance à la balise d’émission et la puissance reçue. Néanmoins, certaines sources d’erreur telles qu’un défaut de polarisation rend ce modèle peu fiable. Notre approche propose une stratégie permettant de s’affranchir de ces erreurs.

Calibration du modèle de propagation

On illustre les limites du modèle en l’évaluant sur un grand nombre d’expériences. Pour chaque expérience réalisée, la position de la trajectoire réelle \mathbf{P}_{reel} est connue, il est donc possible d’appliquer le modèle de l’équation 2.19 et de le comparer avec les mesures réellement observées durant l’expérience. On note alors $z_{\text{mes}}(t)$ la mesure reçue à l’instant t en dBm. La fréquence de pas de la trajectoire PDR peut différer de la fréquence d’échantillonnage des mesures RSS. Afin de garantir une uniformisation des indices temporels,

on réalise une interpolation linéaire comme présenté dans l'équation 2.26 entre chaque instant de mesure. On note $\mathbf{p}_{\text{pdr}}(t)$ (respectivement $\mathbf{p}_{\text{reel}}(t)$) la position inertielle (respectivement réelle) à l'instant t et on note également t_{k_1} (respectivement t_{k_2}) l'instant de pas de la trajectoire inertielle précédant (respectivement suivant) l'instant t . On obtient alors une estimation de la position inertielle à l'instant t :

$$\mathbf{p}_{\text{pdr}}(t) = \frac{t - t_{k_1}}{t_{k_2} - t_{k_1}} \mathbf{p}_{\text{pdr}}(t_{k_2}) + \frac{t_{k_2} - t}{t_{k_2} - t_{k_1}} \mathbf{p}_{\text{pdr}}(t_{k_1}) \quad (2.26)$$

et une estimation de la position réelle à l'instant t :

$$\mathbf{p}_{\text{reel}}(t) = \frac{t - t_{k_1}}{t_{k_2} - t_{k_1}} \mathbf{p}_{\text{reel}}(t_{k_2}) + \frac{t_{k_2} - t}{t_{k_2} - t_{k_1}} \mathbf{p}_{\text{reel}}(t_{k_1}) \quad (2.27)$$

A chaque instant t où une mesure RSS est reçue, on peut associer un triplet $(z_{\text{mes}}(t), \mathbf{p}_{\text{reel}}(t), \mathbf{p}_{\text{pdr}}(t))$. Par souci de simplicité, les instants où une mesure RSS est reçue seront indicés par $1 \leq l \leq L$, et un triplet $(z_{\text{mes}}^l, \mathbf{p}_{\text{reel}}^l, \mathbf{p}_{\text{pdr}}^l)$ est alors défini. On peut alors définir $\mathbf{z}_{\text{mod}} = z_{\text{mod}}^{1:L}$ où $z_{\text{mod}}^l = h(\mathbf{p}_{\text{reel}}^l, \mathbf{b}(l))$ qui correspond au vecteur de puissance en dBm estimé par le modèle selon la trajectoire réelle et L correspond au nombre de mesures totales reçues.

Cette interpolation implique qu'aucune position inertielle n'est considérée pour l'estimation des paramètres durant les instants où aucune mesure RSS n'est reçue. Une exception est à considérer si aucune mesure RSS n'est reçue durant les premiers instants de la trajectoire. La position initiale, qui est une donnée connue de notre problématique, sera bien prise en compte dans la paramétrisation en posant que $\mathbf{p}_{\text{pdr}}^0 = \mathbf{p}_{\text{reel}}^0$ et que $z_{\text{mes}}^0 = z_{\text{mod}}^0$.

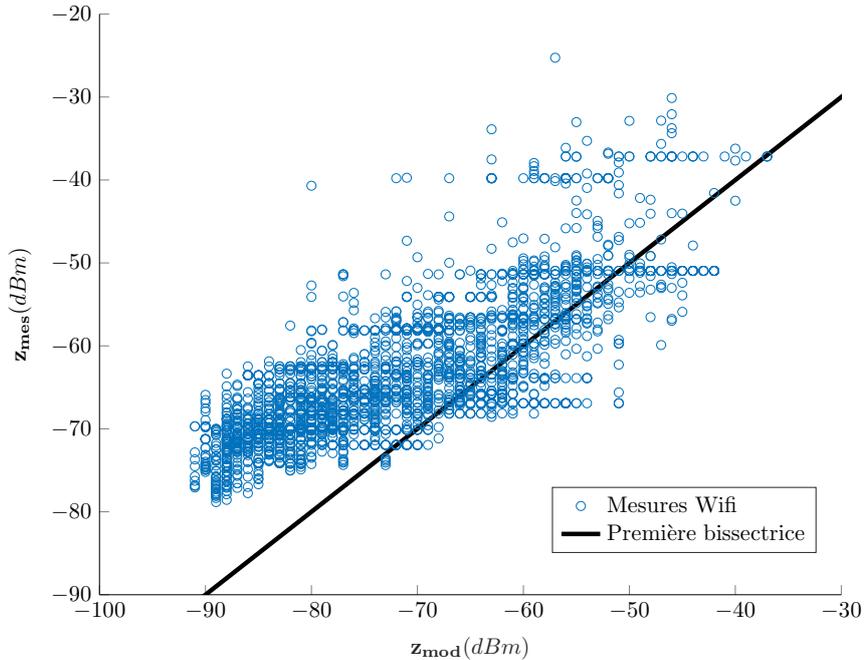


FIGURE 2.14 – Représentation des mesures réelles comparées au modèle appliqué sur la trajectoire réelle. L'expérience de cet exemple est une trajectoire de 380m avec 2900 mesures de puissance répartie sur 10 balises.

La figure 2.14 représente la puissance mesurée en fonction de la puissance estimée par le modèle pour une expérience donnée. Cet exemple démontre les limites du modèle utilisé. En effet, le nuage de points représentant les mesures en fonction des mesures théoriques révèle un décalage. On définit pour chaque expérience i :

$$\text{biais}_{\text{mod}}^i = \overline{(\mathbf{z}_{\text{mes}}^i - \mathbf{z}_{\text{mod}}^i)} \quad (2.28)$$

où $\bar{\mathbf{x}}$ représente la moyenne arithmétique du vecteur \mathbf{x} . Dans l'exemple 2.14, le biais vaut 8dBm.

Il est alors possible de calculer, pour un grand nombre d'expériences, différentes valeurs de biais associées au modèle et obtenir un vecteur $\mathbf{biais}_{\text{mod}} = \text{biais}_{\text{mod}}^{1:n_{\text{exp}}}$ où n_{exp} correspond au nombre d'expériences. La distribution des biais est représentée dans la figure 2.15.

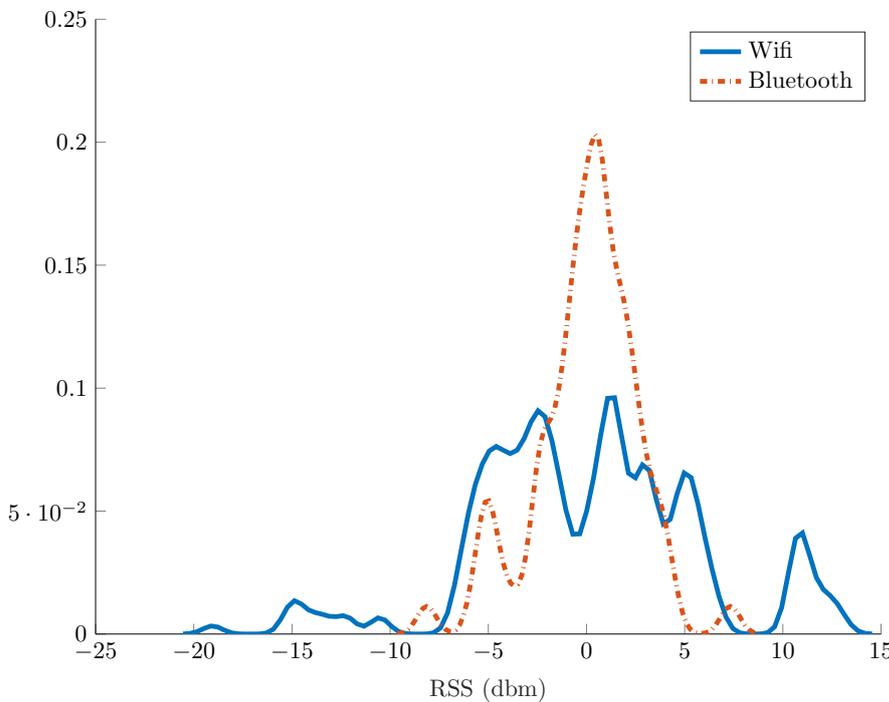


FIGURE 2.15 – Histogramme du biais du modèle pour les mesures wifi et bluetooth, correspondant respectivement à 230 et 70 expériences.

La différence de distribution entre les mesures wifi et bluetooth se justifie par le fait que les mesures bluetooth sont uniquement reçues lorsque le mobile se situe très près de la balise d'émission. Le modèle est donc moins sujet aux limites impliquées par l'environnement. Par ailleurs, on peut observer que le biais pour une mesure wifi peut varier de -15dBm à 10dBm. L'observation de la figure 2.14 incite à recentrer le nuage de points. Pour un ensemble de paramètres θ donné, on définit alors deux modèles pour la trajectoire $\mathbf{P}(\theta)$. Un modèle global que l'on notera M_{glob} qui reprendra simplement l'équation de Friis :

$$\mathbf{z}_{\text{mes}} = h(\mathbf{P}(\theta), \mathbf{B}) + \eta_{\text{mes}} \quad (2.29)$$

et un modèle adaptatif :

$$\mathbf{z}_{\text{mes}} = h(\mathbf{P}(\theta), \mathbf{B}) + \left[\overline{\mathbf{z}_{\text{mes}}} - \overline{h(\mathbf{P}(\theta), \mathbf{B})} \right] \mathbf{1}_{\mathbf{K}} + v_{\text{mes}} \quad (2.30)$$

où \mathbf{B} correspond à l'ensemble des balises émettant des mesures, $\mathbf{h}(\mathbf{P})$ est le vecteur de mesures correspondant au modèle appliqué à l'ensemble de la trajectoire \mathbf{P} , $\mathbf{1}_K$ le vecteur colonne unitaire de taille K . η_{mes} et \mathbf{v}_{mes} sont des vecteurs de bruits que l'on suppose gaussiens centrés de matrice de covariance $\Sigma_{mes} = \sigma_{mes}^2 \mathbf{I}_d(K)$ où $\mathbf{I}_d(K)$ représente la matrice identité de taille K . Dans la pratique, on fixera un écart-type à $5dBm$ pour les mesures wifi et à $2.5dBm$ pour les mesures bluetooth. Ces écart-types ont été choisis en fonction des différentes erreurs observées sur la base de données. La gaussianité des mesures s'observe sur l'ensemble des mesures des différentes trajectoires d'étude comme représenté dans la figure 2.16. Également, cette figure représente la différence entre les modèles représentés par les équations 2.29 et 2.30.

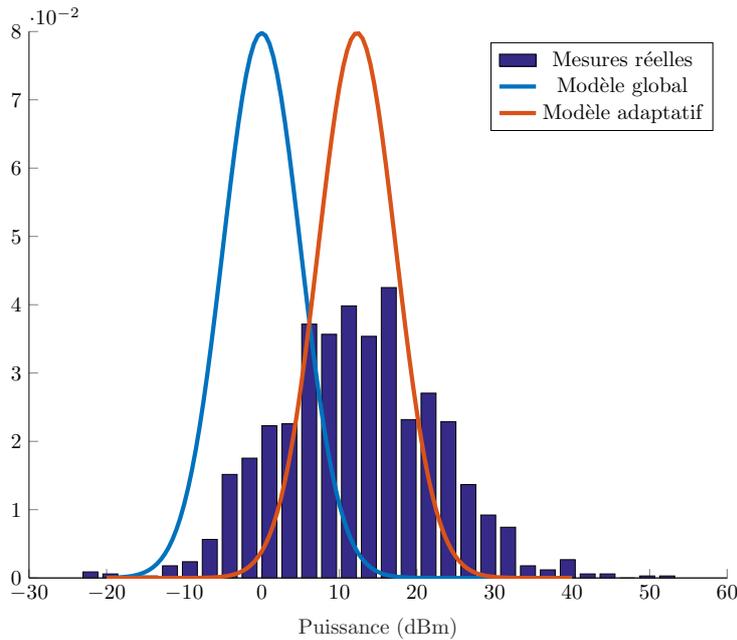


FIGURE 2.16 – Représentation de l'historique de l'erreur entre les puissances mesurées et les mesures du modèle appliquée à la trajectoire du modèle

L'utilisation du modèle adaptatif peut corriger le biais mais ne permet pas de donner une position absolue. En effet, en centrant le modèle sur l'erreur, seule la variance de l'erreur permettra d'ajuster les paramètres, et l'information moyenne sera supprimée. On illustrera dans la partie 2.3.2.1 que ceci pourra impliquer des problèmes d'identifiabilité sur l'estimation du paramètre, si la trajectoire transformée se trouve être trop éloignée de la trajectoire de recherche. Il est nécessaire de définir un modèle complémentaire pouvant estimer une position absolue permettant de retrouver l'information moyenne. On définit un modèle de proximité.

Ajustement global de l'estimation de la trajectoire via des mesures de proximité

L'information de la position absolue se situe notamment dans les mesures très élevées de puissances reçues. En effet, une mesure de puissance très forte, par exemple supérieure à $-50dBm$, signifie que la distance entre la balise d'émission et le récepteur est relativement faible, de l'ordre de $5m$. Il est donc possible de définir des mesures de proximité en considérant les mesures RSS supérieures à un certain seuil. Par ailleurs, on note n_{prox} le nombre de ces mesures de proximité que l'on considérera. On présente alors un

nouveau modèle, pour une mesure de proximité n , avec $1 \leq n \leq n_{prox}$, on suppose :

$$v_{\theta}^n = \mathbf{d}\{\mathbf{b}(k_{prox}(n)), \mathbf{p}_{\theta}^{\mathbf{k}_{prox,n}}\} \sim \mathcal{N}(d_{prox}, \sigma_{prox}^2) \quad (2.31)$$

où $k_{prox}(n)$ correspond à l'indice de la mesure associée à la mesure de proximité n , $\mathbf{b}(k)$ étant la position de la balise d'émission qui envoie la mesure k , d_{prox} étant une distance moyenne à définir, σ_{prox}^2 la variance associée. Ce modèle donne alors uniquement une information de proximité et ne permet plus d'obtenir de relation entre la distance à la balise et la puissance de la mesure RSS. Deux mesures différentes, par exemple de $-40dBm$ ou $-50dBm$, seront alors traitées identiquement par ce modèle. Lorsqu'une mesure est forte, le modèle d'émission est remplacé par le modèle de l'équation 2.31. Ici, l'erreur est donc mesurée par un écart de distance et non par un écart de puissance comme précédemment. Dans notre implémentation, les mesures fortes ont été considérées avec le modèle de proximité mais également avec le modèle adaptatif, impliquant une redondance d'information ainsi qu'une dépendance entre les modèles. Néanmoins, ceci n'impacte pas numériquement les résultats. Par ailleurs, le choix des mesures de proximité se fait en considérant les mesures de puissances supérieures à un seuil. Afin de pouvoir par la suite ajuster l'estimation des paramètres avec le modèle adaptatif qui est plus précis car il permet de corriger les erreurs du modèle de propagation, on limite le nombre de mesures de proximité. En effet, les mesures utilisées par le modèle adaptatif sont celles qui n'auront pas été utilisées par le modèle de proximité, limiter le nombre des mesures de proximité garantit alors un nombre de mesures pertinent pour le modèle adaptatif. En outre, on fixe également un seuil de puissance minimal afin d'assurer que pour une trajectoire n'ayant uniquement capté des mesures RSS très faibles, aucune ne soit considérée comme une mesure de proximité. Le modèle d'observation couplant un modèle adaptatif, défini par l'équation 2.30 et par des mesures de proximité, défini par l'équation 2.31, sera noté M_{adapt} .

2.3.2 Estimateur du maximum *a posteriori*

Le paramètre θ sera estimé par l'estimateur MAP, décrit dans la partie 1.3.1, celui-ci étant obtenu en maximisant la loi *a posteriori*.

$$\theta_{MAP} = \underset{\theta \in \Theta}{\operatorname{argmax}} L(\theta | \mathbf{z}_{mes}) \pi(\theta) \quad (2.32)$$

2.3.2.1 Définition des fonctions de coût

Pour le modèle M_{glob} , on obtient que la fonction à optimiser est :

$$\begin{aligned} L_{glob}(\theta | \mathbf{z}) \pi(\theta) &\propto \frac{1}{K_{prior}} \exp \left[-\frac{1}{2} \operatorname{err}_{prior}(\theta) \right] \times \frac{1}{K_{RSS}} \exp \left[-\frac{1}{2} \operatorname{err}_{RSS}(\theta) \right] \\ &\propto \exp \left[-\frac{1}{2} (\operatorname{err}_{RSS}(\theta) + \operatorname{err}_{prior}(\theta)) \right] \end{aligned} \quad (2.33)$$

où $K_{prior} = (2\pi)^{n_{br}} |\Sigma_{prior}|^{1/2}$ et $K_{RSS} = (2\pi)^{L/2} |\Sigma_{mes}|^{1/2}$ correspondent à des constantes de normalisation ne dépendant pas du paramètre θ , et en définissant l'écart au modèle *a priori* noté $\operatorname{err}_{prior}$:

$$\operatorname{err}_{prior}(\theta) = (\theta_{prior} - \theta)^T \Sigma_{prior}^{-1} (\theta_{prior} - \theta) \quad (2.34)$$

et l'écart quadratique sur le modèle global noté $\text{err}_{\text{RSS}}(\theta)$:

$$\text{err}_{\text{RSS}}(\theta) = (\mathbf{z}_{\text{mes}} - \mathbf{z}(\theta))^T \Sigma_{\text{mes}}^{-1} (\mathbf{z}_{\text{mes}} - \mathbf{z}(\theta)) \quad (2.35)$$

où $\mathbf{z}(\theta) = \mathbf{h}(\mathbf{P}(\theta), \mathbf{B})$. Résoudre le problème d'optimisation de l'équation 2.35 est équivalent à minimiser la fonction que l'on définira comme la fonction de coût :

$$\text{cout}_{\text{glob}}(\theta) = \text{err}_{\text{RSS}}(\theta) + \text{err}_{\text{prior}}(\theta) \quad (2.36)$$

Il est possible de définir une fonction de coût liée au second modèle M_{adapt} . On définit d'abord l'erreur de proximité err_{prox} :

$$\text{err}_{\text{prox}}(\theta) = (\mathbf{v}_\theta - d_{\text{prox}} \mathbf{1}_{n_{\text{prox}}})^T \Sigma_{\text{prox}}^{-1} (\mathbf{v}_\theta - d_{\text{prox}} \mathbf{1}_{n_{\text{prox}}}) \quad (2.37)$$

où $\Sigma_{\text{prox}} = \sigma_{\text{prox}}^2 \mathbf{I}_{n_{\text{prox}}}$ est la matrice de covariance et $\mathbf{v}_\theta = v_\theta^{1:n_{\text{prox}}}$ correspond au vecteur de distance entre les balises d'émission et la position estimée. On définit également l'erreur quadratique centrée que l'on notera $\text{err}_{\text{adapt}}$:

$$\text{err}_{\text{adapt}}(\theta) = (\tilde{\mathbf{z}}_{\text{mes}} - \tilde{\mathbf{z}}(\theta) - \tilde{\boldsymbol{\mu}}(\theta))^T \tilde{\Sigma}_{\text{mes}}^{-1} (\tilde{\mathbf{z}}_{\text{mes}} - \tilde{\mathbf{z}}(\theta) - \tilde{\boldsymbol{\mu}}(\theta)) \quad (2.38)$$

où $\tilde{\boldsymbol{\mu}}(\theta) = \left[\overline{\tilde{\mathbf{z}}_{\text{mes}}} - \overline{\tilde{\mathbf{z}}(\theta)} \right] \mathbf{1}_{\mathbf{K}}$ correspond au vecteur unitaire multiplié par la moyenne empirique de l'écart de l'erreur aux mesures sur la trajectoire et où $\tilde{\mathbf{z}}_{\text{mes}}$, (respectivement $\tilde{\mathbf{z}}(\theta)$ et $\tilde{\Sigma}_{\text{mes}}$) correspond au vecteur \mathbf{z}_{mes} (respectivement au vecteur \mathbf{z}_θ et à la matrice Σ_{mes}) dont il a été retiré les lignes (et colonnes pour la matrice) correspondant aux mesures de proximité.

$$\text{cout}_{\text{adapt}}(\theta) = \text{err}_{\text{adapt}}(\theta) + \text{err}_{\text{prox}}(\theta) + \text{err}_{\text{prior}}(\theta) \quad (2.39)$$

Centrer le modèle global se révèle être probant en terme de performance d'estimation (cf partie 2.4), mais a des conséquences directes sur la fonction de coût et sur l'optimisation. En effet, on choisit de représenter l'évolution de la fonction de coût selon les paramètres β et ϵ . Dans cet exemple simple ne comportant qu'un seul segment, il est possible d'observer la conséquence de centrer le modèle, puis d'ajouter les mesures de proximité sur la fonction de coût. La fonction de coût associée au modèle global présente une fonction convexe et simple à minimiser dans la figure 2.17.

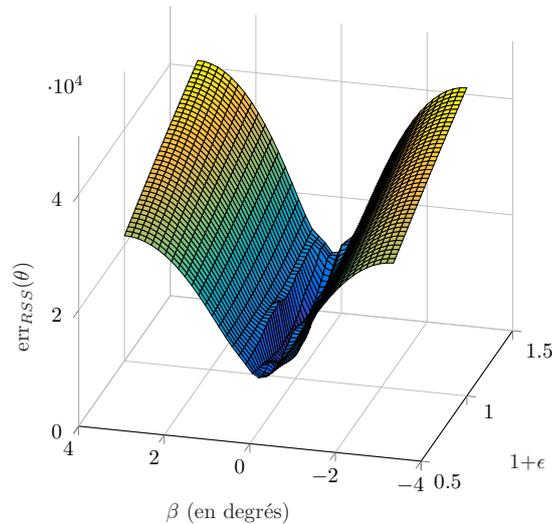


FIGURE 2.17 – Représentation de la fonction de coût associée au modèle globale en fonction de β et ϵ

La fonction de coût associée au modèle centré, sans les mesures de proximité de la figure 2.18 présente certaines difficultés. Le centrage des mesures élimine l'information de distance absolue du modèle, seule une notion de distance relative au travers des variations de puissance est préservée. Ceci justifie l'aspect plat des fonctions de coût lorsque les paramètres se trouvent loin de la solution. Par ailleurs, le modèle centré présente une sensibilité plus marquée à l'angle puisqu'on observe que la fonction de coût se trouve être très élevée aux alentours de la zone convexe. Cette sensibilité accrue traduit une meilleure précision d'estimation du paramètre à partir de ce modèle. Ceci illustre que l'utilisation du modèle adaptatif, même s'il améliore sensiblement l'estimation des paramètres de transformation, ne peut uniquement faire sens lorsque la trajectoire estimée se situe à proximité de la trajectoire réelle.

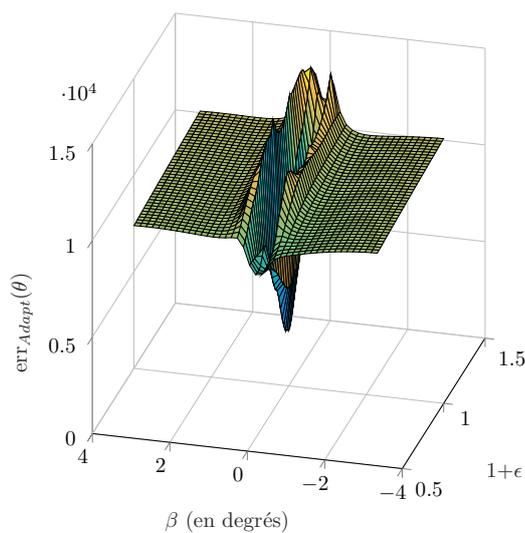


FIGURE 2.18 – Représentation de la fonction de coût associée au modèle centré sans les mesures de proximité en fonction de β et ϵ

L'utilisation d'un modèle adaptatif peut améliorer la précision d'estimation des para-

mètres à condition toutefois de pouvoir initialiser la recherche dans une zone de convergence réduite. Ceci justifie l'utilisation des mesures de proximité. En effet, la figure 2.19 illustre bien que la fonction de coût est convexe sur l'espace des paramètres, tout en conservant la précision sur les paramètres d'intérêt.

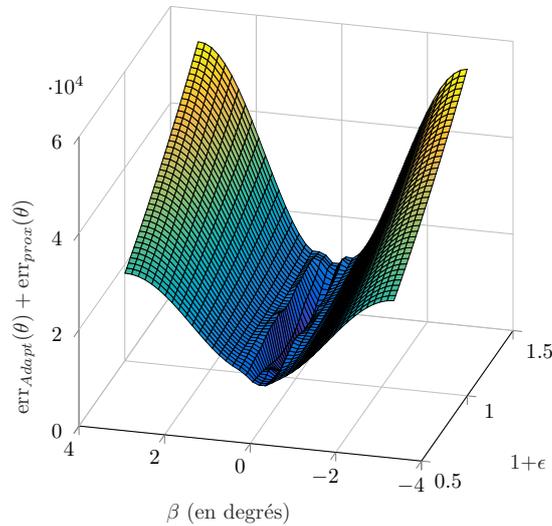


FIGURE 2.19 – Représentation de la fonction de coût associée au modèle centré avec les mesures de proximité en fonction de β et ϵ

Il n'est pas possible d'exprimer la solution de ces problèmes d'optimisation de façon analytique, il est nécessaire d'utiliser un algorithme de résolution numérique.

2.3.3 Résolution numérique de l'optimisation

2.3.3.1 Résolution de l'optimisation via l'algorithme de Levenberg-Marquardt

Différents algorithmes permettent de résoudre numériquement un problème d'optimisation. Certains considèrent un premier paramètre θ_0 puis additionnent de façon itérative un vecteur noté $h(i)$ définissant alors une suite θ_i :

$$\theta_{i+1} = \theta_i + h(i) \quad (2.40)$$

Ces algorithmes définissent une suite $h(k)$ permettant de faire converger θ_k vers l'optimum recherché. Par exemple, l'algorithme de la descente du gradient définit à chaque instant le vecteur $h_{grad}(k)$ dans la direction de la plus forte pente de gradient. Toutefois, le temps de convergence peut être long en terme d'itération. Également, l'algorithme de Gauss-Newton qui est un algorithme dont la vitesse de convergence est quadratique, linéarise localement la fonction à minimiser et implique une optimisation plus rapide. Néanmoins, cet algorithme nécessite de se trouver dans la bonne zone de convergence. L'algorithme de Levenberg-Marquardt [MORÉ \[1978\]](#) est une combinaison entre la méthode de la descente du gradient et l'algorithme de Gauss-Newton. En effet, lorsque les paramètres à estimer se trouvent trop loin de l'optimum, l'algorithme favorise la méthode du gradient. En revanche, la méthode de Gauss-Newton sera privilégiée lorsque les paramètres à estimer se trouveront près d'un minimum, c'est-à-dire lorsque la pente du gradient est faible. Cet algorithme permet de diminuer le temps d'itération nécessaire par rapport à chacun de ces algorithmes. Par ailleurs, d'autres algorithmes d'optimisation

existent, tels que les méta-heuristiques comme le recuit simulé ou les colonies de fourmis qui sont basées sur des résolutions stochastiques. Un algorithme génétique qui applique la théorie de l'évolution à une population de solutions possibles, a été implémenté mais n'a pas été probant. Le choix s'est porté sur un algorithme de Levenberg-Marquardt (LM) pour les gains de convergence par rapport aux méthodes classiques. A chaque itération, le vecteur $h_{lm}(i)$ est donné par :

$$h_{lm}(i) = [\mathbf{J}_{\theta_i}^T \mathbf{W} \mathbf{J}_{\theta_i} + \lambda \text{Diag}(\mathbf{J}_{\theta_i}^T \mathbf{W} \mathbf{J}_{\theta_i})]^{-1} \mathbf{J}_{\theta_i}^T \mathbf{W} [\mathbf{y}_m - \mathbf{y}(\theta_i)] \quad (2.41)$$

où $\text{Diag}(\mathbf{M})$ correspond à la matrice diagonale contenant uniquement les éléments diagonaux de la matrice carrée \mathbf{M} , λ est le paramètre de poids pondérant la combinaison de l'algorithme de Gauss-Newton (pour un λ petit) ou de la descente du gradient (quand λ est grand), $\mathbf{y}(\theta)$ correspond à la fonction paramétrique que l'on compare avec \mathbf{y}_m , $\mathbf{J}_\theta = \frac{\partial}{\partial \theta} \mathbf{y}(\theta)$ est la fonction jacobienne associée à la fonction $\mathbf{y}(\theta)$ et \mathbf{W} est une matrice de pondération.

On présentera l'application de LM pour la minimisation de la fonction de coût $adapt(\theta)$. On exprime la fonction paramétrique $\mathbf{y}(\theta)$ en fonction des données du problème :

$$\mathbf{y}(\theta) = \begin{pmatrix} \tilde{\mathbf{z}}(\theta) - \overline{\tilde{\mathbf{z}}(\theta)} \mathbf{1}_L \\ \mathbf{v}_\theta \\ \theta \end{pmatrix} \quad (2.42)$$

$\mathbf{y}(\theta)$ est donc un vecteur de taille $L + 2n_r$. Ce vecteur est alors comparé au vecteur d'observations \mathbf{y}_m :

$$\mathbf{y}_m = \begin{pmatrix} \tilde{\mathbf{z}}_{mes} - \overline{\tilde{\mathbf{z}}_{mes}} \mathbf{1}_K \\ d_{prox} \mathbf{1}_{n_{prox}} \\ \theta_{prior} \end{pmatrix} \quad (2.43)$$

De même, il est possible de réécrire le jacobien \mathbf{J}_θ et la matrice de pondération \mathbf{W} :

$$\mathbf{J}_\theta = \begin{pmatrix} \mathbf{J}_z \\ \mathbf{J}_v \\ \mathbf{J}_p \end{pmatrix} \text{ et } \mathbf{W} = \begin{pmatrix} \mathbf{W}_z & 0 & 0 \\ 0 & \mathbf{W}_v & 0 \\ 0 & 0 & \mathbf{W}_p \end{pmatrix} \quad (2.44)$$

où $\mathbf{J}_z = \frac{\partial}{\partial \theta} [\tilde{\mathbf{z}}(\theta) - \overline{\tilde{\mathbf{z}}(\theta)} \mathbf{1}_L]$ de taille $2n_r \times (L - n_{prox})$, $\mathbf{J}_v = \frac{\partial}{\partial \theta} \mathbf{v}_\theta$ de taille $2n_r \times n_{prox}$, $\mathbf{J}_p = \mathbf{I}_d(2n_r)$ la matrice identité de taille $2n_r$, $\mathbf{W}_z = \Sigma_{mes}^{-1}$, $\mathbf{W}_v = \Sigma_{prox}^{-1}$ et $\mathbf{W}_p = \Sigma_{prior}^{-1}$. Les calculs des jacobiens sont développés en annexe A.2.

2.3.3.2 Problème de minimum local

Le problème d'optimisation étudié peut présenter un certain nombre de minima locaux. En effet, le modèle que l'on considère tient uniquement compte de la distance entre la balise et la position du mobile. De fait, beaucoup de situations de symétrie existent, comme illustré dans la figure 2.20. En effet, sur cet exemple simple, avec une seule balise et sans aucun virage, il existe deux solutions symétriques. Ce problème de minimum traduit le fait que le segment estimé ne se situe pas du bon côté de la balise. Dans un cas réel, il y a évidemment un plus grand nombre de balises, ainsi que des virages sur la trajectoire, ce qui réduira grandement ce problème de symétrie. Mais, aux alentours de chaque balise, ce phénomène de symétrie subsistera si un segment, par exemple, n'est pas en mesure de traverser la balise à cause de l'aspect local de la fonction de coût.

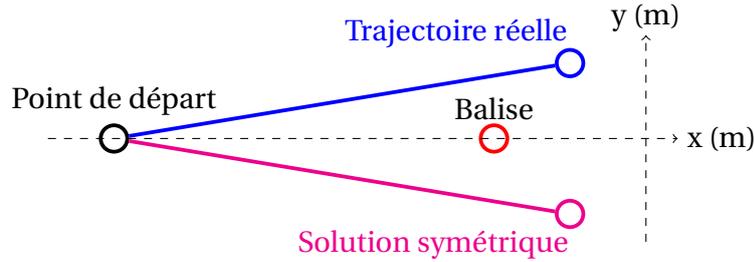


FIGURE 2.20 – Représentation sur un exemple simple des potentiels problèmes de symétrie, avec une seule balise et sans virage dans la trajectoire

2.3.3.3 Heuristique aléatoire d'optimisation

L'algorithme de LM permet de converger rapidement, mais ne garantit pas de trouver le minimum global et peut se retrouver bloqué dans un minimum local. On propose une heuristique aléatoire qui consiste à effectuer une perturbation aléatoire de l'ensemble des paramètres estimés à chaque convergence de l'algorithme de LM afin de sortir de l'espace de convergence, et d'explorer un plus large espace. L'ensemble des paramètres ayant fourni les meilleurs résultats selon la fonction d'erreur sera conservé comme estimation finale. A chaque convergence de l'algorithme de Levenberg-Marquardt, la perturbation aléatoire sur les paramètres s'effectue de la façon suivante :

$$f_{pert}(\theta) = \theta + \frac{1}{2} \left(1 - \frac{p_{pert}}{2}\right) \theta_{pert} \quad (2.45)$$

où $p_{pert} \sim \mathcal{B}\left(\frac{1}{2}\right)$ suit une loi de Bernouilli, $\theta_{pert} \sim \mathcal{N}(\theta_{prior}, \Sigma_{prior})$. Quand $p_{pert} = 0$ (respectivement $p_{pert} = 1$), la variance de la perturbation vaut la moitié (respectivement le quart) de la variance *a priori*. Ces valeurs ont été fixées empiriquement afin de répondre à un compromis : les perturbations doivent être suffisamment importantes pour sortir des zones de convergence, mais doivent également permettre d'explorer l'espace à proximité des meilleures solutions. Ce compromis justifie l'utilisation d'un mélange de gaussiennes avec deux variabilités différentes.

2.3.3.4 Algorithme d'estimation global

On peut proposer un algorithme d'optimisation global, décrit dans l'algorithme 1. Par ailleurs, un nombre d'itérations total du nombre d'étapes cumulé est fixé puisque la convergence ne suffit pas à garantir la bonne estimation des paramètres. Ce nombre d'itérations sera noté i_{stop} . Enfin, pour contrôler s'il y a convergence, on fixe un paramètre $\epsilon_{stop} = 10^{-2}$ qui est un seuil sur la norme du vecteur $h_{lm}(i)$. A chaque fois que la norme du vecteur de perturbation h_{lm} , estimé par l'équation 2.41 est inférieure à ϵ_{stop} , l'heuristique aléatoire est mise en oeuvre.

Algorithme 1 : Algorithme de Levenberg-Marquardt cumulé avec une heuristique aléatoire

Résultat : θ_{est}

```

1   $Err_{est} = 10^{10}$ ,  $\theta_{est} = \theta_0$  &  $i = 0$ ;
2  tant que  $i < i_{stop}$  faire
3       $\lambda_0 = 1$  &  $h_{lm}(0) = [1, \dots, 1, 1]$ ;
4      tant que  $\|h_{lm}(i)\|_2 < \epsilon_{stop}$  faire
5           $i = i + 1$ ;
6           $h_{lm}(i) = [J_0^T W J_0 + \lambda \text{Diag}(J_0^T W J_0)]^{-1} J_0^T W [y_m - y(\theta)]$ ;
7           $\theta_{int} = \theta_i + h_{lm}(i)$ ;
8          si  $Err(\theta_{int}) < Err(\theta_i)$  alors
9               $\theta_{i+1} = \theta_{int}$  &  $\lambda_{i+1} = \lambda_i / 10$ ;
10         sinon
11              $\theta_{i+1} = \theta_i$  &  $\lambda_{i+1} = \lambda_i \times 10$ ;
12     si  $Err(\theta_i) < Err_{est}$  alors
13          $Err_{est} = Err(\theta_i)$  &  $\theta_{est} = \theta_i$ ;
14          $\theta_0 = f_{pert}(\theta_{est})$ 

```

2.4 Résultats

2.4.1 Application numérique

On a fixé à 10 le nombre maximal de mesures de proximité et à $-55dBm$ le seuil de puissance minimal. Ces valeurs ont été fixées à partir de différents essais sur certaines trajectoires. Il ne s'agit pas de paramètres sensible ou impactant grandement les résultats finaux. Enfin, la distance de proximité a été fixée à $d_{prox} = 5m$ avec un écart-type de $\sigma_{prox} = 5m$. De même que pour les autres valeurs du modèle, celles-ci ont été fixées de façon empirique. Néanmoins, la valeur de l'écart-type répond à un compromis spécifique : une variabilité trop grande impliquera que cette mesure de proximité ne permettra pas d'obtenir l'estimation de la position absolue de la trajectoire et une variabilité trop faible ne permettra pas au modèle adaptatif d'estimer les bons paramètres.

2.4.2 Illustration sur une trajectoire sans rupture

2.4.2.1 Présentation de la trajectoire de référence

La première expérience correspond à une trajectoire de 130 mètres qui a été enregistrée dans les locaux d'un bâtiment de bureaux. Pour cette trajectoire, des mesures bluetooth et wifi étaient disponibles et reposaient sur un nombre de 14 et 6 balises respectivement. Celles-ci sont réparties sur l'ensemble de la trajectoire qui est donc totalement couverte.

L'utilisateur effectue une trajectoire fermée représentée dans la figure 2.21. Enfin, pour cette expérience aucune rupture n'est considérée. Le paramètre θ correspond donc uniquement à un angle de rotation et un changement d'échelle.

2.4.2.2 Estimation en temps différé de la trajectoire réelle

La figure 2.22 montre la qualité d'estimation de la trajectoire inertielle dans sa forme globale. Cette figure montre les limites d'une estimation en temps réel (figure en haut

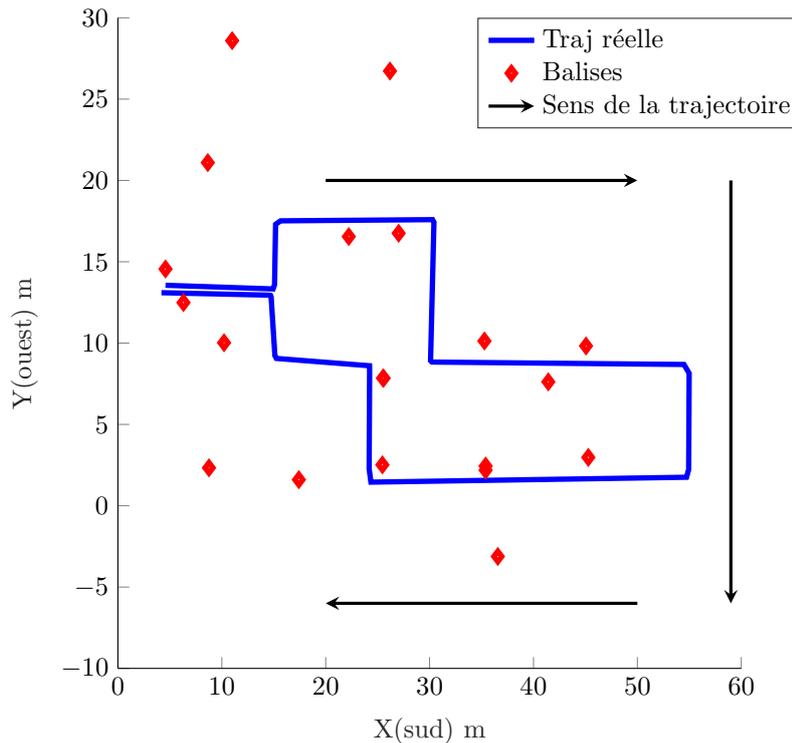


FIGURE 2.21 – Présentation de la trajectoire de référence pour l'exemple 1

à droite) où les positions peuvent présenter des sauts importants. En effet, la trajectoire estimée en temps réel n'est pas garantie d'être continue puisqu'il est possible d'estimer une nouvelle position très différente de la précédente selon l'observation des mesures. De plus, en raison de la faible taille de la fenêtre glissante utilisée pour permettre une estimation en temps réel, à chaque instant, le nombre de mesures RSS est trop faible pour pouvoir appliquer le modèle adaptatif. Seulement 34% de la trajectoire estimée se trouve à moins de 5m de l'estimation en temps réel. Par ailleurs, la qualité d'estimation du modèle global (figure en bas à gauche), démontre bien la nécessité d'ajuster le modèle. En effet, l'estimation de l'angle global proposée par ce modèle est relativement correcte, mais le facteur d'échelle n'est pas satisfaisant impliquant une erreur importante, avec une valeur de 11m rmse pour l'estimation via le modèle global. Enfin, dans cet exemple, en employant le modèle adaptatif (figure en bas à droite) la trajectoire réelle est reconstituée avec une très grande précision puisque la valeur de l'erreur est d'environ 1m. Enfin, l'écart entre la trajectoire estimée et la trajectoire réelle n'excède jamais 5mm.

Trajectoire 1	rmse (m)	%d5 (%)
Estimation temps réel	11.36	33.78
Estimation via modèle global	4.48	67.65
Estimation via modèle adaptatif	1.08	100

TABEAU 2.1 – Évaluation de la fusion IMU et RSS sur une première trajectoire

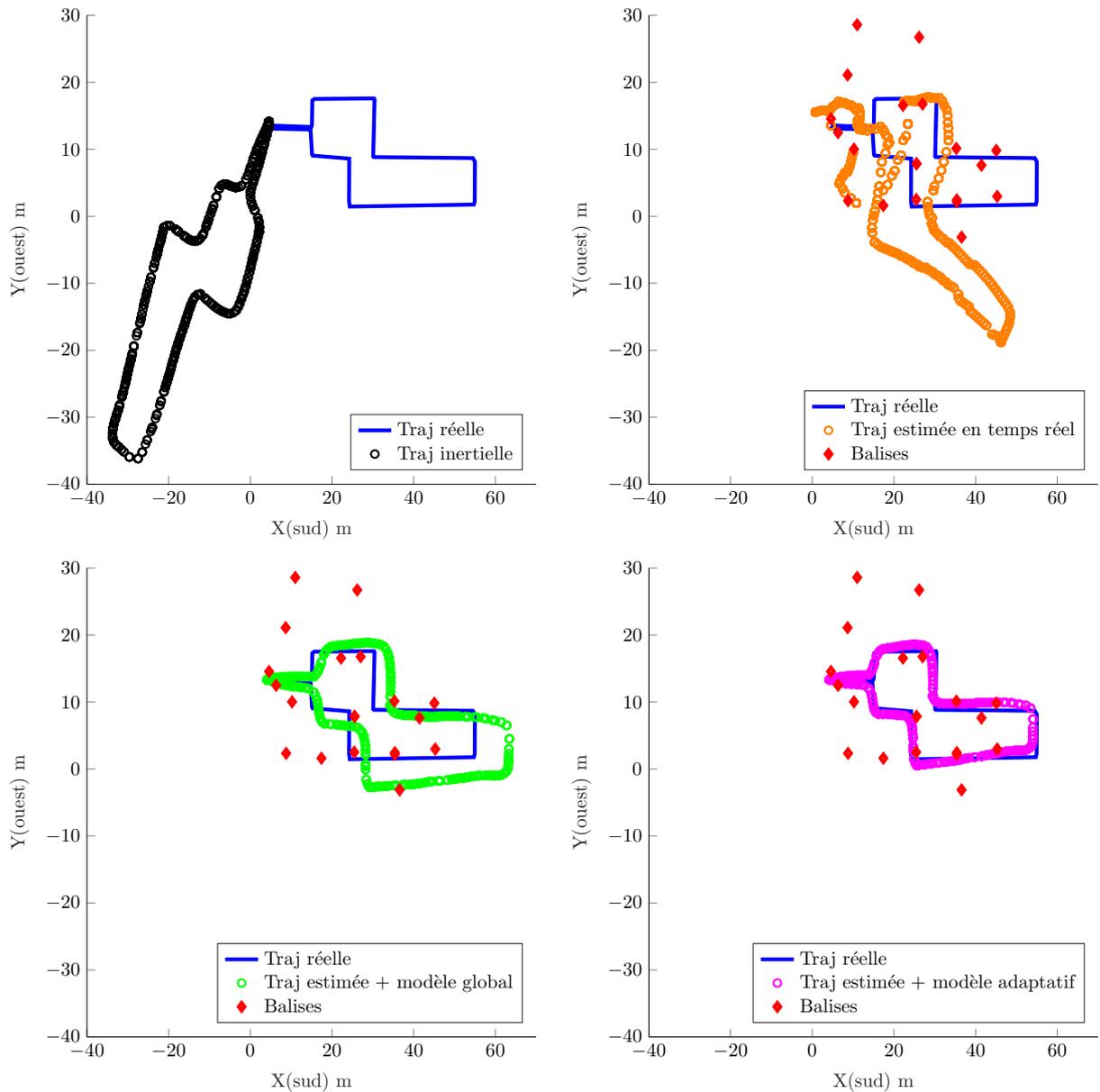


FIGURE 2.22 – Exemple des différentes estimations en fusionnant les mesures RSS ainsi que les mesures IMU.

2.4.3 Illustration sur une trajectoire avec des ruptures

2.4.3.1 Présentation de la trajectoire de référence

La seconde expérience correspond à une trajectoire de 600 mètres qui a été enregistrée dans des locaux de bureaux à Grenoble. Cette trajectoire a été enregistrée avec un téléphone Nexus 4 et seules des mesures wifi étaient disponibles avec un nombre de 6 balises.

Dans un premier temps, l'utilisateur parcourt un long couloir. Ensuite, il parcourt une anse 3 fois consécutivement. Enfin, l'utilisateur parcourt à nouveau le long couloir afin de revenir à son point de départ initial représentée dans la figure 2.23. Pour cette expérience les balises se situent loin du long couloir, ce qui implique que pour le début ainsi que pour la fin de cette expérience, aucune mesure RSS n'est reçue.

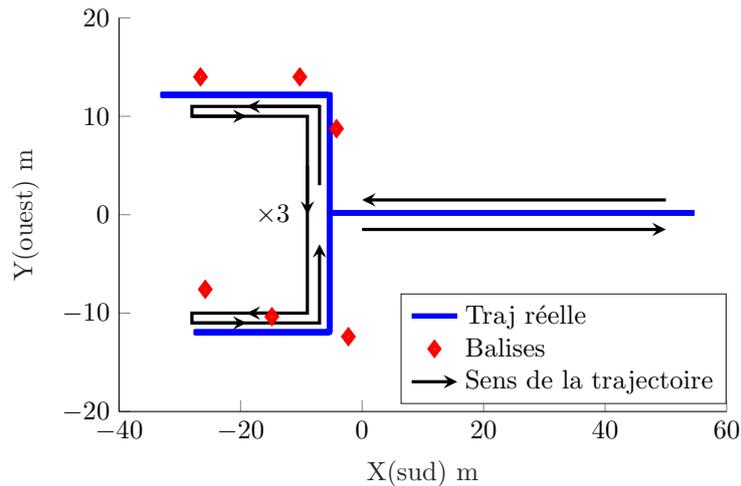


FIGURE 2.23 – Présentation de la trajectoire de référence pour l'exemple 2

Dans le but d'illustrer l'optimisation par morceaux il a été choisi une distance de rupture de 75m, impliquant un nombre de 7 ruptures additionnelles. Le choix de la distance de rupture dans un cadre général est discuté ultérieurement. Il y a 16 paramètres différents à estimer. Ces différentes ruptures artificielles servent ici à corriger la dérive lente de la trajectoire inertielle.

2.4.4 Estimation en temps différé de la trajectoire réelle

La figure 2.24 illustre les limites inhérentes de l'estimation en temps réel en absence de mesure RSS (figure en haut à droite). En effet, sur le long couloir, la trajectoire estimée en temps réel correspond à la trajectoire inertielle, aucune mesure RSS n'étant détectée. Ceci est bien corrigé par l'estimation en temps différé qui permet, au départ, d'aligner l'estimation avec la trajectoire réelle sur le long couloir, mais également à la fin de la trajectoire. En effet, des mesures RSS ont bien été reçues pour le premier segment, ce qui permet de corriger le cap initial. Par ailleurs, le gain apporté par l'utilisation du modèle adaptatif (figure en bas à droite), est significatif puisque la valeur de rmse est de moins de 4m pour le modèle adaptatif contre 11m pour le modèle global, figure en bas à gauche. On note également le gain obtenu entre la meilleure estimation temps différé et l'estimation en temps réel.

