



HAL
open science

Couplage de centrales inertielles et de solutions d'odométrie visuelle pour la navigation bas coût, précise et robuste

Gaëtan Chevrin

► **To cite this version:**

Gaëtan Chevrin. Couplage de centrales inertielles et de solutions d'odométrie visuelle pour la navigation bas coût, précise et robuste. Automatique / Robotique. Université de Haute Alsace - Mulhouse, 2023. Français. NNT : 2023MULH5546 . tel-04573044

HAL Id: tel-04573044

<https://theses.hal.science/tel-04573044>

Submitted on 13 May 2024

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.

Année 2023

N° d'ordre : (attribué par le SCD)

UNIVERSITÉ DE HAUTE-ALSACE
UNIVERSITÉ DE STRASBOURG

THESE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE HAUTE-ALSACE
ECOLE DOCTORALE : Mathématiques, Sciences de l'Information et de l'Ingénieur (ED 269)
Discipline : Automatique

Présentée et soutenue publiquement

par

Gaëtan CHEVRIN

Le 13 septembre 2023

Couplage de centrales inertielle et de solutions
d'odométrie visuelle pour la navigation bas coût,
précise et robuste.

Sous la direction du Prof. Jean-Philippe LAUFFENBURGER et la co-direction du Dr. HDR Sébastien CHANGEY

Jury : Prof. Mohammed Boutayeb, Université de Lorraine (Président)
Dr. HDR, Lydie NOUVELIÈRE Université d'Evry-Paris Saclay (Rapporteuse)
Prof. Pascal VASSEUR, Université de Picardie Jules Verne (Rapporteur)
Dr. HDR Sébastien CHANGEY, Institut Franco-Allemand de recherche de Saint-Louis (Co-directeur de thèse)
Prof. Jean-Philippe LAUFFENBURGER, Université de Haute-Alsace (Directeur de thèse)
Dr. David MONNIN, Institut Franco-Allemand de recherche de Saint-Louis (Encadrant)
Dr. Martin REBERT, Institut Franco-Allemand de recherche de Saint-Louis (invité)
Dr. Stéphane BAZEILLE, Université de Haute-Alsace (invité)

« *Le chemin avant la destination.* »

Brandon Sanderson

Résumé

Gaëtan CHEVRIN

Couplage de centrales inertielles et de solutions d'odométrie visuelle pour la navigation bas coût, précise et robuste.

La navigation visuelle Parallax Beam [103] qui équipe les système STAMINA de l'ISL (Institut Franco-Allemand de Saint-Louis) présente une précision comparable à d'autres algorithmes de navigation à l'estime visuelle de pointe comme 8-points ou l'initialisation d'ORB-SLAM. Cependant, elle est lourdement impactée par les aléas qui peuvent toucher un capteur extéroceptif comme l'occlusion du champ de vision ou les changements de contrastes soudains. De leurs côtés, les centrales inertielles sont immunisées à des aléas de ce type mais souffrent de dérives importantes qui conduisent à des imprécisions sur le long terme lorsqu'elles ne sont pas corrigées. La proposition présentée dans ce document est de compléter la navigation à l'estime visuelle Parallax Beam de STAMINA d'une centrale inertielle visant à améliorer la qualité de la navigation selon des critères de coûts, de précision et de robustesse. Ceci conduit à l'étude d'algorithmes qui appartiennent au champ large des systèmes de navigation visuels et inertiels (VINS). Une étude de ceux-ci nous a conduit à privilégier les solutions de couplage lâche, en particulier celles reposant sur des filtres de Kalman. Nous avons donc proposé et étudié une première architecture construite sur un Error-State Kalman Filter [51] que nous avons testé sur le KITTI vision benchmark [54]. Celui-ci réalise un suivi de l'odométrie visuelle satisfaisante et est en mesure d'assurer la navigation quand le Parallax Beam est en fonctionnement dégradé, simulé par un sous-échantillonnage. Cependant, la faible qualité de la navigation à l'estime inertielle associée a un impact négatif sur la qualité de la fusion proposée. En réponse à ce problème, nous proposons d'employer plutôt une architecture de type MSF-EKF (Multi Sensor Fusion Extended Kalman Filter) [92] que nous avons testé avec le Parallax Beam sur KITTI [48]. Outre une amélioration significative de la précision de la fusion, nous avons constaté une amélioration de la robustesse avec un filtre qui peut assurer une navigation faiblement dégradée pendant plusieurs secondes de coupure de la vision. Comparé avec ESKF, le MSF-EKF est plus performant et se présente comme une réponse adaptée à notre problème initial sur laquelle nous pourrions construire des fonctions supplémentaires.