Trajectoire 2	rmse (m)	%d5 (%)
Estimation temps réel	6.56	71.64
Estimation via modèle global	11.76	40.54
Estimation via modèle adaptatif	3.69	85.73

TABEAU 2.2 – Évaluation de la fusion IMU sur une première trajectoire

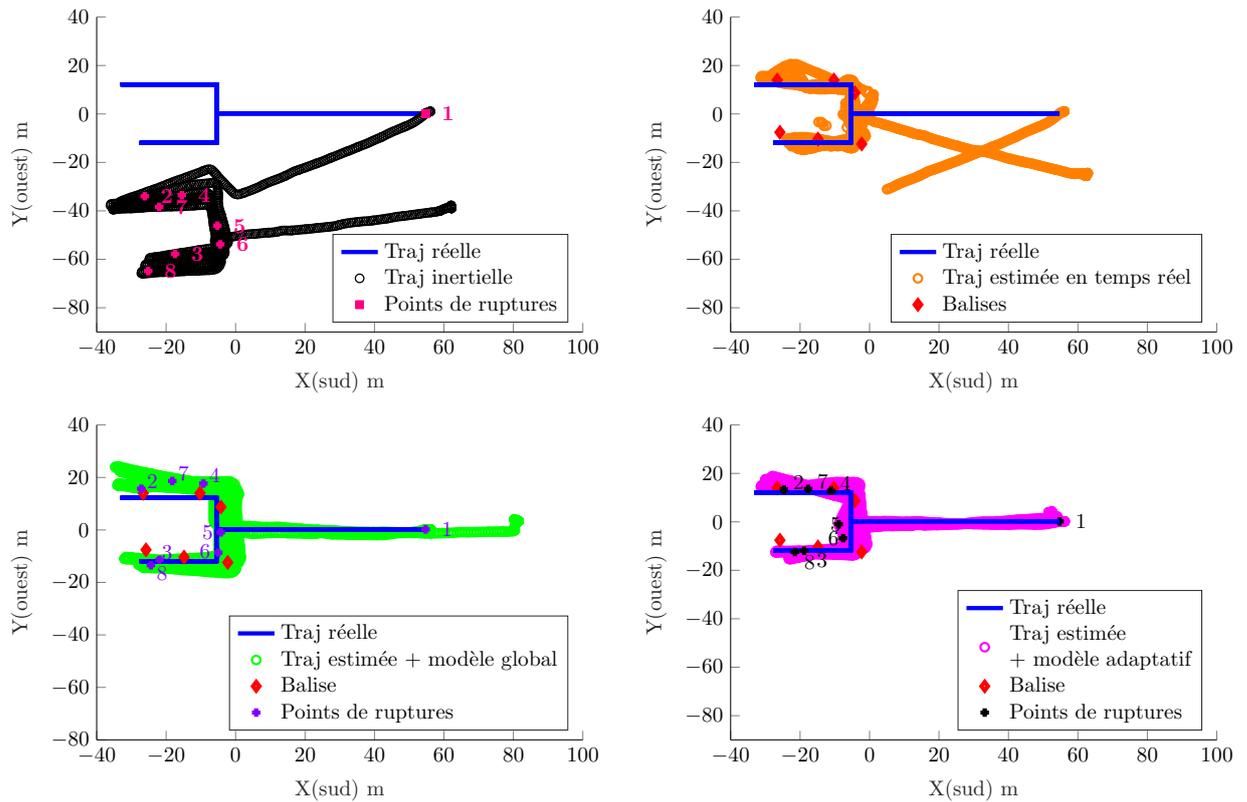


FIGURE 2.24 – Exemple de l’estimation finale en fusionnant les mesures RSS ainsi que les mesures IMU, appliquées au modèle adaptatif, à droite, et au modèle global à gauche

2.4.5 Résultats sur la base de données

Cette méthode est éprouvée sur un grand nombre de trajectoires décrites en 5.1.2 qui représentent 130 expériences sans changement d’étage.

Les résultats sont décrits selon deux paramètres : le nombre d’itérations total lié à l’algorithme d’optimisation ainsi que le nombre et la nature des ruptures artificielles. En effet, trois types différents de ruptures sont comparés : une rupture artificielle à chaque fois qu’une certaine distance est parcourue, ou bien un nombre fixe de ruptures artificielles réparties soit de manière uniforme, soit de manière aléatoire.

Au vu du tableau 2.3, on peut dans un premier temps noter l’importance du nombre d’itérations. Malheureusement, le paramètre i_{stop} est difficile à estimer et doit par conséquent être fixé au préalable. Il n’est pas possible en l’état de faire la différence entre une convergence dans un minimum local ou dans un minimum global. Pour un nombre de ruptures artificielles fixé, la répartition uniforme des points de rupture permet de façon globale une meilleure estimation que la répartition aléatoire. En effet des positions trop proches de ruptures artificielles peuvent impliquer des problèmes d’identification, et même détériorer l’estimation globale. Enfin, le choix d’une rupture artificielle à chaque fois qu’une distance a été parcourue est similaire au choix du nombre de ruptures fixes disposées de façon uniforme. Par ailleurs, la distance entre deux points de rupture a été fixée à 75m. Ce choix se justifie par le fait qu’une distance trop faible impliquerait un trop grand nombre de paramètres à estimer et donc pourrait causer des problèmes de conver-

i_{Stop}	Rupture artificielle	Moyenne %d5	Moyenne %d10	Moyenne rmse(m)
2000	1 (unif)	56,06	84,27	7,48
2000	2 (unif)	57,70	85,21	7,01
2000	3 (unif)	61,43	90,24	6,65
2000	3(aléa)	60,06	87,85	6,45
2000	4 (unif)	61,64	90,32	6,30
2000	4(aléa)	59,35	87,92	6,47
2000	6 (unif)	63,48	90,03	6,28
100	75m	56,15	81,05	8,63
500	75m	60,95	85,91	7,24
1000	75m	64,03	89,55	6,31
100	150m	56,10	83,58	7,40
500	150m	59,36	87,31	6,82
1000	150m	61,18	87,19	6,67
100	225m	52,79	78,84	8,58
500	225m	58,06	85,17	7,26
1000	225m	60,23	85,68	7,02
100	300m	53,04	79,81	8,32
500	300m	54,85	83,29	7,60
1000	300m	55,54	82,79	7,57
Estimation temps réel		35,88	67,47	12,45

TABEAU 2.3 – Résultats de l'estimation en temps différé selon le nombre d'itérations et le type de rupture artificielle sur un ensemble de 130 expériences

gence. Enfin, on note le gain significatif d'utiliser l'estimation en temps différé par rapport à l'estimation en temps réel, puisque la valeur de %d5 (respectivement rmse) passe d'environ 36% (respectivement 12.5m) en temps réel contre 64% (respectivement 6.3m) en temps différé.

2.5 Conclusion

Ce chapitre propose une estimation en temps différé utilisant les mesures RSS et les mesures IMU, sous réserve de connaître la position des balises. Dans la littérature, de nombreux modèles de fusion par filtrage en temps réel ont été éprouvés, en utilisant des approches par *fingerprinting* ou par un modèle de propagation d'ondes. L'approche choisie ici se distingue par le poids plus important accordé à la mesure inertielle brute. L'idée de cette approche est donc d'améliorer l'estimation inertielle par de simples transformations telles que des rotations ou homothéties par morceaux. Les paramètres de ces différentes transformations sont estimés par un maximum *a posteriori* basé sur les mesures de puissance RSS. Un modèle de propagation d'ondes est utilisé et une adaptation de ce modèle est proposée afin de pallier aux défauts impliqués par l'environnement *indoor*. En utilisant uniquement les mesures RSS et les mesures IMU, une meilleure estimation de la trajectoire en temps différé est proposée. Cette estimation est évaluée sur un nombre de 130 expériences pour lesquelles la moyenne de rmse est de 6.3m et la moyenne de %d5 est de 64% ce qui améliore les performances comparativement à une estimation en temps réel avec une moyenne de rmse de 12.5m et une moyenne de %d5

de 36%.

Différentes perspectives pourraient permettre d'affiner cette approche.

- Le nombre et la position des points de rupture sont fixés préalablement. Ceci implique une grande sensibilité sur la stratégie du choix associé à ces paramètres.
- L'adaptation du modèle a été faite de façon globale, c'est-à-dire sur l'ensemble des mesures RSS, avec la séparation wifi et bluetooth. Considérer une adaptation par balise pourrait ainsi améliorer l'estimation.
- Dans notre cas, chaque paramètre est considéré comme étant indépendant, le modèle *a priori* pourrait considérer différentes dépendances hiérarchiques.
- La présence des minima locaux est causée par la symétrie autour des balises. Retirer des balises de la modélisation durant l'optimisation à certaines étapes améliorerait potentiellement le processus de convergence.

D'un point de vue plus global, cette estimation en temps différé présente diverses caractéristiques.

- À partir de la trajectoire inertielle, un faible nombre de paramètres est nécessaire pour obtenir l'estimation en temps différé. Ces différents paramètres ont un lien hiérarchique sur l'estimation. De fait, une mauvaise estimation des premiers paramètres pourrait largement détériorer l'estimation.
- Cette optimisation considère uniquement des transformations basiques, des rotations ou des homothéties. Elle peut ne pas s'adapter à des formes de trajectoires inertielles de très mauvaise qualité qui pourraient nécessiter des transformations curvilignes.

2.6 Références

- BAHL, P. et V. N. PADMANABHAN. 2000, «RADAR : An in-building RF-based user location and tracking system», dans *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, Ieee, p. 775–784. 19
- BEAUREGARD, S. et H. HAAS. 2006, «Pedestrian dead reckoning : A basis for personal positioning», dans *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, p. 27–35. 21
- CHINTALAPUDI, K., A. PADMANABHA IYER et V. N. PADMANABHAN. 2010, «Indoor localization without the pain», dans *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, ACM, p. 173–184. 19
- CRASSIDIS, J. L., F. L. MARKLEY et Y. CHENG. 2007, «Survey of nonlinear attitude estimation methods», *Journal of guidance control and dynamics*, vol. 30, n° 1, p. 12. 24
- DAWES, B. et K.-W. CHIN. 2011, «A comparison of deterministic and probabilistic methods for indoor localization», *Journal of Systems and Software*, vol. 84, n° 3, p. 442–451. 19

- EVENNOU, F. et F. MARX. 2006, «Advanced integration of wifi and inertial navigation systems for indoor mobile positioning», *Eurasip journal on applied signal processing*, vol. 2006, p. 164–164. 26
- FINK, A., H. BEIKIRCH, M. VOSS et C. SCHRÖDER. 2010, «RSSI-based indoor positioning using diversity and inertial navigation», dans *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, IEEE, p. 1–7. 26
- FRIIS, H. T. 1971, «Introduction to radio and radio antennas», *IEEE spectrum*, vol. 8, n° 4, p. 55–61. 20
- HE, S. et S.-H. G. CHAN. 2016, «Wifi fingerprint-based indoor positioning : Recent advances and comparisons», *IEEE Communications Surveys & Tutorials*, vol. 18, n° 1, p. 466–490. 18
- HEURTEFEUX, K. et F. VALOIS. 2012, «Is RSSI a good choice for localization in wireless sensor network?», dans *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, IEEE, p. 732–739. 20
- HIGHTOWER, J., R. WANT et G. BORRIELLO. 2000, «SpotON : An indoor 3D location sensing technology based on RF signal strength», *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA*, vol. 1. vii, 21
- JAHN, J., U. BATZER, J. SEITZ, L. PATINO-STUDENCKA et J. G. BORONAT. 2010, «Comparison and evaluation of acceleration based step length estimators for handheld devices», dans *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, IEEE, p. 1–6. vii, 23
- KHODAYARI, S., M. MALEKI et E. HAMED. 2010, «A RSS-based fingerprinting method for positioning based on historical data», dans *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*, IEEE, p. 306–310. vii, 20
- KLEPAL, M., D. PESCH et collab.. 2007, «Influence of predicted and measured fingerprint on the accuracy of RSSI-based indoor location systems», dans *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*, IEEE, p. 145–151. vii, 19
- LADETTO, Q. 2003, *Capteurs et algorithmes pour la localisation autonome en mode pédestre*, thèse de doctorat, ENAC, Lausanne, doi :10.5075/epfl-thesis-2710. URL <http://vpaa.epfl.ch/page14975-fr.html>. 28
- LEE, M. et D. HAN. 2012, «Voronoi tessellation based interpolation method for wifi radio map construction», *IEEE Communications Letters*, vol. 16, n° 3, p. 404–407. 19
- LIU, T., M. CARLBERG, G. CHEN, J. CHEN, J. KUA et A. ZAKHOR. 2010, «Indoor localization and visualization using a human-operated backpack system», dans *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, IEEE, p. 1–10. 25
- LUO, X., W. J. O BRIEN et C. L. JULIEN. 2011, «Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites», *Advanced Engineering Informatics*, vol. 25, n° 2, p. 355–363. 20

- MADGWICK, S. O., A. J. HARRISON et R. VAIDYANATHAN. 2011, «Estimation of IMU and MARG orientation using a gradient descent algorithm», dans *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, IEEE, p. 1–7. 24
- MAUTZ, R. 2012, *Indoor positioning technologies*, thèse de doctorat. 21
- MAZUELAS, S., A. BAHILLO, R. M. LORENZO, P. FERNANDEZ, F. A. LAGO, E. GARCIA, J. BLAS et E. J. ABRIL. 2009, «Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks», *IEEE Journal of selected topics in signal processing*, vol. 3, n° 5, p. 821–831. 21
- MIROWSKI, P., D. MILIORIS, P. WHITING et T. KAM HO. 2014, «Probabilistic radio-frequency fingerprinting and localization on the run», *Bell Labs Technical Journal*, vol. 18, n° 4, p. 111–133. 20
- MORÉ, J. J. 1978, «The Levenberg-Marquardt algorithm : implementation and theory», dans *Numerical analysis*, Springer, p. 105–116. 42
- NIU, J., B. WANG, L. CHENG et J. J. RODRIGUES. 2015, «Wicloc : An indoor localization system based on wifi fingerprints and crowdsourcing», dans *Communications (ICC), 2015 IEEE International Conference on*, IEEE, p. 3008–3013. 19
- RACKO, J., P. BRIDA, A. PERTTULA, J. PARVIAINEN et J. COLLIN. 2016, «Pedestrian dead reckoning with particle filter for handheld smartphone», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–7. 22
- RENAUDIN, V., M. SUSI et G. LACHAPELLE. 2012, «Step length estimation using handheld inertial sensors», *Sensors*, vol. 12, n° 7, doi :10.3390/s120708507, p. 8507–8525, ISSN 1424-8220. URL <http://www.mdpi.com/1424-8220/12/7/8507>. 23
- SADI, F. et R. KLUKAS. 2013, «New jump trajectory determination method using low-cost MEMS sensor fusion and augmented observations for GPS/INS integration», *GPS solutions*, vol. 17, n° 2, p. 139–152. 25
- SKOG, I., P. HANDEL, J.-O. NILSSON et J. RANTAKOKKO. 2010, «Zero-velocity detection-An algorithm evaluation», *IEEE Transactions on Biomedical Engineering*, vol. 57, n° 11, p. 2657–2666. 23
- SUGANO, M., T. KAWAZOE, Y. OHTA et M. MURATA. 2006, «Indoor localization system using RSSI measurement of wireless sensor network based on ZigBee standard», *Target*, vol. 538, p. 050. 21
- SUN, Z., X. MAO, W. TIAN et X. ZHANG. 2009, «Activity classification and dead reckoning for pedestrian navigation with wearable sensors», *Measurement Science & Technology*, vol. 20, n° 1, p. 15 203. vii, 22
- SUSI, M., V. RENAUDIN et G. LACHAPELLE. 2013, «Motion mode recognition and step detection algorithms for mobile phone users», *Sensors*, vol. 13, n° 2, p. 1539–1562. 22
- VILLIEN, C. 2014, «Method and apparatus for detecting a manipulation of a portable device», US Patent App. 15/106,772. 25, 32
- WANG, B., Q. CHEN, L. T. YANG et H.-C. CHAO. 2016, «Indoor smartphone localization via fingerprint crowdsourcing : Challenges and approaches», *IEEE Wireless Communications*, vol. 23, n° 3, p. 82–89. 19

- WEINBERG, H. 2002, «Using the ADXL202 in pedometer and personal navigation applications», *Analog Devices AN-602 application note*, vol. 2, n° 2, p. 1–6. 23
- WILK, P., J. KARCIARZ et J. SWIATEK. 2015, «Indoor radio map maintenance by automatic annotation of crowdsourced wi-fi fingerprints», dans *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, IEEE, p. 1–8. vii, 26, 27
- WOODMAN, O. J. 2007, «An introduction to inertial navigation», cahier de recherche UCAM-CL-TR-696, University of Cambridge, Computer Laboratory. URL <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>. 21
- XIAO, W., W. NI et Y. K. TOH. 2011, «Integrated wifi fingerprinting and inertial sensing for indoor positioning.», dans *IPIN*, p. 1–6. 26
- ZHUANG, Y. et N. EL-SHEIMY. 2016, «Tightly-coupled integration of wifi and MEMS sensors on handheld devices for indoor pedestrian navigation», *IEEE Sensors Journal*, vol. 16, n° 1, p. 224–234. 26

Chapitre 3

Fusion en temps différé des mesures inertielles et RSS avec la carte des murs du bâtiment

Sommaire

3.1 État de l'art sur la prise en compte des murs d'un bâtiment	56
3.1.1 Contexte de la problématique de <i>map-matching</i>	56
3.1.2 État de l'art des méthodes de <i>map-matching</i> à l'intérieur d'un bâtiment en temps réel	57
3.1.3 État de l'art des méthodes <i>map-matching</i> en temps différé à l'intérieur d'un bâtiment	61
3.1.4 Synthèse de l'état de l'art	65
3.2 Filtre particulaire pour la localisation <i>indoor</i>	65
3.2.1 Théorie du filtre particulaire	65
3.2.2 Algorithme utilisé pour l'estimation en temps réel	68
3.3 Formalisation de l'estimation en temps différé	72
3.3.1 Multi-modalité et problème de continuité	72
3.3.2 Application du lissage sur les <i>clusters</i>	74
3.4 Méthodes de résolution de la problématique du chemin le plus probable	77
3.4.1 Algorithme de Viterbi	77
3.4.2 Approche Bayésienne	78
3.5 Résultats	80
3.5.1 Illustration sur une trajectoire avec des discontinuités	80
3.5.2 Résultats sur la base de données	83
3.5.3 Conclusion	84
3.6 Références	85

3.1 État de l'art sur la prise en compte des murs d'un bâtiment

La problématique de la géolocalisation à l'intérieur d'un bâtiment possède certaines spécificités. En effet, l'espace de déplacement d'un individu est contraint par la présence des murs ou encore par la disposition des cages d'escalier et des ascenseurs lorsque ce dernier souhaite changer d'étage. L'information de la position physique des murs ou des cages d'escalier est donc très précieuse. Par exemple, lorsqu'un changement d'altitude est détecté grâce à une mesure de pression, on sait que l'individu se trouve dans une cage d'escalier ou dans un ascenseur. De même, il devient possible de corriger l'estimation de la trajectoire afin de garantir que cette dernière ne soit pas en contradiction avec les murs du bâtiment. Cette problématique de résolution de contrainte de carte est appelée *map-matching*. Celle-ci peut être traitée en temps réel pour des applications de navigation par exemple, ou en temps différé lorsque l'on s'intéresse à d'autres approches qui ont pour objet d'ajuster une estimation afin qu'elle respecte les contraintes du bâtiment permettant d'améliorer l'estimation. Par exemple, en forçant l'estimation à se déplacer dans un couloir on peut s'affranchir des estimations de trajectoires incorrectes qui présentent des déplacements physiquement impossibles à cause des murs. Après une rapide présentation du contexte général de la problématique de *map-matching*, nous développerons l'état de l'art relatif à son application au cas de la localisation en temps réel à l'intérieur des bâtiments. Enfin, seront développées les différentes approches de *map-matching* en temps différé à l'intérieur d'un bâtiment.

3.1.1 Contexte de la problématique de *map-matching*

3.1.1.1 Application en environnement *outdoor* : l'exemple du GPS

Les méthodes de *map-matching* ont été largement développées dans le cadre de la navigation en environnement *outdoor* pour les automobiles. En effet, il est possible que la position calculée par le GPS ne coïncide pas avec la position des routes. Ceci peut notamment se produire lorsque les mesures GPS n'ont pas été disponibles pendant une certaine période, lorsqu'une voiture traverse un tunnel par exemple. Il peut être utile de corriger une trajectoire en exploitant l'information apportée par la carte. La particularité du cas de la navigation automobile est que, contrairement au cas *indoor*, la largeur des routes est relativement faible au regard d'une voiture ainsi que des besoins de positionnement. De plus, le déplacement d'une voiture est très contraint. En effet, la voiture avance le long d'une route, recule ou change de voie, mais peut difficilement avancer de façon perpendiculaire à cette route au contraire de la navigation *indoor* qui autorise tous les déplacements possibles. Les différentes approches de *map-matching* développées pour un environnement extérieur peuvent être globalement séparées en 3 différentes catégories : l'approche géométrique, l'approche topologique ou l'approche probabiliste. La figure 3.1 présente l'exemple d'une position estimée n'étant pas située sur le réseaux routier.

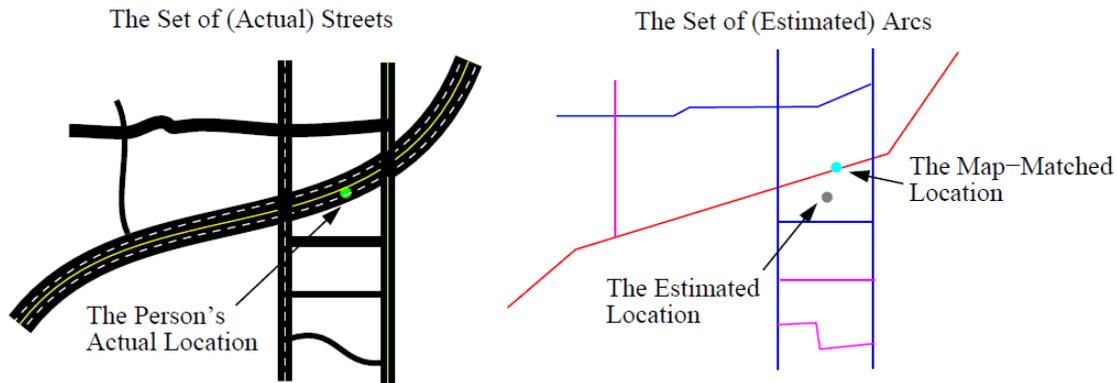


FIGURE 3.1 – Présentation de la problématique de *map-matching outdoor* selon BERNSTEIN et KORNHAUSER [1998]

3.1.2 État de l’art des méthodes de *map-matching* à l’intérieur d’un bâtiment en temps réel

Dans notre cas, l’ensemble des informations contenues dans une carte sont connues, telles que les positions des murs, les dispositions des salles, la présence des cages d’escalier, des ascenseurs ainsi que le nombre d’étages. Par ailleurs, les positions des balises wifi et bluetooth sont également des données d’entrée supposées connues.

Dans la littérature, la problématique de la navigation *indoor* concerne très majoritairement le cas de l’estimation temps réel de la trajectoire. Bien que notre cas d’usage ne soit pas contraint par le temps réel, nous décrivons malgré tout les approches adressant cette problématique car elles sont très nombreuses dans la littérature et peuvent également répondre ou servir de base à notre problème. Intégrer l’information des murs pour la navigation *indoor* permet indéniablement d’améliorer l’estimation en temps réel. Néanmoins, l’estimation de la trajectoire devient bien plus difficile pour deux raisons principales :

- L’information des murs implique des contraintes dans le déplacement d’un individu, celui-ci ne pouvant pas traverser un mur. L’intégration de ces contraintes dans la modélisation peut se révéler analytiquement fastidieuse. En effet, l’utilisation de contraintes sur les murs ne permet pas de définir une fonction analytique dérivable puisqu’il s’agit de vérifier s’il y a une intersection entre deux segments. Le premier segment correspond à chaque mur, et le deuxième segment au déplacement entre deux positions successives.
- De plus, la prise en compte de ces contraintes impliquera que la distribution de la loi *a posteriori* de la position sera multi-modale. Par exemple, si l’individu arrive à une jonction de deux couloirs, il y aura deux possibilités distinctes pour l’estimation. Tenir compte de cette double possibilité se traduit par une multi-modalité de la loi *a posteriori*, chacun des modes étant dans un couloir propre.

L’application des contraintes sur les murs, de même que la multi-modalité justifient alors de mettre en oeuvre une approche de filtrage plus complexe. On pourra alors utiliser un filtre particulière (FP) afin d’estimer la trajectoire en temps réel, tout en intégrant l’information des murs. Un filtre particulière, dont on présentera la théorie dans la partie 3.2.1 est une méthode de filtrage basée sur une estimation par Monte-Carlo séquentielle. Il consiste à simuler un grand nombre de particules, un poids est attribué à chacune de

ces particules en fonction des différents modèles, puis une estimation de la loi *a posteriori* est donnée en fonction de ces particules pondérées. Deux approches principales se distinguent dans l'état de l'art pour l'intégration d'une carte à l'aide d'un filtre particulaire.

- Les particules sont simulées sans tenir compte de la carte. Ensuite, chaque particule qui rencontre un mur sera supprimée.
- Les particules sont simulées uniquement sur des positions ne permettant pas de violer les contraintes de la carte.

3.1.2.1 Correction des particules avec les murs

Lorsque l'on représente les déplacements non autorisés d'une carte, en l'occurrence les murs, on parlera alors de carte négative. Pour cette représentation, pour chaque mise à jour effectuée par l'individu, les particules sont préalablement propagées selon le modèle d'état sans tenir compte des plans physiques du bâtiment. Ensuite, une vérification est effectuée afin de savoir quelles sont les particules qui entrent en conflit avec les murs. Les particules conflictuelles sont alors corrigées, voire supprimées, permettant de garantir une cohérence avec la carte. Dans la figure 3.2, on considère une position initiale dans un coin de la pièce. On considère également une incertitude totale sur le cap de déplacement initial. Des particules sont propagées dans toutes les directions. Les particules rencontrant un mur sont alors supprimées (en rouge dans la figure).

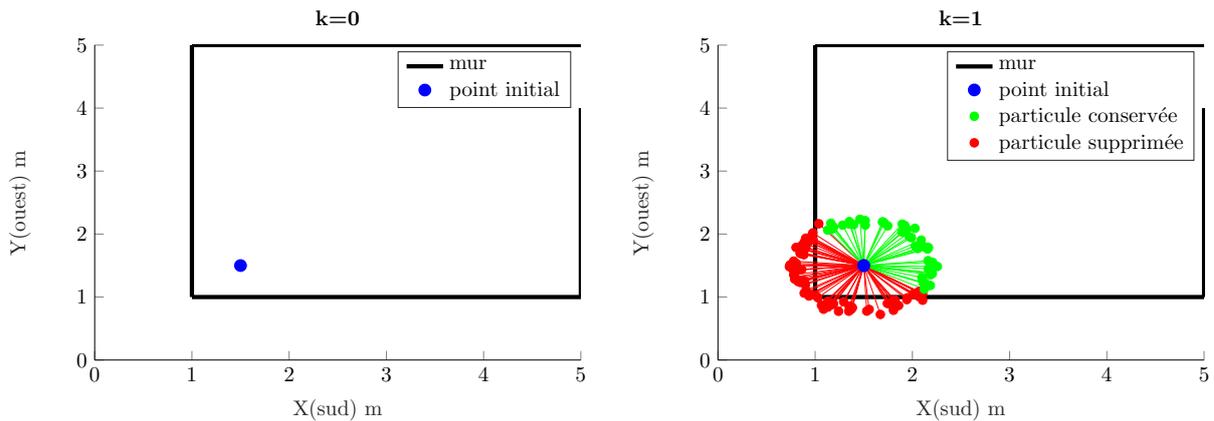


FIGURE 3.2 – Comportement des particules lors d'un conflit avec un mur

L'approche de la représentation négative nécessite de vérifier pour chaque particule si celle-ci a traversé un mur. Le nombre de vérifications est donc, à chaque instant, de $n_p \times n_{mur}$ où n_p (respectivement n_{mur}) est le nombre de particules (respectivement de murs). Néanmoins, il est possible de limiter le nombre de vérifications en considérant uniquement les murs se situant à un voisinage raisonnable de la particule, par exemple, il ne sera pas utile de vérifier si la particule a rencontré un mur se trouvant à 20m. Cette approche sur la vérification des particules peut se traduire sur la pondération associée aux particules par la formule suivante :

$$\omega_k^{(i)} = \begin{cases} 0 & \text{si la particule est en conflit avec un mur} \\ \frac{1}{n_k} & \text{sinon} \end{cases} \quad (3.1)$$

où n_k correspond au nombre de particules n'étant pas en conflit avec les murs à l'instant k , et $\omega_k^{(i)}$ étant le poids de la particule i , comme proposé dans [BEAUREGARD et collab. \[2008\]](#).

3.1.2.2 Discrétisation de l'espace

Discrétisation sur un graphe

D'autres approches intègrent la carte de façon implicite en réduisant l'espace d'état aux positions et transitions autorisées par celle-ci. Par exemple, [EBNER et collab. \[2015\]](#) modélisent la carte sous forme de grille. Dans ce cas, la grille représente un échantillonnage discret des positions possibles. Les particules sont propagées sur la grille uniquement selon les déplacements permis et il n'est plus nécessaire d'appliquer la vérification des particules par rapport aux murs, comme pour l'approche précédente. Néanmoins, discrétiser l'espace entraîne une quantification de l'estimation au pas de la grille, ce pas ne doit pas être trop important. De même, considérer une discrétisation trop fine augmentera largement le coût de calcul. Dans [EBNER et collab. \[2015\]](#), représenté dans la figure 3.3 (à gauche), les noeuds sont les centres des carrés formant une grille régulière de l'espace. La longueur des côtés de ces carrés est de 20cm, afin d'assurer une grande diversité des positions possibles. A chaque noeud est également associé l'ensemble des transitions possibles. Dans [HILSENBECK et collab. \[2014\]](#), l'approche diffère dans la modélisation de l'espace. Dans un premier temps, un squelette du bâtiment est extrait, c'est-à-dire que chaque couloir est représenté par une droite, comme dans la figure 3.3 (figure du milieu), les intersections de ces droites correspondant aux endroits où les couloirs se croisent. Par ailleurs, certaines approches basées sur de l'analyse d'image tendent à automatiser la création d'un tel graphe uniquement à partir de l'image des plans physiques du bâtiment [SCHMITT et collab. \[2016\]](#). Par la suite, les espaces larges sont alors discrétisés selon une grille dont la largeur de maille correspond environ à la longueur moyenne d'un pas. Pour les deux exemples cités, le modèle d'état est donné par une marche aléatoire sur la grille.

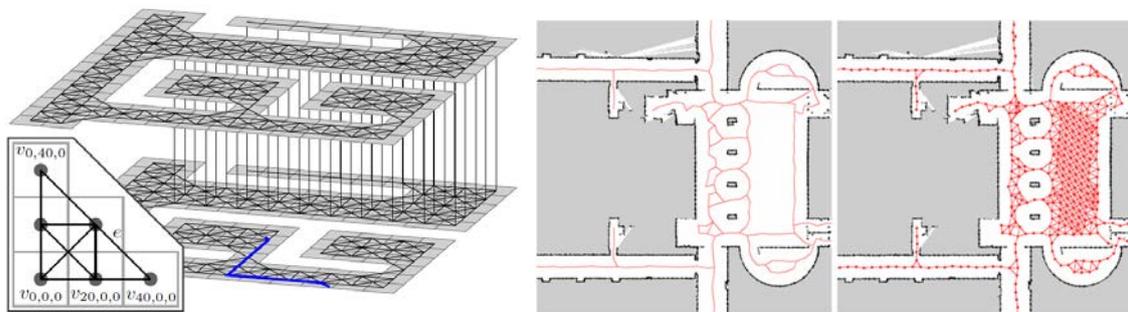


FIGURE 3.3 – Représentation de la discrétisation selon [EBNER et collab. \[2015\]](#) à gauche, et selon [HILSENBECK et collab. \[2014\]](#) à droite.

Discrétisation sur une grille

L'approche présentée dans [LAMY-PERBAL et collab. \[2015\]](#) ou encore dans [GALČÍK et OPIELA \[2016\]](#) est très similaire, mais propose ici une discrétisation sur une grille comme représenté dans la figure 3.4. Ici, la position est donnée par un des éléments de la grille, en l'occurrence un carré, ou un hexagone. Dans cette modélisation, aucun filtre particulière n'a été utilisé. En revanche, à chaque instant, chaque élément de la grille est pondéré selon les différents modèles. L'estimation de la position est alors finalement donnée par une moyenne pondérée. De façon similaire à une discrétisation par graphe, le pas de discrétisation influera directement sur la précision d'estimation, mais également sur le coût de calcul associé.

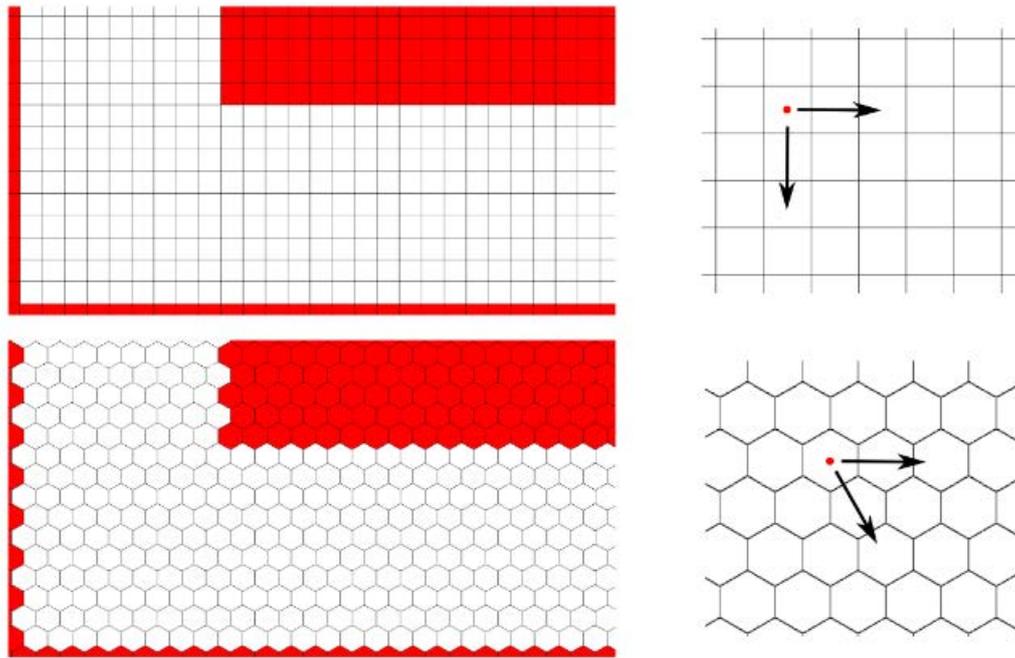


FIGURE 3.4 – Représentation d’une discrétisation en grille selon GALČÍK et OPIELA [2016] avec deux formes différentes.

3.1.2.3 Comparaison des approches

Ces deux approches mettent en avant le compromis entre le coût de calcul nécessaire et la dimension de l’espace d’état. Ce compromis sera mis en avant en s’intéressant à la localisation d’un individu dans un étage d’un bâtiment fictif d’une surface de $500m^2$. De plus, le vecteur d’état qui sera considéré a été décrit dans la partie 2.2.

Le premier point de vue simule les particules sans considérer les murs. Il faut donc effectuer un nombre de $n_p \times n_{mur}$ vérifications. En considérant un nombre de murs environ égal à 100 et le nombre de particules, environ à 500, ceci conduit à un nombre de 5×10^4 vérifications. En revanche, l’espace d’état correspond à un compact de \mathbb{R}^4 . Il n’est pas nécessaire d’allouer de l’espace par rapport à l’espace d’état. Le coût de calcul est important, mais la dimension de l’espace d’état est très réduite.

Le second point de vue discrétise l’espace afin de propager les particules uniquement selon des déplacements autorisés. Aucune vérification n’est donc nécessaire. Toutefois, il faut discrétiser l’espace d’état. Si on considère à nouveau un étage de $500m^2$ pour atteindre une précision à 50cm sur la précision, à 5° sur le biais en angle et de 0.02 sur le biais de longueur de pas, alors la taille de la grille sera environ de $50 \times 50 \times 72 \times 25 \approx 4.5 \times 10^6$. Pour cette approche, il n’y a aucun coût de calcul, mais la taille de mémoire nécessaire pour l’espace d’état est importante dans la mesure où les transitions entre chaque noeud sont également stockées.

3.1.2.4 Approche intermédiaire

ALAOUI et collab. [2016] adoptent une approche intermédiaire par rapport aux précédentes dans laquelle les solutions ne sont que partiellement discrétisées. En effet, la carte est modélisée par un graphe, où les arcs correspondent aux couloirs du bâtiment, et les noeuds aux intersections. Ici, la position est représentée par une abscisse linéaire le long d'un arc. Différemment de la discrétisation par graphe, le déplacement des particules n'est pas limité aux noeuds du graphe de déplacement, les particules pouvant se propager le long des arcs. Un exemple de cette approche est présenté dans la figure 3.5. Cette approche permet alors un gain de coût de calcul, puisque le nombre de possibilités est réduit mais le déplacement reste limité aux déplacements autorisés par le graphe. Dans les espaces larges, un graphe régulier est appliqué.

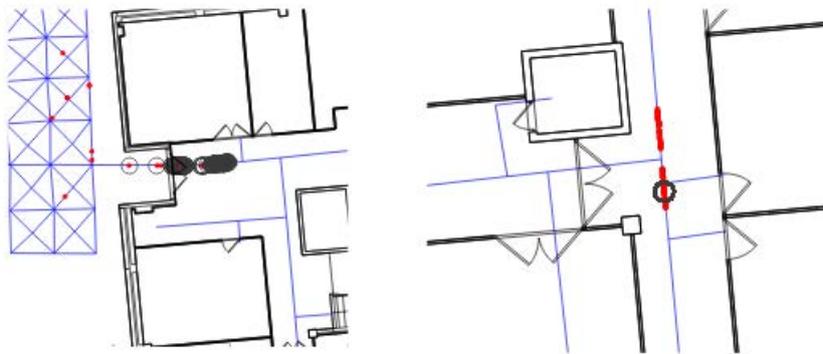


FIGURE 3.5 – Représentation d'une approche intermédiaire selon ALAOUI et collab. [2016]. Sur la figure de gauche, un graphe régulier est utilisé en environnement *outdoor* pour augmenter les possibilités de déplacement.

3.1.3 État de l'art des méthodes *map-matching* en temps différé à l'intérieur d'un bâtiment

L'ensemble des approches de *map-matching* en temps différé à l'intérieur d'un bâtiment dont nous avons connaissance s'organisent en deux axes :

- Faire correspondre la trajectoire inertielle, qui sera linéaire par morceaux, avec les plans de la carte
- Appliquer des approches de lissage particulière suite à l'utilisation d'un filtre particulaire en temps réel

3.1.3.1 Mise en correspondance de la trajectoire inertielle avec une carte

Le principe de cette approche est d'identifier la meilleure correspondance entre un graphe modélisant le bâtiment et un graphe représentant la trajectoire inertielle. La première étape consiste à segmenter la trajectoire inertielle parcourue et à créer un graphe associé. Le graphe représentant la trajectoire inertielle est construit par segmentation : les lignes droites sont représentées par des arcs, les changements de direction par des noeuds. Ici, il est supposé que tous les déplacements se font en ligne droite. Un virage est détecté lorsque le cap de l'utilisateur varie au-delà d'un certain seuil. Un exemple est fourni par la figure 3.6.

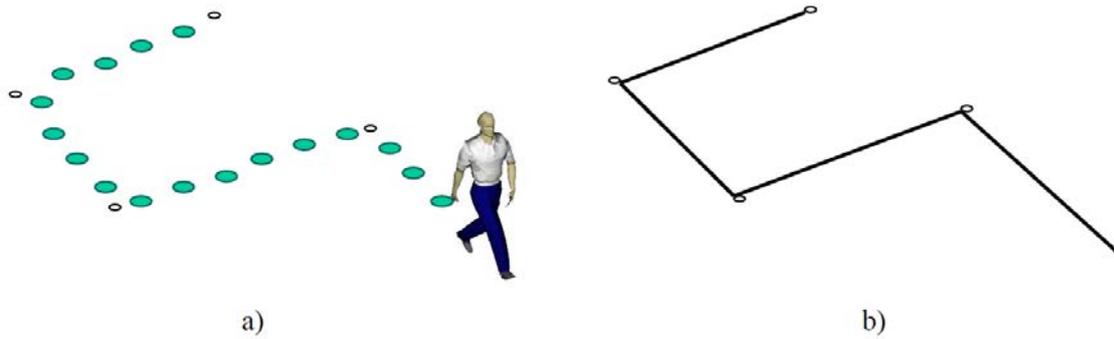


FIGURE 3.6 – Illustration de la segmentation de la trajectoire selon SPASSOV [2007]

Le graphe décrivant la carte considère uniquement la topologie du bâtiment, c'est-à-dire que les couloirs sont représentés par les arcs du graphe, et les points de croisement par les noeuds. Deux exemples sont donnés dans la figure 3.7.

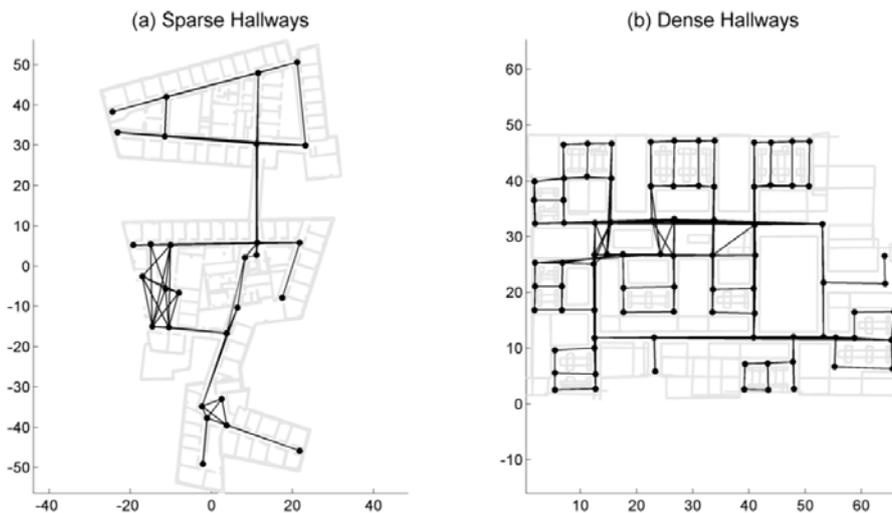


FIGURE 3.7 – Représentation de la carte topologique pour deux bâtiments selon WANG et collab. [2013]

Dans SPASSOV [2007], la problématique traitée considère la mise en correspondance de ces deux graphes, mais également la détermination du point de départ de la trajectoire. En effet, lorsque la trajectoire est suffisamment longue et riche, il est généralement possible de trouver une correspondance unique sans contraindre le point de départ. Ces travaux définissent un modèle de Markov à espace d'état. Le modèle d'état est basé sur la connectivité du graphe, pour deux arcs e_i et e_j du graphe de la carte, on a :

$$f(s_{k+1} = e_i | s_k = e_j) = \begin{cases} 0 & \text{si les arcs } e_i \text{ et } e_j \text{ ne sont pas connectés.} \\ \frac{1}{n(e_j)} & \text{sinon} \end{cases} \quad (3.2)$$

où $n(e_j)$ correspond au nombre d'arcs connectés à e_j , et s_k correspond au k -ème arc du graphe de la trajectoire inertielle. Cette valeur peut être affinée, en fonction du sens de déplacement de la trajectoire, mais également selon d'autres facteurs, tels que l'accessibilité de certains couloirs ou de certaines pièces. Ensuite, le modèle d'observation permet de mesurer la correspondance entre le segment s_k de la trajectoire inertielle avec l'arc e_j ,

$h_k(e_j|s_k)$. Cette correspondance est basée sur des considérations de distance et d'angle. A chaque étape sont définis des poids de vraisemblance notés ω_k^i basés sur cette correspondance pour chaque arc. A l'état initial, chaque arc e_i se voit attribuer une probabilité basée sur la correspondance avec le premier segment $\omega_1^i = h_1(e_i|s_1)$. On note $V(e_i)$ l'ensemble des voisins de cet arc e_i . Le poids pour que le segment s_k soit associé à l'arc e_i vaut alors :

$$\omega_k^i = h_k(e_i|s_k) \sum_{e_j \in V(e_i)} \omega_{k-1}^j f(s_k = e_i | s_{k-1} = e_j) \quad (3.3)$$

Cette probabilité dépend de la correspondance entre le segment s_k et l'arc de la carte e_i , mais également de l'accessibilité par rapport aux arcs voisins, pondérés par leur probabilité de correspondre au segment précédent s_{k-1} . Cette vraisemblance est mesurée pour le premier segment, puis pour les deux premiers segments. Il faut attribuer à chaque couple d'arc voisins la vraisemblance que les deux premiers segments de la trajectoire inertielle correspondent etc. Ce problème, défini comme tel, est combinatoire. Mais à chaque étape, les poids des arcs inférieurs à un certain seuil sont fixés à 0. Cette stratégie permet de conserver uniquement les arcs probables, jusqu'à ce qu'une seule solution soit possible. Cette solution correspond alors à l'estimation de la trajectoire *map-matchée*. [WANG et collab. \[2015\]](#) considèrent une approche similaire mais n'estiment pas le point de départ qui est une donnée connue. Un exemple est présenté dans la figure 3.8.

Toutefois, cette approche présente certaines limites. Dans un premier temps, la trajec-

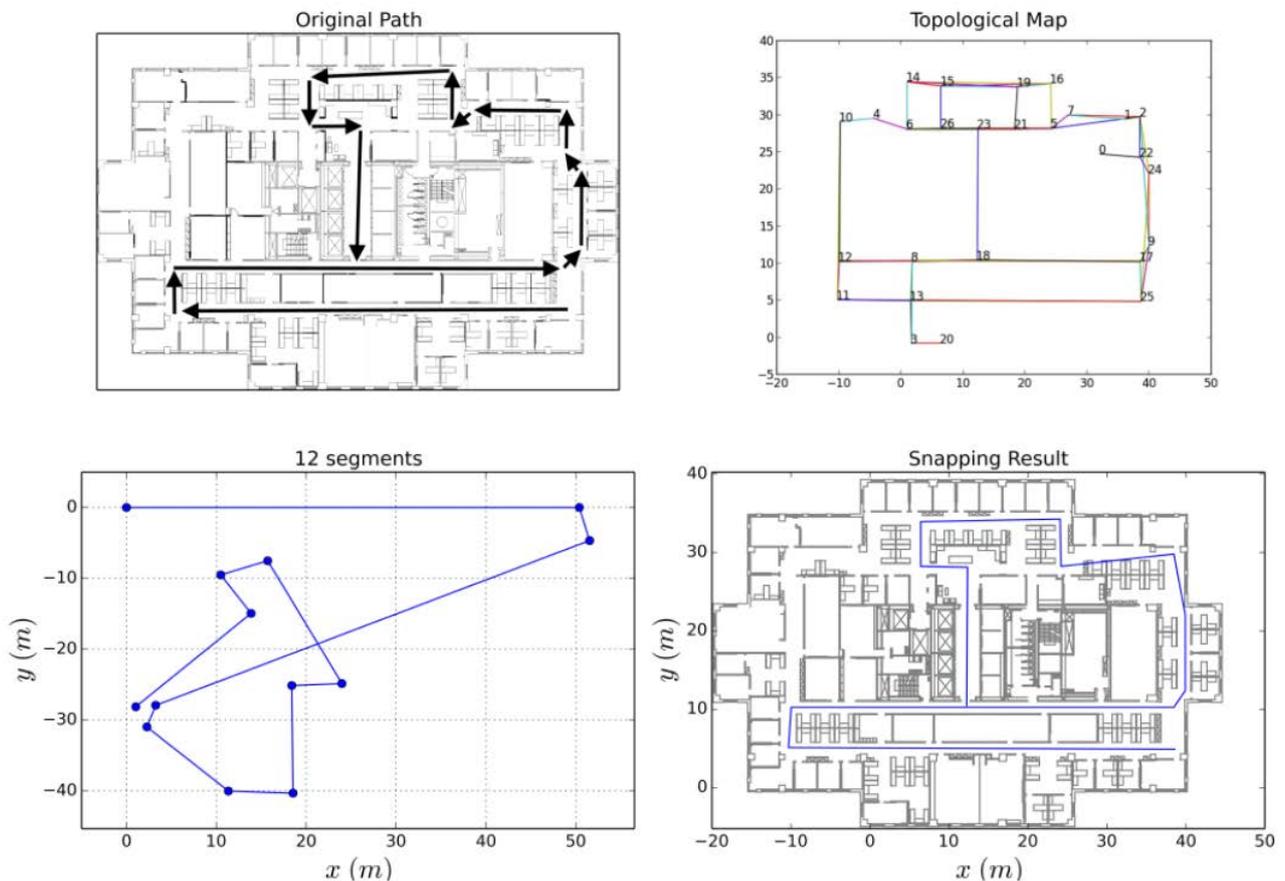


FIGURE 3.8 – Illustration de l'algorithme de *map matching* en environnement indoor selon [WANG et collab. \[2015\]](#)

toire inertielle peut être de mauvaise qualité, et la modélisation linéaire par morceaux

peut s'avérer mauvaise. Par ailleurs, en considérant un graphe sur la carte, les déplacements d'un individu sont uniquement limités aux noeuds du graphe, il ne pourra pas s'arrêter au milieu d'un couloir ou entrer dans une salle. Enfin, cette approche est séquentielle, c'est à dire qu'un couloir est attribué à un premier segment de la trajectoire. Puis, un autre couloir est attribué pour le deuxième segment de la trajectoire, en fonction de l'estimation du couloir précédent. Si l'un des couloirs est mal estimé, alors cette erreur se répercutera sur l'ensemble de la trajectoire.

3.1.3.2 Approche de lissage particulaire

Comme développé dans la sous-section 1.3.2, le lissage permet une formalisation immédiate de la problématique du temps différé. Différentes approches pour la navigation *in-door* développent des méthodes de lissage basées sur les particules estimées en temps réel. [KLEPAL et collab. \[2008\]](#) proposent une estimation lissée et basée sur une fenêtre glissante de taille h , ce qui définit la loi *a posteriori* $P(x_k|x_{1:k+h})$. Ceci signifie que les mesures reçues à l'instant $k + h$ permettent d'affiner l'estimation passée à l'instant k . La figure 3.9 présente une illustration de cette approche. Dans cet exemple, sur la figure en haut à gauche, certaines particules n'ont pas suivi la bonne hypothèse. Lorsque ce groupe de particules entre en collision avec un mur, celles-ci sont supprimées, de même que l'historique de ces particules sur la fenêtre considérée (figure en haut à droite). Enfin, une trajectoire est uniquement évaluée sur les particules qui n'auront pas été supprimées (figure du bas). Par ailleurs, différents algorithmes de lissage particulaire qui seront décrits dans la partie 3.4.2 sont mis en oeuvre dans d'autres approches, [NURMINEN et collab. \[2013\]](#) appliquent le lissage sur l'historique totale de la trajectoire, lorsque [FETZER et collab. \[2016\]](#) présente une comparaison de l'application du lissage sur une fenêtre glissante ou sur l'ensemble de la trajectoire.

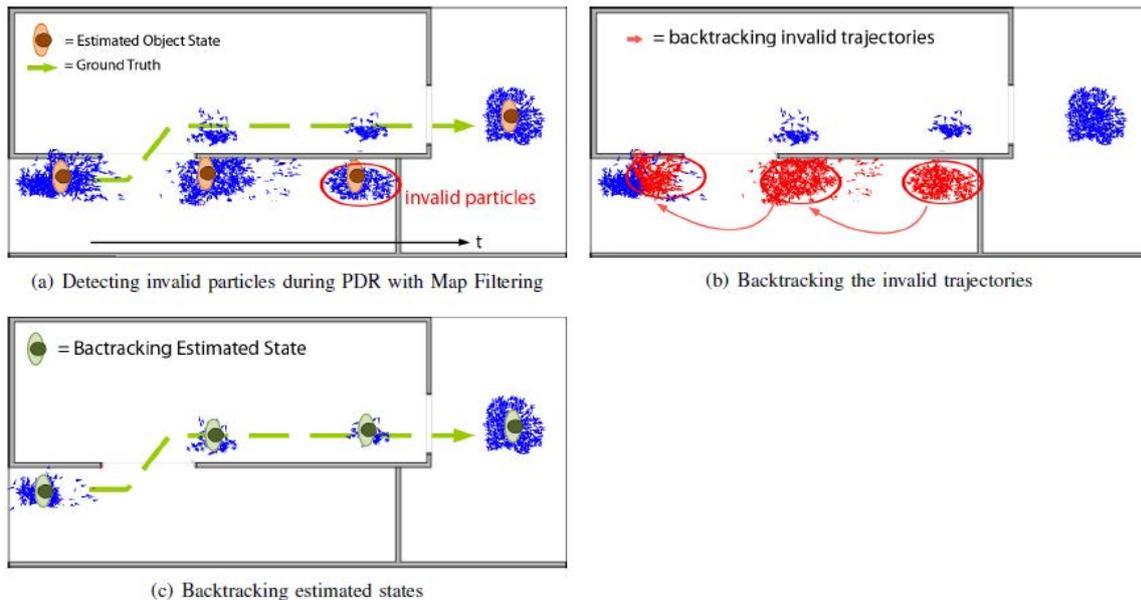


FIGURE 3.9 – Présentation d'une approche de lissage particulaire, tirée de [KLEPAL et collab. \[2008\]](#)

3.1.4 Synthèse de l'état de l'art

La technique du *map-matching* permet d'améliorer significativement les performances d'estimation. En effet, la connaissance de la position des murs du bâtiment limite grandement le déplacement possible d'un individu en imposant des contraintes de déplacement. Toutefois, la problématique devient complexe car il faut savoir intégrer ces contraintes dans l'estimation. De plus, la loi de la position de recherche est alors multi-modale, et les modèles présentés pour l'estimation sans les murs ne sont pas suffisants.

En temps réel, on peut alors mettre en oeuvre un filtre particulière afin d'intégrer l'information de la carte de deux façons : en propageant un ensemble de particules puis en supprimant celles qui sont entrées en collision avec un mur, ou bien en discrétisant l'espace de recherche, et en ne propageant uniquement les particules selon des déplacements permis par les murs. L'utilisation d'un filtre particulière nécessite un coût de calcul important pour la vérification des particules avec chaque mur, dans le premier cas, ou encore le coût et la mémoire nécessaires pour la discrétisation de l'espace dans le second cas. Cette limite de coût est similaire à une autre approche d'estimation bayésienne discrète qui utilise une discrétisation par une grille. Enfin, une limite inhérente à l'estimation en temps réel est qu'il n'est pas possible de garantir des transitions pertinentes à chaque pas. L'exemple de la figure 3.9 (en haut à droite), illustre bien cette limite. Lorsque les particules autour de l'estimation courante rencontrent un mur à un instant t , celles-ci sont supprimées puis échantillonnées à nouveau près des autres particules. Sa position à l'instant $t + 1$ est bien correcte, mais la transition entre l'instant t et $t + 1$ implique un passage à travers les murs. Par ailleurs, un passage d'une position incorrecte à un instant t à une position correcte à un instant $t + 1$ peut même impliquer une transition de l'ordre d'une dizaine de mètres. La trajectoire estimée en temps réel n'est donc pas garantie d'être continue.

En temps différé, à notre connaissance, les méthodes de *map-matching* se distinguent selon deux approches. Une première approche traite de la correspondance entre le graphe de la trajectoire inertielle et le graphe du bâtiment. Une seconde approche applique des méthodes de lissage particulière. Nos travaux seront basés sur la seconde approche en temps différé : celle du lissage particulière. En effet, l'approche par correspondance de graphe se trouve être trop contraignante sur le déplacement d'un individu. On présentera dans un premier temps un filtre particulière pour l'estimation en temps réel. Ce filtre particulière était déjà implémenté au CEA et n'entre pas dans les travaux de cette thèse.

3.2 Filtre particulière pour la localisation *indoor*

3.2.1 Théorie du filtre particulière

Le filtre particulière est une méthode numérique permettant d'estimer une loi de filtrage. Cette approche est basée sur des méthodes de Monte-Carlo. En effet, la loi de filtrage pour des cas complexes ne peut pas être exprimée de façon analytique, celle-ci faisant intervenir des calculs d'intégrales que l'on ne sait pas résoudre. L'ensemble de la théorie du filtre particulière, de même que certains exemples peuvent se trouver dans [DE FREITAS et collab. \[2001\]](#).

3.2.1.1 Méthode de Monte-Carlo

Supposons que l'on cherche à estimer pour toute fonction $u()$ intégrable :

$$I(u) = E(u(x_{0:k})) = \int u(x_{0:k})P(x_{0:k}|y_{1:k})dx_{0:k} \quad (3.4)$$

On suppose qu'il est possible de simuler un échantillon indépendant et identiquement distribué selon la loi exacte de recherche *a posteriori* $P(x_{0:k}|y_{1:k})$. On notera cet échantillon $(x_{0:k}^{(1)}, \dots, x_{0:k}^{(n)})$ de taille n . Par la loi des grands nombres, on obtient alors directement une estimation non-biaisée, dont on peut également garantir une normalité asymptotique par le théorème de la limite Centrale :

$$\hat{I}(u) = \frac{1}{n} \sum_{i=1}^n u(x_{0:k}^{(i)}) \quad (3.5)$$

Lorsque $u()$ est la fonction identité, alors $\hat{I}(u)$ correspond à l'espérance conditionnelle.

3.2.1.2 Échantillonnage préférentiel

Dans la plupart des cas d'application, il n'est pas possible de simuler directement selon cette loi de recherche. On définit alors une loi dite préférentielle $\pi(x_{0:k}|y_{1:k})$, également appelée loi de proposition, dont le support inclut celui de la loi de recherche. En effet, la simulation sera faite par la loi de proposition puis sera ajustée pour correspondre à la loi de recherche. Il est possible de réécrire l'équation 3.4 comme dans l'équation 3.6, le dénominateur étant égal à 1 :

$$I(u) = \frac{\int u(x_{0:k})P(x_{0:k}|y_{1:k})dx_{0:k}}{\int P(x_{0:k}|y_{1:k})dx_{0:k}} \quad (3.6)$$

Cette réécriture permet une factorisation faisant intervenir la loi préférentielle en définissant des poids préférentiels :

$$\omega(x_{0:k}) = \frac{P(x_{0:k}|y_{1:k})}{\pi(x_{0:k}|y_{1:k})} \quad (3.7)$$

on peut alors écrire :

$$I(u) = \frac{\int u(x_{0:k})\omega(x_{0:k})\pi(x_{0:k}|y_{1:k})dx_{0:k}}{\int \omega(x_{0:k})\pi(x_{0:k}|y_{1:k})dx_{0:k}} \quad (3.8)$$

En simulant un nouvel échantillon, selon la loi de proposition $\pi(x_{0:k}|y_{1:k})$, qui sera noté $(x_{0:k}^{(1)}, \dots, x_{0:k}^{(n)})$ de taille n et en utilisant 3.8, il est possible de proposer un nouvel estimateur empirique calculé en simulant d'après la loi $\pi(x_{0:k}|y_{1:k})$:

$$\hat{I}(u) = \frac{\frac{1}{n} \sum_{i=1}^n u(x_{0:k}^{(i)})\omega(x_{0:k}^{(i)})}{\frac{1}{n} \sum_{i=1}^n \omega(x_{0:k}^{(i)})} = \sum_{i=1}^n u(x_{0:k}^{(i)})\tilde{\omega}_k^{(i)} \quad (3.9)$$

où $\tilde{\omega}_k^{(i)}$ correspondent aux poids préférentiels normalisés.

$$\tilde{\omega}_k^{(i)} = \frac{\omega(x_{0:k}^{(i)})}{\sum_{i=1}^n \omega(x_{0:k}^{(i)})} \quad (3.10)$$

On peut également proposer une estimation empirique de la loi *a posteriori*, basée sur l'échantillonnage préférentiel, en notant $\delta(x)$ la loi d'un Dirac sur x :

$$\hat{P}(x_{0:k}|y_{1:k}) = \sum_{i=1}^n \tilde{\omega}_k^{(i)} \delta(x_{0:k}^{(i)}) \quad (3.11)$$

3.2.1.3 Échantillonnage préférentiel séquentiel

L'échantillonnage préférentiel est la forme générale de la méthode de Monte-Carlo, cependant celle-ci n'est pas adaptée à des problèmes récurrents. En effet, la loi de proposition nécessite ici de considérer à chaque instant k l'ensemble de l'historique des mesures $y_{1:k}$. Chaque fois qu'une nouvelle mesure y_{k+1} devient disponible, les poids préférentiels doivent être recalculés sur la séquence entière des mesures. La complexité de cette opération augmente avec le temps. Lorsque la loi x_k est markovienne, une optimisation du calcul est possible. On peut en effet écrire :

$$\pi(x_{0:k}|y_{1:k}) = \pi(x_{0:k-1}|y_{1:k})\pi(x_k|x_{0:k-1}, y_{1:k}) \quad (3.12)$$

En itérant sur k , on obtient alors :

$$\pi(x_{0:k}|y_{1:k}) = f_0 \prod_{j=1}^k \pi(x_j|x_{0:j-1}, y_{1:k}) \quad (3.13)$$

La dépendance entre les poids est donnée par :

$$\tilde{\omega}_k^{(i)} \propto \tilde{\omega}_{k-1}^{(i)} \frac{P(y_k|x_k^{(i)})f(x_k^{(i)}|x_{k-1}^{(i)})}{\pi(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{i:k})} \quad (3.14)$$

Cette approche est également appelée filtre SIS pour *Sequential Importance Sampling*.

3.2.1.4 Échantillonnage préférentiel séquentiel avec rééchantillonnage

L'inconvénient d'un filtre SIS réside dans la problématique de dégénérescence avec le temps qui correspond à un appauvrissement de l'échantillonnage. On considère à chaque instant k un échantillon de taille n selon la loi de proposition. Les éléments de cet échantillon sont appelés les particules. Le poids de chaque particule à un instant k dépend du poids des particules à l'instant $k-1$. Si au cours du temps, une particule $x_k^{(i)}$ se voit attribuer un poids $\omega_k^{(i)}$ très faible voire nul, alors l'ensemble des particules $x_k^{(i+h)}$ qui seront simulées à partir de $x_k^{(i)}$ auront un poids très faible ou nul. Après un certain temps, l'ensemble des poids des particules sera concentré sur un faible sous-ensemble de particules, voire sur une unique particule de poids unitaire. Ceci limite alors totalement l'avantage de mettre en oeuvre une méthode de Monte-Carlo. En effet, les approches de Monte-Carlo tirent un avantage à considérer des échantillons de grandes tailles d'après la convergence presque sûre impliquée par la loi des grands nombres. Pour cette raison, il est proposé à chaque instant k d'effectuer un nouvel échantillonnage à partir des particules existantes, ce rééchantillonnage étant pondéré par les poids préférentiels. Cette approche définit les filtres SISR pour *Sequential Importance Sampling with Resampling*.

3.2.1.5 Filtre *bootstrap*

Le filtre *bootstrap* est un filtre SISR assez courant qui utilise comme loi de proposition le modèle de propagation :

$$\pi(x_k|x_{0:k-1}, y_{1:k}) = f(x_k|x_{k-1}) \quad (3.15)$$

Ce choix signifie que la loi de proposition ne tient plus compte des mesures $y_{1:k}$. Il est possible de simplifier l'équation 3.14 qui permet d'obtenir que :

$$\tilde{\omega}_k^{(i)} \propto \tilde{\omega}_{k-1}^{(i)} P(y_k|x_k^{(i)}) \quad (3.16)$$

Une description complète est donnée dans l'algorithme 2, où $\text{Cat}((x_1, \omega_1), \dots, (x_n, \omega_n))$ représente la loi discrète sur l'ensemble (x_1, \dots, x_n) pondéré par $(\omega_1, \dots, \omega_n)$. Ce filtre est couramment mis en oeuvre dans le contexte de la navigation [GUSTAFSSON \[2010\]](#) et particulièrement dans le cadre de la navigation *indoor*, [KLEPAL et collab. \[2008\]](#), [NURMINEN et collab. \[2013\]](#) ou encore [FETZER et collab. \[2016\]](#).

Algorithme 2 : Algorithme du filtre *bootstrap*

Données : $y_{1:K}$
Résultat : $x_{1:K}^{(1:n)}$

- 1 $k = 0;$
- 2 **pour** $i = 1$ **faire**
- 3 Simuler $x_0^{(i)} \sim f_0;$
- 4 **pour** $k = 1 : K$ **faire**
- 5 **pour** $i = 1 : n$ **faire**
- 6 Simuler $\tilde{x}_k^{(i)} \sim f(x_k | x_{k-1}^{(i)});$
- 7 $\tilde{\omega}_k^{(i)} = P(y_k | \tilde{x}_k^{(i)});$
- 8 Normaliser les poids $\tilde{\omega}_k^{(1:n)};$
- 9 **pour** $i = 1 : n$ **faire**
- 10 $x_k^{(i)} \sim \text{Cat}((\tilde{x}_k^{(1)}, \tilde{\omega}_k^{(1)}), \dots, (\tilde{x}_k^{(n)}, \tilde{\omega}_k^{(n)}));$

3.2.2 Algorithme utilisé pour l'estimation en temps réel

On décrit la solution en temps réel déjà implémentée au CEA. Le modèle d'état de même que le modèle d'observation du filtre sont similaires à ceux présentés dans la partie 2.2 et ne seront pas décrits à nouveau.

3.2.2.1 Vecteur d'état

Le vecteur d'état reste également inchangé, il comprend la position de l'individu (x_k, y_k) , mais également un biais sur l'angle de déplacement β_k et sur la longueur de pas ϵ_k . Il est de dimension 4 :

$$\mathbf{x}_k^{(i)} = \begin{pmatrix} x_k^{(i)} \\ y_k^{(i)} \\ \epsilon_k^{(i)} \\ \beta_k^{(i)} \end{pmatrix} \quad (3.17)$$

3.2.2.2 Loi initiale

On suppose que le point de départ noté \mathbf{M}_0 est relativement bien connu. La loi associée au point de départ choisie est gaussienne :

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \sim \mathcal{N}(\mathbf{M}_0, \Sigma_{\mathbf{M}_0}) \quad (3.18)$$

où $\Sigma_{\mathbf{M}_0} = \sigma_{\mathbf{M}_0}^2 \mathbf{I}_{d^2}$ est la matrice de covariance associée. De plus, il existe une incertitude totale sur le biais sur le cap β_0 ainsi qu'une grande incertitude sur le biais de la longueur

de pas ϵ_0 . Un modèle gaussien pour cette loi est également choisi :

$$\begin{pmatrix} \epsilon_0 \\ \beta_0 \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\epsilon_0}^2 & 0 \\ 0 & \sigma_{\beta_0}^2 \end{pmatrix} \right) \quad (3.19)$$

où $\sigma_{\beta_0}^2 = 180^\circ$ et $\sigma_{\epsilon_0}^2 = 0.2$.

3.2.2.3 Mesure de pression atmosphérique

L'estimation de l'étage utilise généralement la mesure de pression qui équipe aujourd'hui un grand nombre de *smartphones*. A chaque instant, on considère une mesure de pression passant de ρ_{k-1} à ρ_k . On peut déterminer la variation d'altitude Δh_k selon la formule :

$$\Delta h_k = \frac{RT}{gM} \log \left(\frac{\rho_k}{\rho_{k-1}} \right) \quad (3.20)$$

où

- $R = 8.34 \text{ J.K}^{-1}.\text{mol}^{-1}$ est la constante des gaz parfaits
- $g = 9.8 \text{ m.s}^{-2}$ est la constante de gravitation terrestre
- $M = 0.02896 \text{ kg.mol}^{-1}$ correspond à la masse molaire de l'air
- T correspond à la température en Kelvin.

Dans notre cas, lorsque les mesures de pression permettent de détecter un changement d'étage, l'ensemble des particules présent dans une cage d'escalier sont alors échantillonnées autour du point de sortie de la cage d'escalier correspondante. En effet, il existe un retard de l'ordre de 3 secondes entre la détection et le changement d'étage. Échantillonner les particules autour du point de sortie permet de partiellement combler ce retard.

3.2.2.4 Modèle de carte

Une représentation négative de la carte a été choisie. La carte contient l'information des déplacements non-autorisés, c'est-à-dire des murs. L'information du nombre d'étages associé à chaque bâtiment notée n_{etage} est connue. La carte du bâtiment correspond à n_{etage} cartes, chacune associée à un étage. Ensuite, chacune des carte contient les éléments suivants :

- Les murs : chaque mur est représenté par un segment, chacun de ces murs est caractérisé par la position des deux extrémités du segment. A chaque instant, il faut comparer l'intersection du segment représentant le déplacement de la particule avec le segment représentant chaque mur. Afin de réduire le nombre d'opérations, les vérifications seront uniquement effectuées sur les murs dans le voisinage de la particule. On effectue une partition de l'étage afin de définir ces différents voisinages. Chaque élément de la partition correspond à une surface d'environ $50m^2$ caractérisé par un rectangle. Dans un premier temps, on cherche dans quel rectangle de la partition se trouve la particule d'intérêt puis seuls les murs étant inclus sont testés.

- Les cages d'escalier : celles-ci sont représentées par un rectangle sur la carte. Ce rectangle permet d'appliquer un modèle de pas spécifique en raison de la particularité des marches d'escalier. De plus, pour chaque cage d'escalier, à chaque étage, on associe un point de sortie, afin de faciliter la sortie des particules lorsqu'elles se déplacent dans une cage d'escalier. Cet environnement est plus complexe que les couloirs car il s'agit d'un espace très confiné impliquant la collision de nombreuses particules. Les positions des coins du rectangle, ainsi que la position du point d'entrée permettent de caractériser chaque cage d'escalier.
- Les points d'accès wifi ou bluetooth : On caractérise chaque balise par sa position, ainsi que la puissance d'émission et son type (wifi ou bluetooth).

On désignera par \mathbf{M} l'ensemble des informations de la carte.

3.2.2.5 Type de filtre et méthode d'échantillonnage

Le filtre qui a été choisi pour l'estimation en temps réel est le filtre *bootstrap*. Toutefois, deux heuristiques ont été appliquées par rapport à la théorie présentée dans la sous-section 3.2.1.

- Dans l'étape 10 de l'algorithme 2, les particules sont rééchantillonnées à partir des particules qui n'ont pas présenté de collision avec un mur. En notant par exemple $\mathbf{x}_k^{(i)}$ une particule qui a été rééchantillonnée à partir de l'une de ces particules, alors l'heuristique consiste à appliquer une perturbation aléatoire :

$$\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} + d_x \quad (3.21)$$

où d_x correspond à un vecteur simulé selon un modèle aléatoire avec une faible variance. L'objet de cette perturbation aléatoire est d'augmenter la diversité des différentes particules à chaque instant. Les nouvelles particules ne sont pas rééchantillonnées à l'exacte position de la particule initiale, mais dans un hyper-volume de dimension 4 serré autour de cette particule.

- La seconde heuristique s'applique également après l'étape de rééchantillonnage. Dans notre cas, les particules conservent leurs poids d'origine lorsque le filtre *bootstrap* attribue un poids uniforme aux particules qui ont été rééchantillonnées. Ceci accentue le poids des particules fortes et réduit l'impact des particules faibles qui vraisemblablement correspondent à une mauvaise hypothèse.

3.2.2.6 Calcul de l'estimateur

L'estimateur de la position basé sur l'ensemble des particules doit pouvoir tenir compte de la multi-modalité de la loi *a posteriori* de la position. On utilise un algorithme de classification non-supervisé afin de séparer l'ensemble des particules en différents nuages de points, chacun représentant un mode différent de l'estimation. L'algorithme de classification non-supervisé qui est mis en oeuvre est l'algorithme des *k-means*, JAIN [2010]. Il est décrit dans l'algorithme 3. Supposons que l'on dispose d'un ensemble de données $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ dans un espace E que l'on cherche à classifier selon un nombre de n_c classes. On dispose également d'une distance entre deux éléments $d()$ telle que la distance Euclidienne.

L'ensemble des particules déterminera un nombre n_c de *clusters* à chaque instant k , que l'on notera $(\mathbf{C}_1^k, \dots, \mathbf{C}_{n_c}^k)$. L'estimation finale est alors évaluée à partir du *cluster* le

Algorithme 3 : Algorithme des k -means

Données : $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$
Résultat : $\mathbf{C}_1, \dots, \mathbf{C}_{n_c}$

- 1 Initialisation;
- 2 **pour** $i = 1 : n_c$ **faire**
- 3 $\mathbf{C}_i = \emptyset$;
- 4 $n_i = 0$;
- 5 Définir un nombre n_c de centres $(\mathbf{c}_1, \dots, \mathbf{c}_{n_c})$ répartis dans E ;
- 6 **pour** $i = 1 : n$ **faire**
- 7 $j^* = \operatorname{argmin}_{j \in \{1:n_c\}} d(\mathbf{x}_i, \mathbf{c}_j)$;
- 8 $\mathbf{C}_{j^*} = \mathbf{C}_{j^*} \cup \mathbf{x}_i$;
- 9 $n_{j^*} = n_{j^*} + 1$;
- 10 $\mathbf{c}_{j^*} = \frac{1}{n_{j^*}} \sum_{\mathbf{x} \in \mathbf{C}_{j^*}} \mathbf{x}$;

plus lourd au sens du poids des particules qu'il contient. En considérant un ensemble de n_p particules pondérées $\left[(\mathbf{x}_k^{(1)}, \omega_k^{(1)}), \dots, (\mathbf{x}_k^{(n_p)}, \omega_k^{(n_p)}) \right]$, on peut associer à tout *cluster* \mathbf{C}_i^k un poids $\omega(\mathbf{C}_i^k)$:

$$\omega(\mathbf{C}_i^k) = \sum_{\mathbf{x}_k^{(j)} \in \mathbf{C}_i^k} \omega_k^{(j)} \quad (3.22)$$

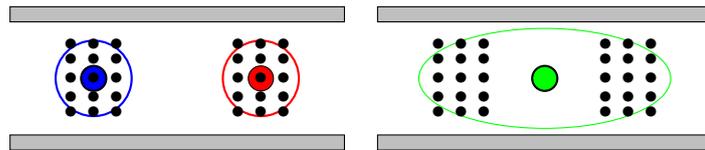
On peut alors évaluer une estimation de la position à l'instant k notée $\hat{\mathbf{x}}_k$ en considérant une moyenne pondérée sur ce *cluster* uniquement :

$$\hat{\mathbf{x}}_k = \frac{1}{\omega(\mathbf{C}_{i_m}^k)} \sum_{\mathbf{x}_k^{(j)} \in \mathbf{C}_{i_m}^k} \omega_k^{(j)} \mathbf{x}_k^{(j)} \quad (3.23)$$

où

$$i_m = \operatorname{argmax}_{i \in \{1:n_c\}} \omega(\mathbf{C}_i^k) \quad (3.24)$$

correspond à l'indice du *cluster* le plus lourd. Par ailleurs, cet algorithme nécessite de déterminer en amont un nombre de *clusters*. L'estimation de la trajectoire sera relativement sensible à ce nombre de *clusters*. En effet, un nombre trop faible de *clusters* ne permettra pas de considérer des hypothèses très différentes. L'exemple de la figure 3.10 présente deux nuages de particules disjoints. Utiliser un unique *cluster* (figure de droite) impliquera une variance très élevée sur ce *cluster* et la moyenne pourrait se trouver entre deux nuages dans une zone sans particule.


 FIGURE 3.10 – Illustration d'un nombre de *clusters* trop faible

A l'inverse, un trop grand nombre de *clusters* impliquera une redondance sur les différentes hypothèses. L'exemple de la figure 3.11 illustre un unique nuage de points qui est

scindé en deux *clusters* distincts. Dans la figure de gauche, chacun de ces *clusters* suivra la même hypothèse, mais le poids de chaque *cluster* sera divisé par deux comparativement à l'unique *cluster* de la figure de droite. Le poids relatif de cette hypothèse sera donc réduit avec deux *clusters*.

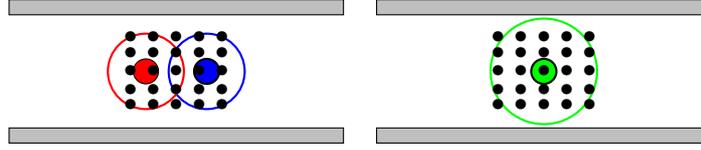


FIGURE 3.11 – Illustration d'un nombre de *clusters* trop grand

Le nombre de *clusters* constituera un paramètre à fixer en fonction des performances de l'estimation.

3.2.2.7 Présentation générale de l'algorithme

On propose l'algorithme permettant d'estimer la position après le pas $k + 1$, notée $\hat{\mathbf{x}}_{k+1}$.

Algorithme 4 : Algorithme d'estimation de la position après le pas $k + 1$

Données : $(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(n_p)}), (\omega_k^{(1)}, \dots, \omega_k^{(n_p)}), l_{k+1}, \alpha_{k+1}, z_{mes}^{k+1}, \rho_{k+1}, \mathbf{M}$

Résultat : $\hat{\mathbf{x}}_{k+1}, (\mathbf{x}_{k+1}^{(1)}, \dots, \mathbf{x}_{k+1}^{(n_p)}), (\omega_{k+1}^{(1)}, \dots, \omega_{k+1}^{(n_p)})$

1 **pour** $i = 1 : n_p$ **faire**

2 Propagation des particules : $\mathbf{x}_{k+1}^{(i)} \sim f(\mathbf{x}|\mathbf{x}_k^{(i)})$ (équation 2.18) à partir de (l_{k+1}, α_{k+1}) ;

3 **si** *Mesure RSS activée* **alors**

4 Pondération mesure RSS à partir de z_{mes}^{k+1} ;

5 Vérifier collision entre $(\mathbf{x}_k^{(i)}, \mathbf{x}_{k+1}^{(i)})$ et \mathbf{M} ;

6 **si** *Collision* **alors**

7 $\omega_{k+1}^{(i)} = 0$;

8 Détermination de l'étage à partir de ρ_{k+1} ;

9 Normalisation des poids ;

10 Rééchantillonnage des particules avec remise, selon leur pondération, chaque particule rééchantillonnée conserve son poids d'origine ;

11 Normalisation des poids ;

12 Classification des particules ;

13 $\hat{\mathbf{x}}_{k+1}$ est le barycentre du *cluster* le plus lourd ;

3.3 Formalisation de l'estimation en temps différé

3.3.1 Multi-modalité et problème de continuité

L'estimation en temps réel de la trajectoire de même que la multi-modalité de la loi *a posteriori* de la position de recherche peut impliquer des problèmes de continuité. Ces discontinuités peuvent provenir de deux situations : à cause des contraintes de la carte ou par l'exploitation des mesures RSS.

- La figure 3.12 illustre un exemple de discontinuité à cause des contraintes de la carte. En effet, à l'instant $k = 0$, cet exemple fictif considère un nombre de 12 particules échantillonnées autour du point de départ. A l'instant $k = 1$, ces particules se séparent en 3 *clusters* pour chacun des 3 différents couloirs. Les *clusters* A, B et C comportent respectivement 5, 4 et 3 particules. En considérant que chaque particule possède le même poids, l'estimation en temps réel est alors donnée par le *cluster* A. A l'instant $k = 2$, chacun de ces *clusters* continuent d'avancer, et l'estimation est toujours donnée par le *cluster* A. Toutefois, à l'instant $k = 3$, certaines des particules du *cluster* A rencontrent un mur et sont alors rééchantillonnées autour des *clusters* B et C. Le *cluster* B possédant plus de particules, il devient la nouvelle estimation de la trajectoire. Entre l'instant 2 et 3, le passage du *cluster* A au *cluster* B implique bien un saut dans la trajectoire.

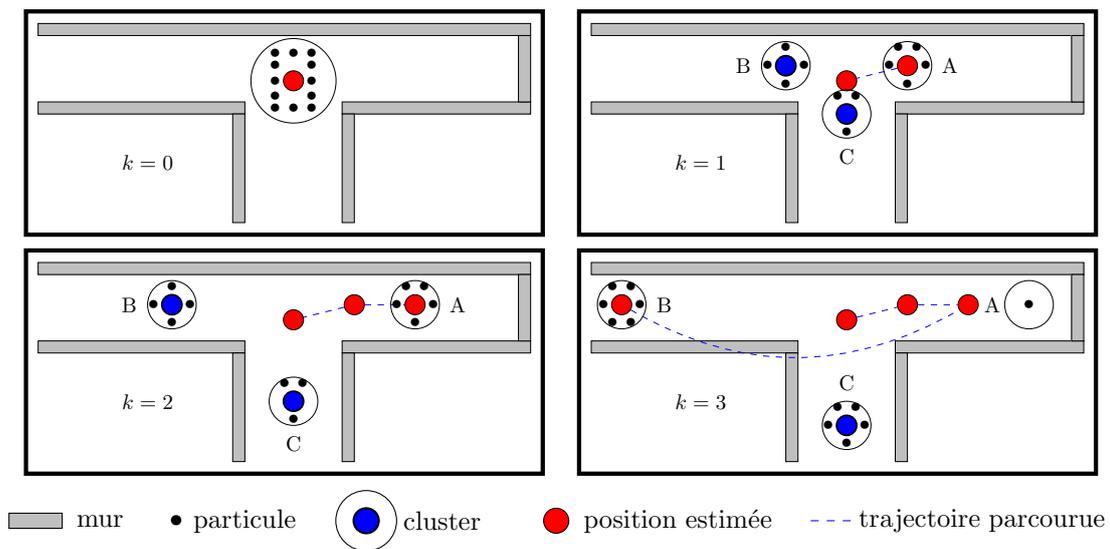


FIGURE 3.12 – Illustration sur un cas simple d'un problème de discontinuité pour l'estimation en temps réel avec un nombre de 12 particules à cause des murs

- La figure 3.13 présente un exemple de discontinuité dû à l'exploitation des mesures RSS. On suppose à un instant $k = 0$ que deux *clusters* A et B se suivent mais constituent deux nuages de particules différents. Cette situation est possible en raison de différentes hypothèses de modèle de longueur de pas. Le *cluster* A, retenu comme estimation car il possède le poids le plus fort, est retardé par rapport à la position réelle qui correspond au *cluster* B dans cet exemple. Néanmoins, à l'instant $k = 3$, la trajectoire réelle se rapproche d'une balise wifi. Certaines particules du *cluster* A sont logiquement rééchantillonnées autour du *cluster* B, faisant de lui la nouvelle estimation en temps réel. Comme dans l'exemple précédent, il y a une discontinuité dans la trajectoire entre l'instant 2 et 3.

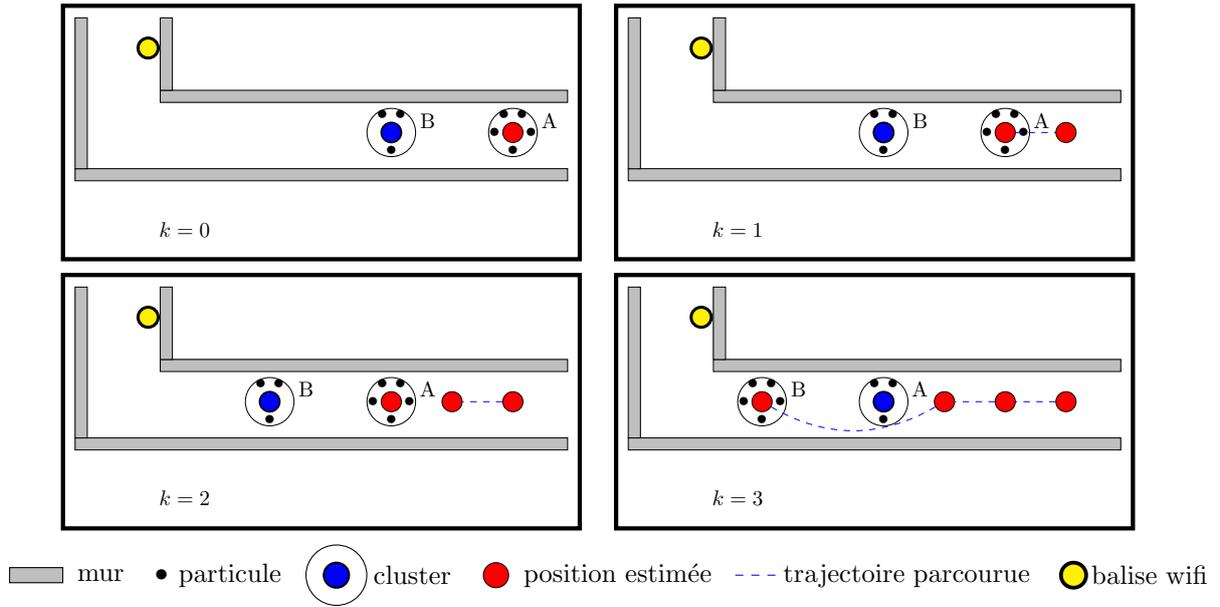


FIGURE 3.13 – Illustration sur un cas simple d'un problème de discontinuité pour l'estimation en temps réel avec un nombre de 8 particules à cause des mesures RSS

Pour chacun de ces exemples, à l'instant $k = 3$, l'information du mur ou des mesures RSS a permis de corriger l'estimation. En temps différé, il est possible de corriger la position aux instants précédents $k = 1 : 2$ en effectuant un lissage. Les différentes approches de lissage en temps différé de l'état de l'art effectuent ce lissage en repondérant les particules qui ont été propagées en temps réel. Dans notre exemple, les particules du *clusters* A aux instants $k = 1 : 2$ auront donc un poids très faible voire seront supprimées. Notre approche propose également un lissage, mais l'applique directement sur les *clusters* définis en temps réel, et non sur les particules.

3.3.2 Application du lissage sur les *clusters*

3.3.2.1 Mélange de modèles gaussiens autour des *clusters*

On rappelle que la loi *a posteriori* en temps réel est estimée à l'aide des particules. A l'instant k , on a :

$$\hat{P}(x_k | y_{1:k}) = \sum_{i=1}^{n_p} \omega_k^{(i)} \delta(\mathbf{x}_k^{(i)}) \quad (3.25)$$

Le point de vu adopté dans ces travaux est de considérer que cette loi *a posteriori* peut être également approximée par un mélange de modèles gaussiens que l'on notera GMM (*Gaussian Mixture Model*). A chaque instant k , l'estimation de la densité $f_k()$ de la loi *a posteriori* est alors donnée par :

$$\hat{f}_k(\mathbf{x}) = \sum_{i=1}^{n_c} \omega(C_i^k) \phi(\mathbf{c}_k^i, \Sigma_k^i) \quad (3.26)$$

où $\bar{\mathbf{c}}_k^i$ et Σ_k^i sont respectivement le barycentre et la matrice de covariance du *cluster* C_k^i , $\omega(C_i^k)$ le poids de ce *cluster* et $\phi(\mu, \Sigma)$ la densité associée à une loi normale de vecteur moyen μ , ici en dimension 4, et de covariance Σ . On représente une illustration simulée dans la figure 3.14 de la loi associée à l'utilisation des particules et également d'un GMM en dimension 1.

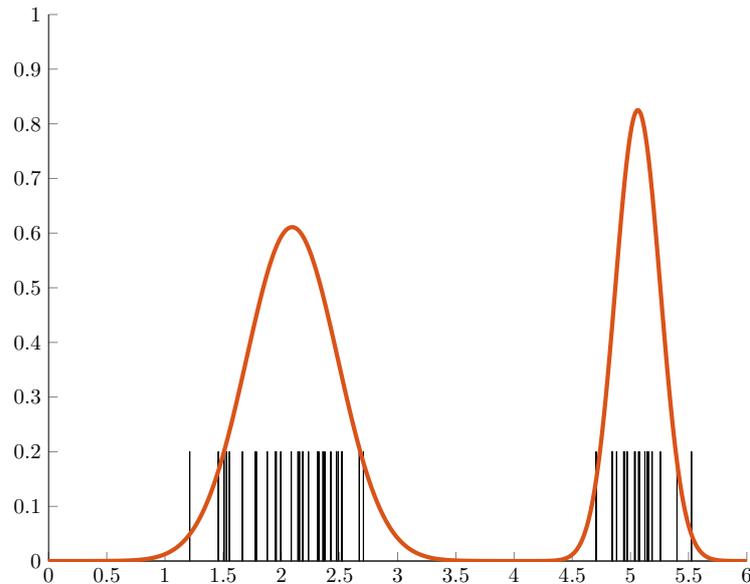


FIGURE 3.14 – Illustration en dimension 1 d'un GMM associé à deux nuages de particules

Utiliser un GMM sur les *clusters* plutôt qu'un ensemble de particules implique une diminution de l'information. En effet, l'information des particules d'un *cluster* sera uniquement résumée par la position centrale du *cluster* ainsi que la covariance associée. Toutefois, nous avons observé en pratique que la perte d'information n'a pas de conséquences importantes quand à l'estimation de la position. En effet, on considère par exemple que la position réelle a bien été estimée par l'une des particules d'un *cluster*. En utilisant les *clusters*, cette information ne sera alors pas conservée. Seule la position centrale du *cluster* auquel appartient cette particule permettra d'estimer la position réelle. Néanmoins, si le nombre de *clusters* n_c est suffisamment élevé, alors la covariance associée à chacun des *clusters* sera relativement faible et la distance entre une particule et le barycentre du *cluster* ne sera pas élevée.

3.3.2.2 Problématique du chemin de *cluster*

La position à chaque instant est estimée par le mode du GMM. Dans le cas où les moyennes des gaussiennes sont considérées disjointes, comme dans l'illustration de la figure 3.14, le mode d'un GMM correspond directement au mode de la gaussienne qui a la plus grande pondération. Néanmoins, il peut exister un mode différent de celui de chaque gaussienne, notamment lorsque les différentes gaussiennes sont trop rapprochées. Pour un GMM en dimension 4, savoir déterminer si les gaussiennes sont trop proches, ou encore exprimer analytiquement le nombre de modes ainsi que leur position peut être complexe comme il est décrit dans [RAY et LINDSAY \[2005\]](#). Dans notre cas, nous considérerons que les nuages de points correspondent bien à des hypothèses de positions suffisamment différentes. L'estimateur de la position finale correspondra bien au barycentre du *cluster* le plus lourd. En reprenant l'exemple de la figure 3.12 et en considérant uniquement le GMM ainsi que les *clusters*, on peut représenter dans la figure 3.15 le chemin de *cluster* qui a été estimé en temps réel (figure de gauche) qui ne correspond donc pas au trajet de l'utilisateur. En revanche, en temps différé il est possible de sélectionner d'emblée la meilleure solution et de réduire les changements d'hypothèses (c'est-à-dire de *clusters*) observés en temps réel (figure de droite).

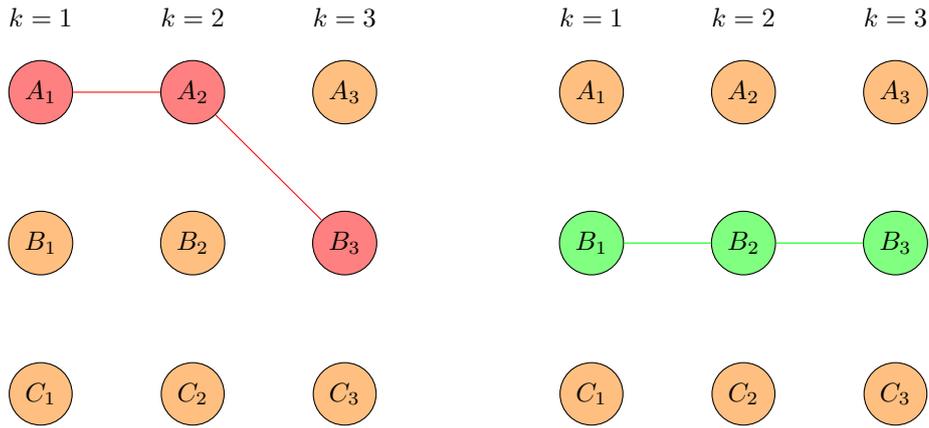


FIGURE 3.15 – Représentation du choix des *clusters* en temps réel d’après l’exemple fictif. L’estimation en temps réel à gauche et l’estimation en temps différé à droite.

Ici, seules les positions des *clusters* à chaque instant ainsi que leurs poids sont sauvegardés au lieu de l’ensemble des particules. Selon l’hypothèse du GMM qui a été formulée, ceci est équivalent à sauver l’ensemble des particules en réduisant considérablement le nombre d’éléments d’intérêt. Cet ensemble des *clusters* forme un espace dans lequel on suppose qu’il existe notre solution. Notre problème est donc de déterminer, dans cet ensemble discret, un chemin (4D) le plus probable. La figure 3.16 représente pour un nombre de $n_c = 3$ *clusters* les différentes possibilités de trajectoires de *clusters* existantes. Une recherche exhaustive du meilleur chemin n’est pas envisageable. En supposant un ensemble de $n_c = 3$ *clusters*, un nombre très faible de $K = 19$ pas conduirait à un nombre de $(n_c)^K > 10^9$ possibilités. On s’intéressera alors à développer des approches intelligentes. Dans un premier temps, on décrira l’algorithme de Viterbi qui permet une évaluation du maximum *a posteriori*. Dans un second temps, des méthodologies de lissage seront étudiées. Enfin, une évaluation sera menée dans la section 3.5.

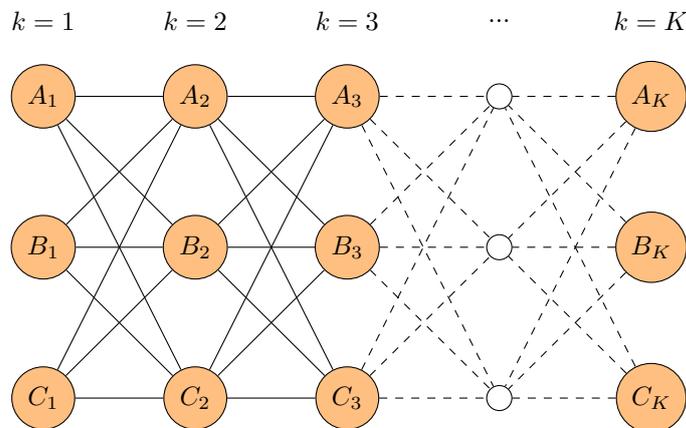


FIGURE 3.16 – Problématique de la sélection du meilleur chemin parmi les 3^K chemins possibles

3.4 Méthodes de résolution de la problématique du chemin le plus probable

3.4.1 Algorithme de Viterbi

En 1966, Viterbi a proposé un algorithme séquentiel VITERBI [1967] permettant d'estimer le maximum *a posteriori* d'un modèle de Markov caché à espace d'état fini. Dans le cas général, en notant E_k l'espace d'état discret de taille m à l'instant k le maximum *a posteriori* est alors défini par :

$$x_{1:K}^{(map)} = \operatorname{argmax}_{x_1 \in E_1, \dots, x_K \in E_K} P(x_{1:K} | y_{1:K}) \quad (3.27)$$

Dans notre cas, l'espace d'état correspond à chaque instant aux *clusters* qui ont été estimés en temps réel. En l'occurrence, cet algorithme a déjà été développé suite à l'utilisation d'un filtre particulaire, les particules échantillonnées en temps réel proposant une discrétisation de l'espace à chaque instant GODSILL et collab. [2001]. L'algorithme de Viterbi utilise une variable instrumentale et se base notamment sur la propriété markovienne de l'espace d'état d'être sans mémoire à l'ordre 1. A l'instant k , pour chaque x_k^i , seule l'information de la valeur précédente x_{k-1}^j qui a maximisé la vraisemblance de la suite $P(x_{1:k} | y_{1:k})$ sera pertinente pour la maximisation globale. Une variable instrumentale de taille m à chaque instant k notée $\mathbf{v}_{1:m}(k)$ permettra de résoudre ce problème d'optimisation. Cette variable est initialisée à partir de la loi initiale associée à x_0 que l'on notera $\mathbf{f}_0 = (f_0(1), \dots, f_0(m))$:

$$v_i(1) = f_0(i)h(y_1 | x_1^i) \quad (3.28)$$

Par la suite, cette variable est estimée de façon récursive en résolvant à chaque étape une maximisation :

$$v_i(k) = \max_{1 \leq j \leq m} \left[v_j(k-1) f(x_k^j | x_{k-1}^i) \right] h(y_k | x_k^j) \quad (3.29)$$

et on définit également la fonction $\operatorname{ind}(i, k)$ qui, pour chaque état i et pour chaque instant $k \geq 2$, associe l'indice de l'état à l'instant $k-1$ qui résout le problème d'optimisation de l'équation 3.30.

$$\operatorname{ind}(i, k) = \operatorname{argmax}_{1 \leq j \leq m} v_j(k-1) f(x_k^j | x_{k-1}^i) \quad (3.30)$$

L'estimation finale à l'instant K du maximum *a posteriori* est donnée par l'état qui maximise la variable instrumentale :

$$\hat{x}_K^{(map)} = x_K^{i^{map,K}} \quad (3.31)$$

où

$$i^{map,K} = \operatorname{argmax}_{1 \leq i \leq m} \mathbf{v}_i(K) \quad (3.32)$$

Le maximum *a posteriori* est alors donné de façon récursive pour un instant $k < K$:

$$\hat{x}_k^{(map)} = x_K^{i^{map,k}} \quad (3.33)$$

où

$$i^{map,k} = \operatorname{ind}(i^{map,k+1}, k+1) \quad (3.34)$$

On choisira d'approximer le calcul du modèle d'observation par le poids d'un *cluster*, c'est-à-dire que $\hat{h}(y_k | x_k^i) = \omega(C_k^i)$. En effet, le poids des *clusters* tient compte des mesures RSS ainsi que de la collision avec les murs.

L'algorithme de Viterbi nécessite un nombre d'opérations de l'ordre de $O(Kn_c^2)$ ainsi qu'une taille de mémoire de $O(n_c K)$.

Algorithme 5 : Adaptation de l'algorithme de Viterbi

Données : $(\bar{\mathbf{c}}_k^i, \omega(C_k^i))_{(1 \leq k \leq K, 1 \leq i \leq n_c)}$
Résultat : $\hat{\mathbf{x}}_{1:K}^{(\text{vit})}$
 1 $k = 1 : \forall 1 \leq i \leq n_c, u_1(i) = \log f_0(i) + \log \omega(C_1^i);$
 2 **pour** $k = 2 : K$ **faire**
 3 **pour** $i = 1 : n_c$ **faire**
 4 $\text{ind}(i, k) = \underset{1 \leq j \leq n_c}{\text{argmax}} u_{k-1}(j) + \log f(\bar{\mathbf{c}}_k^j | \bar{\mathbf{c}}_{k-1}^i);$
 5 $u_k(i) = [u_{k-1}(\text{ind}(i, k)) + \log f(\bar{\mathbf{c}}_k^i | \bar{\mathbf{c}}_{k-1}^{\text{ind}(i, k-1)})] + \log \omega(C_1^i);$
 6 $i_K = \arg \max_{1 \leq i \leq N} L_K(i)$ et $\hat{\mathbf{x}}_K = \mathbf{x}_K^{i_K};$
 7 **pour** $k = K - 1 : 1$ **faire**
 8 $i_k = v_{k-1}(i_{k-1})$ et $\hat{\mathbf{x}}_k = \mathbf{x}_k^{i_k};$

3.4.2 Approche Bayésienne

L'approche bayésienne correspond aux algorithmes de lissage particulière utilisés dans la littérature. Afin de déterminer la meilleure trajectoire de *cluster*, il est possible d'appliquer ces approches sur l'historique pondéré des *clusters*.