Remerciements

Tout d'abord, il paraît indispensable de remercier l'Institut franco-allemand de recherche de Saint-Louis pour avoir financé cette thèse et m'avoir permis de conduire ces travaux jusqu'à ce point et au-delà. Ensuite, je tiens à remercier Mohammed Boutayeb, Lydie Nouvelière, Pascal Vasseur, Martin Rebert et Stéphane Bazeille pour avoir étudié le travail présenté dans ce document et été une source d'échanges enrichissants lors de ma soutenance. Evidemment, je remercie mon encadrement pour son accompagnement au travers de ce sujet difficile. Sébastien Changey pour son soutien et sa gestion qui nous a permis de garder le cap, Jean-Philippe Lauffenburger pour son expertise des choses académiques et David Monnin. Merci également à toutes les personnes avec qui j'ai partagé des espaces de travail : l'équipe du MIAM de l'IRIMAS, toujours accueillant malgré mes trop rares passages, ainsi que l'équipe GNC de l'ISL pour nos pauses café sous le soleil, rare source de vitamine D durant ces quatre années. En particulier, Paul Siclet, dont le stage m'a permis de poser des briques importantes pour le futur de ces travaux. Valentin Riss pour avoir été mon co-bureau durant la majorité de ma présence à GNC. Quentin Jault qui m'a apporté un soutien technique essentiel et Laetitia Deville pour sa relecture orthographique éclairante et nécessaire. Evidemment, mes parents, mon frère et tous mes proches qui constituent ma grande famille du cœur. Merci aux membres de cette grande famille qui ont pu être présent pour me soutenir, de près comme de loin.

Table des matières

Résumé	iii
Remerciements	v
Table des matières	vii
Table des figures	xi
Liste des tableaux	xix
Liste des abréviations	xxi
Liste des symboles	xxiii
1 Introduction générale	1
1.1 Contexte de recherche	1
1.2 Contexte applicatif	3
1.3 Objectifs de la thèse	4
1.4 Diffusions scientifiques et publications	5
1.5 Plan du manuscrit	6
2 La navigation	9
2.1 Introduction	9
2.2 Concepts de la navigation	9
2.2.1 Localisation	10
2.2.2 Planification de trajectoire	12
2.2.3 Cartographie	16
2.3 Outils de la navigation	20
2.3.1 Représentation d'un système	20
2.3.2 Global Navigation Satellite System	21
2.3.3 Systèmes magnéto-inertiels	23
2.3.4 Télémètres	27
2.3.5 Caméras	28
2.3.6 De l'intérêt d'associer plusieurs capteurs : le couplage	31
2.4 Évaluer la navigation	33

2.4.1	Métriques de la navigation	34
2.4.2	Jeux de données courants	35
	KITTI vision	35
	EuRoC	37
2.5	Localisation par la vision à l'ISL	38
2.5.1	Présentation des systèmes STAMINA-UGV	38
2.5.2	Parallax Beam	40
2.5.3	Limitations de STAMINA	42
2.6	Conclusion	44
3	Etat de l'art des systèmes VINS	45
3.1	Introduction	45
3.2	Méthodes de couplage lâche	45
3.2.1	Couplage lâche par filtrage de Kalman	45
	Le couplage EKF et ESKF	46
	Apport des groupes de Lie	49
	Fusion multi capteurs	51
3.2.2	Couplage lâche par filtrage particulière	52
3.2.3	Apports des méthodes d'apprentissage machine	53
3.3	Méthodes de couplage serré	59
3.3.1	Omniprésence du SLAM	59
3.3.2	Méthodes les plus performantes	63
	VINS-Mono	63
	LOVINS	64
	Kimera	66
	Résumé	66
3.4	Couplage et robustesse	67
	Robustesse contre les perturbations des signaux	67
	Robustesse contre les erreurs d'estimations	69
	Indisponibilité d'un capteur	70
	Conclusion	71
3.5	Conclusion : choix d'un couplage lâche	71
4	Fusion Inertielle-Visuelle par filtrage de Kalman à état-erreur	73
4.1	Introduction	73
4.2	Filtre ESKF	73
4.2.1	Équations de mécanisation de l'INS	73
4.2.2	Principe du filtre ESKF	75
4.3	Résultats sur le jeu de données KITTI vision	77
4.3.1	Présentation et analyse des résultats	77
4.3.2	Bilan et pistes d'améliorations	83