3.4.2.1 L'algorithme Forward Filtering Backward Smoothing

On reprend l'équation 1.16 présentée en introduction :

$$P(x_k | y_{1:K}) = P(x_k | y_{1:k}) \int \frac{P(x_{k+1} | y_{1:K}) f(x_{k+1} | x_k)}{P(x_{k+1} | y_{1:k})} dx_{k+1}$$

La loi de lissage fait apparaître deux termes : la loi de filtrage, appelée composante *forward*, ainsi qu'un terme intégral, appelé composante *backward*. Cette décomposition est intéressante puisque la composante *forward* est estimée par le filtre particulière en temps réel et sera utilisée par l'algorithme *Forward Filtering Backward Smoothing* (FFBSm) [DUBARRY \[2012\]](#). L'algorithme FFBSm permet d'estimer la loi de lissage sur le même modèle que les méthodes de Monte-Carlo séquentielles. La loi de filtrage à l'instant k est estimée par un ensemble n_p de particules pondérées :

$$P(x_k | y_{1:k}) \simeq \hat{P}(x_k | y_{1:k}) = \sum_{i=1}^{n_p} \omega_k^{(i)} \delta(x_k^{(i)}) \quad (3.35)$$

L'algorithme FFBSm permet de recalculer une pondération de lissage et d'estimer la loi de lissage à partir des particules simulées en temps réel :

$$P(x_k | y_{1:K}) \simeq \sum_{i=1}^{n_p} \omega_{k|K}^{(i)} \delta(x_k^{(i)}) \quad (3.36)$$

où les poids de lissage sont estimés par :

$$\omega_{k|K}^{(i)} = \omega_k^{(i)} \left[\sum_{j=1}^{n_p} \omega_{k+1|K}^{(j)} \frac{f(x_{k+1}^{(j)} | x_k^{(i)})}{\sum_{l=1}^{n_p} \omega_k^{(l)} f(x_{k+1}^{(l)} | x_k^{(j)})} \right] \quad (3.37)$$

Dans notre cas, il devient possible d'associer à chaque *cluster* une nouvelle pondération en temps différé que l'on notera $\omega(C_{(k|K)}^i)$ pour un *cluster* i à l'instant k . Ensuite, l'estimation de la position par la méthode FFBSm, que l'on notera $\hat{\mathbf{x}}_{1:K}^{(\text{ffbsm})}$, sera donnée par la position du barycentre du *cluster* le plus lourd. L'algorithme FFBSm nécessite un temps de calcul de l'ordre de $O(Kn_c^2)$.

Algorithme 6 : Algorithme FFBSm

Données : $(\bar{\mathbf{c}}_k^i, \omega(C_k^i))_{(1 \leq k \leq K, 1 \leq i \leq n_c)}$

Résultat : $\hat{\mathbf{x}}_{1:K}^{(\text{ffbsm})}$

- 1 $k = K : \omega(C_{(K|K)}^i) = \omega(C_K^i) \forall 1 \leq i \leq n_c;$
- 2 $\hat{\mathbf{x}}_K^{(\text{ffbsm})} = \operatorname{argmax}_{\bar{\mathbf{c}}_K^i \in \text{Cl}_K} \omega(C_{(K|K)}^i);$
- 3 **pour** $k = K - 1 : 1$ **faire**
- 4 **pour** $i = 1 : n_c$ **faire**
- 5 $\omega(C_{(k|K)}^i) = \omega(C_k^i) \left[\frac{\sum_{j=1}^{n_c} \omega(C_{(k+1|K)}^j) \frac{f(\bar{\mathbf{c}}_{k+1}^j | \bar{\mathbf{c}}_k^i)}{\sum_{l=1}^{n_c} \omega(C_k^l) f(\bar{\mathbf{c}}_{k+1}^j | \bar{\mathbf{c}}_k^l)} \right];$
- 6 $\hat{\mathbf{x}}_k^{(\text{ffbsm})} = \operatorname{argmax}_{\bar{\mathbf{c}}_k^i \in \text{Cl}_k} \omega(C_{(k|K)}^i);$

3.4.2.2 L'algorithme Forward Filtering Backward Simulation

L'algorithme *Forward Filtering Backward Simulation* (FFBSi) DUBARRY [2012] est très proche de l'algorithme FFBSm puisqu'il fait intervenir un filtre particulière pour la phase *forward* puis applique un traitement *backward*. En appliquant la formule de Bayes, on obtient que :

$$P(x_k | x_{k+1}, y_{1:k}) = \frac{P(x_{k+1} | y_{1:k}) f(x_{k+1} | x_k)}{P(x_{k+1} | y_{1:k})} \quad (3.38)$$

En intégrant le dénominateur de la formule 3.38 selon x_k , on obtient :

$$P(x_k | x_{k+1}, y_{1:k}) = \frac{P(x_k | y_{1:k}) f(x_{k+1} | x_k)}{\int P(x_k, x_{k+1} | y_{1:k}) dx_k}$$

et par la formule de Bayes, on a alors :

$$P(x_k | x_{k+1}, y_{1:k}) = \frac{P(x_k | y_{1:k}) f(x_{k+1} | x_k)}{\int P(x_k | y_{1:k}) f(x_{k+1} | x_k) dx_k} \quad (3.39)$$

Cette relation définit un noyau à rebours, ou *Backward kernel*, puisqu'elle permet d'obtenir la loi de x_k connaissant l'état suivant x_{k+1} ainsi que les observations jusqu'à l'instant k . On a vu que la loi de filtrage pouvait être estimée par l'équation 3.36. Cette approximation peut être directement insérée dans l'équation 3.39 et définit ce *backward kernel* que l'on notera :

$$\hat{B}(x_k | x_{k+1}) = \sum_{i=1}^{n_c} \frac{\omega_k^{(i)} f(x_{k+1} | x_k^{(i)})}{\sum_{l=1}^{n_c} \omega_k^{(l)} f(x_{k+1} | x_k^{(l)})} \delta(x_k^{(i)}) \quad (3.40)$$

L'idée de cette approche est ensuite d'échantillonner de façon récursive un sous-ensemble de particules à chaque étape. Le processus récursif se découle de la façon suivante :

1. $\tilde{x}_K \sim \hat{P}(x_K | y_{1:K})$ pour l'instant final K
2. $\tilde{x}_k \sim \hat{B}(x_k | \tilde{x}_{k+1})$ pour $k = K - 1 : 1$

En pratique l'algorithme FFBSi considère un nombre m de particules inférieur au nombre de particules initial n_p . L'objet de ce filtre est uniquement de conserver de façon aléatoire, en fonction du modèle d'état, ainsi que du poids des particules, un sous-ensemble représentant les meilleures particules à chaque instant. Dans notre cas, nous avons vu que cela pouvait revenir à chercher le meilleur chemin de *cluster*. Il s'agit donc d'échantillonner à chaque étape un *cluster* qui correspondrait à la position la plus vraisemblable. En pratique, cet algorithme doit être adapté et la composante aléatoire sera supprimée. L'estimation de la position réelle sera donnée par le *cluster* dont la vraisemblance est la plus élevée. Le coût de cet algorithme est alors de l'ordre de $O(Kn_c)$. L'estimation de la trajectoire en temps différé par l'algorithme FFBSi sera alors notée $\hat{\mathbf{x}}_{1:K}^{(\text{ffbsi})}$.

Algorithme 7 : Algorithme FFBSi adapté à notre cas d'application

Données : $(\bar{\mathbf{c}}_k^i, \omega(C_k^i))_{(1 \leq k \leq K, 1 \leq i \leq n_c)}$

Résultat : $\hat{\mathbf{x}}_{1:K}^{(\text{ffbsi})}$

- 1 $k = K : \hat{\mathbf{x}}_K^{(\text{ffbsi})} = \underset{\bar{\mathbf{c}}_i \in \text{Cl}_k}{\text{argmax}} \omega(C_K^i);$
- 2 **pour** $k = K - 1 : 1$ **faire**
- 3 $\left[\hat{\mathbf{x}}_k^{(\text{ffbsi})} = \underset{\bar{\mathbf{c}}_i \in \text{Cl}_k}{\text{argmax}} \omega(C_k^j) f(\hat{\mathbf{x}}_{k+1}^{(\text{ffbsi})} | \bar{\mathbf{c}}_k^i); \right.$

3.5 Résultats

3.5.1 Illustration sur une trajectoire avec des discontinuités

La trajectoire choisie pour illustrer chacune de ces méthodes est la trajectoire de la figure 2.4.2 qui correspond à un bâtiment de bureaux.

Dans cet exemple, nous avons choisi un nombre de *cluster* $n_c = 3$. On posera également que $\sigma_{M_0}^2 = 1\text{m}$. Le problème de continuité observé pour l'estimation en temps réel correspond à la correction par les mesures RSS. La figure 3.18 présente l'estimation en temps réel pour laquelle on observe deux discontinuités en raison de la mauvaise qualité des mesures RSS.

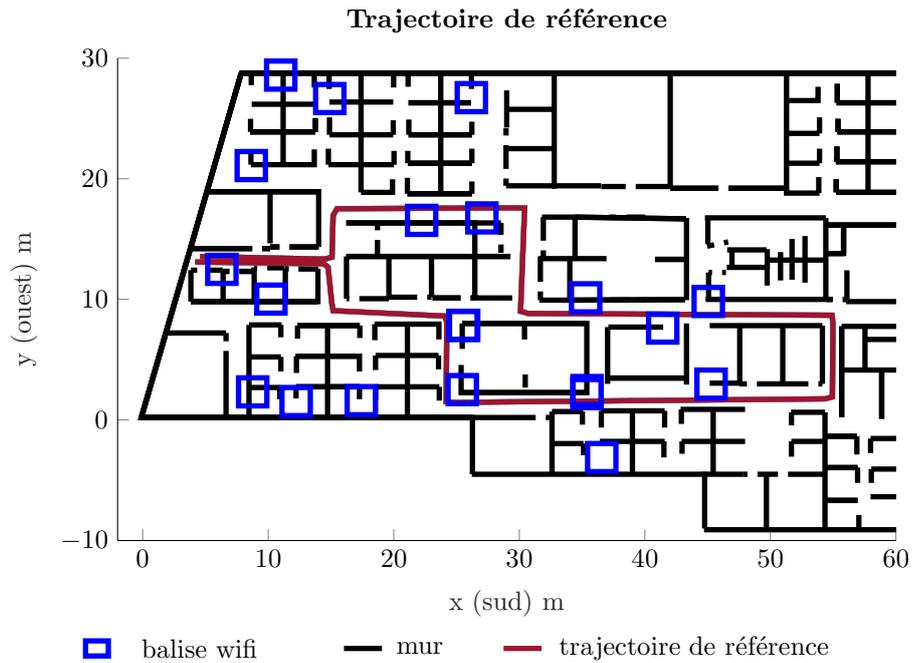


FIGURE 3.17 – Représentation de la trajectoire de référence

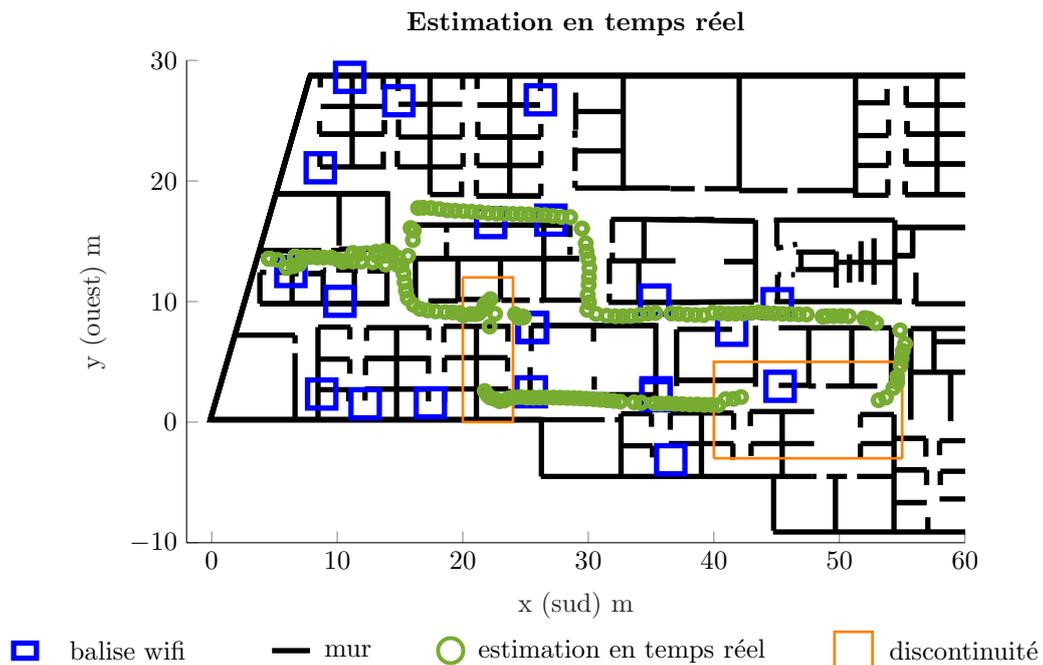
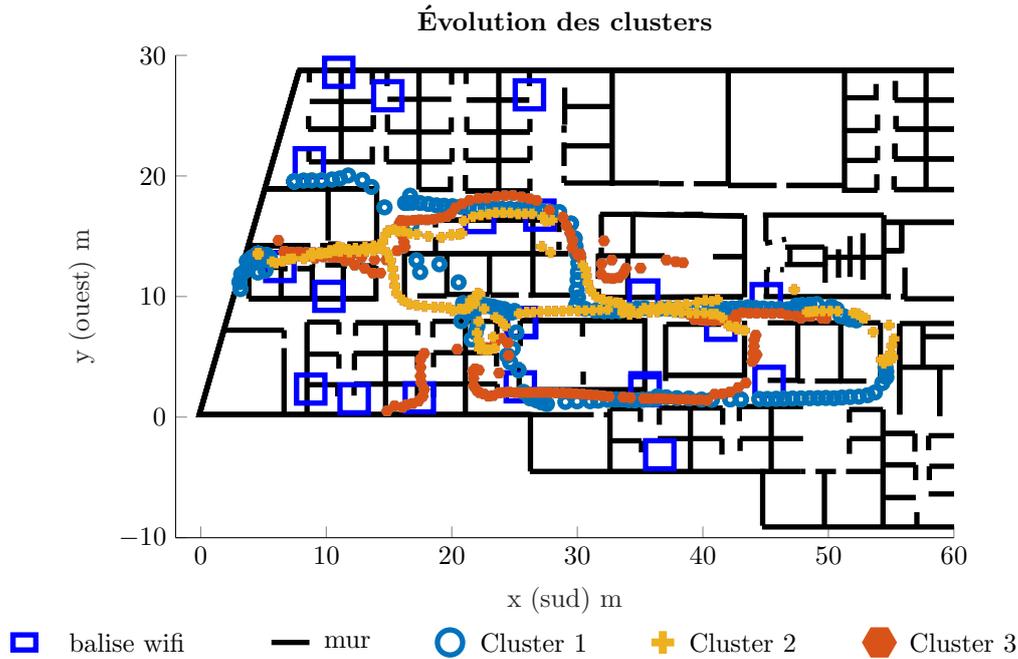


FIGURE 3.18 – Estimation en temps réel d’une trajectoire *map-matchée*. Cette estimation présente deux discontinuités dues à la recalibration de la trajectoire par les mesures RSS.

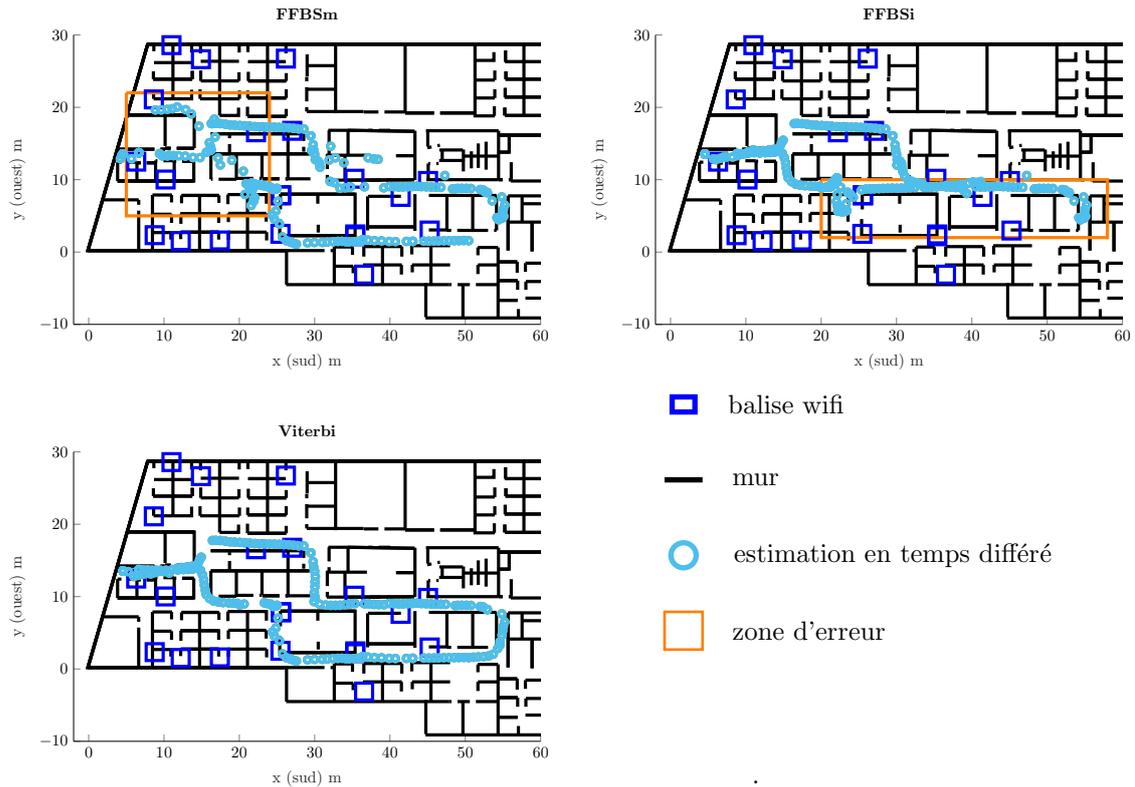
La figure 3.19 montre l’évolution des *clusters* correspondant à cette trajectoire. Sur cet exemple, au moins un des *clusters* a pu, à chaque instant, estimer la position correcte.


 FIGURE 3.19 – Évolution des 3 *clusters* estimés en temps réel

On peut appliquer les différentes approches représentées par la figure 3.20, les résultats numériques étant présentés dans le tableau 3.1. On observe dans un premier temps que l'algorithme de Viterbi (figure en bas à gauche) permet d'estimer la trajectoire correcte, la valeur du %d5 étant de 100%. Par ailleurs, l'algorithme FFBSm (figure en haut à gauche) améliore l'estimation globale de la trajectoire, notamment en corrigeant les deux discontinuités de la figure 3.18. Néanmoins, sur la fin de la trajectoire, cette méthode détériore l'estimation finale en considérant un mauvais *cluster*. Enfin, la méthode FFBSi (figure en haut à droite) détériore la trajectoire en temps réel. Ceci se justifie par l'aspect séquentiel de cette méthode qui n'est pas en mesure de se corriger au cours du temps. Il reste nécessaire de tester ces méthodes sur un grand nombre de trajectoires afin de dégager des tendances globales sur l'efficacité de chacune des méthodes.

Trajectoire	rmse (m)	%d5 (%)	Coût	Mémoire
Estimation temps réel	3.77	80.38	-	-
FFBSm	2.63	90.62	$O(Kn_c^2)$	-
FFBSi	4.82	77.05	$O(Kn_c)$	-
Viterbi	1.05	100	$O(Kn_c^2)$	$O(Kn_c)$

TABLEAU 3.1 – Évaluation de la méthode de sélection du chemin sur un cas applicatif ainsi que les coûts de calcul associés


 FIGURE 3.20 – Estimations en temps différé par la méthode de sélection des *clusters*

3.5.2 Résultats sur la base de données

Chacun des algorithmes sera mis en oeuvre sur un ensemble de 130 expériences. Les conditions expérimentales pour chaque enregistrement sont présentées dans la partie 5.1.2 ainsi que le protocole pour obtenir la trajectoire réelle. Ces résultats sont présentés selon l'évolution du nombre de *clusters* n_c dans le tableau 3.2

Méthode	n_c	Moyenne %d5	Moyenne %d10	Moyenne rmse(m)
FFBSm	3	82.40	93.88	5.35
FFBSm	10	75.79	89.26	7.89
FFBSm	20	75.67	89.98	7.53
FFBSi	3	70.90	87.54	8.61
FFBSi	10	71.09	87.93	8.53
FFBSi	20	74.12	89.10	7.69
Viterbi	3	84.70	95.09	4.15
Viterbi	10	82.21	93.68	5.11
Viterbi	20	81.97	94.03	5.62
Estimation temps réel		83.29	94.12	5.81

 TABLEAU 3.2 – Résultats de l'estimation en temps différé selon la méthode de sélection ainsi que le nombre de *clusters* sur un ensemble de 130 expériences

On observe ainsi que l'algorithme de Viterbi est la méthode la plus probante avec $n_c = 3$ *clusters* concernant la sélection du meilleur chemin, puisque la valeur du rmse est

de 4.15m contre 5.81m en temps réel, ce qui correspond à gain de 28%. De même la valeur du %d5 est globalement diminuée, même si l'effet est moins important, 84.7% contre 83.3% en temps réel. Cette approche est également très sensible au nombre de *clusters*, puisqu'elle se détériore en considérant un trop grand nombre de *clusters*. En effet, dans le cadre de la navigation *indoor*, et en tenant compte des murs, il n'est pas nécessaire de considérer un nombre d'hypothèses possibles trop grand. En effet, ceci peut impliquer une redondance de l'information de même qu'une perte relative des poids qui peut détériorer l'estimation. Par la suite, on fixera la méthodologie de la sélection du meilleur chemin à l'algorithme de Viterbi pour un nombre de $n_c = 3$ *clusters* qui correspond au nombre maximum d'hypothèses concurrentes pertinentes à chaque instant.

Par ailleurs, les approches bayésiennes FFBSm et FFBSi ne permettent pas d'obtenir des résultats satisfaisants. En effet, l'algorithme de FFBSi tel qu'il a été appliqué détériore la qualité globale de l'estimation. Ces approches se basent principalement sur une estimation par Monte-Carlo qui ne correspond pas réellement à notre choix d'utiliser des *clusters*. Toutefois, la méthode FFBSm dégrade la valeur du %d5, 82.4% contre 83.3% en temps réel, mais améliore légèrement la valeur de rmse, 5.35m contre 5.81m en temps réel. De façon globale, elle ne permet pas d'améliorer la trajectoire réelle de façon significative. Ces méthodes sont pertinentes en étant appliquées sur des particules selon différents travaux de l'état de l'art, leur application sur des *clusters* ne garantit toutefois pas les mêmes résultats, contrairement à l'algorithme de Viterbi. Par la suite, on notera VPF la trajectoire estimée en appliquant l'algorithme de Viterbi.

3.5.3 Conclusion

L'estimation en temps réel ne prémunit pas des potentiels problèmes de discontinuité. En effet, celle-ci peut proposer une estimation en temps réel correspondant à une hypothèse fautive, puis retrouver l'hypothèse correcte. Durant le laps de temps où l'estimation en temps réel s'est égarée, l'erreur globale d'estimation de la position augmente. Avoir accès à un horizon plus large de l'information peut permettre de s'affranchir de ces problèmes de continuité, en choisissant, *a posteriori*, l'hypothèse qui présentait la meilleure vraisemblance de façon globale. Dans notre cas, ces hypothèses sont représentées par des *clusters* permettant de considérer l'aspect multi-modal de la loi *a posteriori* de la position de l'utilisateur. La problématique associée à ces discontinuités consiste à déterminer le meilleur chemin de *cluster* parmi les *clusters* estimés en temps réel. L'algorithme de Viterbi permet finalement d'obtenir des résultats probants, en impliquant une diminution de la valeur moyenne du rmse globale de 28%. Toutefois, cette approche reste perfectible sur certains aspects.

- En effet, l'estimation en temps différé se base ici uniquement sur les barycentres des *clusters* qui ont été estimés en temps réel. Cette estimation en temps différé sera intrinsèquement limitée en terme de potentielle amélioration par rapport au temps réel. Si par exemple aucune particule n'a pu suivre la bonne hypothèse en temps réel, et *a fortiori* aucun *cluster*, l'approche en temps différé ne saura pas retrouver cette position.
- Le nombre de *clusters* joue un rôle significatif pour cette approche. Il serait intéressant de considérer un nombre de *clusters* adaptatif en fonction de la variance associée à chacun des *clusters*. Le nombre de *clusters* augmenterait lorsque différentes

hypothèses claires se démarqueront, et celui-ci diminuerait lorsque certaines de ces hypothèses ne seront plus considérées comme viables.

- Enfin, dans notre cas, les algorithmes ont été appliqués sur l'ensemble de la trajectoire. Mais ils auraient pu être appliqués sur des fenêtres de taille réduite, uniquement lorsque qu'une discontinuité en temps réel aurait été détectée. Cette approche aurait nécessité de savoir correctement délimiter la taille de fenêtre pertinente autour de cette discontinuité pour qu'elle soit correctement corrigée.

3.6 Références

- ALAOUI, F. T., D. BETAILLE et V. RENAUDIN. 2016, «A multi-hypothesis particle filtering approach for pedestrian dead reckoning», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–8. viii, 61
- BEAUREGARD, S., M. KLEPAL et collab.. 2008, «Indoor PDR performance enhancement using minimal map information and particle filters», dans *Position, Location and Navigation Symposium, 2008 IEEE/ION*, IEEE, p. 141–147. 58
- BERNSTEIN, D. et A. KORNHAUSER. 1998, «An introduction to map matching for personal navigation assistants», . viii, 57
- DE FREITAS, N., C. ANDRIEU, P. HØJEN-SØRENSEN, M. NIRANJAN et A. GEE. 2001, «Sequential Monte Carlo methods for neural networks», dans *Sequential Monte Carlo methods in practice*, Springer, p. 359–379. 65
- DUBARRY, C. 2012, *Méthodes de lissage et d'estimation dans des modèles à variables latentes par des méthodes de Monte-Carlo séquentielles*, thèse de doctorat, Institut National des Télécommunications. 78, 79
- EBNER, F., T. FETZER, F. DEINZER, L. KÖPING et M. GRZEGORZEK. 2015, «Multi sensor 3D indoor localisation», dans *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, IEEE, p. 1–11. viii, 59
- FETZER, T., F. EBNER, F. DEINZER, L. KÖPING et M. GRZEGORZEK. 2016, «On Monte Carlo smoothing in multi sensor indoor localisation», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–8. 64, 68
- GALČÍK, F. et M. OPIELA. 2016, «Grid-based indoor localization using smartphones», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–8. viii, 59, 60
- GODSILL, S., A. DOUCET et M. WEST. 2001, «Maximum a posteriori sequence estimation using Monte Carlo particle filters», *Annals of the Institute of Statistical Mathematics*, vol. 53, n° 1, p. 82–96. 77
- GUSTAFSSON, F. 2010, «Particle filter theory and practice with positioning applications», *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, n° 7, p. 53–82. 68
- HILSENBECK, S., D. BOBKOV, G. SCHROTH, R. HUITL et E. STEINBACH. 2014, «Graph-based data fusion of pedometer and wifi measurements for mobile indoor positioning», dans *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, p. 147–158. 59

- JAIN, A. K. 2010, «Data clustering : 50 years beyond K-means», *Pattern recognition letters*, vol. 31, n° 8, p. 651–666. 70
- KLEPAL, M., S. BEAUREGARD et collab.. 2008, «A backtracking particle filter for fusing building plans with PDR displacement estimates», dans *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, IEEE, p. 207–212. viii, 64, 68
- LAMY-PERBAL, S., N. GUÉNARD, M. BOUKALLEL et A. LANDRAGIN-FRASSATI. 2015, «A HMM map-matching approach enhancing indoor positioning performances of an inertial measurement system», dans *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, IEEE, p. 1–4. 59
- NURMINEN, H., A. RISTIMAKI, S. ALI-LOYTTY et R. PICHÉ. 2013, «Particle filter and smoother for indoor localization», dans *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, IEEE, p. 1–10. 64, 68
- RAY, S. et B. G. LINDSAY. 2005, «The topography of multivariate normal mixtures», *Annals of Statistics*, p. 2042–2065. 75
- SCHMITT, S., L. ZECH, T. WILLEMSSEN, H. STERNBERG et M. KYAS. 2016, «Conditional erosion to estimate routing graph out of floor plans», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–8. 59
- SPASSOV, I. 2007, «Algorithms for map-aided autonomous indoor pedestrian positioning and navigation, thèse de doctorat, École Polytechnique Fédérale de Lausanne», . viii, 62
- VITERBI, A. 1967, «Error bounds for convolutional codes and an asymptotically optimum decoding algorithm», *IEEE transactions on Information Theory*, vol. 13, n° 2, p. 260–269. 77
- WANG, R., R. SHROFF, Y. ZHA, S. SESHAN et M. VELOSO. 2015, «Indoor trajectory identification : Snapping with uncertainty», dans *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, p. 4901–4906. viii, 63
- WANG, R., M. VELOSO et S. SESHAN. 2013, «Iterative snapping of odometry trajectories for path identification», *Lecture Notes in Computer Science*, vol. 8371, p. 432–443. viii, 62

Chapitre 4

Heuristiques d'amélioration en temps différé d'un filtre particulaire

Sommaire

4.1 Méthode de répétition d'un filtre particulaire	88
4.1.1 Présentation de l'approche	88
4.1.2 Caractérisation non-supervisée de la trajectoire	89
4.1.3 Augmentation de la variabilité du filtre	94
4.1.4 Résultats	96
4.1.5 Synthèse	101
4.2 Couplage de la fusion paramétrique des mesures RSS et IMU avec un filtre particulaire	102
4.2.1 Expression de la trajectoire WIB	102
4.2.2 Algorithme WIB + PF	102
4.2.3 Résultats	105
4.2.4 Synthèse	109
4.3 Méthode de sélection des trajectoires	110
4.3.1 Présentation de l'approche	110
4.3.2 Formalisation de la problématique	111
4.3.3 Méthodes d'apprentissage statistique supervisé	112
4.3.4 Résultats	115
4.3.5 Synthèse	118
4.4 Références	119

4.1 Méthode de répétition d'un filtre particulaire

L'approche de lissage décrite dans le chapitre 3 a permis une première estimation de la trajectoire en temps différé, en choisissant le meilleur chemin parmi les barycentres des *clusters* de particules estimées en temps réel. Néanmoins, les performances d'une telle approche sont limitées par l'estimation en temps réel. En effet, si la bonne hypothèse n'a pas été estimée par un *cluster* en temps réel, le lissage ne permettra pas de l'estimer en temps différé. On propose alors une heuristique qui consistera à répéter plusieurs fois un filtre particulaire afin d'obtenir différentes estimations de trajectoire. Ces estimations seront alors comparées et la trajectoire finale estimée en temps différée, que l'on notera $\hat{\mathbf{x}}_{1:K}^{mcpf}$, sera alors donnée par la meilleure estimation.

4.1.1 Présentation de l'approche

On note p_{est} la simulation d'une trajectoire en utilisant le filtre particulaire et n_{mc} le nombre de répétitions du filtre. Cette comparaison se fait séquentiellement, et l'approche globale de la répétition du filtre est présentée dans le schéma de la figure 4.1 :

1. Dans un premier temps, une première réalisation du filtre permet de définir la trajectoire courante $\hat{\mathbf{x}}_{1:K}^{mcpf} \sim p_{est}$.
2. Pour chaque nouvelle répétition :
 - une nouvelle trajectoire est simulée $\hat{\mathbf{x}}_{1:K}^{(i)} \sim p_{est}$
 - les trajectoires $\hat{\mathbf{x}}_{1:K}^{mcpf}$ et $\hat{\mathbf{x}}_{1:K}^{(i)}$ sont comparées
 - la trajectoire courante est alors remplacée par la meilleure trajectoire
3. La trajectoire finale correspond à la trajectoire courante après n_{mc} itérations

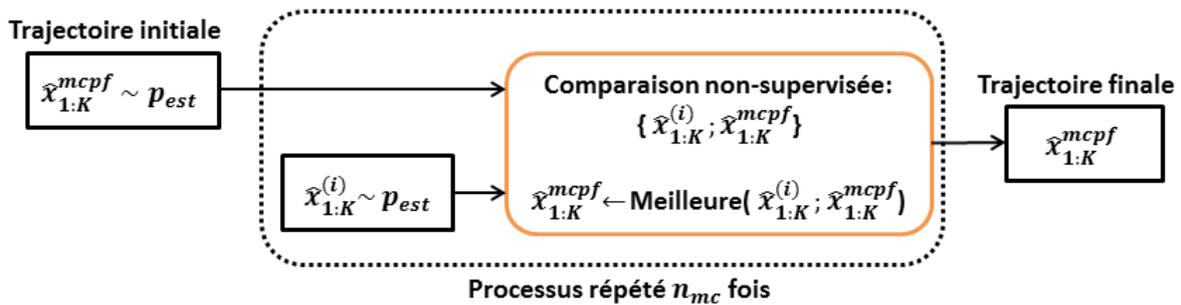


FIGURE 4.1 – Schéma de l'heuristique de répétition du filtre

Cette approche présente une problématique centrale. A chaque itération, la trajectoire courante est comparée avec une nouvelle estimation de trajectoire. Cette comparaison doit être faite sans la connaissance de la trajectoire de référence, c'est-à-dire de façon non-supervisée. La section suivante présentera les deux scores qui seront mis en oeuvre afin de déterminer si la nouvelle estimation de trajectoire est meilleure que la trajectoire courante ainsi que la règle de décision associée.

Par ailleurs, cette approche n'est pas équivalente à un filtre particulaire possédant $n_{mc} \times n_p$ particules. Dans un premier temps, la mémoire nécessaire à un très grand nombre de particules pourra rapidement devenir trop importante. De plus, il a été observé que des répétitions indépendantes d'un filtre offrent une plus grande diversité de trajectoires qu'en augmentant le nombre de particules ou le nombre de *clusters*.

4.1.2 Caractérisation non-supervisée de la trajectoire

L'objectif de cette partie est de répondre à la problématique qui est de classer différentes estimations d'un même filtre de façon non-supervisée. Un premier score naturel consiste à s'intéresser à la continuité, c'est-à-dire l'absence de saut, mais des cas ambigus subsistent. Donc un deuxième score basé sur les mesures RSS est utilisé. On présentera dans un premier temps l'évaluation de chacun des scores, puis dans un second temps, la façon dont ces scores sont couplés.

4.1.2.1 Fonction de score sur la continuité

Comme présenté dans le chapitre 3, l'estimation en temps réel peut impliquer des changements de *clusters* et causer des discontinuités. Pour une trajectoire $\tilde{\mathbf{x}}_{1:K}^{(i)}$, on définit le vecteur d'estimation de la longueur de pas que l'on note $\tilde{p}_{1:K}^{(i)}$, où

$$\tilde{p}_k^{(i)} = \sqrt{(\tilde{x}_k^{(i)} - \tilde{x}_{k-1}^{(i)})^2 + (\tilde{y}_k^{(i)} - \tilde{y}_{k-1}^{(i)})^2} \quad (4.1)$$

et $(\tilde{x}_k^{(i)}, \tilde{y}_k^{(i)})$ correspond à la position estimée à l'instant k . Des discontinuités dans la trajectoire, comme représentées dans les exemples des figures 3.12 et 3.13, vont impliquer une augmentation de la variance empirique sur les longueurs de pas notée $s_p^2(i)$:

$$s_p^2(i) = \sum_{k=1}^K \frac{(\bar{p}^{(i)} - \tilde{p}_k^{(i)})^2}{K} \quad (4.2)$$

On proposera donc un modèle de pas permettant d'évaluer la vraisemblance associée à cette variance empirique. La vraisemblance sur cette variable pourra définir un score sur la continuité. La longueur du pas mesurée par les capteurs inertiels l_k ainsi que le biais de longueur de pas ϵ_k sont considérés comme des variables aléatoires gaussiennes. La longueur de pas réel est alors donnée par :

$$p_k = (1 + \epsilon_k)l_k \quad (4.3)$$

Déterminer la loi des p_k selon ce modèle devient fastidieux puisqu'il fait intervenir la loi du produit de deux variables aléatoires normales indépendantes. De plus, en considérant ce modèle, le score tiendra compte de la valeur du changement d'échelle qui ne nous intéresse pas dans le cas présent, seules les discontinuités présentant un intérêt. En outre, si

on imagine que l'utilisateur se déplace avec une allure constante, alors on peut supposer que le *cluster* qui aura estimé la position en temps réel correspond bien au déplacement de l'individu. On suppose alors le modèle suivant :

$$\tilde{p}_k^{(i)} \sim \mathcal{N}\left(p_0, \sigma_p^2\right) \quad (4.4)$$

où σ_p^2 est la variance associée à la longueur de pas. p_0 correspond à la valeur moyenne de la longueur de pas du trajet d'un individu qui est une variable aléatoire qu'on ne cherchera pas à estimer. Une limite de ce modèle simplifié réside dans l'hypothèse de l'allure constante de l'utilisateur. En pratique, cette limite n'impacte pas significativement les résultats. En considérant chaque pas comme indépendant, on obtient alors la loi associée à la variance empirique :

$$\frac{K}{\sigma_p^2} s_p^2(i) \sim \chi^2(K-1) \quad (4.5)$$

Pour K grand, on peut obtenir l'approximation suivante :

$$\sqrt{2K} \frac{s_p(i)}{\sigma_p} - \sqrt{2(K-2)} \sim \mathcal{N}(0, 1) \quad (4.6)$$

et en considérant que pour un grand nombre de pas K , $K \simeq K-2$, on a alors :

$$\sqrt{2K} \left[\frac{s_p(i)}{\sigma_p} - 1 \right] \sim \mathcal{N}(0, 1) \quad (4.7)$$

On définit directement la vraisemblance associée à la variance empirique :

$$L(s_p(i)) = \frac{1}{\sqrt{2\pi}} \exp \left[-K \left(\frac{s_p(i)}{\sigma_p} - 1 \right)^2 \right] \quad (4.8)$$

Cette vraisemblance n'est pas monotone selon la variance empirique. En effet, deux variances différentes peuvent aboutir à la même valeur de vraisemblance. Si on considère deux variances $s_1 \neq s_2$ telle que $s_1 + s_2 = 2\sigma_p$, alors $L(s_1) = L(s_2)$. Pour cette raison, on définit une nouvelle fonction de score, également basée sur le modèle de l'équation 4.4. Ce score est bien décroissant en fonction de la variance des pas. Le détail des calculs de ce score a été reporté dans l'annexe A.3.

$$s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)}) = -K \left[\log(\sigma_p \sqrt{2\pi}) + \frac{s_p^2(i)}{\sigma_p^2} \right] \quad (4.9)$$

4.1.2.2 Fonction de score sur les mesures RSS

Une seconde fonction de score est également mise en oeuvre, basée sur l'ensemble des mesures RSS. Les mesures RSS associées sont mesurées à différents instants, $\mathbf{z}_{mes} = z_{mes}(t_{1:N})$ où N correspond au nombre de mesures total, et t_n l'instant temporel de la n -ième mesure RSS. On a pu observer dans le chapitre 2 que l'utilisation d'un modèle centré comme dans l'équation 2.30 permettait de calibrer le modèle de propagation. On définira un modèle considérant un biais de calibration noté z_0 . Par exemple, z_0 peut être causé par une mauvaise connaissance du gain de réception du téléphone. Ce biais est une variable aléatoire que l'on ne cherche pas à estimer. On suppose alors le modèle suivant :

$$z_{mes}(t_n) - z_{mod}^{(i)}(t_n) \sim \mathcal{N}(z_0, \sigma_z^2) \quad (4.10)$$

où $z_{mod}^{(i)}(t_n)$ est calculé grâce au modèle décrit par l'équation 2.19 associé à la position de la trajectoire $\tilde{\mathbf{x}}_{1:K}^{(i)}$ à l'instant t_n . De façon analogue au score sur la continuité, on peut définir une fonction de score sur les mesures RSS noté $s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(i)})$:

$$s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(i)}) = -N \left[\log(\sigma_z \sqrt{2\pi}) + \frac{s_z^2(i)}{\sigma_z^2} \right] \quad (4.11)$$

où $s_z^2(i)$ est la variance empirique du vecteur de l'écart entre le modèle et les mesures :

$$s_{rss}^2(i) = \frac{1}{N} \sum_{n=1}^N \left(\left[\mathbf{z}_{mes}^{(i)} - \mathbf{z}_{mod}^{(i)} \right] - \left[z_{mes}^{(i)}(t_n) - z_{mod}^{(i)}(t_n) \right] \right)^2. \quad (4.12)$$

4.1.2.3 Couplage des scores

Ces deux scores sont complémentaires, et l'on décide d'adopter la règle de décision suivante : pour qu'une réalisation $\tilde{\mathbf{x}}_{1:K}^{(i)}$ soit meilleure qu'une autre $\tilde{\mathbf{x}}_{1:K}^{(j)}$, elle doit l'être au sens du score de continuité et du score RSS :

$$\tilde{\mathbf{x}}_{1:K}^{(i)} \text{ est meilleure que } \tilde{\mathbf{x}}_{1:K}^{(j)} \iff s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)}) > s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(j)}) \ \& \ s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(i)}) > s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(j)}) \quad (4.13)$$

L'utilisation conjointe de deux scores différents permet une robustesse quant à la caractérisation non-supervisée. Néanmoins, le choix de cette règle de décision ne permet pas de définir une relation d'ordre entre toutes les trajectoires. Par exemple, si une trajectoire A possède un meilleur score de continuité qu'une trajectoire B, mais un moins bon score RSS, alors, on ne saura déterminer la meilleure trajectoire. Selon notre algorithme, comme les deux trajectoires A et B ne sont pas comparables, la meilleure estimation sera donnée par l'estimation courante. Ceci signifie que si la trajectoire A a été estimée avant la trajectoire B, alors la meilleure estimation sera donnée par la trajectoire A. Toutefois, on formule l'hypothèse que la trajectoire correcte est meilleure au sens des deux scores qu'une trajectoire qui aura divergé. Cette hypothèse a été vérifiée sur un grand nombre d'exemples et sera également illustrée dans l'exemple de la figure 4.2 dans la partie suivante. Partant de cette hypothèse, on suppose que, pour un nombre de répétitions suffisamment grand, une trajectoire correcte au sens des deux scores sera estimée et permettra d'améliorer l'estimation courante. Cette approche est décrite dans l'algorithme 8.

Algorithme 8 : Méthode de répétition du filtre et règle de décision

Données : $z_{mes}(t_{1:N})$
Résultat : $\hat{\mathbf{x}}_{1:K}^{(mc)}$

- 1 $s_{cont}(max) = -10^{10}$;
- 2 $s_{rss}(max) = -10^{10}$;
- 3 **pour** $i = 1 : n_{mc}$ **faire**
- 4 Simuler $\tilde{\mathbf{x}}_{1:K}^{(i)} \sim p_{est}$;
- 5 **si** $s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)}) > s_{cont}(max)$ & $s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(i)}) > s_{rss}(max)$ **alors**
- 6 $s_{cont}(max) = s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)})$;
- 7 $s_{rss}(max) = s_{rss}(\tilde{\mathbf{x}}_{1:K}^{(i)})$;
- 8 $\hat{\mathbf{x}}_{1:K}^{(mc)} = \tilde{\mathbf{x}}_{1:K}^{(i)}$;

4.1.2.4 Illustration des scores

On applique les différents scores sur un exemple applicatif afin d'illustrer notamment l'hypothèse selon laquelle la bonne trajectoire est bien meilleure que les autres au sens des deux scores. On considère dans la figure 4.2 5 trajectoires différentes que l'on cherche à comparer en fonction des différents scores. Ces scores, de même que les performances associées à chacune des trajectoires, sont représentés dans le tableau 4.1. On a fixé les valeurs $\sigma_p = 50\text{cm}$ et $\sigma_z = 7\text{dBm}$ pour évaluer les scores dans cet exemple, mais également pour la suite des travaux. Ces valeurs correspondent à une réalité physique pour le cas des mesures RSS, observées sur un ensemble de trajectoires, mais également à un compromis permettant une stabilité numérique pour le score de continuité. En effet, un écart-type de 10cm correspond à une échelle de grandeur pertinente pour un écart-type associé à une longueur de pas, mais impliquera un score trop sensible dans la mesure où des discontinuités de parfois 10m peuvent subvenir. Ceci justifie des écarts dans les ordres de grandeurs des deux scores. On peut faire différentes remarques :

- La trajectoire de référence (en haut à gauche) présente le score le plus élevé pour la continuité, mais le score RSS le plus grand est obtenu par l'estimation 1 (en haut à droite). Elles ne sont donc pas comparables selon la règle conjointe des scores. Toutefois, on remarque que ces deux trajectoires sont presque équivalentes puisque l'erreur rmse est de 1m et la valeur du %d5 est de 100%.
- L'estimation 2 (au milieu à gauche) est la figure qui présente la plus grande discontinuité, et implique logiquement le score continuité le plus faible. Néanmoins, son score RSS reste le plus élevé si l'on excepte la trajectoire de référence, ainsi que l'estimation 1, car la position globale de cette trajectoire est correcte.
- L'estimation 3 (au milieu à droite) a un meilleur score continuité que l'estimation 2. En revanche, l'estimation 3 a un moins bon score RSS car une partie de cette trajectoire a suivi un chemin, à la fin de la trajectoire, qui ne correspond à la trajectoire parcourue. On note également que les performances pour l'estimation 2 et 3 sont comparables en terme d'erreur rmse mais également de %d5.
- L'estimation 4 (en bas à gauche) est la plus mauvaise estimation de trajectoire. Toutefois, en comparaison des scores de continuité, elle est mieux classée que les estimations 2 et 3, car elle présente des sauts de trajectoire plus faibles. Par ailleurs, le score RSS associé est le moins élevé car la position globale de cette trajectoire se trouve éloignée de la trajectoire parcourue à plusieurs reprises.
- La règle conjointe que nous avons définie ne permet pas de classer les estimations 2, 3 et 4 entres elles. De même, la trajectoire de référence et l'estimation 1 ne peuvent être ordonnées. Néanmoins, la trajectoire de référence, ainsi que l'estimation 1 sont meilleures que les estimations 2, 3 et 4.

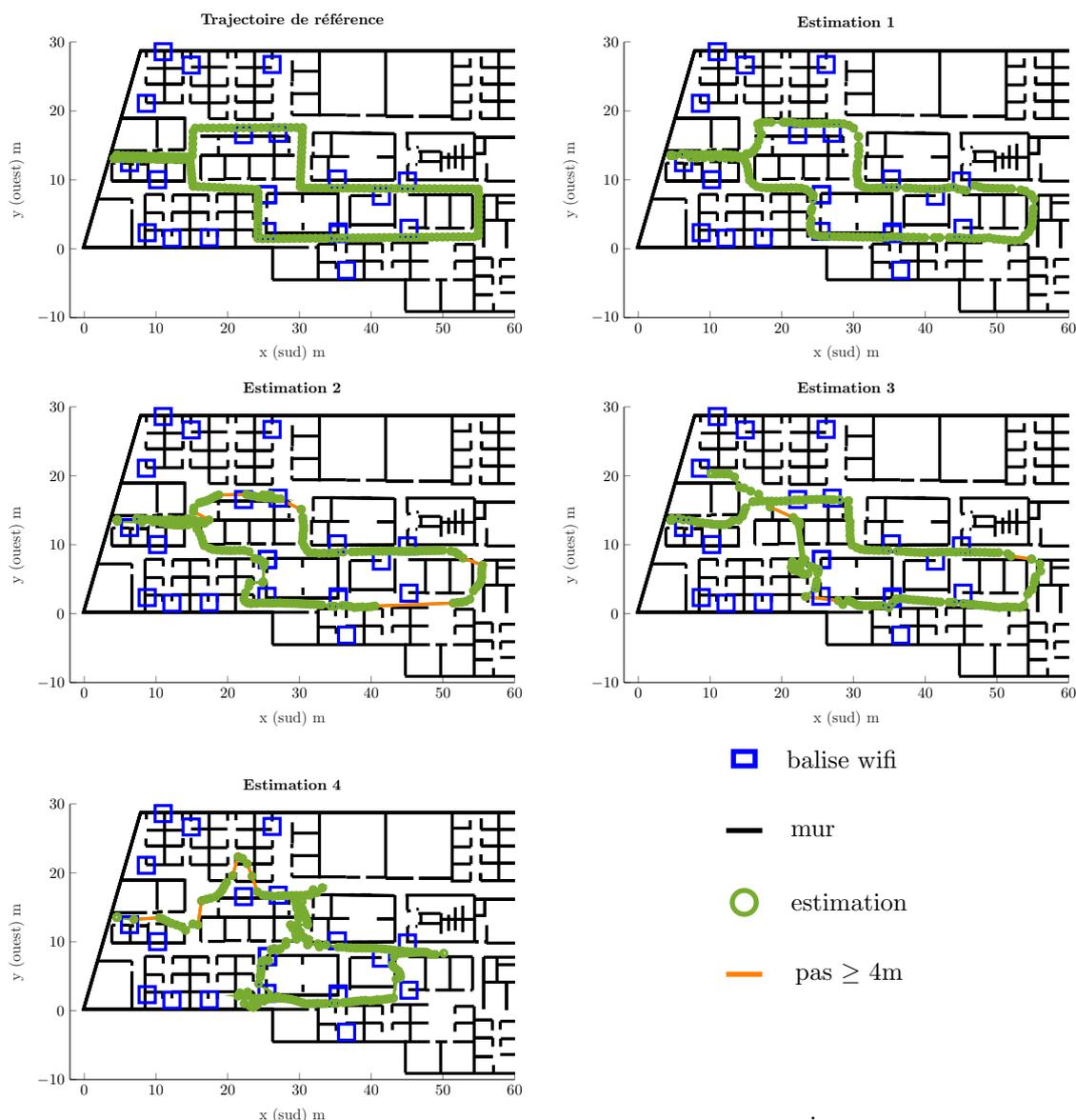


FIGURE 4.2 – Illustration des scores combinés sur un cas applicatif concret sur différentes estimations de la trajectoire réelle ainsi que sur la trajectoire de référence. En orange sont représentés les pas supérieurs à 4 mètres.

Trajectoire	Score continuité	Score RSS	rmse(m)	&d5 (%)
Référence	-0.52	-8.67×10^4	0	100
Estimation 1	-16.19	-8.15×10^4	1.11	100
Estimation 2	-617	-9.39×10^4	3.29	86.23
Estimation 3	-437	-10.7×10^4	3.49	83.81
Estimation 4	-395	-13.7×10^4	9.23	50.58

TABLEAU 4.1 – Représentation des scores sur différentes estimations d'un cas applicatif concret

4.1.3 Augmentation de la variabilité du filtre

L'objectif de cette partie est d'augmenter la variabilité des trajectoires estimées par le filtre. En effet, si la variabilité est trop faible et que les trajectoires estimées sont très proches, le gain obtenu par la répétition du filtre se trouvera limité. Néanmoins, l'augmentation de la variabilité doit tout de même permettre d'estimer une trajectoire correcte avec un nombre raisonnable d'itérations. Par exemple, si on considère le cas extrême où l'estimation serait donnée uniquement par une marche aléatoire sur la carte du bâtiment, la bonne trajectoire finira par être estimée et la variabilité des trajectoires sera importante. Mais cette stratégie nécessitera un trop grand nombre de répétitions afin d'estimer une trajectoire correcte. Différentes stratégies ont été mises en oeuvre pour augmenter la variabilité de la loi p_{est} . Par exemple, un nombre de particules plus faible a été testé, mais ce choix détériore grandement la qualité d'estimation de la trajectoire et ne permet pas systématiquement de sélectionner une estimation satisfaisante. Également, une augmentation de la variabilité du bruit d'état implique le même phénomène, c'est-à-dire une grande variabilité d'estimation associée à une moins bonne qualité. On propose alors une stratégie qui consiste à traiter la trajectoire par morceaux.

4.1.3.1 Traitement par morceaux de la trajectoire

L'idée de cette stratégie est de traiter la trajectoire par morceaux, c'est-à-dire de séparer la trajectoire en sous-trajectoires. Par exemple, une trajectoire de 100m peut être séparée en 2 trajectoires plus courtes de 50m. On peut alors appliquer l'approche de répétition d'un filtre particulaire, comme il a été présenté dans la section précédente, mais sur chacune des sous-trajectoires. La figure 4.3 illustre cette approche en considérant une trajectoire séparée en 3 sous-trajectoires (ST), et avec un nombre de répétitions du filtre $n_{mc} = 4$.

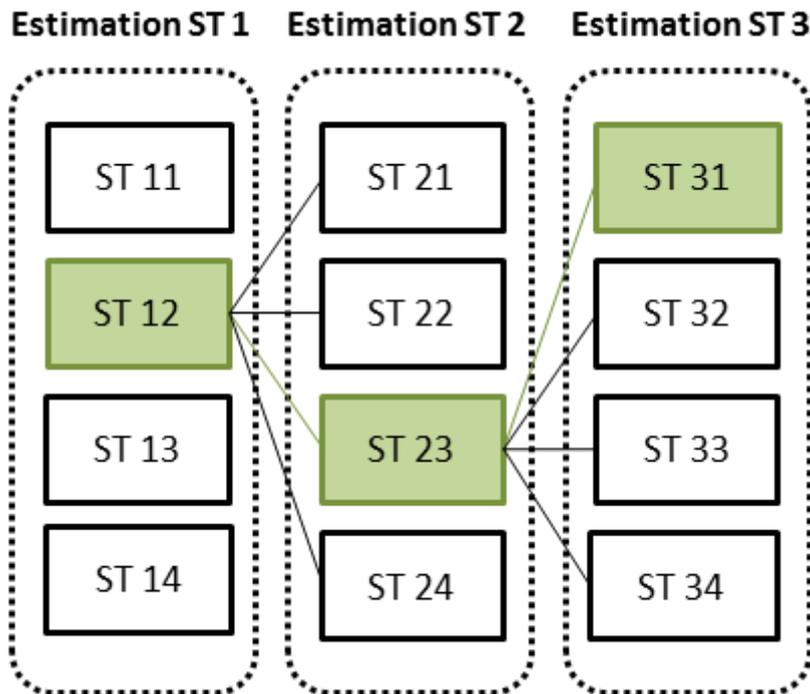


FIGURE 4.3 – Schéma de l'approche de la séparation en sous-trajectoires (ST).

Dans un premier temps, la ST 1 est estimée en appliquant l'algorithme 8 et une esti-

mation de cette ST est obtenue. Dans notre exemple, celle-ci correspond à la deuxième réalisation du filtre. Puis, le point de départ de la ST 2 est fixé comme étant le point d'arrivée de la ST 1. L'algorithme 8 est de nouveau appliqué pour obtenir une estimation de la ST 2. Le point d'arrivée de l'estimation de ST 2 devient le point de départ de ST 3, et l'algorithme 8 est appliqué une dernière fois. L'estimation de la trajectoire finale est obtenue en concaténant les sous-trajectoires estimées en occurrence [ST 12;ST 23;ST 31].

Cette séparation en sous-trajectoires permet deux avantages.

- Une longue trajectoire présente un risque plus élevé de diverger qu'une trajectoire courte. Si une trajectoire de 100m a été correcte, mais n'a divergé qu'à partir de 80m, alors l'ensemble de cette trajectoire ne sera pas conservé et la première partie correctement estimée de la trajectoire ne sera pas conservée. Utiliser des sous-trajectoires réduit la problématique d'estimation, en considérant deux sous-trajectoires de 50m. Il s'agit alors de résoudre deux problèmes d'estimation plus simple.
- La variabilité du filtre est augmentée. En effet, répéter un nombre n_{mc} de filtres sur deux sous-trajectoires permet d'obtenir un plus grand nombre de possibilités qu'avec un filtre répété n_{mc} fois sur une seule trajectoire.

Ici, l'aspect séquentiel dans l'estimation des sous-trajectoires a son importance. En effet, si une sous-trajectoire a été mal estimée, alors le point de départ permettant d'estimer la sous-trajectoire suivante sera faux. Cette erreur risque alors de se propager et de détériorer également les sous-trajectoires suivantes. Par ailleurs, dans le modèle d'état qui est utilisé pour le filtre particulaire, on considère un bruit d'état important au point de départ, notamment sur le cap initial. Dans notre cas, pour chaque nouveau segment, ce cap initial aura déjà été estimé par le segment précédent. Pour le segment suivant, ce cap sera bien intégré dans le modèle, notamment en considérant une valeur de variance initiale plus faible. On détaille alors les étapes dans l'algorithme 9. On note $k_0, \dots, k_{n_{sep}}$ les indices de temps permettant de séparer la trajectoire totale, que l'on appellera les instants de séparation et n_{sep} correspond au nombre de séparations. On définit alors une sous-trajectoire comme étant la partie de la trajectoire comprise entre deux instants successifs de séparation. On définit également $p_{est}^{k_{n-1}:k_n}$ le filtre particulaire appliqué entre les instants de séparation k_{n-1} et k_n , et on note $\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)}$ une réalisation de ce filtre.

Par ailleurs, comme pour la partie 2.3.1.1, on définit une distance d_{sep} qui permettra de définir le nombre de sous-trajectoires :

$$n_{sep} = e\left(\frac{d_{tot}}{d_{mc}}\right) \quad (4.14)$$

où d_{tot} est la distance totale de la trajectoire et $e()$ est la fonction partie entière. Choisir une distance trop petite augmente la variabilité des trajectoires, mais augmente également le risque qu'une sous-trajectoire soit mal estimée. En effet, les scores continuité ou RSS mesurent des variances, et sont plus robustes en étant appliqués sur des sous-trajectoires plus longues. Comme pour la partie 2.3.1.1, les différents instants de séparation sont fixés *a priori*, puis la répétition des filtres est mise en oeuvre. Le nombre d'opérations de cette méthodologie utilisant des sous-trajectoires est le même que pour la répétition du filtre sans séparation. En effet, le nombre d'estimations est multipliés par n_{sep} mais chacune de ces estimations s'applique sur des trajectoires plus petites. Néanmoins,

Algorithme 9 : Méthode de répétition du filtre et règle de décision avec une séparation de trajectoire

Données : $Z_{mes}(t_{1:N})$
Résultat : $\hat{\mathbf{x}}_{1:K}^{(mc,sep)}$

- 1 **pour** $n = 1 : n_{sep}$ **faire**
- 2 $s_{cont}(max) = -10^{10};$
- 3 $s_{r_{ss}}(max) = -10^{10};$
- 4 **pour** $i = 1 : n_{mc}$ **faire**
- 5 Simuler $\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)} \sim p_{est}^{k_{n-1}:k_n};$
- 6 **si** $s_{cont}(\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)}) > s_{cont}(max)$ & $s_{r_{ss}}(\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)}) > s_{r_{ss}}(max)$ **alors**
- 7 $s_{cont}(max) = s_{cont}(\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)});$
- 8 $s_{r_{ss}}(max) = s_{r_{ss}}(\tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)});$
- 9 $\hat{\mathbf{x}}_{k_{n-1}:k_n}^{(mc,sep)} = \tilde{\mathbf{x}}_{k_{n-1}:k_n}^{(i)};$

dans notre application de cette méthode, le temps de calcul effectif s'avère être plus important en considérant une séparation en sous-trajectoires. En effet, un temps inhérent pour l'initialisation des données, et également de mise en forme des résultats, est nécessaire pour chaque utilisation d'un filtre. L'utilisation de $n_{mc} \times n_{sep}$ filtres particuliers, même sur des trajectoires plus courtes, augmentera sensiblement le temps effectif de réalisation par rapport à l'utilisation de n_{mc} filtres particuliers sur des trajectoires plus longues (environ $\times 0.8n_{sep}$).

4.1.4 Résultats

4.1.4.1 Illustration sur une expérience

Présentation de la trajectoire de référence

L'expérience choisie pour l'illustration correspond à environ 230m de marche, la trajectoire de référence étant présentée dans la figure 4.4. La couverture wifi était totale pour ce scénario avec la présence de 10 balises wifi. L'utilisateur effectue deux boucles du même trajet. Par ailleurs, durant l'expérience, l'utilisateur suivra certains scénarios, tels que répondre au téléphone ou s'arrêter pour lire une affiche.

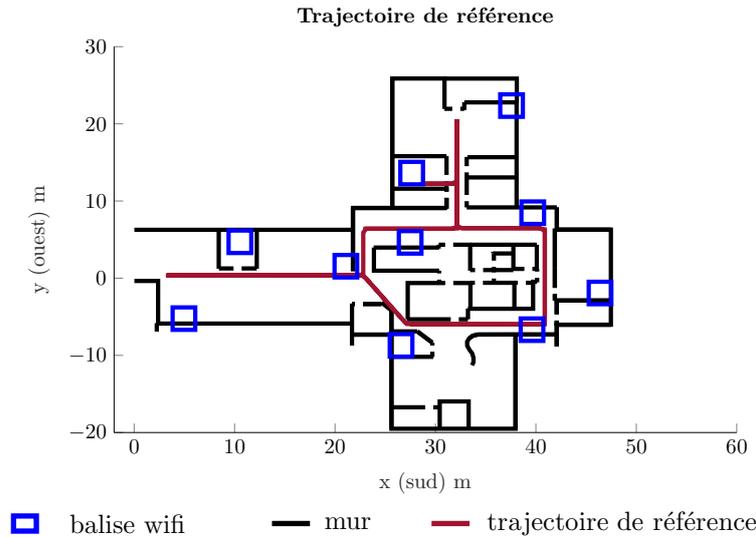


FIGURE 4.4 – Représentation de la trajectoire de référence

Résultats sur l'exemple d'illustration

Dans la figure 4.5, sont représentées 4 estimations de la trajectoire réelle. La première correspond à l'estimation en temps réel par le filtre particulaire. La seconde correspond à $n_{mc} = 30$ répétitions d'un filtre particulaire sans séparation de la trajectoire. La troisième (respectivement quatrième) correspond à $n_{mc} = 30$ répétitions d'un filtre particulaire en considérant des sous-trajectoires avec une distance de séparation de 50m (respectivement 100m). On pose également les valeurs suivantes, $\sigma_{\beta_0}(1) = 180^\circ$ pour la première sous-trajectoire, et pour tout autre segment $n > 1$, on posera $\sigma_{\beta_0}(n) = 30^\circ$. Les performances associées à chacune de ces estimations sont reportées dans le tableau 4.2. Dans un premier temps, on observe les performances ci-dessous :

1. La trajectoire estimée en temps réel (en haut à gauche) propose une première estimation que l'on peut encore améliorer. En effet, cette trajectoire évaluée en temps réel a divergé sur la seconde moitié de la trajectoire impliquant une valeur de %d5 d'environ 55%.
2. L'estimation 1 (en haut à droite) a répété 30 fois un filtre particulaire. Ici, l'estimation diverge également, mais diverge plus tard que l'estimation en temps réel. Il y a bien un gain sur les performances puisque la valeur de %d5 est de 68% par rapport à l'approche en temps réel.
3. L'estimation 2 (en bas à gauche) met en oeuvre une séparation en deux sous-trajectoires. La première sous-trajectoire a été correctement estimée sur le premier parcours. La seconde sous-trajectoire présente des erreurs sur certains couloirs, mais propose une estimation globale satisfaisante sur la trajectoire parcourue, avec une erreur %d5 de 84%, ce qui améliore la répétition d'un filtre de l'estimation par Monte-Carlo 1.
4. Enfin, l'estimation 3 (en bas à droite) considère 5 sous-trajectoires et la qualité globale de cette estimation se révèle moins bonne qu'avec 2 sous-trajectoires, avec une valeur de %d5 de 65%. Cette estimation est même moins bonne que l'estimation 1. En effet, une sous-trajectoire a divergé, et cette erreur s'est répandue sur les sous-trajectoires finales. Le point de départ de la dernière sous-trajectoire (en rouge dans

l'image) présente une erreur d'environ 12m sur la position réelle, et n'a plus la possibilité de retrouver la trajectoire correcte.

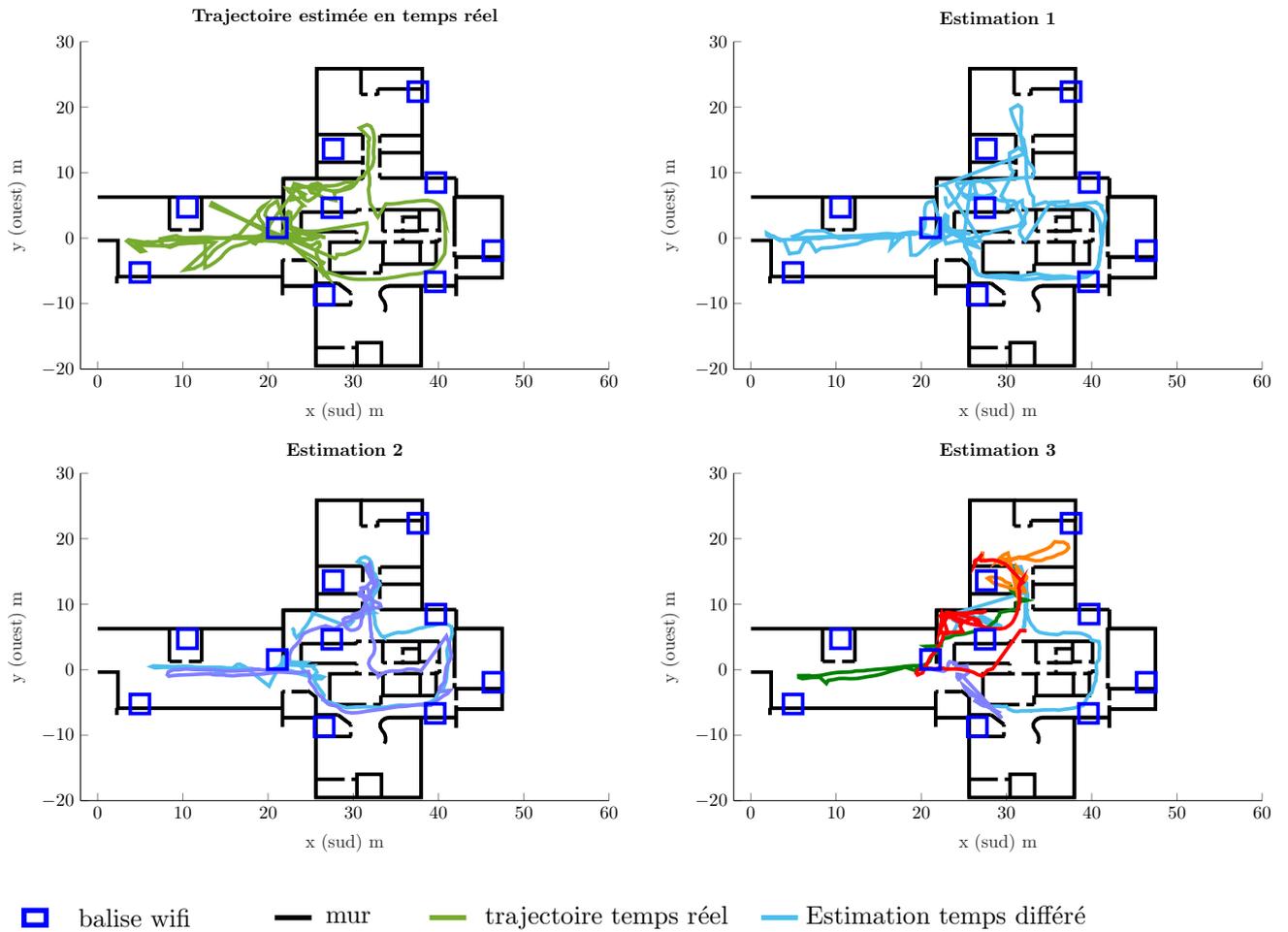


FIGURE 4.5 – Illustration de la méthode de répétition du filtre. Le nombre de répétition est fixé à $n_{mc} = 30$. Pour l'estimation 1, il n'y a pas de séparation de la trajectoire. Pour l'estimation 2, la distance de séparation est de $d_{sep} = 100m$. Pour l'estimation 3, la distance de séparation est de $d_{sep} = 50m$. Chaque couleur différente couleur correspond à la séparation des segments pour les estimations 2 et 3.

Trajectoire	rmse (m)	%d5 (%)	n_{mc}	d_{sep}
Estimation temps réel	7.87	54.86	-	-
Monte-Carlo 1	4.57	68.15	30	-
Monte-Carlo 2	3.79	84.34	30	100m
Monte-Carlo 3	7.14	65.84	30	50m

TABLEAU 4.2 – Évaluation de la méthode de répétition du filtre

On peut ainsi analyser certains éléments à partir des premiers résultats de cette approche :

- La trajectoire estimée en temps réel illustre que l'estimation en temps réel n'est pas robuste. En effet, si lorsque les *clusters* se perdent et que l'hypothèse correcte n'est plus suivie, comme c'est le cas dans l'exemple de la figure 4.4 (en haut à gauche), la qualité d'estimation de cette trajectoire sera ainsi détériorée, notamment dans les cas où les conditions expérimentales sont plus complexes. Utiliser une répétition du filtre permet d'augmenter la robustesse de l'estimation comparativement au temps réel puisque la meilleure des n_{mc} trajectoires au sens des scores est conservée.
- Mais cette approche de répétition du filtre reste limitée par les performances du filtre en temps réel sur l'ensemble de la trajectoire. Dans l'estimation 1, la première moitié de la trajectoire a bien été correctement estimée. Pourtant, les réalisations suivantes du filtre n'en tiennent pas compte et estiment à nouveau la trajectoire totale.
- Ceci justifie l'utilisation de sous-trajectoires. En effet, la probabilité que la réalisation d'un filtre soit correcte sur la moitié d'une trajectoire est plus grande que la probabilité que la réalisation du filtre soit correcte sur l'ensemble de cette trajectoire. Ici, le gain de performance est donc plus important puisque la première moitié a été correctement estimée, et le point de départ du second segment est également correcte (une erreur de 2m). Le gain de séparer la trajectoire en deux est significatif puisque le ratio %d5 augmente d'environ 25% pour le même nombre de répétitions du filtre.
- Toutefois, considérer des segments trop petits détériore la qualité des scores qui ne sont plus pertinents, et la question de la robustesse se pose à nouveau, puisque l'estimation 3 est de moins bonne qualité que l'estimation 2. Il suffit qu'un segment soit mal estimé pour que les sous-trajectoires suivantes ne soit pas correctement estimées. De plus, les scores sont plus pertinents pour des segments longs car ils sont généralement associés à un grand nombre de mesures RSS.
- Cette approche de répétition du filtre est probante comparativement à l'approche temps réel, mais elle est basée sur un compromis entre le coût de calcul et la robustesse de cette méthode. La robustesse du filtre est directement croissante en fonction de n_{mc} mais décroît selon d_{sep} . A l'inverse, le coût de calcul croît directement avec n_{mc} et est très sensible à la distance de séparation, comme illustré dans l'exemple de la figure 4.4.
- Ces choix de paramètres seront fixés de façon globale en appliquant cette approche sur un grand nombre d'expériences. Pour cet exemple, le choix très faible de la distance de séparation de 50m était délibérément choisi afin d'illustrer la perte de performance que peut impliquer un nombre de séparations trop faible et également la sensibilité de cette méthode par rapport à cette distance de séparation. En pratique, cette distance doit s'adapter selon deux critères :
 - la disponibilité des mesures RSS
 - l'épaisseur des couloirs.

En effet, dans des conditions d'estimation plus complexes, il faut considérer de grandes distances de séparations, c'est-à-dire au moins de l'ordre de 200m voire ne

pas appliquer de séparation de la trajectoire. Lorsque les conditions de mesures impliquent des cas d'estimation plus simples, on peut alors considérer des distances de séparation plus fines, de l'ordre de 100m. Ces valeurs reposent sur une application de ces approches à différentes trajectoires dans des situations d'estimation parfois complexes.

4.1.4.2 Résultats sur la base de données

Ces différentes méthodes sont testées sur un ensemble de 130 expériences. Les différents paramètres selon lesquels seront présentés les résultats sont : le nombre de particules de chaque filtre n_p , la distance de séparation pour les sous-segments d_{sep} ou encore le nombre d'itérations totale de chaque filtre, noté n_{mc} . Les résultats numériques sont reportés dans le tableau 4.3.

n_{mc}	d_{sep}	n_p	Moyenne %d5	Moyenne %d10	Moyenne rmse(m)
1	-	20	51.32	66.25	14.54
10	-	20	72.03	85.46	7.77
10	250m	20	73.26	86.66	7.00
30	250m	20	79.19	90.80	5.43
1	-	50	65.17	79.11	11.02
10	-	50	79.99	92.64	5.02
10	250m	50	81.31	92.89	5.32
30	250m	50	84.63	95.89	5.13
5	75m	500	83.76	94.39	4.78
5	250m	500	84.67	94.87	4.88
10	-	500	86.81	95.61	4.43
30	-	500	86.93	96.35	4.50
30	75m	500	85.71	95.69	4.48
30	250m	500	86.81	96.91	3.78
30	350m	500	87.58	97.16	4.00
Estimation temps réel		500	83.29	94.12	5.81

TABLEAU 4.3 – Résultats de la répétition du filtre en temps différé selon le nombre d'itérations, le nombre de particules et la distance de séparation sur un ensemble de 130 expériences

Dans un premier temps, un faible nombre de particules $n_p = 20$ a été testé. Les estimations sont fortement dégradées et restent inférieures à l'estimation en temps réel classique. En effet, l'erreur moyenne de %d5 pour les meilleures performances de 79.2% reste bien inférieure à la valeur de l'estimation en temps réel qui est de 83.3%. Toutefois, pour un nombre plus important de particules $n_p = 50$, en considérant un nombre de répétitions suffisamment grand, $n_{mc} = 30$, la moyenne de %d5 qui vaut 84.6% est améliorée par rapport au temps réel. Ceci illustre la problématique de la variabilité du filtre. Un faible nombre de particules aura une grande variance associée à de mauvaises performances, lorsqu'un filtre, utilisant $n_p = 50$ particules et avec $n_{mc} = 30$ répétitions, permet de meilleurs résultats que le filtre en temps réel. En fixant le nombre de particules à $n_p = 500$, la distance de séparation influe grandement sur la qualité globale de l'estimation. En effet, considérer une distance de séparation relativement faible $d_{sep} = 75m$ améliore l'estimation en temps réel avec une erreur moyenne de %d5 de 85.7%, mais les meilleurs résultats sont obtenus pour une distance de séparation de $d_{sep} = 350m$ avec une

erreur de 87.6%. On obtient des résultats intéressants sans la séparation des trajectoires, avec une erreur moyenne de %d5 de 86.9%. La trajectoire estimée par cette approche sera par la suite notée MCPF. Ceci est bien conforme avec les analyses développées dans l'exemple applicatif de la figure 4.4.

4.1.5 Synthèse

Afin d'améliorer l'approche précédente de la sélection du meilleur chemin, l'approche ici proposée permet d'explorer un espace plus large, permettant de corriger le cas où l'estimation en temps réel n'aura pas su estimer les bons couloirs par exemple. Cette approche tire profit de la composante aléatoire du filtre particulaire et considère ensuite une optimisation en répétant ce filtre. Dans notre cas, cette optimisation se fera de façon non-supervisée en considérant deux scores différents : l'un utilisant les mesures RSS et le second s'intéressant à la variance des pas. A partir du filtre, un certain nombre de trajectoires sont estimées, et la meilleure aux sens des scores servira d'estimation en temps différé. Il s'agit d'obtenir un compromis intéressant entre la diversité des trajectoires estimées et leur qualité. Pour cette raison, une stratégie de séparation des trajectoires en sous-trajectoires a été proposée. La conclusion est qu'une répétition globale du filtre sans la séparation permet d'obtenir de très bon résultats obtient par rapport au temps réel. Pour un nombre de 30 itérations, la valeur de %d5 augmente de %d5 83.29% à 86.93% mais également selon l'erreur rmse qui diminue de 5.81m à 4.50m. Les meilleurs résultats sont obtenus en considérant une séparation de la trajectoire en sous-trajectoires. Toutefois, pour des considérations de coût de calcul effectif, par la suite, l'estimation par la répétition du filtre sera considérée sans séparation de la trajectoire.

Cette approche de répétition du filtre résout la problématique de la robustesse en temps réel, mais reste améliorable, notamment sur deux aspects principaux :

- La répétition des filtres est faite de façon indépendante entre chaque réalisation. Ceci signifie qu'il n'a pas été possible de faire dépendre une nouvelle estimation de la meilleure estimation courante. Investiguer une dépendance entre les différentes réalisations représente une piste d'amélioration très intéressante. Par exemple, on peut en considérer des points de séparation adaptatifs ou en définir un modèle *a priori* centré autour de la meilleure estimation courante. La nouvelle estimation combinerait ainsi l'utilisation d'un filtre particulaire et cette loi *a priori*.
- Par ailleurs, le modèle utilisé avec les deux scores permet de présenter une hiérarchie entre les trajectoires correctes et les autres estimations de trajectoire. Ces scores ne définissent pas de relation d'ordre directe entre chaque trajectoire en les utilisant de façon conjointe. Un algorithme d'apprentissage, telle qu'une régression linéaire entre les deux scores et les performances réelles, permettrait de pondérer l'importance de chacun des scores et définirait une relation d'ordre. Toutefois, l'avantage de cette approche comme étant non-supervisée serait perdu suite à l'utilisation d'un modèle statistique.

4.2 Couplage de la fusion paramétrique des mesures RSS et IMU avec un filtre particulaire

Dans ce chapitre, nous présentons une heuristique couplant les méthodes du chapitre 2 et du filtre particulaire. En effet, l'optimisation du chapitre 2 à l'aide des mesures RSS et IMU permet d'obtenir une trajectoire inertielle corrigée, que l'on appellera trajectoire WIB (*wifi batch*). En effet, la trajectoire inertielle dérive avec le temps car elle présente un biais sur le modèle de pas ainsi que sur le cap de la trajectoire. L'approche proposée consiste à appliquer un filtre particulaire sur cette trajectoire WIB et non sur la trajectoire inertielle. Il s'agit bien d'une estimation en temps différé, puisque la trajectoire WIB n'est pas disponible en temps réel. En utilisant la trajectoire WIB, ainsi qu'un filtre particulaire, on définit alors une nouvelle estimation appelée trajectoire WIB + PF.

4.2.1 Expression de la trajectoire WIB

Dans un premier temps, on cherche à exprimer la trajectoire WIB sur laquelle sera appliqué le filtre particulaire. Dans le chapitre 2, on a défini un modèle paramétrique, permettant une estimation des paramètres $\hat{\theta}^t = (\hat{\epsilon}_1, \hat{\beta}_1, \dots, \hat{\epsilon}_{n_r}, \hat{\beta}_{n_r})$ associés à différents instants de rupture k_1, \dots, k_{n_r} . Pour un instant k , on définit une nouvelle longueur de pas l_k^{wib} :

$$l_k^{wib} = \prod_{k_i < k} (1 + \hat{\epsilon}_{k_i}) l_k \quad (4.15)$$

et un nouvel angle de pas α_k^{wib} :

$$\alpha_k^{wib} = \alpha_k + \sum_{k_i < k} \hat{\alpha}_{k_i} \quad (4.16)$$

qui cumulent les rotations et changements d'échelles qui ont été appliqués avant l'instant k . La trajectoire WIB est alors définie par l'ensemble des couples $(l_{1:K}^{wib}, \epsilon_{1:K}^{wib})$. Utiliser la trajectoire WIB nécessite également de modifier la variance d'état à l'instant initial. En effet, dans le cadre de l'estimation en temps réel, l'incertitude sur le cap initial est totale, donc la variance initiale σ_{β_0} sera très élevée. Pour la trajectoire WIB, ce cap initial a été estimé, on utilise donc une variance plus faible $\sigma_{\beta_0}^{wib}$ telle que $\sigma_{\beta_0}^{wib} < \sigma_{\beta_0}$. De même, on considérera une variance d'état différente associée au paramètre d'échelle $\sigma_{\epsilon_0}^{wib}$ plus petite que celle utilisée pour le temps réel.

4.2.2 Algorithme WIB + PF

L'application de l'approche WIB+PF peut améliorer globalement l'estimation en temps réel, mais reste très sensible à l'estimation de la trajectoire WIB. En effet, il est possible qu'un mauvais paramètre d'échelle soit estimé. Le modèle paramétrique du chapitre 2 fait intervenir un problème d'optimisation présentant de nombreux minima locaux. La convergence vers un minimum global de tous les paramètres n'est pas systématiquement garantie. Une grande confiance accordée à ce paramètre pourra alors détériorer l'estimation finale comme illustré dans la figure 4.6. Cette figure représente une illustration de la loi du paramètre d'échelle *a priori*, donc la loi utilisée par le PF en temps réel, la loi *a posteriori* de ce paramètre, donc après l'optimisation WIB ainsi que la valeur réelle. Dans la figure du haut, la distribution *a posteriori* se trouve bien au voisinage de la valeur réelle. L'estimation finale sera améliorée en utilisant la distribution *a posteriori*, il s'agit donc

d'un cas favorable à l'utilisation de la trajectoire WIB. En bas, le cas est bien défavorable, puisque la valeur de recherche se trouve éloignée de la distribution *a posteriori*. La distribution *a priori* choisie en temps réel est alors plus intéressante.

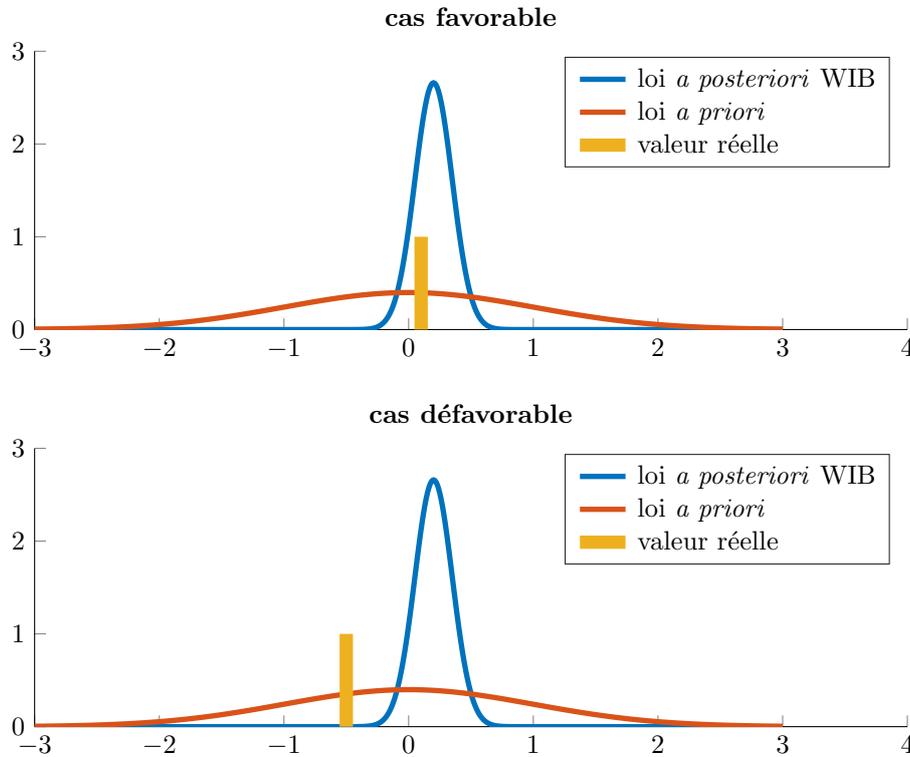


FIGURE 4.6 – Illustration des différents cas possibles pour l'ajustement du modèle d'état

Par ailleurs, le paramètre d'échelle, de même que le cap de la trajectoire ne sont pas des variables connues. Il faut savoir déterminer de façon non-supervisée si l'on se trouve dans un cas favorable à l'utilisation de la trajectoire WIB ou non. Pour répondre à cette problématique, on emploiera deux méthodes concurrentes : une approche PF telle qu'elle a été appliquée pour l'estimation en temps réel, et l'approche WIB+PF telle qu'elle a été présentée. Ceci conduira à deux estimations de trajectoires concurrentes qui seront alors comparées en appliquant le score RSS défini par l'équation 4.11. L'estimation finale sera donnée par la meilleure trajectoire au sens du score RSS. La figure 4.7 présente un schéma de cette méthodologie qui compare les deux approches concurrentes. L'estimation finale sera alors noté $\hat{x}_{1:K}^{wpf}$.

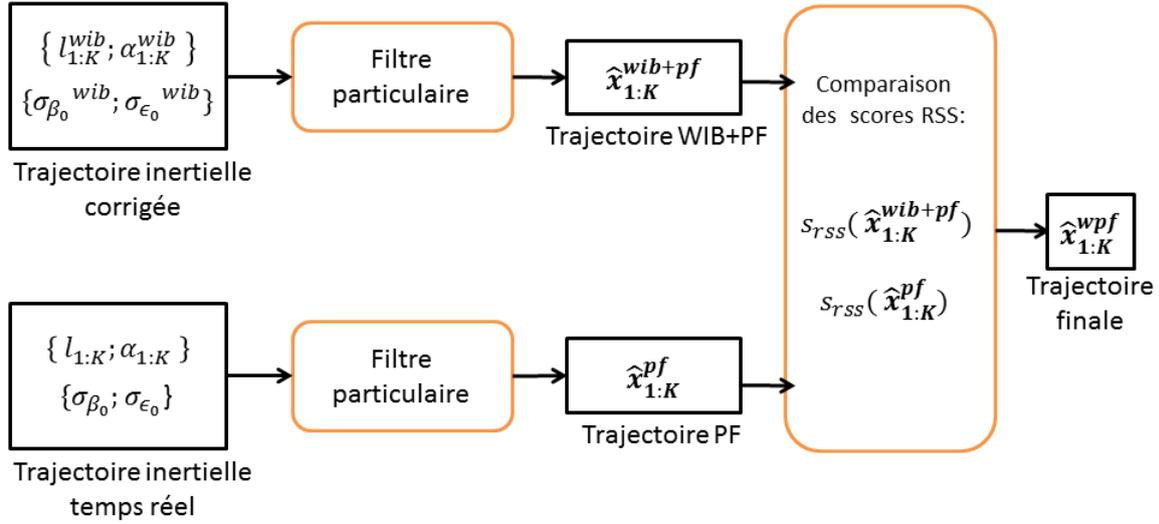


FIGURE 4.7 – Schéma représentant l'algorithme de couplage entre la trajectoire WIB et le filtre particulaire

Le détail de cette méthodologie est décrit dans l'algorithme 10. Ici, le score de continuité n'a pas été appliqué. En effet, il était nécessaire de pouvoir désigner une meilleure estimation de trajectoire entre l'estimation PF et l'estimation WIB+PF. Dans la section 4.1.2.4 l'utilisation conjointe des deux scores n'a pas permis de définir une relation d'ordre sur les estimations de trajectoires. Enfin, le score RSS permet d'évaluer la qualité globale de la trajectoire, il permet ici d'obtenir de meilleures performances que le score de continuité.

Algorithme 10 : Algorithme d'ajustement du modèle d'état

Données : $z_{mes}^{1:K}, l_{1:K}, \alpha_{1:K}$

Résultat : $\hat{x}_{1:K}^{wipf}$

- 1 $\hat{x}_{1:K}^{pf} \sim p_{est}$;
 - 2 $s_{rss}^{pf} = s_{rss}(\hat{x}_{1:K}^{pf})$;
 - 3 Appliquer l'optimisation par morceaux du chapitre 2 et déterminer :
 $\hat{\theta}^t = (\hat{\epsilon}_1, \hat{\beta}_1, \dots, \hat{\epsilon}_{n_r}, \hat{\beta}_{n_r})$;
 - 4 Calculer $l_{1:K}^{wib}, \alpha_{1:K}^{wib}$ selon les équations 4.15 et 4.16 ;
 - 5 $\hat{x}_{1:K}^{wib,pf} \sim p_{wib}$;
 - 6 $s_{rss}^{wib,pf} = s_{rss}(\hat{x}_{1:K}^{wib,pf})$;
 - 7 **si** $s_{rss}^{pf} > s_{rss}^{wib,pf}$ **alors**
 - 8 $\hat{x}_{1:K}^{wipf} = \hat{x}_{1:K}^{pf}$;
 - 9 **sinon**
 - 10 $\hat{x}_{1:K}^{wipf} = \hat{x}_{1:K}^{wib,pf}$;
-

4.2.3 Résultats

4.2.3.1 Application à un cas illustratif

Trajectoire de référence

La trajectoire de référence est une trajectoire de 215m dans le site 1 qui est un bâtiment de bureaux pour lequel le plan a été modifié, comme décrit dans la partie 5.1.2. Le nombre de balises wifi est de 26, et celles-ci sont réparties sur l'ensemble de la trajectoire, ce qui garantit une couverture totale de la trajectoire parcourue. La trajectoire de référence consiste en une trajectoire fermée décrite dans la figure 4.8. Durant l'acquisition de la trajectoire, l'utilisateur a effectué un changement de position du téléphone.

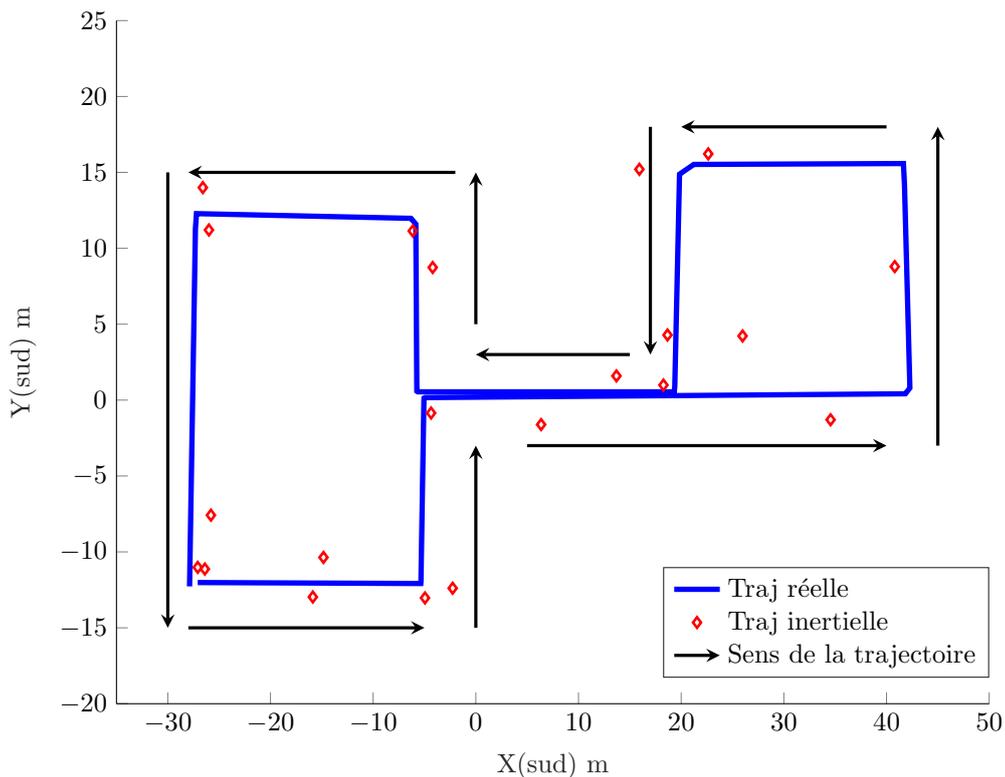


FIGURE 4.8 – Parcours de la trajectoire de référence

Application de la fusion RSS et IMU

Dans un premier temps, on illustre l'estimation de la trajectoire WIB, comme il a été décrit dans le chapitre 2. Le choix de la distance de séparation $d_{sep} = 300m$ étant supérieure à la distance totale de la trajectoire, il n'y aura pas de point de rupture supplémentaire mais le changement de position implique bien un point de rupture.

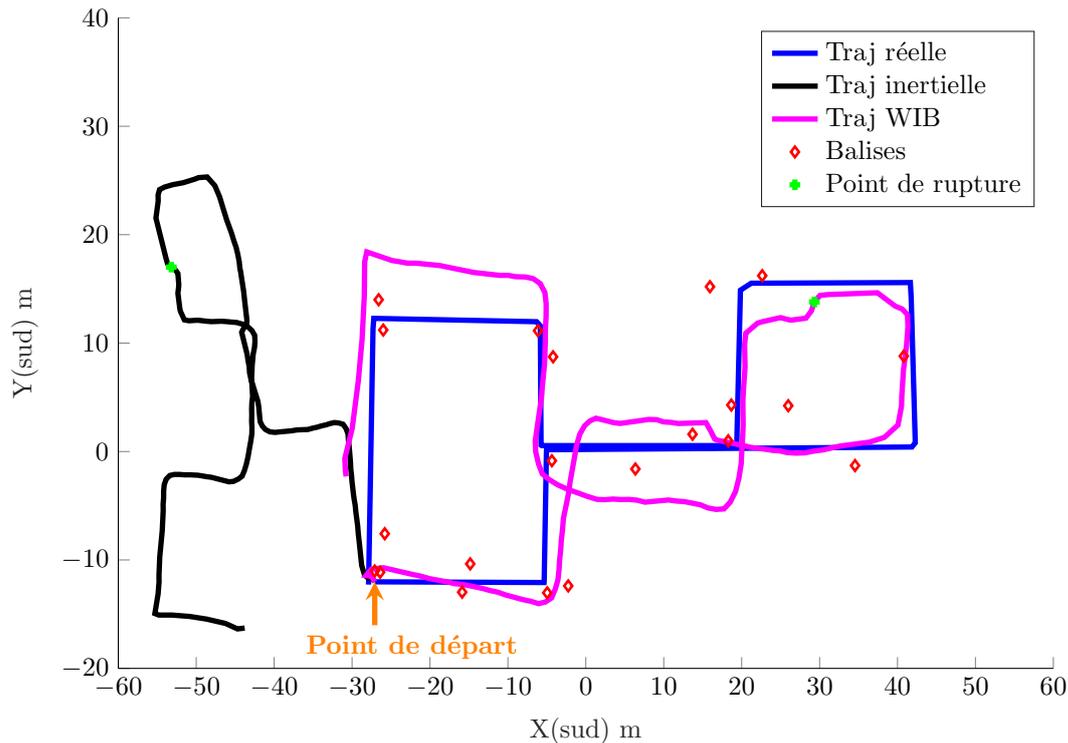


FIGURE 4.9 – Représentation de la calibration de la trajectoire inertielle

Différentes estimations de trajectoires sont représentées dans la figure 4.9.

- La trajectoire noire correspond à la trajectoire inertielle estimée en temps réel. Le cap initial présente une erreur d'environ 90° et le facteur d'échelle global est également erroné.
- La trajectoire rose est la trajectoire WIB. Elle correspond à l'optimisation via les mesures RSS en considérant une unique rupture sur l'ensemble de la trajectoire. On observe que le cap initial a bien été corrigé par l'estimation WIB. Concernant le paramètre d'échelle, il a également été globalement corrigé.

Application du filtre particulaire

D'après la figure 4.10, la trajectoire estimée en temps réel présente une estimation très mauvaise dans les zones encadrées en orange. En effet, en observant la trajectoire inertielle de la figure 4.9, on remarque bien que cette dernière est très différente de la trajectoire réelle, et notamment que le modèle de pas n'est pas bon. La longueur totale de la trajectoire inertielle est de 160m contre 215m pour la trajectoire de référence et 225m pour la trajectoire WIB. Une estimation basée sur la trajectoire inertielle aura tendance à sous-estimer les distances parcourues et les bons couloirs ne seront pas explorés, certaines particules tournant trop tôt. Cette figure illustre l'impact que peut avoir une trajectoire inertielle de mauvaise qualité sur l'estimation en temps réel.

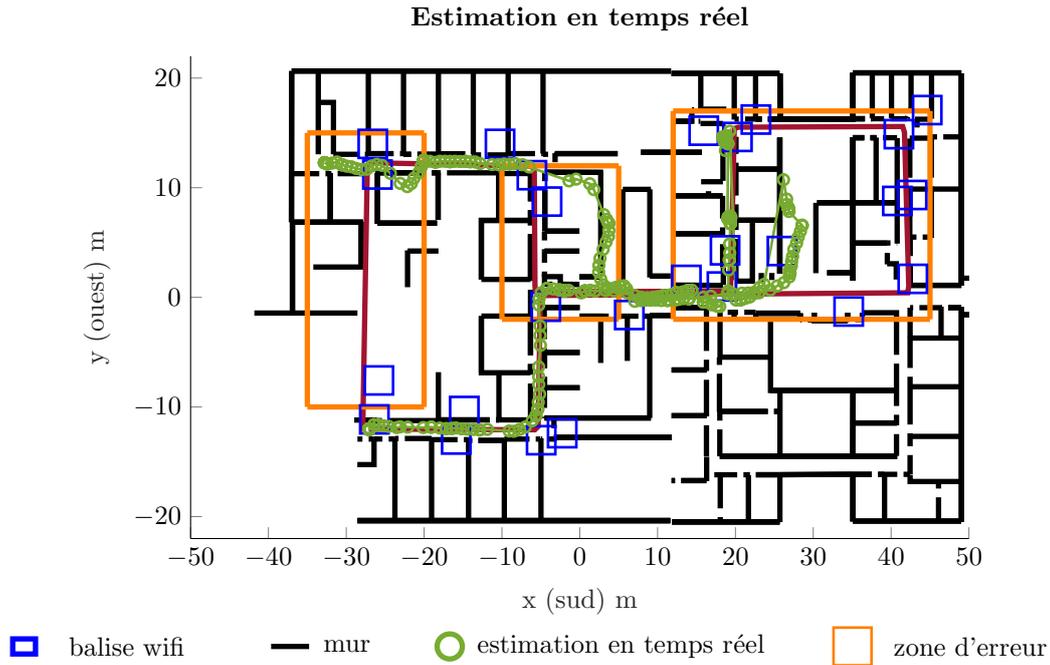


FIGURE 4.10 – Estimation en temps réel de la trajectoire estimée par le filtre particulaire appliqué à la trajectoire inertielle

En appliquant ensuite le filtre sur la trajectoire WIB dans la figure 4.11, l'estimation est grandement améliorée. En effet, les distances parcourues sont sur-estimées pour la trajectoire WIB, ce qui implique certaines zones d'erreur, certaines particules tournant trop tard. Néanmoins, les bons couloirs auront été systématiquement explorés par ce filtre. Dans cet exemple, le gain impliqué par l'ajustement du modèle est très important en terme de rmse, environ 10m pour l'estimation en temps réel, contre 3.6m pour la trajectoire WIB+PF, mais également de %d5, qui passe de 43% à 83%. Par ailleurs, ceci est logiquement validé par la valeur du score RSS, bien plus importante pour la trajectoire WIB + PF. Pour cet exemple, on a utilisé les valeurs suivantes, $\sigma_{\beta_0}^{wib} = 10^\circ$ et $\sigma_{\epsilon_0}^{wib} = 0.05$.