4.4	Navigation inertielle par IEKF	83
4.4.1	L'IEKF	83
4.4.2	Intégration d'un IEKF pour la fusion INS/VISION	85
4.4.3	Conclusion	86
5	Multi-Sensor Fusion Extended Kalman Filter	87
5.1	Introduction	87
5.2	MSF-EKF : fusion multi-capteurs par filtrage de Kalman Etendu	87
5.2.1	Principe du MSF-EKF	88
	Représentation d'états	88
	Modèle de mesure	91
5.3	Évaluation quantitative sur KITTI	93
5.3.1	Test avec le Parallax Beam sans coupures	93
5.3.2	Test de coupure	96
5.4	Conclusion	99
6	Conclusion et Perspectives	101
6.1	Conclusion	101
6.2	Étude comparative	101
	Précision	102
	Robustesse	103
	Complexité combinatoire	106
	Synthèse	107
6.3	Perspectives	107
6.3.1	De nouveaux moyens d'essais	107
6.3.2	Diagnostiquer les erreurs en amont de la fusion	109
6.4	Conclusion	109
	Bibliographie	111
A	Les filtres de Kalman	121
A.1	Filtre de Kalman Continu	121
A.2	Filtre de Kalman Discret	123
A.3	Filtre de Kalman Etendu	125
B	ESKF sur KITTI - figures supplémentaires	127
	ESKF sur la séquence 04 avec Parallax Beam à 10 Hz	128
	ESKF sur la séquence 06 avec Parallax Beam à 10 Hz	129
	ESKF sur la séquence 09 avec Parallax Beam à 10 Hz	130
	ESKF sur la séquence 10 avec Parallax Beam à 10 Hz	131
	ESKF sur la séquence 04 avec Parallax Beam à 5 Hz	132
	ESKF sur la séquence 06 avec Parallax Beam à 5 Hz	133

ESKF sur la séquence 09 avec Parallax Beam à 5 Hz	134
ESKF sur la séquence 10 avec Parallax Beam à 5 Hz	135
IEKF sur le jeu de données KITTI vision	136
C MSF-EKF sur KITTI - figures supplémentaires	137
MSF-EKF sur la séquence 04 sans coupure	138
MSF-EKF sur la séquence 09 sans coupure	139
MSF-EKF sur la séquence 10 sans coupure	140
MSF-EKF sur la séquence 04 avec coupure	141
MSF-EKF sur la séquence 09 avec coupure	142
MSF-EKF sur la séquence 10 avec coupure	143

Table des figures

1.1	Autonomous Real-Time Experimental platform of MIPS ou ARTEMIPS.	2
1.2	Le STAMINA-Aurochs.	3
2.1	La navigation et ses tâches.	10
2.2	L'objectif de la localisation : exprimer la position d'un mobile dans un repère de référence \mathbf{R}_{ref} en disposant d'informations capteur mesurées dans le repère capteur \mathbf{R}_c mouvant avec le repère corps (body) \mathbf{R}_b défini sur le centre de gravité du véhicule.	10
2.3	Les sous-tâches de la localisation : la perception fournit à la prédiction et à la correction les données nécessaires. La prédiction réalise une première estimation, que la correction met à jour.	11
2.4	À gauche, un spar islandais, utilisé pour la navigation via la polarisation de la lumière du soleil. À droite, des caméras polarisées, équivalent moderne du spar islandais.	11
2.5	Un loch à plateau, premier outil pour réaliser un dead-reckoning. La bobine est pourvue de noeuds utilisés pour mesurer la vitesse d'un navire.	12
2.6	Infographie de la société des ingénieurs en automobiles sur les différents niveaux d'autonomies d'un véhicule.	13
2.7	Schéma présentant un cas d'application de l'algorithme A*.	14
2.8	Exemple d'une navigation par point de passage.	15
2.9	Illustration de l'approche par champ de potentiel artificiel. Sur l'image (a), on peut voir la carte des potentiels exprimés. Sur l'image (b), la manière de déterminer la direction que le véhicule doit emprunter. . . .	15
2.10	Illustration de l'intérêt de la cartographie. Le robot présent sur l'image a) a défini les différents obstacles comme des amers sur sa carte en image b). Ainsi notifiées, leurs positions sont connues du robot qui pourra utiliser cette information pour sa navigation.	17
2.11	À gauche, la sonde martienne Persévérance et à droite l'aspirateur autonome Rumba i7. Deux systèmes qui utilisent des méthodes de SLAM pour la navigation.	17
2.12	Illustration du problème du SLAM : localiser un véhicule et des amers pour la cartographie dont les positions dépendent alors de la bonne localisation du véhicule.	18