Trajectoire	rmse (m)	%d5 (%)	S_{RSS}
Estimation temps réel	9.96	43.23	-1.25×10^5
Trajectoire WIB + PF	3.59	83.96	-4.98×10^4

TABLEAU 4.4 – Évaluation de l'ajustement du modèle d'état sur un exemple applicatif

4.2.3.2 Résultats sur la base de données

Cette approche est alors éprouvée sur un nombre de 130 expériences, présentées dans la partie 5.1.2. On s'intéresse également au pourcentage de trajectoires pour lesquelles il a été décidé, de façon non-supervisée, que la trajectoire WIB+PF était meilleure selon le score RSS que la trajectoire PF.

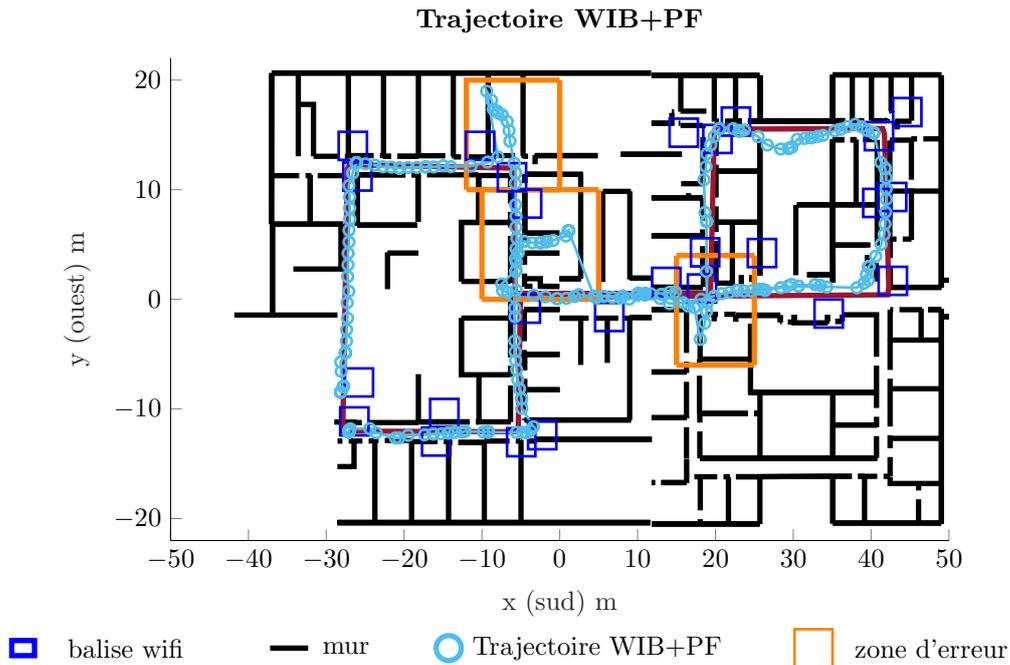


FIGURE 4.11 – Estimation en temps réel de la trajectoire estimée par le filtre particulaire appliqué à la trajectoire WIB

Rupture artificielle	% ratio WIB+PF	Moyenne %d5	Moyenne %d10	Moyenne rmse(m)
-	53.1	83.91	95.44	4.40
300m	44.6	84.88	95.53	4.63
75m	72.3	84.50	94.86	4.81
Estimation temps réel		83.29	94.12	5.81

TABEAU 4.5 – Résultats de l'estimation en temps différé selon le nombre d'itérations et le type de rupture artificielle sur un ensemble de 130 expériences

Les meilleurs résultats ont bien été obtenus en considérant une rupture artificielle chaque 300m. Il n'est pas nécessaire d'utiliser une distance de séparation trop faible. En effet, les erreurs sur la trajectoire inertielle qui sont corrigées sont relativement grossières, comme dans le cas de l'exemple illustratif. Le ratio d'application de l'ajustement du modèle d'état est donc relativement variable. Si la trajectoire inertielle est de très bonne qualité, il n'est pas nécessaire d'ajuster le modèle d'état. Pour ces trajectoires, l'estimation PF, ainsi que l'estimation WIB+PF seront très proches, et la discrimination entre ces deux trajectoires n'impactera pas sur les résultats. Il est intéressant d'avoir un ratio d'ajustement faible, tout en garantissant une amélioration globale des performances. Le gain obtenu par l'ajustement est particulièrement observable sur la moyenne de rmse, 5.81m en temps réel, contre 4.40m en appliquant notre approche.

4.2.4 Synthèse

Dans le chapitre 2, on a proposé une méthode permettant de corriger la trajectoire inertielle à partir des mesures RSS. En effet, cette trajectoire inertielle présente un biais sur la longueur de pas, ainsi que sur le cap initial. La trajectoire corrigée à l'aide des mesures RSS permet de définir la trajectoire WIB. Par ailleurs, dans la partie 3.2.2, une estimation en temps réel était basée sur un filtre particulaire qui utilise la trajectoire inertielle pour définir le modèle d'état. Une combinaison logique en temps différé consistait donc à utiliser la trajectoire corrigée puis de lui appliquer un filtre particulaire.

- En combinant ces deux approches, on peut obtenir une bonne estimation de la trajectoire parcourue malgré une trajectoire inertielle de mauvaise qualité. Dans notre cas, on peut même s'affranchir de la taille de l'individu pour l'estimation de la longueur de chaque pas, puisqu'une homothétie permet de corriger la trajectoire inertielle.
- Cette approche présente tout de même un risque de détérioration de la précision d'estimation par rapport à une approche en temps réel. En effet, une mauvaise estimation de la trajectoire WIB impliquerait une mauvaise estimation de la trajectoire estimée par l'approche WIB+PF. On utilise le score RSS non-supervisé de l'équation 4.11, qui a été développé dans la partie 4.1.2, afin de comparer une trajectoire obtenue par l'approche WIB+PF avec une trajectoire en temps réel estimée par un PF. La trajectoire WPF correspond alors à la meilleure trajectoire au sens du score RSS.

4.3 Méthode de sélection des trajectoires

On se place dans cette partie dans un contexte où l'on cherche à effectuer une sélection des trajectoires, c'est-à-dire qu'il est possible d'éliminer des trajectoires qui auront été mal estimées. Par exemple, dans le cadre du *crowdsourcing*, on suppose qu'un très grand nombre d'utilisateurs viendront alimenter une base de données. Il est possible de s'affranchir des estimations de trajectoires que l'on considère comme étant mauvaises, et de travailler uniquement avec les estimations jugées correctes. En effet, s'il y a un grand nombre d'utilisateurs, alors on a accès à une quantité importante de données. Il devient plus important de privilégier la qualité des données à leur quantité, d'où la nécessité de trier et de conserver uniquement les trajectoires jugées comme correctement estimées. Cette problématique du contrôle de qualité des mesures pour des approches de *crowdsourcing* est discutée ici dans un cadre plus générique LEASE [2011]. On propose un modèle d'apprentissage statistique afin de déterminer une règle de décision et de juger si une nouvelle estimation de trajectoire est correcte ou non.

4.3.1 Présentation de l'approche

Comme représenté dans la figure 4.12, il est nécessaire dans un premier d'avoir une base d'apprentissage. Cette base de données est composée de trajectoires dont on sait si elles ont été correctement estimées ou non, c'est-à-dire si elles correspondent à la trajectoire réelle. Il s'agit d'un apprentissage supervisé. A chacune de ces trajectoires, on associe un ensemble de caractéristiques qui permettront de décrire cette trajectoire comme la durée du parcours par exemple. L'algorithme d'apprentissage cherchera alors à distinguer à partir de cet ensemble de variables quelles trajectoires ont été correctement estimées. Une fois l'apprentissage terminé, cet algorithme sera en mesure de déterminer pour une nouvelle trajectoire, dont on ne connaît que les caractéristiques, si l'estimation est correcte.

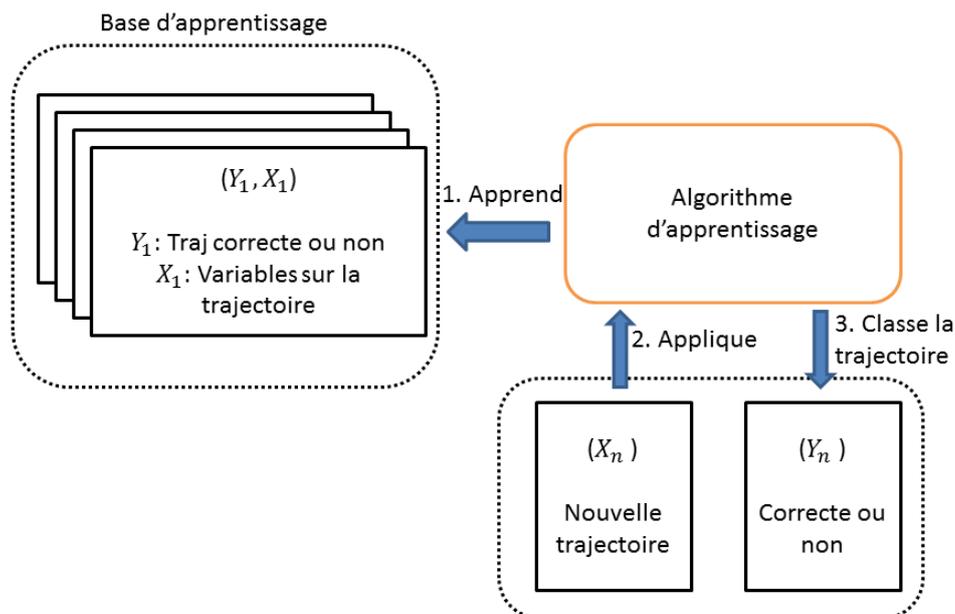


FIGURE 4.12 – Schéma de l'application d'un algorithme d'apprentissage supervisé pour la sélection de trajectoire

La section suivante présentera la formalisation du problème ainsi que les variables d'intérêt. Puis, on décrira les algorithmes d'apprentissage mis en oeuvre. Enfin, une évaluation sera réalisée sur l'ensemble de la base de données.

4.3.2 Formalisation de la problématique

4.3.2.1 Définition d'une trajectoire correcte

On note l'échantillon des différentes estimations de trajectoire que l'on cherche à classer $(\mathbf{x}_{1:K(1)}^1, \dots, \mathbf{x}_{1:K(n)}^n)$ où n correspond au nombre de trajectoires. Ici, chaque trajectoire correspond à une expérience et à un utilisateur potentiellement différent. $K(i)$ est le nombre de pas de la trajectoire i . $\mathbf{x}_{1:K(i)}^i$ est estimé *via* une approche qui couple les différentes heuristiques et qui sera présentée dans la partie 5.2. On définira également la notion de trajectoire correcte selon des critères sur l'erreur rmse et sur le %d5. Une estimation de trajectoire sera correcte au sens du rmse (respectivement %d5) si sa valeur de rmse (respectivement %d5) est inférieure à 5m (respectivement supérieure à 80%). Ces seuils sont fixés de façon arbitraire, selon un choix qui semble raisonnable en fonction des différentes observations sur la base de données. On considérera qu'une estimation de trajectoire sera correcte si elle est simultanément correcte au sens du rmse et du %d5. On peut associer à chaque trajectoire $\mathbf{x}_{1:K(i)}^i$ une valeur y_i telle que $y_i = 1$ si l'estimation est correcte, et $y_i = 0$ sinon :

$$y_i = 1 \iff rmse(\mathbf{x}_{1:K(i)}^i) < 5m \ \& \ \%d5(\mathbf{x}_{1:K(i)}^i) > 80 \quad (4.17)$$

où $rmse()$ (respectivement $\%d5()$) correspond à la fonction qui mesure la rmse (respectivement la valeur de %d5) par rapport à la trajectoire de référence. La base de données contient un ensemble de $n = 190$ expériences différentes, parmi lesquelles 62% sont correctes. Pour chaque trajectoire, on considère un nombre p de caractéristiques, et on a alors $\mathbf{x}_{1:K(i)}^i = (C_1(x_{1:K(i)}^i), \dots, C_p(x_{1:K(i)}^i))$ où $C_j(x_{1:K(i)}^i)$ correspond à la valeur de la j ème variable de la i ème trajectoire.

4.3.2.2 Description des variables

On définit 7 variables permettant de caractériser les trajectoires, celles-ci décrivant 4 aspects différents : la durée de la trajectoire, la correspondance avec l'optimisation WIB, le score de continuité ainsi que le score RSS. Certaines de ces variables sont très fortement corrélées puisqu'elles évaluent les mêmes caractéristiques de la trajectoire. Dans un modèle d'apprentissage, il est préférable de ne pas utiliser de variables trop corrélées. On présentera une routine permettant de sélectionner parmi les différentes variables corrélées celles qui seront les plus pertinentes pour l'apprentissage du modèle. Dans l'annexe B sont représentées les distributions empiriques de chaque variable selon que la trajectoire est correcte ou non.

- **Longueur de la trajectoire**

Les deux premières variables représentent la durée totale de trajectoire notée *temps* en secondes ainsi que le nombre de pas total associé à cette trajectoire *nbPas* qui vaut $K(i)$. En effet, selon la longueur mesurée en seconde ou mesurée en nombre de pas de cette trajectoire, le modèle statistique pourra adapter ses critères de décisions. Pour une trajectoire courte, il est normal de s'attendre à moins d'erreur sur son estimation. Ces deux variables sont donc naturellement très corrélées, les nuances pouvant subsister lorsque l'utilisateur piétine un peu ou lorsqu'il

marque un temps d'arrêt. Enfin, le temps de la trajectoire ne dépend pas de l'algorithme de détection de pas.

- **Fonction de score**

Les scores qui ont été définis dans la partie 4.1.2 par les équations 4.9 et 4.11 sont également mis en oeuvre. Ces scores seront notés s_{cont} et s_{rss} .

- **Distance d'estimation avec WIB**

Une première estimation de la trajectoire est obtenue par $\mathbf{x}_{1:K(i)}^i$ qui utilise la carte. Une seconde estimation est donnée par la trajectoire WIB estimée selon la fusion du chapitre 2, $\mathbf{x}_{1:K(i)}^i(\mathbf{wib})$. Lorsque ces deux estimations sont proches, il y a de grandes chances que ces deux estimations soient simultanément correctes. Cette idée est d'ailleurs également présente dans [FETZER et collab. \[2017\]](#) qui combinent un filtre particulaire tenant compte des murs ainsi qu'un second filtre en temps réel en parallèle qui est uniquement basé sur les mesures wifi. Lorsque l'écart entre ces deux filtres, mesuré par la divergence de Kullback-Leibler, est important alors on considère que le premier filtre particulaire a divergé. Dans notre cas, cette comparaison sera faite à partir de 3 mesures de distance entre la trajectoire WIB et la trajectoire en temps différé. Chacune de ces distances seront très corrélées, il s'agira alors de déterminer la distance la plus pertinente. L'une des distances qui sera utilisée est la distance de Fréchet. On considérera également l'erreur $rmse$ et le %d5 entre les trajectoires $\mathbf{x}_{1:K(i)}^i$ et $\mathbf{x}_{1:K(i)}^i(\mathbf{wib})$ évaluée selon les équations 1.17 et 1.18. Ces variables seront notées $frechet$, $rmse$ et $d5$.

4.3.3 Méthodes d'apprentissage statistique supervisé

Dans la littérature, les différentes méthodes d'apprentissage statistique sont nombreuses [HASTIE et collab. \[2009\]](#), telles que la régression logistique, les *Support Vector Machine*, les méthodes non-paramétriques, comme les méthodes à noyau ou encore les plus-proches-voisins, mais également des méthodes dites de *deep learning* telles que les réseaux de neurones. Pour notre problématique, nous avons choisi de mettre en oeuvre, dans un premier temps, une méthode de classification par arbre que l'on décrira dans la section suivante, introduite dans [STONE et collab. \[1984\]](#), puis un algorithme des forêts aléatoires, introduit dans [BREIMAN \[2001\]](#). Le choix d'un arbre de classification se justifie par la volonté de proposer des seuils pour définir des règles de décision simples. L'utilisation des forêts aléatoires constitue ici une suite logique puisqu'une forêt aléatoire est construite à partir d'un ensemble d'arbres de décision. Enfin, ces deux méthodes ne nécessitent pas de traiter les mesures en amont et sont également très robustes aux données aberrantes.

4.3.3.1 Arbre de décision

Présentation d'un arbre de décision

L'idée de la classification par arbre est de partitionner à l'aide de séparateurs binaires l'espace des variables de façon récursive en fonction de la variable à discriminer. Dans notre cas, l'espace des variables est \mathbb{R}^p qui correspond aux variables des trajectoires, et la variable à discriminer est y qui détermine la pertinence d'une estimation de trajectoire. Ces séparateurs binaires vont permettre de séparer l'espace en branches, qui seront séparées ensuite en sous-branches. On définit les feuilles d'un arbre (F_1, \dots, F_N) comme étant les éléments terminaux de l'arbre, c'est-à-dire les ensembles qui ne seront plus séparés en

deux. N représente ici le nombre de feuilles de l'arbre. Ces feuilles constituent alors une partition de l'espace.

- $F_i \cap F_j = \emptyset$ pour $1 \leq i \neq j \leq N$
- $\bigcup_{i=1}^N F_i = \mathbb{R}^p$

Par la suite, chaque feuille F_i est classée selon deux catégories : les estimations de trajectoires de cette feuille sont considérées correctes $g(F_i) = 1$ ou non-correctes $g(F_i) = 0$, $g()$ étant une fonction étiquette binaire. Pour une nouvelle observation $\mathbf{x}_{1:K(i)}^i \in \mathbb{R}^p$, on cherche la feuille contenant $\mathbf{x}_{1:K(i)}^i$, que l'on note $F(\mathbf{x}_{1:K(i)}^i)$. La nouvelle estimation de la trajectoire sera alors considérée comme correcte (respectivement non-correcte) si $g(F(\mathbf{x}_{1:K(i)}^i)) = 1$ (respectivement si $g(F(\mathbf{x}_{1:K(i)}^i)) = 0$). Un schéma représentant un arbre est donné dans la figure 4.13.

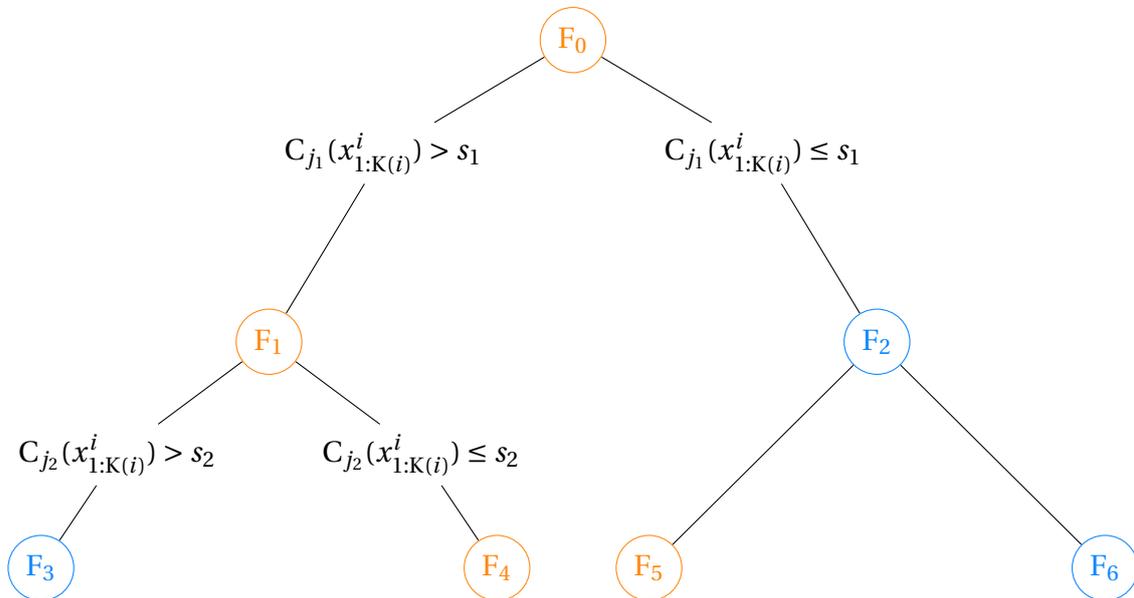


FIGURE 4.13 – Illustration d'un arbre de classification

Construction d'un arbre

La partition de l'espace, qui correspond à la construction de l'arbre, s'effectue récursivement de la façon suivante :

1. On considère l'ensemble de l'espace \mathbb{R}^p contenant toutes les observations $(\mathbf{x}_{1:K(1)}, \dots, \mathbf{x}_{1:K(n)})$ comme étant la racine principale de l'arbre, notée F_0 .
2. On détermine un seuil s_1 sur l'une des variables j , qui permet de définir deux nouvelles feuilles F_1 et F_2 telles que pour un élément $\mathbf{x}_{1:K(i)}^i = (C_1(x_{1:K(i)}^i), \dots, C_p(x_{1:K(i)}^i)) \in F_0$:

- $\begin{cases} \mathbf{x}_{1:K(i)}^i \in F_1 & \text{si } C_j(x_{1:K(i)}^i) > s_1 \\ \mathbf{x}_{1:K(i)}^i \in F_2 & \text{sinon} \end{cases}$
- $g(F_1) = 1 - g(F_2)$

On construit l'arbre de décision en répétant d'abord cette démarche pour la population F_1 que l'on sépare en F_3 et F_4 , et pour F_2 que l'on sépare en F_5 et F_6 comme représenté dans la figure 4.13. Pour cet exemple, il y a donc finalement 4 feuilles (F_3, F_4, F_5, F_6). Les différentes trajectoires disponibles ($\mathbf{x}_{1:K(1)}, \dots, \mathbf{x}_{1:K(n)}$) sont réparties dans chacune de ces feuilles. Pour une feuille F , un vote majoritaire permet de classifier cette feuille, en notant $n_1(F)$ (respectivement $n_0(F)$) le nombre de trajectoires correctes (respectivement non-correctes) dans la feuille F . Le nombre total d'éléments dans la feuille F est noté $n(F)$.

$$g(F) = \operatorname{argmax}_{j \in \{0,1\}} n_j(F) \quad (4.18)$$

Les feuilles oranges (respectivement bleues) correspondant aux feuilles F telles que $g(F) = 0$ (respectivement $g(F) = 1$). Pour la suite, on notera \mathcal{A} un arbre de décision, et pour une nouvelle observation $\mathbf{x}_{1:K(i)}$, on notera $h_{\mathcal{A}}(\mathbf{x}_{1:K(i)})$ la prédiction associée selon cet arbre.

Critère d'impureté

L'objet d'un arbre de classification est de proposer une partition de l'espace qui permettra de construire des classes homogènes. On définit un critère d'impureté qui permet de mesurer l'homogénéité d'une classe tel que l'indice de Gini que l'on définit pour une feuille F :

$$i(F) = \frac{n_1(F)n_0(F)}{n(F)^2} \quad (4.19)$$

$i(F)$ est maximisé quand $n_1(F) = n_0(F)$. Plus l'indice de Gini est faible, et plus la feuille F est homogène. La séparation recherchée pour chaque feuille est celle qui permettra de maximiser une perte d'impureté que l'on définira alors pour une séparation de la branche F en deux feuilles F_1 et F_2 :

$$\Delta i(F) = - \sum_{j \in \{0,1\}} \left(\frac{n_j(F)}{n(F)} \right)^2 + \frac{n(F_1)}{n(F)} \sum_{j \in \{0,1\}} \left(\frac{n_j(F_1)}{n(F_1)} \right)^2 + \frac{n(F_2)}{n(F)} \sum_{j \in \{0,1\}} \left(\frac{n_j(F_2)}{n(F_2)} \right)^2 \quad (4.20)$$

Différents critères permettent d'arrêter la construction de l'arbre à un certain niveau tels que la profondeur maximale de l'arbre, ou encore la taille minimale d'une feuille $n(F)$ autorisée. En effet, construire un arbre trop profond impliquera un problème de sur-apprentissage, c'est-à-dire que l'arbre saura parfaitement classifier les données utilisées pour la construction de l'arbre, mais très mal classifier de nouvelles données indépendantes.

4.3.3.2 Forêt aléatoire

Une forêt consiste à agréger un ensemble d'arbres de décision tel que décrit dans la partie 4.3.3.1 afin d'augmenter la robustesse de la prédiction. Lorsque ces arbres sont construits avec une composante aléatoire, on définit alors une forêt aléatoire, le nombre d'arbres sera noté n_a . La composante aléatoire pour chaque arbre \mathcal{A} se situe à deux niveaux :

- Pour chaque arbre \mathcal{A} , un sous-échantillonnage sans remise de $m < n$ estimations de trajectoire parmi l'échantillon ($\mathbf{x}_{1:K(1)}, \dots, \mathbf{x}_{1:K(n)}$) est effectué.
- À chaque noeud à couper de l'arbre, $l < p$ variables sont tirées aléatoirement, et la coupure est déterminée parmi l'une de ces l variables.

Ces deux éléments permettent d'obtenir des arbres aléatoires différents. Pour une nouvelle observation $\mathbf{x}_{1:\mathbf{K}(i)}^i$, la prédiction selon la forêt aléatoire que l'on notera $h_{\mathcal{F}, \mathcal{A}}(\mathbf{x}_{1:\mathbf{K}(i)}^i)$ sera simplement donnée par un vote majoritaire des prédictions de chaque arbre.

$$h_{\mathcal{F}, \mathcal{A}}(\mathbf{x}_{1:\mathbf{K}(i)}^i) = \operatorname{argmax}_{j \in \{0;1\}} \sum_{i=1}^{n_a} \mathbb{1}\{h_{\mathcal{A}_i}(\mathbf{x}_{1:\mathbf{K}(i)}^i) = j\} \quad (4.21)$$

Cette méthode implique une diminution de la variance du prédicteur final comparative-ment à un unique arbre de décision. De plus, il est possible de déterminer une évaluation de l'importance des variables pour la discrimination de la variable y en utilisant des forêts aléatoires. Cette importance est mesurée à partir de la perte d'impureté calculée dans l'équation 4.20. En effet, pour chaque variable j , on peut considérer l'ensemble des n_j arbres qui ont utilisé cette variable pour la séparation de certaines feuilles, et évaluer la perte d'impureté totale impliquée par cette variable pour chaque arbre que l'on notera $\Delta(\mathcal{A}_i, j)$. L'importance d'une variable correspond alors à la moyenne des pertes d'impureté totales sur l'ensemble de ces arbres :

$$\operatorname{Imp}(j) = \frac{1}{n_j} \sum_{i=1}^{n_j} \Delta(\mathcal{A}_i, j) \quad (4.22)$$

4.3.4 Résultats

L'apprentissage a été effectuée à l'aide du logiciel R et du package *RandomForest*. Le nombre d'arbres choisi est de $n_a = 500$. La durée d'apprentissage pour une base de moins de 200 mesures est ici inférieure à 3 secondes.

4.3.4.1 Importance des variables

Dans un premier temps, on calcule l'importance des variables en utilisant l'algorithme des forêt aléatoires. Les importances des variables sont représentées dans la figure 4.22 afin de sélectionner, parmi les variables très corrélées, celles permettant d'obtenir les meilleures performances. Le nombre d'arbres utilisé par les forêts aléatoires pour ce calcul d'importance a été fixé à $n_a = 5000$. En pratique, pour une base de données de taille aussi réduite que la nôtre, il n'est pas nécessaire d'utiliser un très grand nombre d'arbres pour la classification. Ce choix, dans le cas présent, permet uniquement d'assurer une stabilité dans l'évaluation de l'importance des variables. Celle-ci est représentée dans la figure 4.14. On peut noter que les variables évaluant la distance entre la trajectoire WIB et la trajectoire en temps différé présentent la plus grande importance. Parmi ces distances, c'est la variable *rmse* qui sera conservée. De même, la variable du nombre de pas *nbPas* a une importance plus grande que la variable *temps*, cette dernière sera donc retirée du modèle. On propose alors un second modèle d'apprentissage après avoir sélectionné les variables *nbPas*, *rmse*, *scont* et *s_{rss}*. L'importance de ces variables est également mesurée selon un protocole identique dans la figure 4.15. Quatre modèles différents seront comparés : un arbre de classification ainsi que des forêts aléatoires sur toutes les variables puis sur les variables sélectionnées.

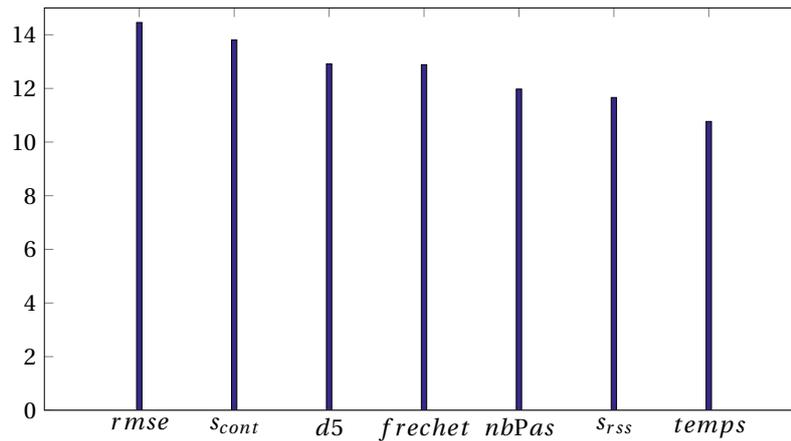


FIGURE 4.14 – Représentation de l'importance des variables pour le modèle global

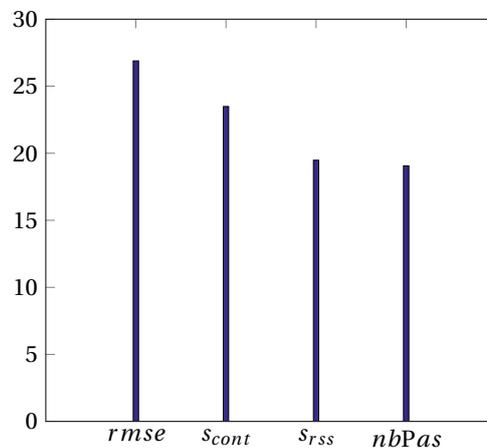


FIGURE 4.15 – Représentation de l'importance des variables pour le modèle réduit

4.3.4.2 Protocole d'évaluation

Pour l'évaluation d'un modèle statistique, il est nécessaire de distinguer les données permettant d'apprendre le modèle, que l'on appellera les données d'apprentissage, des données sur lesquelles sera évalué ce modèle, qui sont les données de validation. Il existe différentes stratégies de séparation des données telles que la méthode de validation croisée. Selon cette méthode, l'ensemble des données, que l'on notera D , est séparé en un nombre k de données sous-échantillonnées (D_1, \dots, D_k) de taille similaire. Ensuite, pour chaque sous base D_i de données, le modèle est appris sur l'ensemble des autres bases $(D_1, \dots, D_{i-1}, D_{i+1}, \dots, D_k)$ puis est évalué sur la base D_i . Lorsque k est égal à n et correspond au nombre d'observations, cette stratégie est appelée *leave-one-out*. Une autre stratégie d'échantillonnage aléatoire (*Random sampling*) consiste à séparer la base en deux de façon aléatoire : une base d'apprentissage D_a de taille pn et une base de validation D_v de taille $(1-p)n$. Cette séparation aléatoire peut être effectuée un nombre n_{rs} de fois, il s'agit ensuite d'agréger les différents résultats associés. L'utilisation de l'une ou l'autre de ces stratégies ne présente pas d'avantage significatif. On choisira alors de façon arbitraire d'évaluer notre approche avec la dernière stratégie en fixant $p = 0.7$ et $n_{rs} = 30$.

4.3.4.3 Critère d'évaluation

Afin de quantifier les performances des algorithmes présentés, il faut dans un premier temps définir une erreur qui soit adaptée au cas d'utilisation. En effet, il existe deux types d'erreur, les faux positifs et les faux négatifs. Un faux positif correspond à une trajectoire dont l'estimation n'est pas correcte, mais qui aura été jugée comme correcte selon notre modèle et un faux négatif correspond à une trajectoire dont l'estimation est correcte, mais qui aura été jugée comme incorrecte. Dans notre cas, on a considéré qu'il était possible de s'affranchir des trajectoires de mauvaise qualité. L'erreur commise associée à un faux négatif aura moins d'importance qu'un faux positif. Par exemple, pour une application dans un contexte de *crowdsourcing*, le faux positif, qui est donc une trajectoire incorrecte, pourra détériorer l'analyse avec des informations fausses. En revanche, le faux négatif constituera uniquement une information perdue.

Ceci signifie que les erreurs ne sont pas symétriques. Il est possible d'apprendre différemment les modèles en considérant des poids d'erreur associés à chacune des classes. Il est également possible de modifier le critère de décision pour la prédiction afin de considérer l'asymétrie des erreurs. Dans l'équation 4.21, mais également pour un arbre de classification, on considère un vote majoritaire pour décider si une trajectoire sera considérée comme correcte ou non, c'est-à-dire qu'il existe un seuil $s = 0.5$ sur les votes à partir duquel la trajectoire est classée comme étant correcte ou non. Si on fixe le seuil à $s = 1$, alors aucune trajectoire ne sera considérée comme correcte. Il n'y a donc aucun faux positifs mais beaucoup de faux négatifs. A contrario, si $s = 0$, toutes les trajectoires seront considérées comme correctes, et le nombre de faux positifs sera important, contre aucun faux négatifs. Dans la figure 4.16, on a représenté le taux de faux positifs noté r_{fp} en fonction du taux de faux négatifs r_{fn} qui sont évalués en faisant varier le seuil s entre 0 et 1. La diagonale représente le pire cas possible de classification, c'est-à-dire si la classification est faite de façon totalement aléatoire donc sans aucune information *a priori*. Visuellement, on remarque dans un premier temps que les forêts aléatoires améliorent la classification par rapport à l'utilisation d'un unique arbre. En effet, pour des valeurs de faux positifs équivalentes, les taux de faux négatifs sont moins importants pour les modèles des forêts aléatoires.

On a fixé de façon arbitraire que le taux de faux positifs doit être inférieur à 10%. Ceci signifie que pour 10 estimations de trajectoires classées correctes, au moins 9 doivent l'être réellement. Normalement, le choix de cette valeur doit résoudre un problème d'optimisation faisant intervenir les différents coûts associés à chaque type d'erreur. Un compromis permet de fixer les taux d'erreur acceptables. Dans notre cas, il n'est pas possible d'explicitier ces fonctions de coût. On pourra alors classer les modèles selon ceux qui présentent le taux de faux négatifs le moins élevé associé à un taux de faux positifs de 10%. On a représenté la droite correspondant à la valeur arbitraire fixée à un taux de faux positifs de 10%. On peut alors directement visualiser les valeurs des taux de faux négatifs associées, représentées dans le tableau 4.6. Par ailleurs, comme attendu, il n'y a pas de gain significatif à utiliser l'ensemble des variables comparativement au modèle utilisant les variables sélectionnées, les variables du modèle étant très corrélées. On pourra considérer pour la suite uniquement l'algorithme des forêts aléatoires avec les variables sélectionnées, impliquant un taux de faux négatifs de 44%. Ceci signifie qu'avec cette solution, 44% des bonnes trajectoires ne seront pas considérées et ne seront pas utilisées pour du *crowdsourcing*.

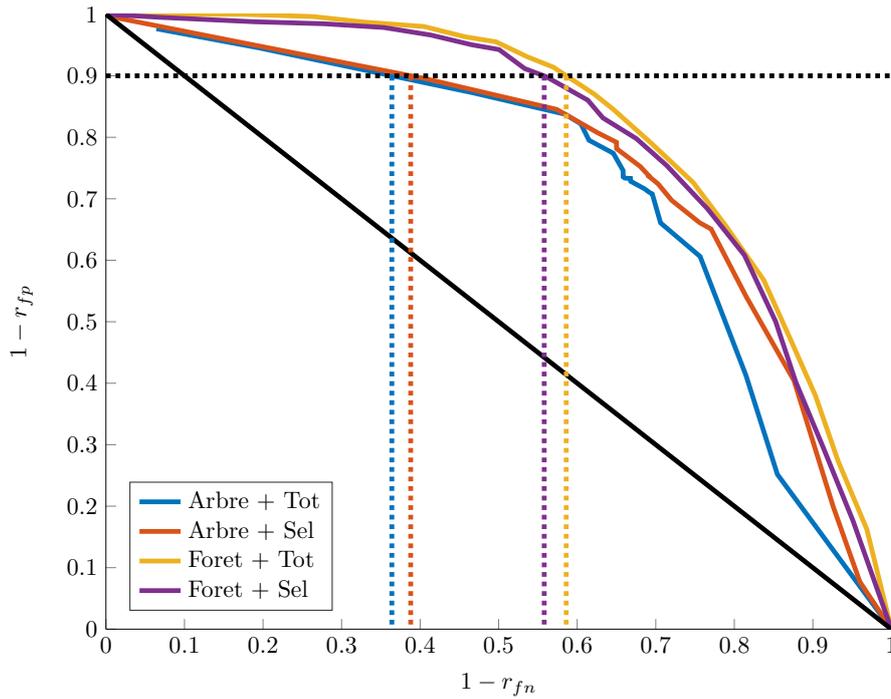


FIGURE 4.16 – Représentation du taux de faux positifs en fonction du taux de faux négatifs. Les méthodes par arbre et par forêt aléatoire sont comparées sur l'ensemble des variables (Tot) ou sur les variables sélectionnées (Sel).

Modèle	$1 - r_{fn}$	r_{fn}	r_{fp}
Arbre	0.364	0.636	0.10
Arbre + sélection	0.388	0.62	0.10
Forêt	0.586	0.414	0.10
Forêt + sélection	0.558	0.442	0.10

TABLEAU 4.6 – Comparaison du taux de faux négatifs pour un taux de faux positifs fixé à 10% pour chacune des méthodes

4.3.5 Synthèse

On propose d'appliquer des forêts aléatoires sur l'ensemble des expériences de la base de données afin de savoir classifier une estimation selon qu'elle soit correcte ou non. En utilisant 4 variables qui sont le nombre de pas de la trajectoire, les scores de continuité ou de mesures RSS, ou encore l'écart entre l'estimation en temps différé avec la carte du bâtiment et l'estimation de la trajectoire WIB, il est possible de classifier les trajectoires avec un taux de faux négatifs de 44% associé à un taux de faux positifs de 10%. Toutefois, bien que satisfaisants, les résultats obtenus sous-estiment le potentiel d'utiliser un modèle d'apprentissage.

- On utilise la distance entre deux estimations différentes de trajectoires en utilisant la variable $rmse$. Cette variable a un caractère très discriminant, et a une grande importance pour la sélection des variables. Néanmoins, dans le cas où l'estimation WIB est mauvaise, l'estimation en temps différé sera classée comme mauvaise alors qu'elle pourrait proposer une très bonne estimation de la trajectoire de l'utilisateur. Une étude de sensibilité sur cette variable permettrait d'améliorer le modèle.

- Enfin, dans notre cas, tous les bâtiments ont été confondus pour le modèle d'apprentissage afin d'avoir un nombre de trajectoires suffisant, mais également pour considérer une large variété de trajectoires. Pour un cas applicatif réel, utiliser un modèle par bâtiment améliorerait grandement les résultats même si construire des bases de test pour chaque bâtiment reste fastidieux.

4.4 Références

- BREIMAN, L. 2001, «Random forests», *Machine learning*, vol. 45, n° 1, p. 5–32. 112
- FETZER, T., F. EBNER, F. DEINZER et M. GRZEGORZEK. 2017, «Recovering from sample impoverishment in context of indoor localisation», dans *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, p. 18–21. 112
- HASTIE, T., R. TIBSHIRANI et J. FRIEDMAN. 2009, *The elements of statistical learning : Data mining, inference and prediction*, 2^e éd., Springer Series in Statistics, Springer, New York, doi :10.1007/978-0-387-84858-7. Pages 460–461. 112
- LEASE, M. 2011, «On quality control and machine learning in crowdsourcing.», *Human Computation*, vol. 11, n° 11. 110
- STONE, C. J., J. FRIEDMAN, L. BREIMAN et R. OLSHEN. 1984, «Classification and regression trees», *Wadsworth International Group*, vol. 8, p. 452–456. 112

Chapitre 5

Résultats généraux

Sommaire

5.1	Protocole d'évaluation des méthodes	122
5.1.1	État de l'art	122
5.1.2	Description des campagnes de mesure	123
5.1.3	Post-traitement des mesures et outil de benchmark	128
5.2	Couplage des méthodes	129
5.2.1	Description du couplage des méthodes	129
5.2.2	Application à une longue trajectoire	130
5.2.3	Application à la base de données	136
5.2.4	Conclusion	141
5.3	Références	142

5.1 Protocole d'évaluation des méthodes

Ce chapitre final comparera les approches qui ont été introduites durant ces travaux de thèse. Dans un premier temps, le protocole expérimental sera décrit.

5.1.1 État de l'art

Les conditions d'expérimentation ont une grande importance sur l'évaluation d'une méthodologie d'estimation, et à différents niveaux que l'on décrira brièvement.

- Le choix du bâtiment : notamment pour les problématiques de *map-matching*, le choix du bâtiment impacte grandement les performances d'estimation. Par exemple, un espace avec des couloirs étroits, tels que les bâtiments de bureaux qui sont très souvent utilisés, facilitera l'estimation. En effet, les déplacements possibles seront plus contraints contrairement à de grands espaces tels que les centres commerciaux, ou encore les halls de gare.
- Le protocole expérimental : pour le même site de campagne, les conditions peuvent être très variables. Par exemple, selon la fréquentation du bâtiment au moment des tests, la trajectoire peut se faire dans un environnement vide, ou doit au contraire s'adapter à des flux importants d'individus. De même, la position du téléphone, ainsi que le comportement de l'utilisateur a son importance. Par exemple, certaines méthodologies souhaitent détecter les changements de position du téléphone et forceront l'utilisateur à avoir un comportement spécifique lors de l'expérimentation.
- La qualité des appareils et des capteurs : la précision des capteurs utilisés influence directement la qualité de l'estimation de la trajectoire.
- La trajectoire de référence : en environnement *indoor*, certaines approches coûteuses pour déterminer la trajectoire de référence sont basées sur l'utilisation de caméra [TRACKING \[2016\]](#) ou [VICON \[2014\]](#). Dans [LE SCORNEC et collab. \[2017\]](#) est également proposé un module d'évaluation *foot-mounted*. Cette trajectoire de référence est souvent estimée en définissant des points de passage à annoter, ce qui constitue potentiellement une source d'erreur.

Différentes études telles que [BRENA et collab. \[2017\]](#) visent à comparer les différentes approches de la littérature. Néanmoins, ils ne sont pas en mesure de se positionner sur les performances numériques associées en raison des différences sur les conditions d'expérience. Par ailleurs, différentes compétitions internationales, [POTORTÌ et collab. \[2017\]](#) ou encore [LYMBERPOULOS et collab. \[2015\]](#), sont organisées à l'occasion de conférences centrées sur la localisation *indoor*. Il devient possible de confronter directement les différents algorithmes sur des données identiques dans des conditions similaires et en utilisant les mêmes critères. Enfin, dans le domaine des méthodes d'apprentissage statistiques, des centaines de bases de données en lignes existent [LEARNING \[2007\]](#). De façon identique, [MONTOLIU et collab. \[2017\]](#) ou encore [VAN HAUTE et collab. \[2015\]](#) proposent des plate-formes similaires contenant différentes bases de données issues de différents capteurs pour la localisation *indoor*. Dans notre cas, nos résultats seront basés uniquement sur des mesures réalisées par nos soins et effectuées au cours de différentes campagnes qui seront décrites dans la partie 5.1.2.

5.1.2 Description des campagnes de mesure

5.1.2.1 Sites des tests

Les différentes expériences mises en oeuvre pour évaluer les algorithmes ont été effectuées à travers 3 campagnes différentes totalisant 190 expériences. Celles-ci ont été effectuées dans 4 différents sites, 3 d'entre eux correspondent à des bâtiments de bureaux sans grand espace large, donc avec des couloirs plutôt étroits et une zone de déplacement limitée. Le site 1 à Grenoble (figure 5.1) se répartit sur 4 étages différents, le site 2 (figure 5.2) correspond à un autre bâtiment de bureaux sur un seul étage et le site 3 (figure 5.3) se trouve également à Grenoble sur un seul étage. Le site 4 (figure 5.4) correspond au *World Trade Center* à Grenoble, qui est un centre de congrès sur 3 étages et qui correspond à un environnement similaire à un centre commercial, donc avec de larges espaces. Enfin, pour certaines expériences réalisées sur le site 1, certains murs de la carte ont été supprimés afin de simuler un bâtiment avec de plus larges espaces, ce bâtiment sera désigné par site 1 mod.

Site 1 : bâtiment de bureaux sur 4 étages

Le site de la figure 5.1 correspond à des bâtiments de bureaux, avec des couloirs étroits d'environ 1.5m de largeur, et des salles de bureaux d'environ 20m². Ce site correspond à un ensemble de trois bâtiments d'une surface d'environ 800m² chacun. Pour le 3ème étage dans lequel la plupart des tests ont été effectués, on a placé 6 balises wifi dans le bâtiment de gauche, ce qui correspond à une densité d'un balise par 130m².

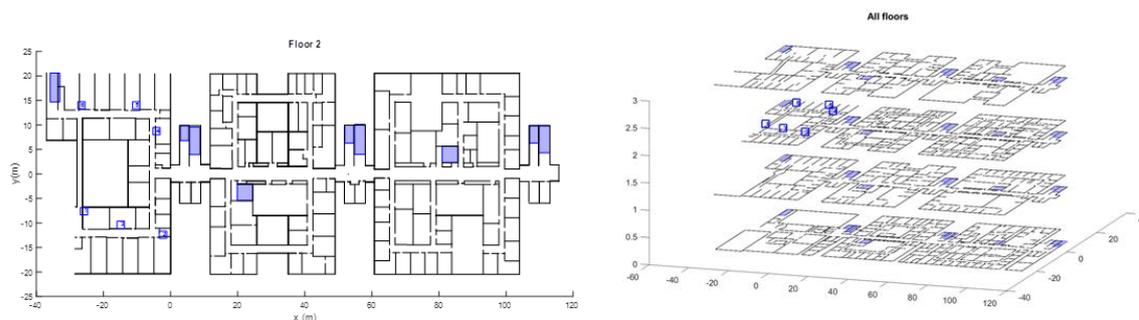


FIGURE 5.1 – Plan des locaux du site 1 à Grenoble. Dans la figure de gauche est représenté le 3e étage qui correspond à l'étage principal où les tests ont été menés. La figure de droite représente l'ensemble des 4 étages.

Site 2 : bâtiment de bureaux avec des couloirs étroits

Le bâtiment de la figure 5.2 correspond à un bâtiment de bureaux, et est constitué de certains espaces en *open-space* mais également de certaines salles de bureaux fermées. La largeur des couloirs dans lesquels les utilisateurs se sont déplacés durant les expérimentations est d'environ 3m. Pour cet étage, un nombre de 20 balises (bluetooth et wifi) ont été disposées afin d'obtenir une couverture totale de la trajectoire parcourue ce qui correspond à une densité de balise d'environ 1 balise par 75m². Pour cette carte, il n'y a pas eu de transition entre les différents étages.

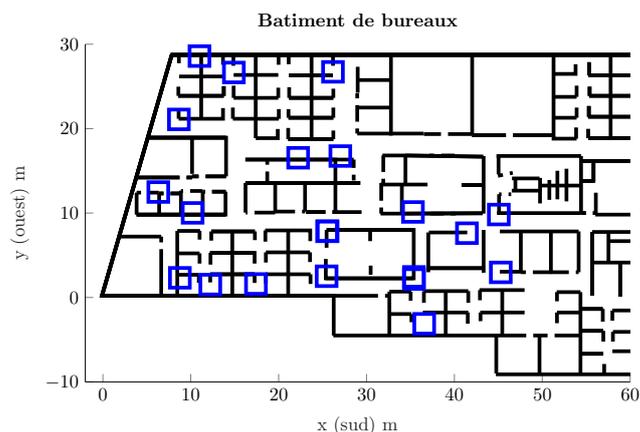


FIGURE 5.2 – Plan des locaux d'un bâtiment de bureaux

Site 3 : bâtiment de bureaux avec des couloirs larges

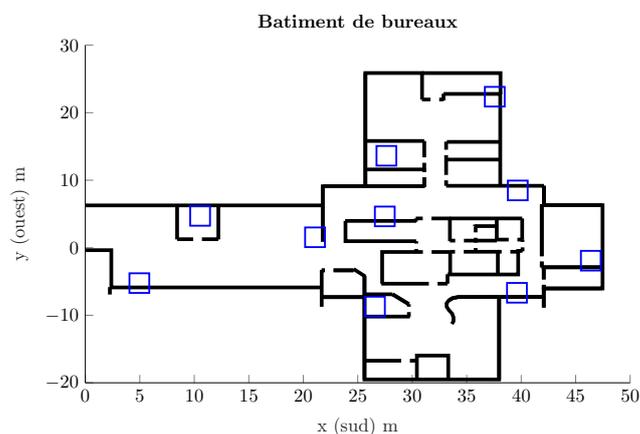


FIGURE 5.3 – Plan des locaux du site 3 à Grenoble

La figure 5.3 correspond aux locaux du site 3 à Grenoble. Pour cette partie du bâtiment, les couloirs sont bien plus larges puisqu'ils peuvent atteindre plus de 10m de largeur à certains endroits. Ici, 10 balises wifi uniquement sont réparties sur environ 1200m². Les balises sont réparties uniformément de telle sorte que chaque expérience sera totalement couverte par les mesures wifi. Comparativement aux cartes précédentes des figures 5.1 et 5.2, les expériences dans ce bâtiment seront plus complexes. Ceci est justifié par les déplacements moins contraints, les couloirs étant plus larges, mais également par la faible densité de balises.