2.13	Illustration du principe de fonctionnement de la technologie GNSS à 4 satellites.	22
2.14	Schéma de la structure interne d'un gyromètre laser.	24
2.15	Schéma d'une centrale inertielle : on peut constater qu'outre l'IMU composée d'accéléromètres et de gyromètres, l'INS comporte une modélisation du champ de gravité qui permet de faire fonctionner un processeur de navigation pour déduire la position, la vitesse et l'attitude des données IMU.	25
2.16	Schéma-bloc de l'algorithme de navigation à l'estime inertielle.[36]	26
2.17	Illustration du principe de fonctionnement d'un capteur de portée infrarouge.	27
2.18	Concept du balayage d'un lidar [43]	28
2.19	À gauche, un UGV équipé d'un lidar SICK et à droite un exemple d'une cartographie lidar selon le constructeur Velodyne, acteur majeur du marché du lidar.	29
2.20	Cas exemple d'une application de détection d'obstacle : un piéton est détecté et mis en évidence par une boîte l'encadrant sur l'image issue du flux vidéo.	29
2.21	Exemple du fonctionnement d'un détecteur de Harris. Si la matrice (carré bleu) autour du point étudié est inchangée dans toutes les directions, rien n'est détecté (zone plate). Si un changement de contraste est exprimé dans une seule direction, une arrête est détectée. Si un changement de contraste s'opère dans toutes les directions, alors un coin est détecté. . . .	30
2.22	Applications du détecteur de Harris à l'image de gauche. On constate la détection de nombreux coins sur l'image. Le nombre de coins détectés dépend du seuil décidé par l'expérimentateur.	31
2.23	Exemple de drone volant équipé de plusieurs capteurs pour la navigation [109]. Ici, on retrouve un capteur proprioceptif (IMU) et deux capteurs extéroceptifs le complétant (une caméra monoculaire, et un capteur de portée infrarouge).	32
2.24	Schéma illustrant le principe d'un couplage lâche : chaque capteur est traité séparément pour obtenir plusieurs estimations indépendantes, lesquelles sont ensuite couplées.	32
2.25	Schéma illustrant le principe d'un couplage serré : les données des capteurs sont directement couplées avec peu ou pas d'estimation intermédiaire.	33
2.26	Anniway, le véhicule équipé de capteurs utilisé pour la réalisation du jeu de données jeu de données KITTI vision. On peut également voir les différents repères attachés aux capteurs.	36

2.27	Vérité-terrain en 2D de la séquence de validation n° 9 du jeu de données KITTI vision. Les axes sont exprimés en mètre.	36
2.28	Les différents STAMINA-UGV. Dans le sens de lecture : le Wolf/P4, le μ Scout, l'Aurochs et le Patrol.	39
2.29	Schéma des capacités de navigation de STAMINA : navigation autonome et replanification de trajectoire en cas d'obstacle à éviter.	39
2.30	Schéma bloc du fonctionnement de l'algorithme Parallax Beam, extrait de [103]	41
2.31	Trajectoire estimée par Parallax Beam, en vert, sur la séquence de validation n° 9 du jeu de données KITTI vision. Axes en mètre.	42
2.32	Illustration des cas limites d'un algorithme de navigation à l'estime visuelle seule : l'occultation du champ de vision ou la mauvaise orientation de la caméra à cause du terrain à naviguer peuvent entraîner une indisponibilité de la fonction de navigation.	42
2.33	Séquence 9 du jeu de données KITTI vision, Parallax Beam (en vert) subit une coupure du flux d'image au bout de 20 secondes pendant respectivement 2, 5 et 10 secondes. Axes en mètres.	43
2.34	Illustration de l'objectif d'un couplage visuel-inertiel : être capable de compenser la dérive (drift) inertielle et assurer la navigation en cas d'estimation aberrante de la navigation visuelle.	44
3.1	Schéma-bloc typique d'un couplage lâche GNSS/INS, dont la structure en couplage lâche est adaptable à un couplage VINS [56].	46
3.2	Schéma-bloc d'un couplage lâche entre une IMU, une caméra monoculaire et un GNSS [25].	48
3.3	Résultats obtenus par des méthodes basées sur les groupes de Lie et un UKF testés sur six trajectoires de EuRoC [16]. Á gauche les RMSE, à droite les temps de calculs moyens pour 10 s de vol pour les étapes de prédictions et de mise à jour des filtres [12].	50
3.4	Résultats obtenus par des filtres basés sur les groupes de Lie (notées S-UKF-LG et S-IEKF) sur différentes séquences de EuRoC [16]. A gauche les RMSE, à droite les charges processeur maximales [13].	50
3.5	Illustration du ring buffer utilisé dans une MSF-EKF pour compenser les délais de mesures [92].	51
3.6	Schéma de principe de l'organisation de la tâche de localisation du mobile [120].	52
3.7	Schéma bloc de l'Adaptative Fuzzy Strong Tracking Kalman Filter [70].	53
3.8	Schéma bloc d'un IEKF augmenté par un réseau de neurones pour estimer une des matrices de covariances des bruits (notées N) [11].	54