Site 4 : bâtiment sur 3 étages avec des espaces très larges

La figure 5.4 représente le site 4 qui est le centre du *Word Trade Center*. Il est réparti sur 3 étages. Le premier étage, d'une surface d'environ 900m², possède 4 balises. Le second étage, d'une surface d'environ 2500m², possède 6 balises d'émission. Enfin, la surface de test du 3ème étage vaut près de 1400m² et contient 4 balises. Ce site constitue un cas d'estimation très difficile pour plusieurs raisons : les couloirs sont parfois très larges ou correspondent à des espaces libres, et la densité des balises, pouvant atteindre 1 balise par 400m², est très faible.

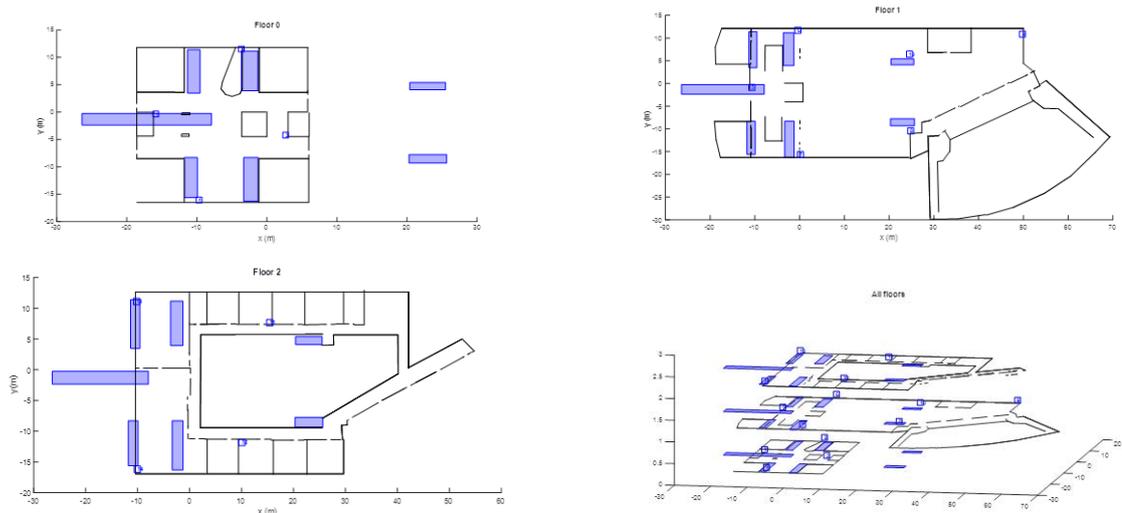


FIGURE 5.4 – Plan du site 4 à Grenoble

5.1.2.2 Description des campagnes

L'ensemble des campagnes a été effectué par 50 utilisateurs différents, et correspondent à 190 trajectoires au total. De plus, ces trajectoires ont été menées au cours de différentes campagnes, impliquant des conditions de mesures très diversifiées, notamment au regard du nombre et de la disposition des balises wifi et bluetooth. La densité des balises pour chacune des campagnes correspond à minima à une couverture plutôt classique, avec une balise par 300m². Sur cet ensemble de trajectoires, la distance moyenne parcourue était d'environ 420m pour une durée moyenne de près de 10 minutes, les détails pour chacun des bâtiments étant décrits dans le tableau 5.1.

Dans notre cas d'étude, 130 expériences sont effectuées sans changement d'étage, et 60 expériences sur plusieurs étages. Les transitions entre ces étages s'effectuent uniquement par des escaliers, des escalators ou encore des ascenseurs. Par ailleurs, 30 des expériences effectuées sur plusieurs étages correspondent au bâtiment du *Word Trade Center*, donc avec de larges couloirs, ce qui implique des trajectoires plus complexes.

Bâtiment	Nombre expériences	Durée moyenne (min)	Distance moyenne (m)	Nombre étages	Nb balises wifi/ble
Site 1	19	10	841	1	6
Site 1	19	10	841	1	6
Site 1 mod	34	13	254	1	35
Site 3	30	10	384	1	10
Site 3	24	3	130	1	21
Site 2	4	3	147	1	20
Site 1	30	13	476	4	20
Site 4	30	10	347	3	14

TABLEAU 5.1 – Description des différentes campagnes de mesures

5.1.2.3 Capteurs d'utilisation et conditions expérimentales

Les différents téléphones qui ont été utilisés pour ces campagnes de mesures sont un *Nexus 4*, un *Nexus 5*, un *Galaxy Note 3* mais également une plate-forme équipée de capteurs de qualité grand public. Par ailleurs, afin de rendre ces expériences plus réalistes, certaines conditions particulières sur la position du téléphone ont été imposées, mais également concernant le comportement de l'utilisateur. Les différentes positions qui ont été testées lors des différentes campagnes de mesure sont : téléphone en position portrait, en position paysage, en position appel téléphonique, dans la poche, dans le sac à dos, dans une poche sur la poitrine ou encore en balancier dans la main. Par ailleurs, différentes situations étaient imposées à l'expérimentateur afin d'approximer le comportement d'un individu en train de marcher. Par l'exemple, l'utilisateur s'arrête pour répondre au téléphone, pour discuter, ou encore pour chercher un élément dans une affiche.

5.1.2.4 Définition de la trajectoire de référence

Comme également décrit dans [DE LA OSA et collab. \[2016\]](#), la trajectoire de référence est définie de façon supervisée. Un protocole d'expérimentation est défini en amont et fixe au préalable la trajectoire que l'utilisateur va parcourir. Cette trajectoire est définie selon un certain nombre de points de référence qui correspondent aux virages de la trajectoire et également aux entrées et sorties des cages d'escalier. Chaque test est effectué à l'aide d'un utilisateur et d'un superviseur. A chaque fois que l'utilisateur se situera au niveau d'un point de référence, le superviseur devra l'indiquer à l'aide d'une application développée par le CEA spécialement pour l'acquisition de données, mais également pour produire un démonstrateur. Le superviseur possède une plate-forme reliée par bluetooth à l'ensemble des autres appareils et effectue alors des annotations à distance simultanément pour l'ensemble des autres téléphones, ceci permettant d'effectuer les campagnes avec le téléphone dans le sac à dos ou dans la poche. L'interface de cette application est représentée dans la figure 5.5. Dans le cercle vert est représenté le numéro du point de référence, et le trajet qui a été préalablement défini est représenté par la ligne verte. Entre chaque point de référence, l'utilisateur se déplace en ligne droite. Par la suite, entre chaque point de référence, une interpolation est effectuée, en considérant une vitesse constante de déplacement, sauf en cas d'arrêt. Cette trajectoire de référence reste perfectible, notamment parce que l'utilisateur n'a pas un déplacement strictement rectiligne, lors des virages par exemple, ou s'il doit esquiver un obstacle comme un objet ou un autre individu sur son trajet. Toutefois, ce décalage entre la trajectoire réelle et cette trajectoire de référence reste

marginal.



FIGURE 5.5 – Interface de l'application permettant l'acquisition des données

5.1.2.5 Calibration des capteurs

Chaque capteur est calibré avant la réalisation d'une expérimentation. L'accéléromètre est calibré une fois uniquement en mesurant l'apesanteur. Le magnétomètre est calibré une fois par campagne en effectuant des mouvements de huit dans toutes les directions, comme il est décrit dans [BONNET et collab. \[2009\]](#). Enfin, le gyroscope est calibré avant chaque expérience en considérant une phase statique, le téléphone étant posé au sol quelques secondes.

5.1.2.6 Acquisition des données

La fréquence maximale d'acquisition des capteurs des téléphones est d'environ 200Hz, mais également pour la plate forme propriétaire. Un fichier est créé à chaque expérimentation contenant l'ensemble des mesures captées par l'appareil. Ce fichier contient tout d'abord les éléments caractéristiques de l'expérience :

- le nom et la taille de la personne, la taille permettant d'estimer la longueur de pas
- la position, la puissance des balises et le type de balise (wifi ou bluetooth)
- la description du nombre d'étages

Par la suite, lorsque l'expérience démarre, des enregistrements de mesures successifs incrémentent le fichier de mesure. On a choisi, pour des raisons de simplification de l'implémentation, de traiter les mesures RSS successivement. Pour les mesures arrivant en même temps, on enregistre un maximum d'une mesure par pas de temps, les mesures étant alors réparties sur les pas de temps suivants jusqu'à ce qu'il n'y ait plus de mesures, ceci pouvant d'ailleurs introduire un léger retard. Par ailleurs, si lors de l'enregistrement de mesures à un instant t , la valeur d'un capteur n'a pas été mise à jour, alors la valeur précédente à l'instant $t - 1$ sera simplement également reportée à l'instant t dans le fichier. On obtient un fichier composé des éléments suivants :

1. le temps
2. les capteurs MEMs calibrés
3. l'annotation d'un changement de position du téléphone

4. l'identifiant de la balise d'émission et la puissance mesurée
5. la position et précision du GPS

5.1.3 Post-traitement des mesures et outil de benchmark

Un outil de post-traitement développé sous le logiciel *Matlab* permet d'appliquer les différents algorithmes pour chacune des expériences. Pour augmenter la vitesse de calcul, un outil de compilation a permis d'obtenir un code C pour l'estimation en temps réel. Un outil de benchmark permet ensuite de traiter cet ensemble de 190 trajectoires. L'ensemble des fichiers est réparti selon un nombre de 3 campagnes qui sont elles-mêmes décomposées en 3 ou 4 cas d'utilisation. Le benchmark répartit les cas d'utilisation sur l'ensemble des coeurs de l'ordinateur afin de les traiter parallèlement. Pour chaque expérience, tous les algorithmes sont testés, c'est-à-dire les algorithmes en temps réel sans les plans du bâtiment, en utilisant le filtre particulière, ou encore en considérant un algorithme en temps différé. Les résultats associés à chaque cas d'utilisation sont reportés dans un tableau. Les métriques utilisées pour caractériser les résultats sont l'erreur rmse, le %d5 ou encore le %d10.

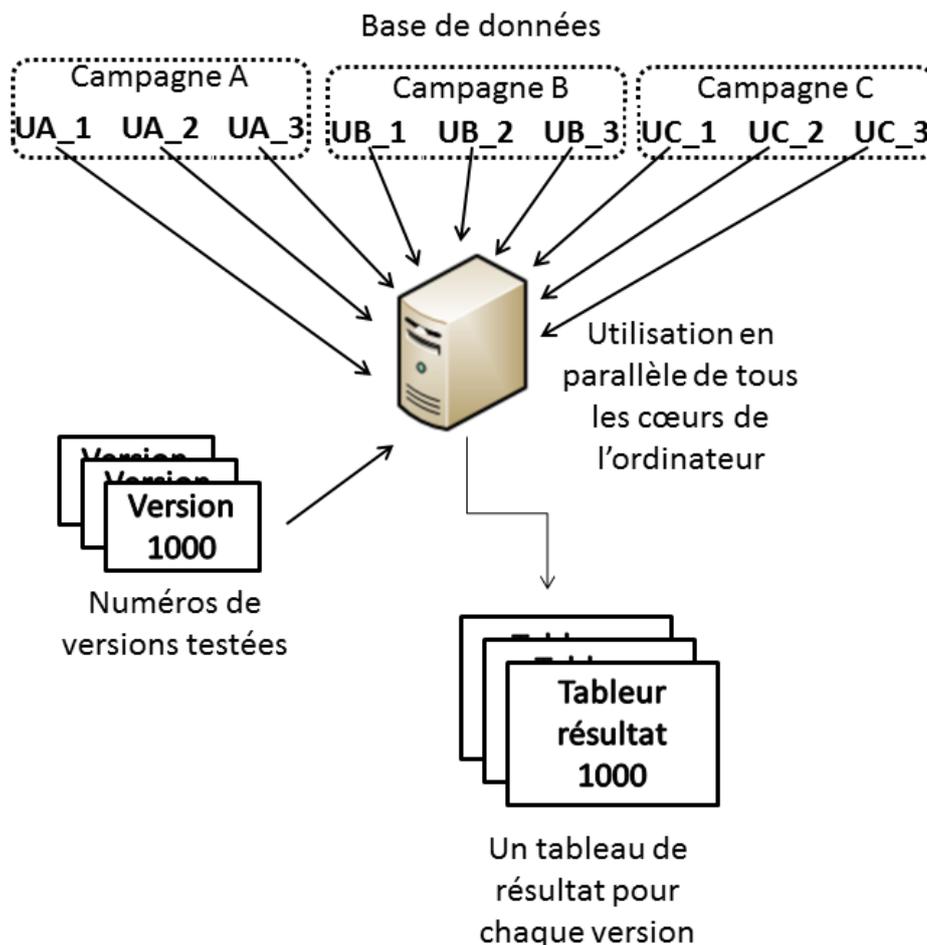


FIGURE 5.6 – Schéma de l'outil du benchmark pour tester les différentes versions compilées

Enfin, un outil de *versionning*, ici *svn*, permet, au-delà de la tenue du code, d'optimiser les paramètres du filtre. En effet, on peut fixer un ensemble de paramètres, puis appliquer

un *commit* est obtenu une version du code, par exemple numérotée 1000. Ensuite, on peut programmer l'outil du benchmark pour qu'il s'active selon une version donnée du code, par exemple la version 1000. Les paramètres qui auront été enregistrés à la version 1000 seront ceux appliqués. Le fichier est alors appelé "NumVersion+Date". Il devient ensuite possible d'enregistrer différents jeux de paramètres à l'aide de versions successives, par exemple, la version 1000 effectuera les estimations avec un nombre de 10 particules, puis la version 1001 avec 1000 particules. Il suffit de comparer les différents fichiers de sortie finaux. Le schéma de la figure 5.6 récapitule l'outil de post-traitement des mesures.

5.2 Couplage des méthodes

5.2.1 Description du couplage des méthodes

Trois approches utilisant le filtre particulaire ont été développées.

- Dans la section 4.2, on a ajusté le modèle d'état en corrigeant la trajectoire inertielle à partir des mesures RSS et on a ainsi défini la trajectoire WPF. Cet ajustement considère également la prise de décision présentée en 4.2.2, et correspondant à la sortie de l'algorithme 10.
- Dans la partie 4.1, une répétition du filtre permettait de définir la trajectoire MCPF (*Monte-Carlo Particle Filter*).
- Enfin, l'algorithme de Viterbi a permis d'estimer le maximum *a posteriori* de l'estimation de la trajectoire, basé sur l'historique des différents *clusters* de l'estimation en temps réel et d'introduire la trajectoire VPF.

Ces algorithmes peuvent également être couplés pour améliorer la précision.

1. Dans un premier temps, on estime la trajectoire WIB. La règle de décision, définie en 4.2.2 à l'aide du score RSS, permet de choisir si l'estimation sera donnée à partir de la trajectoire inertielle ou de la trajectoire WIB.
2. Ensuite, on peut appliquer la répétition du filtre particulaire sur la trajectoire WIB si celle-ci s'est révélée probante au sens du score RSS à l'étape précédente. Sinon le filtre est utilisé sur la trajectoire inertielle initiale.
3. Enfin, on applique l'algorithme de Viterbi aux *clusters* de la trajectoire déterminée à l'étape 2.

Le couplage de ces différentes étapes est schématisé dans la figure 5.7 et sera par la suite désigné par l'approche finale. et la trajectoire sera notée la trajectoire AF. Dans un premier temps, on appliquera cet algorithme à une trajectoire longue. Par la suite, cette approche sera éprouvée sur l'ensemble de la base de données et sera également comparée avec les algorithmes en temps différé déjà définis.

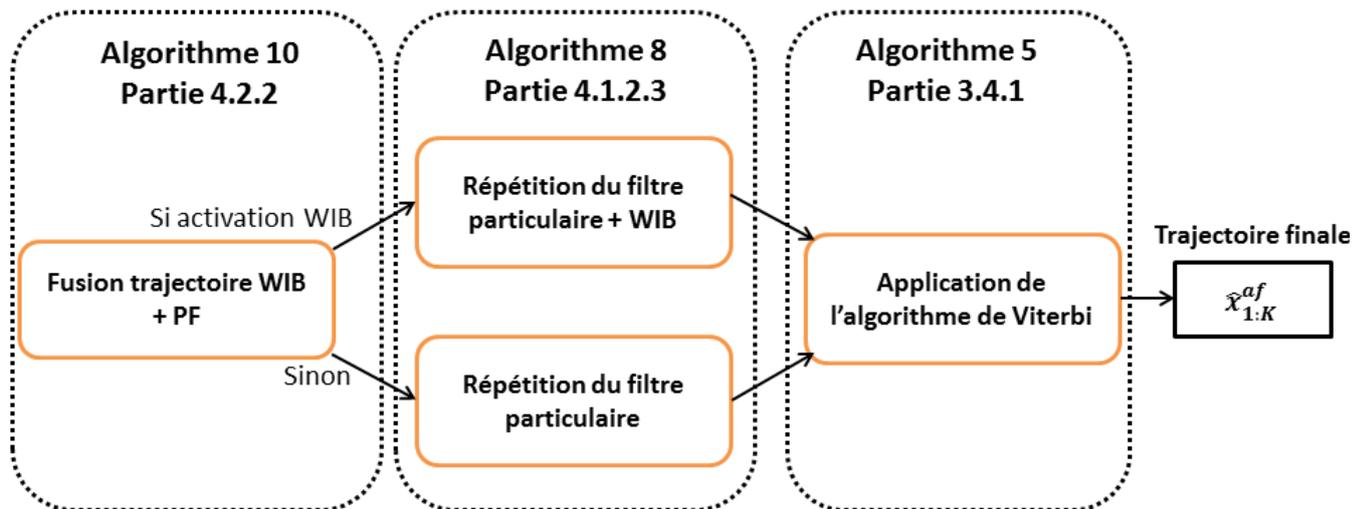


FIGURE 5.7 – Schéma de l'approche de l'estimation final en temps différé

Les notations sont reprises dans le tableau 5.2. On note par ailleurs qu'il aurait également été possible de coupler les algorithmes deux à deux. Ces couplages ne seront pas approfondis durant cette thèse, l'approche finale étant déjà un algorithme de couplage pertinent.

Notation	Étape	Méthode
PF	-	Filtre particulaire
WPF	1	Trajectoire WIB + Filtre particulaire
MCPF	2	Répétition du filtre particulaire
VPF	3	Filtre particulaire + Algorithme de Viterbi
AF	1+2+3	Trajectoire WIB + Répétition du filtre particulaire + Algorithme de Viterbi

TABEAU 5.2 – Notation des différents algorithmes

5.2.2 Application à une longue trajectoire

L'approche finale est appliquée sur un exemple de trajectoire longue qui est similaire à l'expérience présentée dans la figure 4.4. Le nombre de particules n_p sera fixé à 500 pour la suite. Dans un premier temps, on applique un filtre particulaire en temps réel. Ensuite, on mettra en oeuvre l'approche finale tout en détaillant chaque gain obtenu par chacune des étapes comparativement au temps réel.

5.2.2.1 Approche en temps réel

La figure 5.8 représente l'estimation de la trajectoire estimée par un filtre particulaire en temps réel. La valeur de l'erreur rmse est de 5.95m et le ratio de %d5 est de 64.33%. Dans cette figure, lorsque l'erreur est supérieure à 5m, l'estimation en temps réel, représentée normalement en vert, est remplacée par des points rouges.

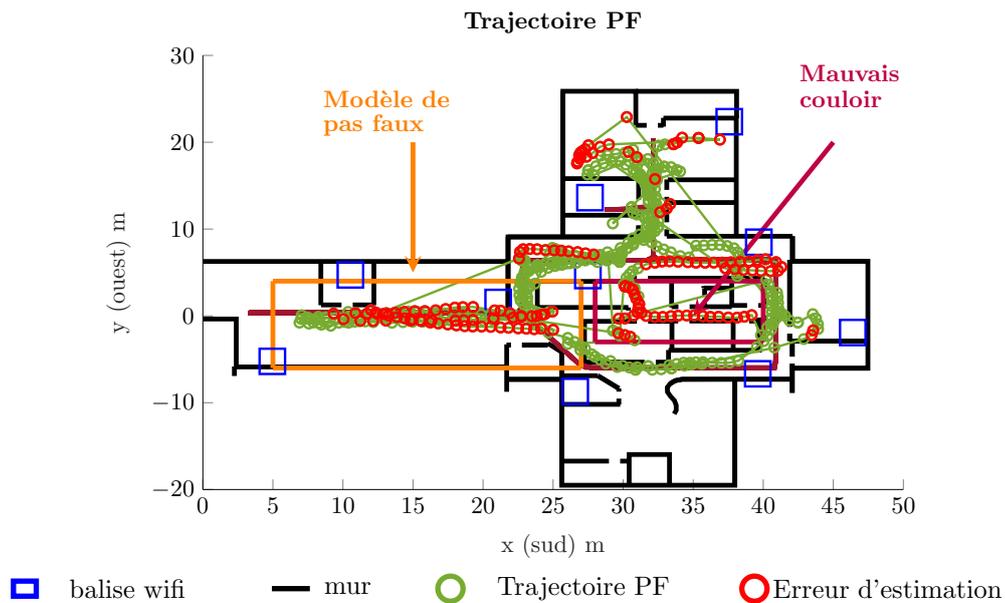


FIGURE 5.8 – Estimation de la trajectoire en temps réel

De plus, dans la figure 5.8, on a représenté l'erreur en mètre à chaque pas entre la trajectoire de référence et l'estimation en temps réel.

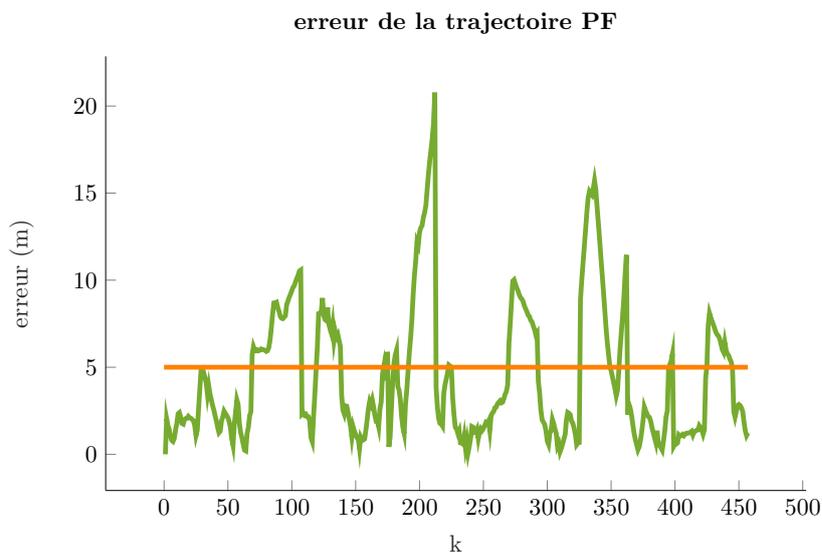


FIGURE 5.9 – Évolution de l'erreur le long de la trajectoire de l'estimation en temps réel

En observant la figure 5.8, on observe que les sources des erreurs en temps réel sont majoritairement dues à certains facteurs inhérents au temps réel :

- La carte du bâtiment possède de larges couloirs. Lorsque les couloirs sont étroits, la trajectoire est suffisamment contrainte pour que les paramètres de biais sur le cap, mais également sur le paramètre d'échelle se fixent autour de la bonne valeur. Dans cet exemple, ça n'est pas le cas. Si l'on observe l'erreur dans le couloir de gauche, dans le cadre orange, on remarque en effet que l'erreur est ici causée par un retard de la trajectoire estimée, sur la trajectoire de référence.

- Une seconde source d'erreur du temps réel concerne le mauvais choix de couloir ou d'hypothèse. En effet, on représente dans le cadre violet une mauvaise hypothèse qui a été suivie. En temps réel, cette mauvaise hypothèse continuera d'être suivie tant qu'il n'y aura pas de contradiction avec les murs ou alors de mise à jour à partir des mesures RSS.
- Enfin, également liée à la source d'erreur précédente, il existe quelques sauts de *clusters*, représentés par les longs segments verts qui correspondent à des changements d'hypothèse.

On applique par la suite, étape par étape, l'approche finale afin d'observer dans quelle mesure une approche en temps différé permet de corriger ces erreurs.

5.2.2.2 Application de l'étape 1 de l'approche finale

On applique alors l'optimisation WIB. Les paramètres choisis sont une distance de séparation de $d_{sep} = 300\text{m}$ associée à un nombre d'itérations $i_{stop} = 800$. Dans la figure 5.10 est représentée la trajectoire WPF. Ici, le score RSS associé à la trajectoire WIB+PF était plus élevé que le score associé à la trajectoire PF seule.

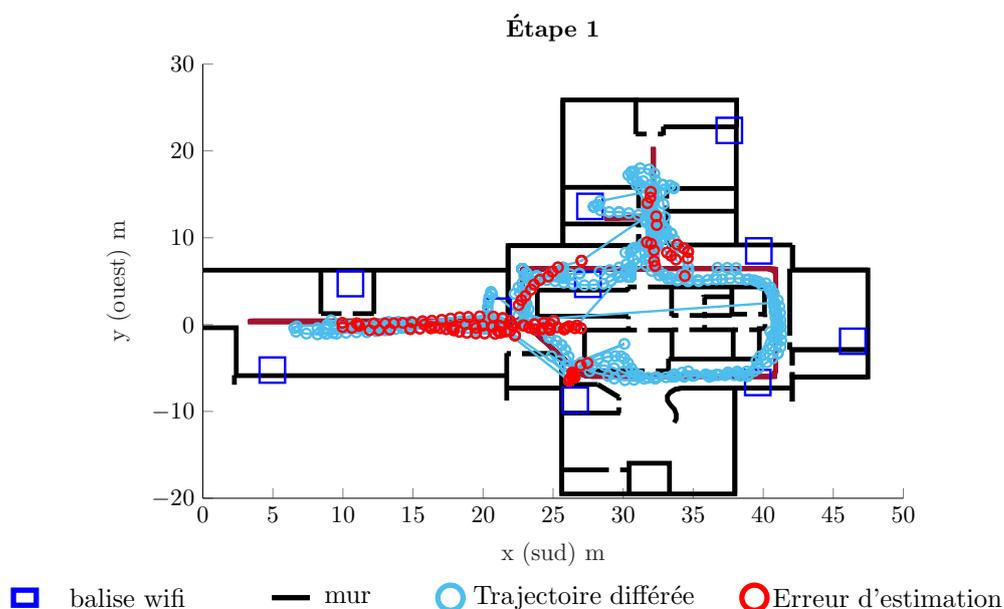


FIGURE 5.10 – Estimation après l'étape 1 de l'approche finale

L'erreur rmse après l'étape 1 est de 5.27m, et le ratio %d5 est de 72.85%. Pour chacune des erreurs, on observe un gain par rapport à l'estimation en temps réel.

- L'utilisation de la trajectoire WIB en amont d'un filtre particulière a permis, comme c'était également le cas dans l'exemple de la figure 4.11, de parcourir les bons couloirs, ce qui justifie le gain de performance.
- Néanmoins, l'ajustement du modèle de pas n'a pas su corriger l'erreur dans le couloir à l'intérieur du rectangle orange dans la figure 5.8. Ceci illustre la rigidité de considérer en amont une distance de séparation fixe, ici de 300m. Par exemple, si deux points de rupture existaient juste à l'entrée, ainsi qu'à la sortie du rectangle orange, l'ajustement du modèle aurait potentiellement permis de correctement estimer la position dans cette zone avec un paramètre d'échelle spécifique.

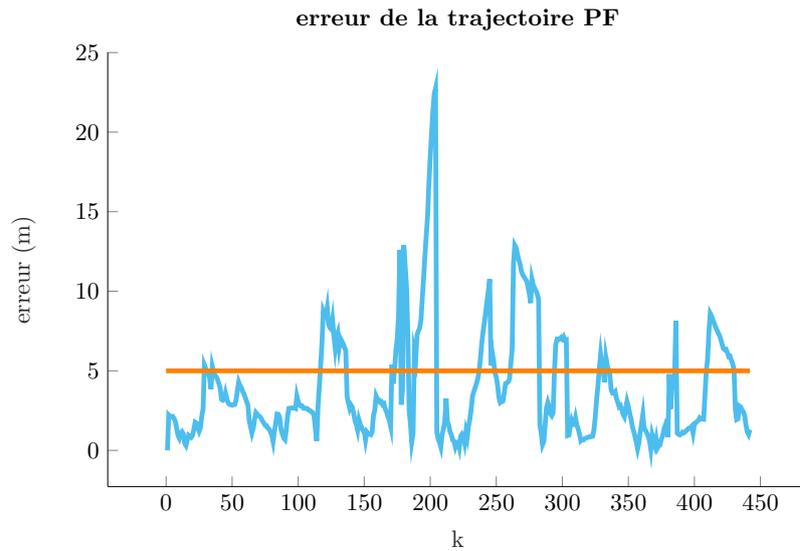


FIGURE 5.11 – Évolution de l’erreur le long de la trajectoire de l’estimation après l’étape 1

5.2.2.3 Application de l’étape 2 de l’approche finale

Ici, la répétition du filtre est appliquée sans considérer de sous-trajectoires. Le nombre de particules reste bien fixé à $n_p = 500$ et le nombre de répétitions du filtre est de $n_{mc} = 30$.

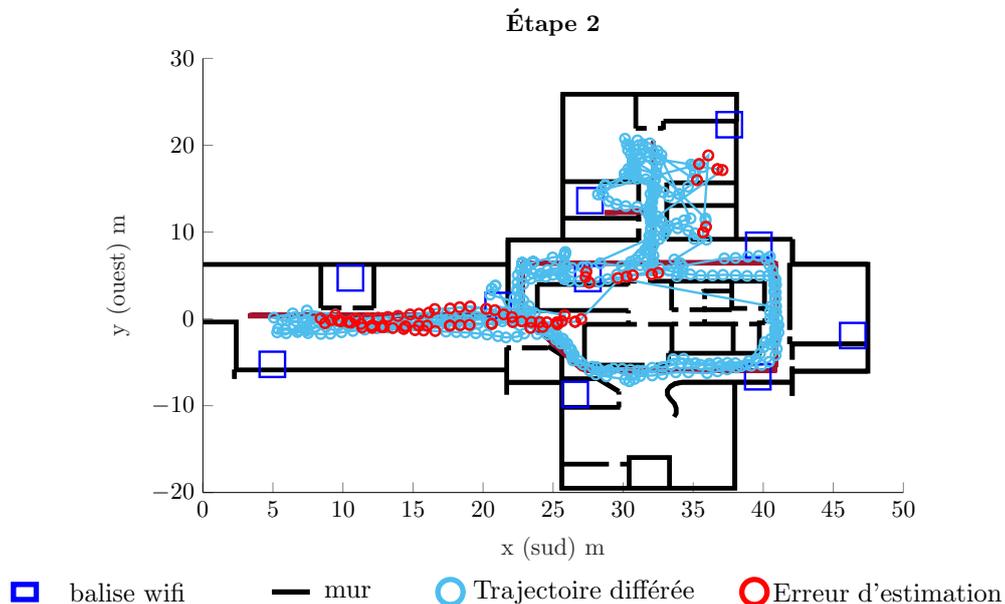


FIGURE 5.12 – Estimation après l’étape 2 de l’approche finale

L’erreur rmse est ici de 3.60m et le ration %d5 est de 82.80%. Si on considère le critère qui a été défini dans la partie 4.3.2.1 sur la sélection des trajectoires, alors cette trajectoire est considérée comme correcte.

On a représenté sur la figure 5.13 la fonction d’erreur associée à l’estimation suite à l’étape 1 afin de mieux visualiser le gain obtenu.

- Le gain d’utiliser une répétition du filtre s’observe directement sur l’écart maximal existant avec la trajectoire entre l’estimation après l’étape 2, qui est de 12m, contre

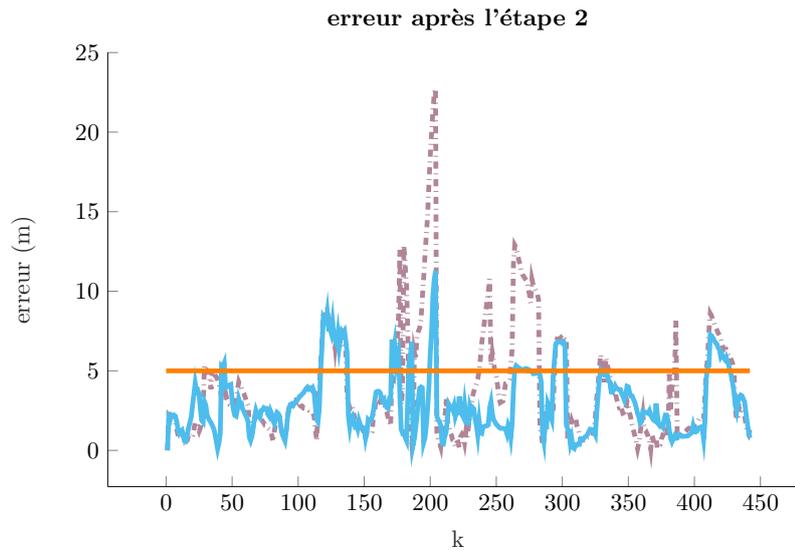


FIGURE 5.13 – Évolution de l'erreur le long de la trajectoire de l'estimation après l'étape 2. En pointillé est représentée l'erreur de l'estimation de l'étape précédente.

20m pour les estimations précédentes. Toutefois, sur un nombre de 30 itérations, seules 2 améliorations ont été acceptées et il subsiste des erreurs.

- L'utilisation d'une répétition du filtre n'a pas non plus été en mesure d'estimer correctement le couloir à gauche. Néanmoins, cette erreur est largement réduite.
- L'utilisation des sous-trajectoires aurait peut-être permis d'estimer correctement cette portion de trajectoire. Mais comme pour l'optimisation, il aurait fallu pouvoir déterminer de façon précise, mais non-supervisée, les instants de rupture pertinents pour la sous-trajectoire.

5.2.2.4 Application de l'étape 3 de l'approche finale

On applique l'algorithme de Viterbi à la meilleure estimation des filtres au sens des deux scores. Le nombre des *clusters* a été fixé à $n_c = 3$. Par rapport à l'étape 2, la moyenne rmse est stable et s'élève à 3.62m mais le ratio de %d5 augmente et vaut 85.74%.

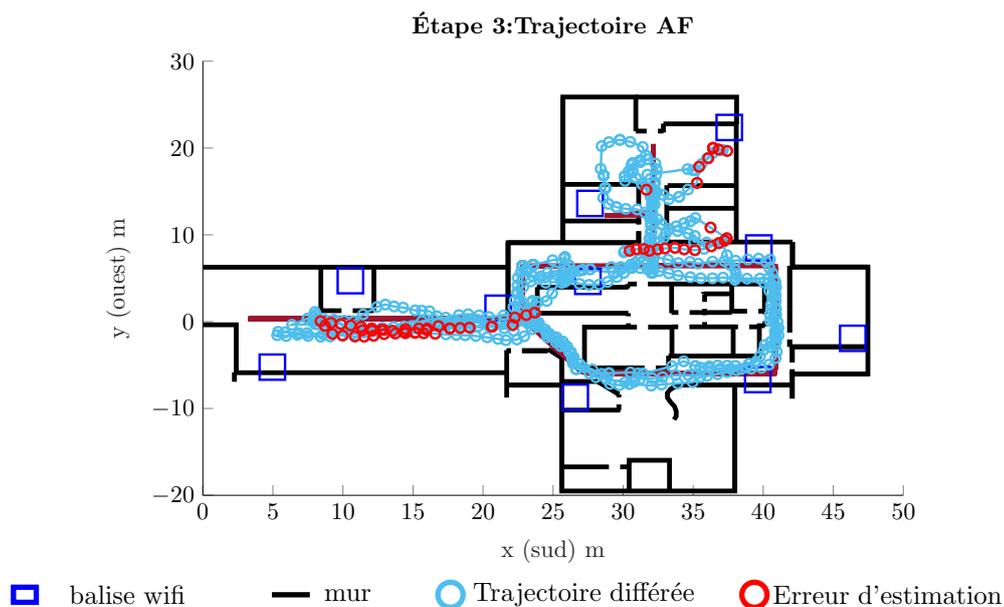


FIGURE 5.14 – Trajectoire AF

Comme pour l'étape précédente, on pourra comparer directement l'évolution de l'écart entre la position estimée et la position de référence dans la figure 5.15.

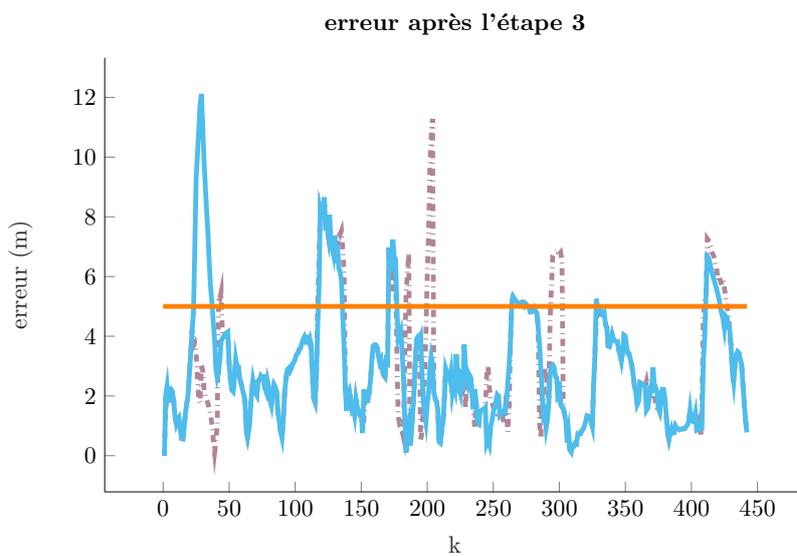


FIGURE 5.15 – Évolution de l'erreur le long de la trajectoire AF. En pointillé est représentée l'erreur de l'estimation de l'étape précédente.

- contrairement à toutes les estimations précédentes, on observe qu'il n'y a pas de sauts de *clusters*. Cette limite inhérente au temps réel a été corrigée sur l'estimation de la trajectoire en temps différé.
- Par rapport à l'estimation de la trajectoire après l'étape 2, 72% des positions sont identiques, c'est-à-dire que parmi le chemin de *cluster* estimé par Viterbi, 72% possédait le poids le plus lourd en temps réel. La trajectoire AF est plus proche de la trajectoire de référence que l'estimation après l'étape 2 dans 16%, des cas mais est plus loin dans 12% des cas. Contrairement aux autres estimations, le couloir a mal été estimé lors du premier passage (aux environs du 30e ème pas) dans la figure 5.15, mais a correctement estimé le second passage (aux alentours du 200ème pas). Ceci démontre bien que l'algorithme de Viterbi n'a pas été en mesure de déterminer le meilleur chemin possible parmi le chemin de *cluster* malgré l'amélioration globale de la trajectoire estimée. Ceci constitue une limite de cette approche.

5.2.2.5 Synthèse

1. Dans un premier temps, cet exemple illustre la complémentarité de chacune des trois étapes puisque les erreurs ont systématiquement été réduites. L'ajustement WIB a permis de faire parcourir les bons couloirs, la répétition du filtre a grandement diminué l'écart avec la trajectoire de référence, et l'algorithme de Viterbi a corrigé les discontinuités.
2. L'erreur rmse associée à la trajectoire PF était de 5.27m contre 3.62m pour la trajectoire AF, ce qui correspond à un gain d'environ 45%. De même, le ratio pour la trajectoire PF était de 64.33% contre une valeur de 85.74% pour la trajectoire AF correspondant à un gain de 34%.
3. Néanmoins, l'utilisation cumulée des approches n'a pas permis d'estimer la bonne trajectoire dans sa totalité. De plus l'historique des *clusters* aurait pu permettre d'obtenir une meilleure solution que celle proposée par Viterbi puisque pour chaque passage dans le couloir, au moins un *cluster* a correctement suivi la bonne hypothèse.
4. Enfin, certains paramètres doivent être fixés en amont et ceci constitue une limite potentielle. Par exemple, pour l'optimisation WIB ou encore la répétition d'une sous-trajectoire, il aurait été intéressant de savoir détecter les portions complexes ou mal estimées afin d'insister spécifiquement sur ces portions.

5.2.3 Application à la base de données

5.2.3.1 Comparaison des temps de calcul

On s'intéresse à comparer les différents coûts de calcul théoriques en fonction des différents paramètres d'intérêt. Une autre métrique permettant d'évaluer le coût de calcul est le temps d'exécution sur l'ensemble de la base de données. S'intéresser au temps de calcul permet d'avoir une bonne idée des coûts effectifs mais dépend grandement de l'architecture du code. L'algorithme de localisation implémenté au CEA a été développé pour une approche en temps réel. Une adaptation a permis d'estimer des trajectoires en temps différé, mais cette adaptation n'est pas garantie d'être optimale.

Les conditions d'évaluation sont identiques à celles présentées dans la partie 5.2.2, c'est-à-dire :

- $i_{stop} = 800$, le nombre d'itérations pour l'optimisation WIB
- $n_p = 500$, le nombre de particules
- $n_c = 3$, le nombre de *clusters*
- $n_{mc} = 30$, le nombre de répétitions du filtre
- $n_{mur} \simeq 20$, correspond environ au nombre de murs testés par chaque particule.

Les coûts de calcul sont présentés dans le tableau 5.3. La référence du temps de calcul est de 10 minutes. Cette durée correspond au temps total pour que le benchmark présenté dans la partie 5.1.3 puisse fournir un fichier de résultat contenant l'ensemble des 190 expériences pour le filtre particulaire en temps réel. L'algorithme de Viterbi, appliqué sur les *clusters*, et sans tenir compte des murs, est presque équivalent en coût de calcul à un filtre particulaire en temps réel, c'est-à-dire 10 minutes. En revanche, l'optimisation WIB est coûteuse, 3 fois plus longue que le filtre en temps réel, car le nombre d'itérations qui a été fixé est grand (800), afin de garantir le mieux possible la convergence vers le minimum global. Enfin, la répétition du filtre reste la méthode la plus lourde en terme de calcul, 5 fois plus longue que le PF. Néanmoins, pour la répétition de 30 filtres, ce rapport est faible. Ceci se justifie par la structure du code. En effet, les fonctions les plus chronophages sont l'initialisation avant le lancement du filtre ainsi que l'écriture des résultats dans les fichiers correspondants. Ces fonctions n'étant appelées qu'une fois pour la répétition du filtre, le temps de calcul reste raisonnable sur la base de données totale. On définira alors le temps de calcul propre à chaque approche, qui correspond au temps de calcul total auquel aura été soustrait le temps de calcul structurel, qui correspond environ à 9 minutes sur l'ensemble de la base de données, afin de mieux comparer le coût associé à chaque approche. Le benchmark met 80 minutes afin de traiter et d'estimer l'ensemble des 190 trajectoires de la base de données à l'aide de l'approche finale.

Notation	Coût de calcul	Temps de calcul total	Temps de calcul propre à l'approche
PF	$O(Kn_p n_{mur})$	10min	1min20
WPF	$O(K(2n_p n_{mur} + i_{stop}))$	30min	21min
MCPF	$O(Kn_p n_{mur} n_{mc})$	49min	40min
VPF	$O(K(n_p n_{mur} + n_c^2))$	10min	1min25
AF	$O(K((n_{mc} + 2)n_p n_{mur} + i_{stop} + n_c^2))$	80min	71min

TABLEAU 5.3 – Comparaison des temps de coûts de calcul et des temps de calcul effectifs

5.2.3.2 Application numérique sur l'ensemble de la base

Résultats par site de campagne sur un étage

On représente respectivement dans les tableaux 5.4 et 5.5 les valeurs moyennes du %d5 et du rmse par site de campagne des expériences sur un seul étage. A chaque valeur, on a également associé le gain en performance.

Site	<i>nb</i>	PF	WPF	MCPF	VPF	AF
Site 1	38	80,30 -	83,49 3,97%	84,49 5,22%	83,54 4,04%	86,25 7,41%
Site 1Mod	34	85,99 -	87,76 2,06%	88,62 3,06%	87,20 1,4%	90,14 4,83%
Site 2	4	95,54 -	96,05 0,54%	100 4,67%	100 4,67%	100 4,67%
Site 3	54	81,27 -	83,02 2,16%	86,58 6,54%	82,95 2,08%	87,29 6,90%

TABLEAU 5.4 – Valeurs de la moyenne du %d5 par site de campagne pour les trajectoires sur un seul étage

Site	<i>nb</i>	PF	WPF	MCPF	VPF	AF
Site 1	38	10,50 -	6,34 39,61%	7,58 27,84%	4,63 55,89%	4,53 56,86%
Site 1Mod	34	4,42 -	3,81 13,83%	3,43 22,47%	3,62 18,13%	3,25 26,60%
Site 2	4	2,48 -	2,00 19,23%	1,03 58,41%	1,10 55,49%	1,10 55,59%
Site 3	54	4,58 -	4,31 5,87%	3,57 22,14%	4,40 3,98%	3,39 26,13%

TABLEAU 5.5 – Valeurs de la moyenne du rmse en mètre par site de campagne pour les trajectoires sur un seul étage

- Pour le site 1 de la figure 5.1, les balises wifi se situent uniquement dans le bâtiment de gauche de la carte. Si le point de départ se trouvait loin de ces balises et que le cap initial qui a été suivi par l'estimation en temps réel était incorrect, alors la mauvaise hypothèse sera suivie durant un long trajet, par exemple sur 40m, avant d'être corrigée par les mesures RSS. Cette erreur sur le cap initial, qui constitue la principale source d'erreur liée à ce site, est corrigée par chacune des approches en temps différé. La trajectoire WPF estimera le cap initial grâce à la trajectoire WIB. L'une des répétitions du filtre pour la trajectoire MCPF choisira éventuellement la bonne hypothèse. Enfin, la trajectoire VPF dont l'objet consiste spécifiquement à supprimer ces sauts de *clusters* permettra d'obtenir le plus grand gain en terme d'erreur rmse par rapport au temps réel.
- Le site 1 modifié, ou encore le site 3, correspondent à des bâtiments de bureaux avec plus de situations d'espaces larges que le site 1. Pour ces expériences, c'est l'estimation MCPF qui permet d'obtenir les meilleurs résultats puisqu'elle peut ainsi tirer

profit des espaces plus larges pour augmenter la variabilité du filtre. En effet, deux trajectoires différentes peuvent parcourir le même couloir tout étant relativement espacées.

- Enfin, 4 expériences ont été estimées pour le site 2 de la figure 5.2. Pour ces expériences assez courtes, d’une moyenne de 150m environ, la couverture wifi est totale, et les couloirs sont très resserrés. Il s’agit de conditions idéales facilitant l’estimation en temps réel. Sur ces 4 expériences, 3 ont une valeur de %d5 de 100%, et la 4ème expérience a une valeur de 81%. Pour cet exemple, l’estimation VPF a permis d’obtenir une valeur de %d5 de 100% pour les 4 expériences. L’apport de l’approche finale par rapport à l’approche VPF devient nul pour un coût de calcul qui reste élevé, ce qui constitue une limite de l’utilisation du couplage.

Résultats par site de campagne sur plusieurs étages

Les tableaux 5.6 et 5.7 contiennent les performances associées aux tests réalisés sur plusieurs étages.

Site	<i>nb</i>	PF	WPF	MCPF	VPF	AF
Site 1	30	72,74	70,58	76,79	72,87	77,63
		-	-2,97%	5,57%	0,18%	6,72%
Site 4	30	48,34	46,66	52,59	49,56	52,95
		-	-3,48%	8,79%	2,54%	9,54%

TABLEAU 5.6 – Valeurs de la moyenne du %d5 par site de campagne pour les trajectoires sur plusieurs étages

Site	<i>nb</i>	PF	WPF	MCPF	VPF	AF
Site 1	30	8,31	8,12	6,64	7,79	5,58
		-	-2,25%	20,08%	6,25%	32,82%
Site 4	30	10,22	10,75	9,04	10,06	8,80
		-	-5,22%	11,54%	1,57%	13,92%

TABLEAU 5.7 – Valeurs de la moyenne du rmse en mètre par site de campagne pour les trajectoires sur plusieurs étages

- Pour les trajectoires sur plusieurs étages, l’approche WPF détériore l’approche en temps réel. En effet, celle-ci n’a pas été adaptée à plusieurs étages. L’optimisation WIB a été effectuée sur le premier étage uniquement, mais la trajectoire inertielle a été corrigée sur l’ensemble de la trajectoire. Cette utilisation implique que cette approche améliore l’estimation du premier étage par rapport au temps réel mais dégrade l’estimation sur les étages suivants.
- Les performances associées au site 4 de la figure 5.4 sont largement inférieures comparativement aux autres sites. Ceci se justifie par la faible densité de balise, mais également par la présence d’espaces libres qui ne permettent pas de contraindre l’estimation de la trajectoire lorsque celle-ci diverge. Pour ce site, même si l’approche finale améliore les résultats du temps réel, elle ne permet pas d’estimer correctement la trajectoire parcourue, ce qui illustre les limites de cette approche dans des situations d’estimation difficiles.

Résultats sur l'ensemble des expériences

Les méthodes sont comparées sur l'ensemble des 190 expériences de la base de données. En plus des métriques habituellement utilisées, on s'intéressera également aux ratios des trajectoires correctes au sens du rmse et au sens du %d5 comme définis dans la partie 4.3.2.1. Les résultats sont représentés dans le tableau 5.8, mais également graphiquement dans la figure 5.16.

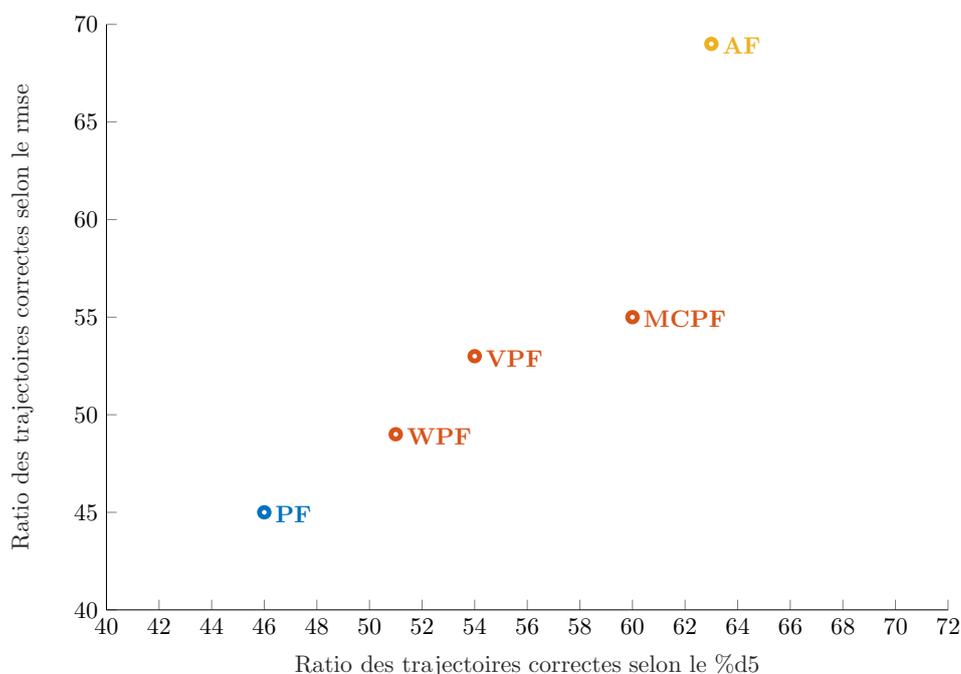


FIGURE 5.16 – Représentation du ratio de trajectoires correctes selon le rmse en fonction du ratio de trajectoires correctes selon %d5 pour chaque approche sur un ensemble de 190 expériences sur un ou plusieurs étages

La figure 5.16 illustre bien dans un premier temps que chacune des estimations en temps différé améliore la trajectoire PF en terme d'erreur rmse, mais également en terme de ratio %d5. En comparant chacune des approches individuellement, la meilleure estimation, qui est également la plus coûteuse, est donnée par l'approche MCPF, s'agissant en effet de l'approche la plus robuste. L'approche WPF peut détériorer l'estimation de la trajectoire selon la qualité des mesures RSS, et l'algorithme de Viterbi est limité par l'historique des *clusters* qui ont été estimés en temps réel. Néanmoins, l'algorithme de Viterbi, et notamment son application sur la *clusters* permise par la modélisation GMM, présente un gain d'environ 22% sur l'erreur moyenne rmse contre un coût de calcul quasiment similaire.

La figure 5.16 confirme également l'analyse de l'exemple applicatif de la partie 5.2.2. Ces approches sont en effet complémentaires et permettent de définir une estimation finale, qui est la trajectoire AF et qui aboutit aux meilleurs résultats. Le résultat obtenu par l'estimation AF est un gain de l'erreur rmse de 34% contre un coût de calcul effectif total multiplié par 8 en considérant le coût structurel de l'outil du benchmark. La moyenne du ratio %d5 varie très peu quand on s'intéresse à la moyenne parce que de nombreuses trajectoires ont correctement été estimées par le filtre temps réel qui est déjà un outil très performant de localisation. Néanmoins, quand on s'intéresse au ratio de trajectoires correctes selon le %d5, alors l'approche finale permet d'améliorer un plus grand nombre de

trajectoires, 63% contre 46% pour le PF. Ce résultat met en perspective la résolution des limites inhérentes au temps réel, telles que les discontinuités, ou la non-connaissance du cap initial.

Trajectoire	Moyenne %d5 (%)	Moyenne rmse (m)	Ratio %d5 (%)	Ratio rmse (%)
PF	75,72	7,15	46	45
WPF	76,60	6,16	51	49
MCPF	79,95	5,54	60	55
VPF	77,38	5,82	54	53
AF	80,96	4,71	63	69

TABLEAU 5.8 – Comparaison de toutes les approches sur un ensemble de 190 expériences sur un ou plusieurs étages

5.2.4 Conclusion

Nous avons développé 3 approches basées autour d'un filtre particulaire, outil très répandu et performant dans le cadre de la navigation *indoor*. La première approche, WPF, améliore l'erreur rmse d'environ 14%, en corrigeant la trajectoire inertielle selon certaines transformations simples qui sont des rotations et des homothéties, puis en appliquant un filtre particulaire. Cette approche permet d'obtenir une bonne estimation de la trajectoire finale malgré une mauvaise qualité de la trajectoire inertielle. Dans notre cas, il devient alors possible de s'affranchir de la taille d'un individu. La seconde approche de répétition du filtre implique un gain sur l'erreur rmse d'environ 23%. Cette approche permet d'augmenter la variabilité des solutions possibles, tout en garantissant une robustesse accrue du filtre pour un coût de mémoire presque identique à celui nécessaire à un filtre particulaire, mais associé à un temps de calcul effectif multiplié d'un facteur 5. Ce chiffre est à nuancer en raison du coût structurel qui est commun à chacune de nos approches. Enfin, l'algorithme de Viterbi, qui a été appliqué suite à une modélisation par des GMM, corrige les changements de *clusters* et donc les discontinuités, impliquant un gain de rmse de 19% pour un coût de calcul presque identique à un filtre particulaire en temps réel. L'approche finale combine ces 3 approches qui sont tout à fait complémentaires, et permet un gain sur le rmse de 34% contre un coût de calcul multiplié par un facteur 8. Néanmoins, le cadre de la localisation *indoor* en temps différé est bien moins limitatif en coût de calcul que pour le temps réel, qui nécessite des réponses instantanées. Différentes perspectives permettraient d'améliorer cet algorithme :

- Cette trajectoire AF est estimée à partir d'un ensemble d'approches qui nécessitent de fixer de façon empirique certains paramètres, tels que le nombre de *clusters* pour l'algorithme de Viterbi, ou encore la distance de séparation pour l'optimisation WIB et pour la répétition du filtre en utilisant des sous-trajectoires. Dans notre cas, nous avons à notre disposition une grande base de données qui a permis d'ajuster certains choix des paramètres, et de se forger la conviction de la robustesse de ces méthodes. Dans une optique de type *crowdsourcing* par exemple, pouvoir proposer des approches automatisées permettrait un gain significatif.
- La trajectoire AF est essentiellement centrée autour d'un filtre particulaire. Dans le cadre de la localisation *indoor*, le filtre particulaire est un outil pertinent car il

permet d'intégrer les contraintes de la carte, mais également de proposer des distributions multi-modales. Néanmoins, le filtre particulière peut être mis à défaut dans des configurations complexes, telles que les espaces très larges. L'intérêt du couplage avec la trajectoire WIB réside bien dans cette possibilité d'adaptation à des environnements *a priori* complexes.

5.3 Références

- BONNET, S., C. BASSOMPIERRE, C. GODIN, S. LESECQ et A. BARRAUD. 2009, «Calibration methods for inertial and magnetic sensors», *Sensors and Actuators A : Physical*, vol. 156, n° 2, p. 302–311. 127
- BRENA, R. F., J. P. GARCÍA-VÁZQUEZ, C. E. GALVÁN-TEJADA, D. MUÑOZ-RODRIGUEZ, C. VARGAS-ROSALES et J. FANGMEYER. 2017, «Evolution of indoor positioning technologies : A survey», *Journal of Sensors*, vol. 2017. 122
- LE SCORNEC, J., M. ORTIZ et V. RENAUDIN. 2017, «Foot-mounted pedestrian navigation reference with tightly coupled GNSS carrier phases, inertial and magnetic data», dans *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*, IEEE, p. 1–8. 122
- LEARNING, M. 2007, «Machine learning repository», URL <http://archive.ics.uci.edu/ml/index.php>, accessed : 2017-11-22. 122
- LYMBEROPOULOS, D., J. LIU, X. YANG, R. R. CHOUDHURY, V. HANDZISKI et S. SEN. 2015, «A realistic evaluation and comparison of indoor location technologies : Experiences and lessons learned», dans *Proceedings of the 14th international conference on information processing in sensor networks*, ACM, p. 178–189. 122
- MONTOLIU, R., E. SANSANO, J. TORRES-SOSPEDRA et O. BELMONTE. 2017, «IndoorLoc platform : A public repository for comparing and evaluating indoor positioning systems», dans *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*, IEEE, p. 1–8. 122
- DE LA OSA, C. M., G. G. ANAGNOSTOPOULOS, M. TOGNERI, M. DERIAZ et D. KONSTANTAS. 2016, «Positioning evaluation and ground truth definition for real life use cases», dans *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, IEEE, p. 1–7. 126
- POTORTÌ, F., S. PARK, A. R. JIMÉNEZ RUIZ, P. BARSOCCHI, M. GIROLAMI, A. CRIVELLO, S. Y. LEE, J. H. LIM, J. TORRES-SOSPEDRA, F. SECO et collab.. 2017, «Comparing the performance of indoor localization systems through the EvAAL framework», *Sensors*, vol. 17, n° 10, p. 2327. 122
- TRACKING, A. R. 2016, «ARTTRACK system», URL <https://ar-tracking.com/technology/motion-tracking/>, accessed : 2017-11-22. 122
- VAN HAUTE, T., E. DE POORTER, F. LEMIC, V. HANDZISKI, N. WIRSTRÖM, T. VOIGT, A. WOLISZ et I. MOERMAN. 2015, «Platform for benchmarking of RF-based indoor localization solutions», *IEEE Communications Magazine*, vol. 53, n° 9, p. 126–133. 122
- VICON. 2014, «Vicon Object Tracking», URL <https://www.vicon.com/motion-capture/engineering>, accessed : 2017-11-22. 122

Conclusion générale

Si les récentes années ont vu se développer un très grand nombre de recherches sur la localisation *indoor* en temps réel, l'approche en temps différé doit encore être largement approfondie. Les travaux de cette thèse se sont portés sur l'estimation en temps différé d'une trajectoire réalisée par un individu dans un environnement *indoor*, à partir des capteurs présents dans un *smartphone*, mais également des informations sur la carte du bâtiment.

Dans un premier temps, on a développé un modèle paramétrique afin de fusionner les mesures inertielles avec les mesures RSS. En effet, les mesures inertielles permettent d'estimer une trajectoire relative correcte mais mauvaise en erreur absolue car elle dérive avec le temps, lorsque les mesures RSS permettent d'obtenir une trajectoire qui ne dérive pas avec le temps mais associée à une précision faible. Cette précision est par ailleurs dégradée par l'utilisation d'un modèle de propagation d'ondes à l'intérieur d'un bâtiment qui est sujet à de nombreuses perturbations. L'approche en temps différé définit un modèle qui combine des mesures de proximité pour les mesures RSS de puissance fortes, et un modèle adaptatif afin de palier les limites de l'environnement intérieur pour les autres mesures reçues. Ce modèle est alors couplé à un modèle paramétrique par morceaux estime pour chaque morceau de trajectoire une rotation ainsi qu'une homothétie. A partir de cette approche uniquement basée sur les mesures inertielles et les mesures RSS, on obtient une moyenne de 6.3m d'erreur rmse. Cette approche améliore le temps réel de près de 50%, mais nécessite de définir des paramètres de façon empirique tels que la distance de séparation entre chaque morceau de trajectoire ou encore le nombre de mesures de proximité. Une perspective d'amélioration réside dans la détermination automatisée des paramètres optimaux. Également, les seules transformations autorisées pour les morceaux de la trajectoire sont des rotations et des homothéties. Une amélioration serait alors de proposer des transformations curvilignes.

Dans un second temps, une approche utilisant les plans physiques du bâtiment comme source d'information supplémentaire permet une estimation de la trajectoire en temps réel basée sur l'utilisation d'un filtre particulaire. Cette méthode d'estimation permet déjà d'obtenir une estimation moyenne satisfaisante de 5.8m de rmse évaluée sur des trajectoires sans changement d'étage. Toutefois, elle reste sujette à des problèmes de continuité, une limite inhérente à une estimation en temps réel. En temps réel, on applique à l'ensemble des particules une classification non-supervisée en différents *clusters* afin de représenter, par chaque *cluster*, une hypothèse de localisation différente. On considère ensuite une réduction de l'information des particules à travers l'historique de *clusters* en définissant un modèle de mélange de gaussiennes. La problématique devient alors de déterminer, parmi cet historique de *clusters*, le chemin le plus probable. Des approches de lissage particulaire redéfinissent *a posteriori* de nouveaux poids associés à chaque *clusters* et augmentent la précision moyenne en temps réel de 7%,

ce qui correspond à 5.4m d'erreur rmse. Une estimation du maximum *a posteriori* par Viterbi, impliquant une précision moyenne de 4.2m d'erreur rmse et un gain de 28%, a été appliquée sur ces *clusters*. On observe tout de même que sur certains exemples, malgré une amélioration significative des résultats, l'algorithme de Viterbi n'estimait pas systématiquement le meilleur chemin de *cluster*.

Par la suite, des heuristiques numériques ont été mises en oeuvre afin d'améliorer l'estimation en temps différé. Une première heuristique consiste à répéter plusieurs fois un filtre particulaire en temps réel. Un filtre particulaire possédant une composante aléatoire, chaque réalisation sera différente. Une stratégie de score non-supervisée, basée sur un score RSS et sur un score de continuité, est alors définie afin de sélectionner la meilleure réalisation du filtre parmi les différentes répétitions. Cette heuristique a permis d'obtenir une erreur moyenne de 4.5m rmse ce qui correspond à un gain de 22% sur le temps réel. La limite de cette heuristique réside dans son coût de calcul très lourd comparativement au temps réel. En effet, la répétition du filtre particulaire est naïve. Une amélioration réside dans la construction d'un modèle de simulation qui tire profit de la meilleure réalisation du filtre pour guider les réalisations suivantes.

Une seconde heuristique applique un filtre particulaire à partir d'un modèle d'état qui aura été corrigé par le modèle paramétrique adaptatif défini en temps différé utilisant uniquement les mesures RSS et IMU. L'ajustement de ce modèle réduit l'erreur moyenne de rmse à 4.6m, donc un gain de 21% par rapport au temps réel. Toutefois, la correction de la trajectoire inertielle par le modèle paramétrique ne garantit pas d'améliorer l'utilisation d'un filtre particulaire, et peut même détériorer la qualité d'estimation. On utilise alors le score RSS non-supervisé afin de définir une règle de décision quant à l'utilisation d'un couplage du modèle paramétrique avec le filtre particulaire. Néanmoins, cette approche est trop dépendante de la qualité des mesures RSS. Dans certains cas complexes, notamment lorsque le nombre de balises est très faible, le score RSS ne garantit pas une règle de décision pertinente.

Pour certaines applications, par exemple dans le cadre d'applications utilisant le *crowdsourcing* pour lesquelles un grand nombre de données est disponible, il n'est pas toujours nécessaire de traiter la totalité des données recueillies. Dans cette optique, on propose un modèle statistique de sélection des trajectoires correctement estimées qui est basé sur des forêts aléatoires. Les différentes caractéristiques permettant de discriminer les trajectoires correctement estimées sont la durée de la trajectoire, les scores RSS et de continuité, ainsi que la correspondance entre deux estimations différentes en temps différé; l'estimation uniquement obtenues à partir des mesures RSS et inertielles, et l'estimation couplant l'ensemble des approches et heuristiques définies autour d'un filtre particulaire. Les résultats obtenus sur ce modèle sont un taux de vrais positifs d'environ 56% associé à un taux de vrais négatifs de 90%. Ceci signifie que l'on peut limiter le taux d'erreur sur la classification des trajectoires correctes à 10%, tout en conservant plus de la moitié des expériences. La définition d'un modèle d'apprentissage plus spécifiquement basé sur des variables environnementales améliorerait ces résultats. Toutefois, ceci nécessite un nombre d'expériences supervisées conséquent pour chaque bâtiment, ce qui n'est pas toujours réalisable.

Enfin, en couplant l'ensemble des heuristiques et approches définies en temps différé, on obtient une estimation finale avec une moyenne de rmse de 3.6m, ce qui correspond à un gain de 38% par rapport au temps réel. Si cette estimation finale permet un gain significatif par rapport au temps réel, elle cumule les limites de chacune des différentes approches qui la composent, notamment le coût de calcul effectif, ainsi que

le choix empirique des différents paramètres.

Au-delà des choix des axes de recherches développés par ces travaux, différentes pistes constitueraient des défis intéressants au regard de la problématique de localisation en temps différé :

L'estimation en temps différé des trajectoires comportant des changements d'étage a été faite en appliquant les approches aux trajectoires sur un seul étage sans aucune adaptation, lorsqu'une réflexion spécifique permettrait de tirer profit des changements d'étage. Par exemple, on sait à chaque changement d'étage que l'utilisateur passe par la sortie d'une cage d'escalier. On peut donc définir des points de passage intermédiaire, et ainsi reformuler la problématique : quel est le trajet parcouru d'un utilisateur dont on connaît le point de départ ainsi qu'un point de passage ?

En outre, ces travaux pouvant servir pour une application de *crowdsourcing*, une suite serait d'appliquer l'approche d'estimation finale, ainsi que l'algorithme de sélection de trajectoires donc sur un grand nombre d'utilisateurs, afin, par exemple, de développer des cartes de fréquence d'un bâtiment.

Annexe A

Développements mathématiques

A.1 Preuve par récurrence de la formule 2.24

En posant $\beta_0 = 0$, $\epsilon_0 = 0$ et $\mathbf{p}_{\text{pdr}}^{\mathbf{k}_0} = \mathbf{p}_{\text{pdr}}^0 = 0$:

$$\mathbf{p}_0^{\mathbf{k}} = \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^i \left[\prod_{j=1}^l (1 + \epsilon_{j-1}) \mathbf{R}(\beta_{j-1}) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_l} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{l-1}}) \quad (\text{A.1})$$

A.1.1 Initialisation de la récurrence :

Pour $i = 1$, $\mathbf{p}_0^{\mathbf{k}} \in \mathbf{C}_1(\theta)$ et ne dépend donc uniquement des paramètres (β_1, ϵ_1) , ainsi :

1. On effectue une translation pour centrer le repère sur le point de départ de la trajectoire pour pouvoir effectuer la rotation :

$$\mathbf{I}_1 = \mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^1$$

2. On effectue la rotation et l'homothétie de paramètres (β_1, ϵ_1) :

$$\begin{aligned} \mathbf{I}_2 &= (1 + \epsilon_1) \mathbf{R}(\beta_1) \mathbf{I}_1 \\ &= (1 + \epsilon_1) \mathbf{R}(\beta_1) \left[\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^1 \right] \end{aligned}$$

3. On effectue la translation inverse et on obtient la position finale recherchée :

$$\begin{aligned} \mathbf{p}_0^{\mathbf{k}} &= \mathbf{I}_2 + \mathbf{p}_{\text{pdr}}^1 \\ &= (1 + \epsilon_1) \mathbf{R}(\beta_1) \left[\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^1 \right] + \mathbf{p}_{\text{pdr}}^1 \end{aligned}$$

A.1.2 Hérité de la récurrence :

On suppose que pour $1 \leq i < n_r$, on a pour tout k tel que $\mathbf{p}_{\text{pdr}}^{\mathbf{k}} \in \mathbf{S}_i$:

$$\mathbf{p}_0^{\mathbf{k}} = \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_l} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{l-1}})$$

On cherche alors à calculer la position pour un élément du segment \mathbf{S}_{i+1}

1. On applique les transformations appliquées au segment \mathbf{S}_i selon l'hypothèse de récurrence :

$$\mathbf{J}_1 = \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_l} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{l-1}})$$

2. On effectue une translation pour centrer le repère sur $\mathbf{p}_0^{\mathbf{k}_{i+1}}$, c'est-à-dire $\mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}}$ auquel a été appliqué les transformations appliquées au segment \mathbf{S}_i selon l'hypothèse de

récurrence :

$$\begin{aligned}
 \mathbf{J}_2 &= \mathbf{J}_1 - \mathbf{p}_0^{\mathbf{k}_{i+1}} \\
 &= \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}) \\
 &\quad - \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) - \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}) \\
 &= \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}})
 \end{aligned}$$

3. On effectue ensuite les transformations associées au segment \mathbf{S}_{i+1} :

$$\begin{aligned}
 \mathbf{J}_3 &= (1 + \epsilon_{i+1}) \mathbf{R}(\beta_{i+1}) \mathbf{J}_2 \\
 &= \left[\prod_{j=1}^{i+1} (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}})
 \end{aligned}$$

4. On effectue la translation inverse, pour trouver la position finale :

$$\begin{aligned}
 \mathbf{p}_0^{\mathbf{k}} &= \mathbf{J}_3 + \mathbf{p}_0^{\mathbf{k}_{i+1}} \\
 &= \left[\prod_{j=1}^{i+1} (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}}) + \left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_{i+1}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) \\
 &\quad + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}) \\
 \mathbf{p}_0^{\mathbf{k}} &= \left[\prod_{j=1}^{i+1} (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} + \sum_{l=1}^i \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}))
 \end{aligned}$$

ce qui conclue la récurrence.

A.2 Évaluation des matrices jacobiennes

A.2.1 Évaluation de la matrice jacobienne \mathbf{J}_z

$$\mathbf{J}_z = \frac{\partial}{\partial \theta} \left[\mathbf{z}(\theta) - \overline{\mathbf{z}(\theta)} \mathbf{1}_L \right] = \frac{\partial}{\partial \theta} \mathbf{z}(\theta) - \frac{\partial}{\partial \theta} \overline{\mathbf{z}(\theta)} \mathbf{1}_L = \frac{\partial}{\partial \theta} \mathbf{z}(\theta) - \frac{1}{K} \sum_{i=1}^L \frac{\partial}{\partial \theta} \left[z_\theta^i \right] \mathbf{1}_K$$

La position d'un segment $\mathbf{S}_i(\theta)$ ne dépend pas des paramètres (ϵ_j, β_j) pour tout $i < j \leq n_r$, par composition de fonctions z_θ^i ne dépend pas de (ϵ_j, β_j) , et on a donc bien que $\frac{\partial}{\partial \beta_j} z_\theta^i = \frac{\partial}{\partial \epsilon_j} z_\theta^i = 0$.

Soit $k \leq K$ tel que $\mathbf{p}_\theta^k \in \mathbf{S}_i(\theta)$, on pose les notations suivantes :

$$z_\theta^k = a + b \log[(\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))] \quad (\text{A.2})$$

où

- $a = P_{tr} + G_t + G_r + 10\alpha_{\text{RSS}} \log \left[\frac{\lambda}{4\pi} \right]$
- $b = -5\alpha_{\text{RSS}}$

Ainsi, on a donc :

$$\begin{aligned} \frac{\partial}{\partial \theta} z_\theta^k &= b \frac{\partial}{\partial \theta} \log[(\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))] \\ &= \frac{2b}{[(\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))]} \left[\frac{\partial}{\partial \theta} (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k})) \right]^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k})) \\ &= \frac{2b}{[(\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k}))]} \left[\frac{\partial}{\partial \theta} \mathbf{p}_\theta^k \right]^\top (\mathbf{p}_\theta^k - \mathbf{b}(\mathbf{k})) \end{aligned} \quad (\text{A.3})$$

Calcul de la dérivée partielle selon un changement d'échelle ϵ_m pour $m \leq i$:

$$\begin{aligned} \frac{\partial}{\partial \epsilon_m} \mathbf{p}_\theta^k &= \frac{\partial}{\partial \epsilon_m} \left(\left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^k - \mathbf{p}_{\text{pdr}}^{k_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{k_j} - \mathbf{p}_{\text{pdr}}^{k_{j-1}}) \right) \\ &= \mathbf{R}(\beta_i) \left(\left[\prod_{j=1, j \neq m}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^k - \mathbf{p}_{\text{pdr}}^{k_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1, j \neq m}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{k_j} - \mathbf{p}_{\text{pdr}}^{k_{j-1}}) \right) \\ &\quad \times \frac{\partial}{\partial \epsilon_m} [1 + \epsilon_m] \\ &= \left(\frac{1}{1 + \epsilon_m} \mathbf{p}_\theta^k \right) \frac{\partial}{\partial \epsilon_m} [1 + \epsilon_m] \text{ car } m \leq i \text{ et on ne considère pas la cas pathologique } \epsilon_m = -1 \end{aligned}$$

d'où

$$\frac{\partial}{\partial \epsilon_m} \mathbf{p}_\theta^k = \frac{1}{1 + \epsilon_m} \mathbf{p}_\theta^k \quad (\text{A.4})$$

Calcul de la dérivée partielle selon l'angle β_m pour $m \leq i$

$$\begin{aligned}
 \frac{\partial}{\partial \beta_m} \mathbf{p}_\theta^{\mathbf{k}} &= \frac{\partial}{\partial \beta_m} \left(\left[\prod_{j=1}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}) \right) \\
 &= \frac{\partial}{\partial \beta_m} \mathbf{R}(\beta_m) \\
 &\times [1 + \epsilon_m] \left(\left[\prod_{j=1, j \neq m}^i (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_i}) + \sum_{l=1}^{i-1} \left[\prod_{j=1, j \neq m}^l (1 + \epsilon_j) \mathbf{R}(\beta_j) \right] (\mathbf{p}_{\text{pdr}}^{\mathbf{k}_j} - \mathbf{p}_{\text{pdr}}^{\mathbf{k}_{j-1}}) \right) \\
 &= \mathbf{R}(\beta_m + \frac{\pi}{2}) \mathbf{R}(-\beta_m) \mathbf{p}_\theta^{\mathbf{k}}
 \end{aligned}$$

d'où

$$\frac{\partial}{\partial \beta_m} \mathbf{p}_\theta^{\mathbf{k}} = \mathbf{R}\left(\frac{\pi}{2}\right) \mathbf{p}_\theta^{\mathbf{k}} \quad (\text{A.5})$$

En insérant ces deux résultats dans l'équation A.3, on obtient alors :

$$\begin{cases} \frac{\partial}{\partial \epsilon_m} z_\theta^{\mathbf{k}} = \frac{2b}{1 + \epsilon_m} \left[\frac{\mathbf{p}_\theta^{\mathbf{k}^T} (\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))}{(\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))^T (\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))} \right] \\ \frac{\partial}{\partial \beta_m} z_\theta^{\mathbf{k}} = 2b \left[\frac{\mathbf{R}(\frac{\pi}{2}) \mathbf{p}_\theta^{\mathbf{k}^T} (\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))}{(\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))^T (\mathbf{p}_\theta^{\mathbf{k}} - \mathbf{b}(\mathbf{k}))} \right] \end{cases} \quad (\text{A.6})$$

A.2.2 Évaluation de la matrice jacobienne \mathbf{J}_v

Soit une mesure de proximité $n \leq n_r$, on considère que $\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} \in \mathbf{S}_i(\theta)$. Ainsi, on a bien, comme précédemment, que pour $i < j \leq n_r$, $\frac{\partial}{\partial \beta_j} v_\theta^n = \frac{\partial}{\partial \epsilon_j} v_\theta^n = 0$.

$$\begin{aligned}
 \frac{\partial}{\partial \theta} v_\theta^n &= \frac{\partial}{\partial \theta} \left[\left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right) \right]^{1/2} \\
 &= \frac{1}{\left[\left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right) \right]^{1/2}} \left[\frac{\partial}{\partial \theta} \mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} \right]^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)
 \end{aligned}$$

En réutilisant les expressions précédentes des équations A.4 et A.5, on obtient pour un changement d'échelle ϵ_m et un angle β_m pour $m \leq i$:

$$\begin{cases} \frac{\partial}{\partial \epsilon_m} v_\theta^n = \frac{1}{1 + \epsilon_m} \frac{\left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)}{\left[\left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right) \right]^{1/2}} \\ \frac{\partial}{\partial \beta_m} v_\theta^n = \frac{\left(\mathbf{R}(\frac{\pi}{2}) \mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)}{\left[\left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right)^T \left(\mathbf{p}_\theta^{\mathbf{k}_{\text{prox},n}} - \mathbf{b}(\mathbf{k}_{\text{prox},n}) \right) \right]^{1/2}} \end{cases} \quad (\text{A.7})$$

A.3 Calcul du score de continuité s_{cont}

On s'intéresse à la moyenne empirique :

$$\bar{p}^{(i)} = \frac{1}{K} \sum_{k=1}^K \tilde{p}_k^{(i)} \sim \mathcal{N}\left(p_0, \frac{\sigma_p^2}{K}\right) \quad (\text{A.8})$$

On obtient ainsi directement :

$$\bar{p}^{(i)} - \tilde{p}_k^{(i)} = \frac{K-1}{K} \tilde{p}_k^{(i)} - \frac{1}{K} \sum_{j=1, j \neq k}^K \tilde{p}_j^{(i)} \quad (\text{A.9})$$

ce qui donne alors que $\bar{p}^{(i)} - \tilde{p}_k^{(i)}$ suit une loi normale d'espérance nulle et de variance égale à $\left(\frac{K-1}{K}\right) \sigma_p^2$. En considérant la variable intermédiaire u_k , on obtient la loi suivante :

$$u_k = \frac{1}{\sigma_l} \sqrt{\frac{K}{K-1}} \left(\bar{p}^{(i)} - \tilde{p}_k^{(i)}\right) \sim \mathcal{N}(0, 1) \quad (\text{A.10})$$

Finalement, le score d'intérêt considérera le produit des vraisemblance associées à u_k comme s'il n'y avait pas de dépendance. On obtient alors, en notant ce produit $M(s_p^2(i))$:

$$\begin{aligned} M(s_p^2(i)) &= \prod_{k=1}^K \frac{1}{\sigma_p \sqrt{2\pi}} \exp\left[\frac{-u_k^2}{2}\right] \\ &= \frac{1}{(\sigma_p \sqrt{2\pi})^K} \exp\left[\sum_{k=1}^K \left[\frac{-u_k^2}{2}\right]\right] \\ &= \frac{1}{(\sigma_p \sqrt{2\pi})^K} \exp\left[\frac{-K}{2\sigma_p^2(K-1)} \sum_{k=1}^K \left(\bar{p}^{(i)} - \tilde{p}_k^{(i)}\right)^2\right] \\ &= \frac{1}{(\sigma_p \sqrt{2\pi})^K} \exp\left[\frac{-K^2}{2\sigma_p^2(K-1)} \sum_{k=1}^K \frac{\left(\bar{p}^{(i)} - \tilde{p}_k^{(i)}\right)^2}{K}\right] \\ &= \frac{1}{(\sigma_p \sqrt{2\pi})^K} \exp\left[\frac{-K^2}{2\sigma_p^2(K-1)} s_p^2(i)\right] \end{aligned}$$

si K est grand, on peut faire l'approximation que $\frac{K}{K-1} \simeq 1$:

$$M(s_p^2(i)) \simeq \frac{1}{(\sigma_p \sqrt{2\pi})^K} \cdot \exp\left[-K \frac{s_p^2(i)}{\sigma_p^2}\right] \quad (\text{A.11})$$

L'objet de ce score étant de pouvoir comparer les différentes estimations des trajectoires, on peut alors uniquement considérer la valeur logarithme de cette expression et finalement définir le score continuité $s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)})$:

$$\begin{aligned} s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)}) &= \log\left(\frac{1}{(\sigma_p \sqrt{2\pi})^K} \cdot \exp\left[-K \frac{s_p^2(i)}{\sigma_p^2}\right]\right) \\ &= -K \log(\sigma_p \sqrt{2\pi}) - K \frac{s_p^2(i)}{\sigma_p^2} \\ s_{cont}(\tilde{\mathbf{x}}_{1:K}^{(i)}) &= -K \left[\log(\sigma_p \sqrt{2\pi}) + \frac{s_p^2(i)}{\sigma_p^2} \right] \quad (\text{A.12}) \end{aligned}$$

Annexe B

Distribution des variables du modèle d'apprentissage

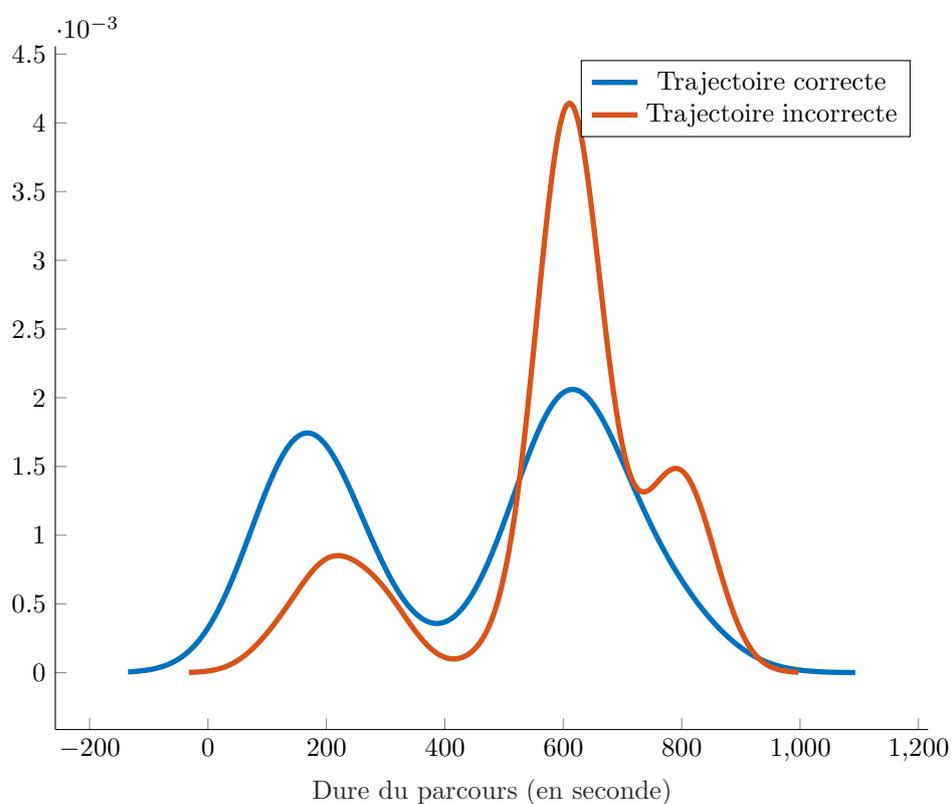


FIGURE B.1 – Distribution de la durée du parcours selon la qualité d'estimation de la trajectoire

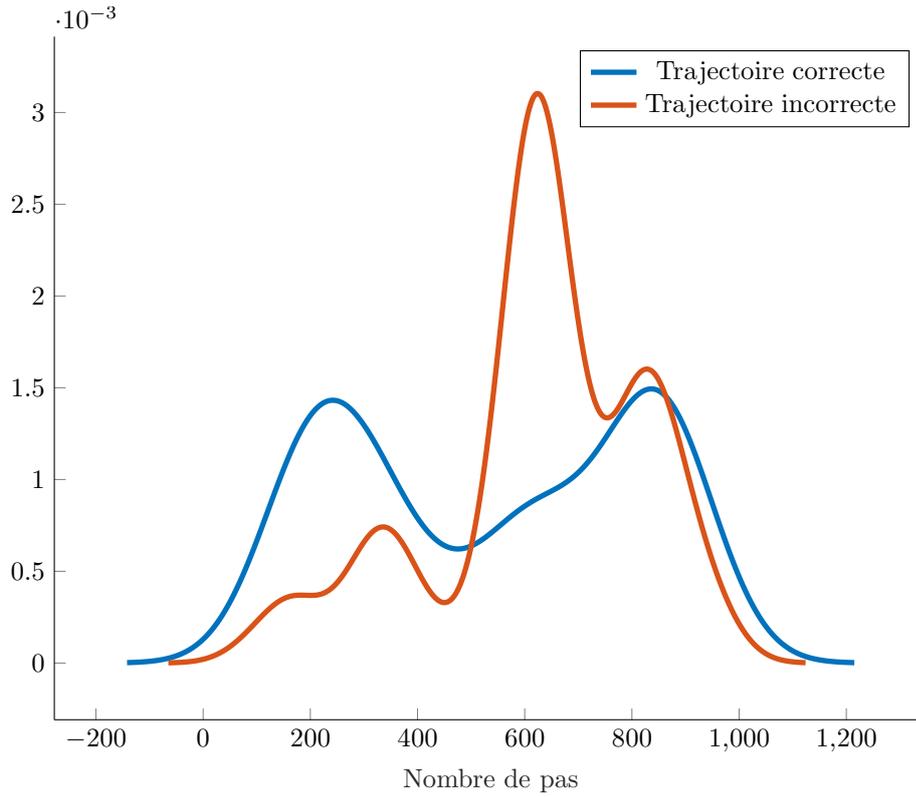


FIGURE B.2 – Distribution du nombre de pas selon la qualité d'estimation de la trajectoire

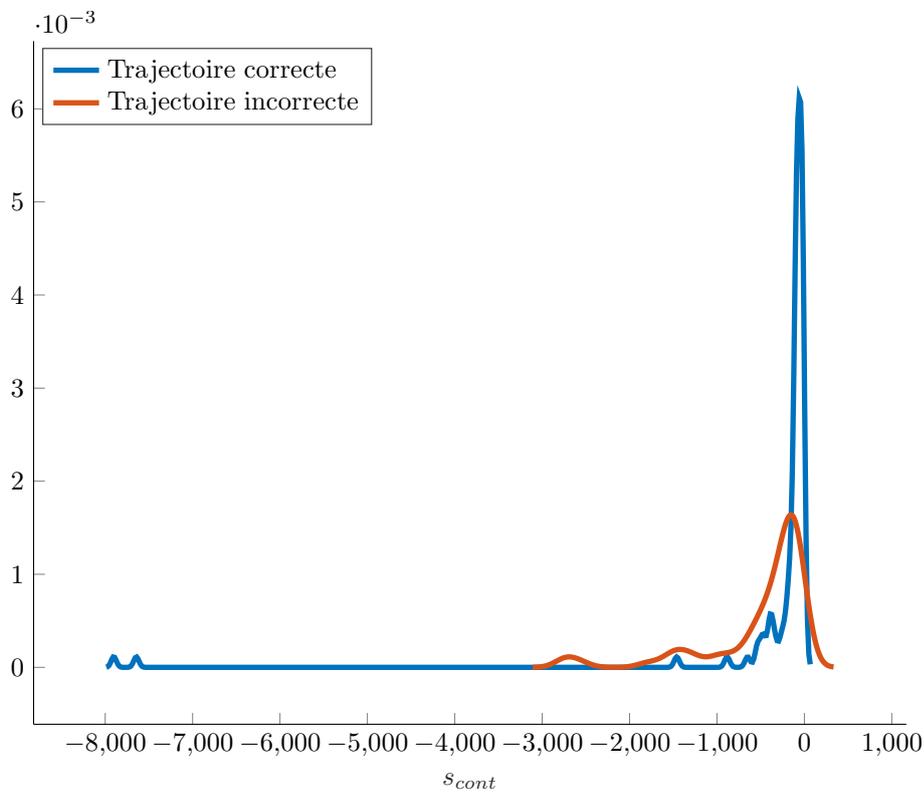


FIGURE B.3 – Distribution du score de continuité selon la qualité d'estimation de la trajectoire

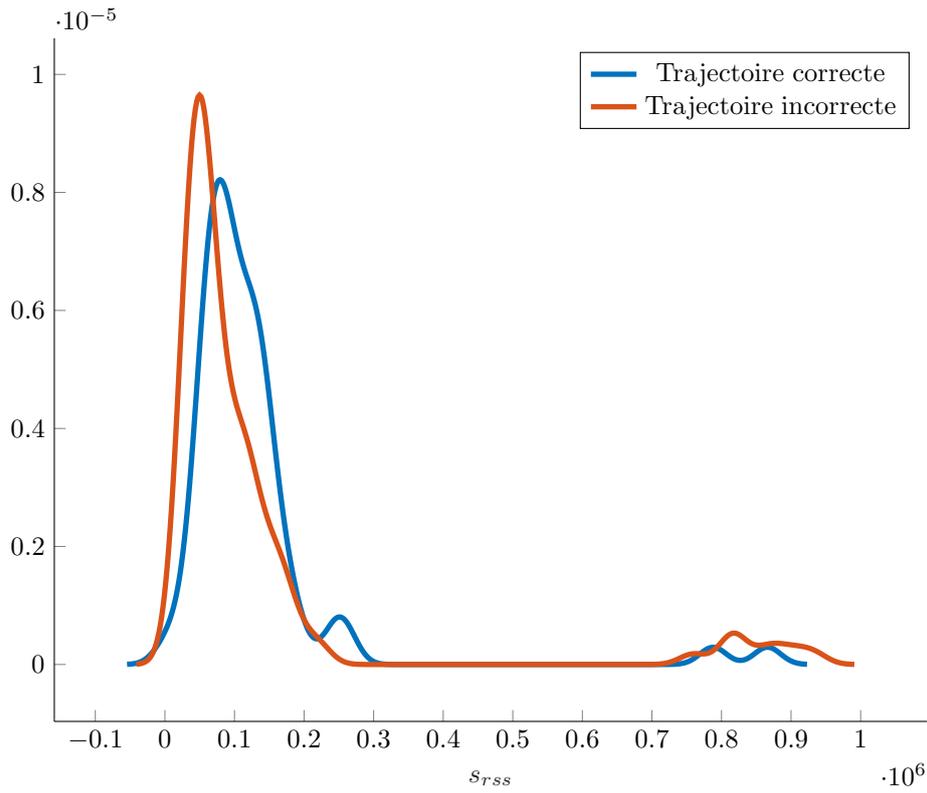


FIGURE B.4 – Distribution du score RSS selon la qualité d'estimation de la trajectoire

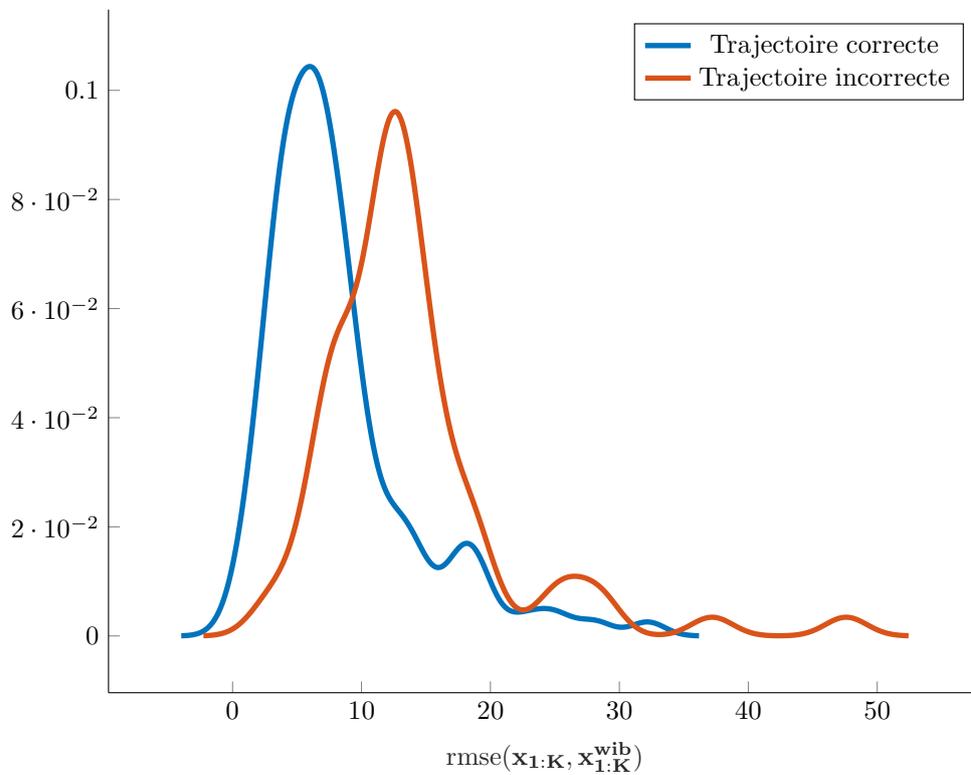


FIGURE B.5 – Distribution de l'erreur rmse entre l'estimation en temps différé et la trajectoire WIB selon la qualité d'estimation de la trajectoire

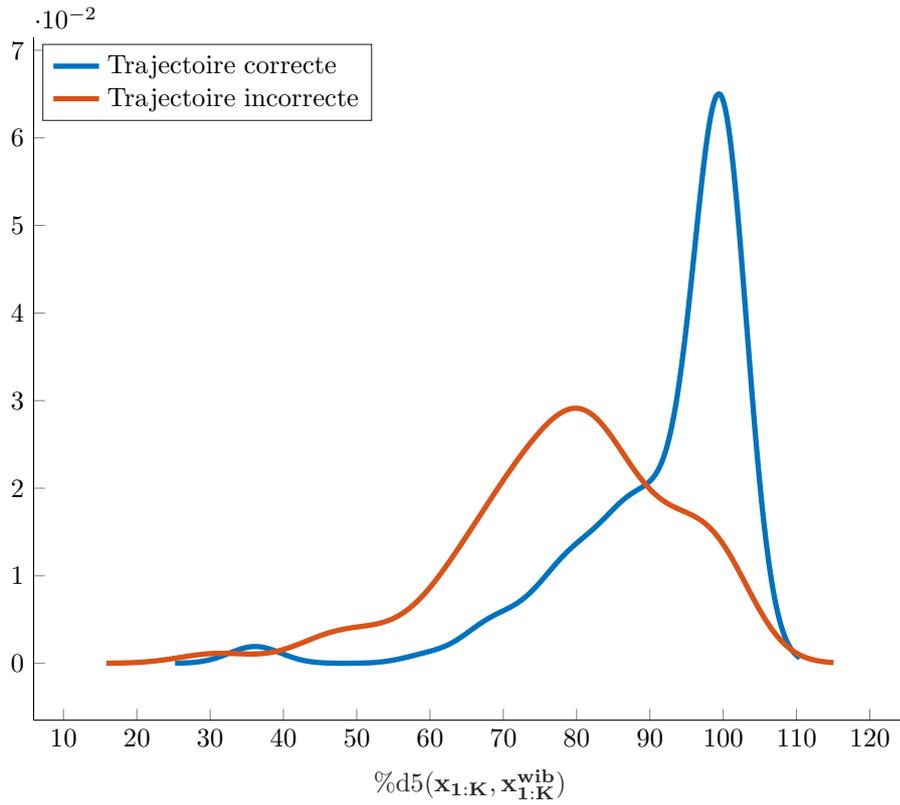


FIGURE B.6 – Distribution du ratio $\%d5$ entre l'estimation en temps différé et la trajectoire WIB selon la qualité d'estimation de la trajectoire

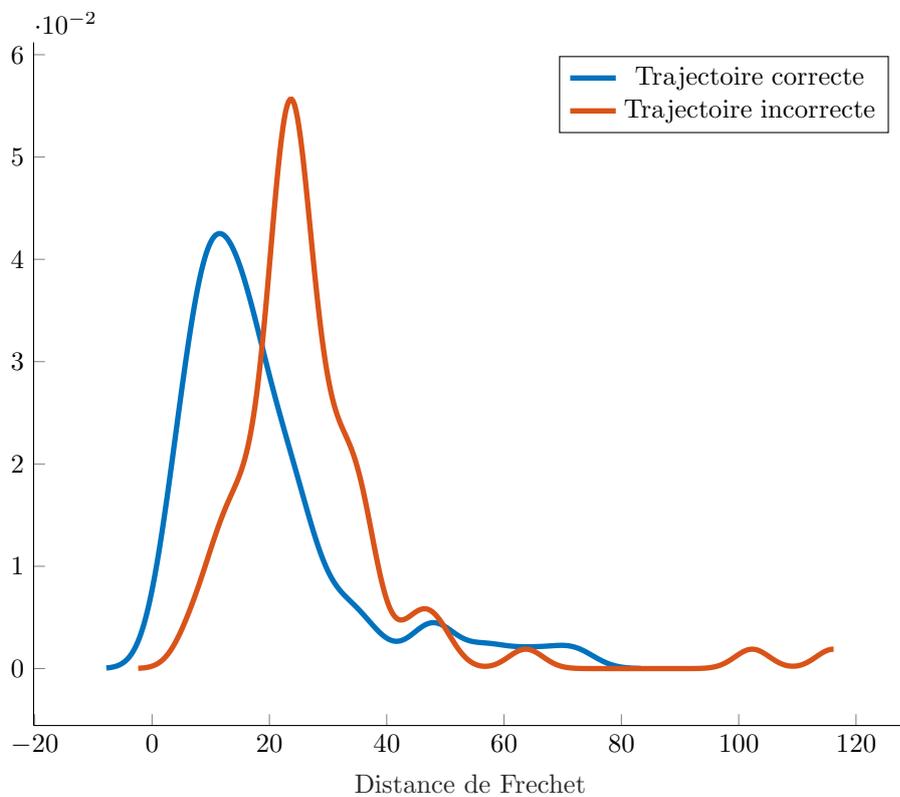


FIGURE B.7 – Distribution de la distance de Fréchet entre l'estimation en temps différé et la trajectoire WIB selon la qualité d'estimation de la trajectoire