3.9	Schéma illustrant les avantages d'une solution d'apprentissage profond pour la fusion, ici VINet (Visual-Inertial Net) [27]. En bleu, on trouve tous les paramètres qui requièrent une action de l'utilisateur du système.	55
3.10	Architecture de VINet [27] qui réalise une navigation VINS uniquement à l'aide de réseaux de neurones.	56
3.11	Résultats de VINet [27] sur une séquence de KITTI. Sont comparés VINet sans apport inertiel en bleu, VINet complet en vert et un MSF-EKF en rouge.	56
3.12	Schémas blocs (le premier pour l'apprentissage, le second pour l'utilisation en cas de perte) d'un système de couplage INS/GNSS qui utilise un LSTM pour apprendre le comportement du GNSS et le remplacer lors de coupures du signal [128].	57
3.13	Résultats d'un filtre Deep EKF [64] sur la séquence #34 de KITTI comparée à un EKF standard.	58
3.14	Exemple d'une structure de navigation incluant des outils d'évaluation du contexte afin d'améliorer les fonctions de navigation [41].	58
3.15	Historique des technologies V-SLAM divisée en 3 périodes : une époque classique des technologies SLAM, une époque de développement du V-SLAM à proprement parler (en rouge) et enfin une époque d'amélioration de la robustesse du V-SLAM et de l'introduction du VI-SLAM [67].	60
3.16	Schéma descriptif du principe du MSCKF. On voit clairement l'appariement entre les amers 3D ("3D feature") et les poses caméras ("Camera Nav State")[60].	61
3.17	Estimée du trajet réalisé dans [95] afin valider le MSCKF. On constate l'écart de l'estimée à l'arrivée (le carré rouge, symbolisant aussi le départ)[95].	62
3.18	Benchmark sur différentes plate-formes comparant plusieurs méthodes de navigation VINS dont ROVIO [7], OKVIS [84], MSCKF [95] ou encore VINS-Mono [100] par un calcul de RMSE sur KITTI [32].	64
3.19	Schéma bloc de VINS-Mono [100].	64
3.20	Schéma illustrant l'alignement métrique entre la caméra et l'IMU (trajectoire en jaune) [100].	65
3.21	Trajectoires réalisées par OKVIS et VINS-mono. On constate en particulier l'utilité de la fermeture de boucle qui réalise un recalage nécessaire à la qualité de l'estimation [100].	65
3.22	Résultats (RMSE) de LOVINS sur le jeu de données TUM VI [75] et comparé à ORB-SLAM et VINS-mono [49].	66
3.23	Résultats (RMSE) de Kimera sur le jeu de données EuRoC [16] et comparé à OKVIS, MSCKF, ROVIO et VINS-mono. On constate clairement la performance de Kimera [104].	67

3.24	Représentation de la variance d'Allan [101].	68
3.25	Position dans le plan Est-Nord lors d'une coupure GPS (gauche). Erreur de position dans ce même plan (droite) [101].	69
3.26	Détection de valeurs aberrantes (cerclée en noir) d'angles par filtre médian [119].	69
3.27	Exemple d'architecture à multiples modèles interactifs à deux modèles [126].	70
4.1	Schéma-bloc illustrant l'algorithme ESKF employé pour un couplage lâche VINS.	77
4.2	Trajectoire estimées sur la séquence 05 du jeu de données KITTI vision	78
4.3	Trajectoire estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axes du repère de référence.	79
4.4	Termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquences 05 du jeu de données KITTI vision.	79
4.5	Trajectoire estimée sur la séquence 5 du jeu de données KITTI vision. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.	81
4.6	Trajectoire estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.	81
4.7	Termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquence 05 du jeu de données KITTI vision. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.	82
4.8	Trajectoires estimées sur la séquence 10 du jeu de données KITTI vision avec un IEKF. La navigation à l'estime inertielle (Dead-Reckoning), visuelle (Parallax Beam) et la référence (benchmark) sont également affichées pour comparaison.	84
4.9	Schéma-bloc illustrant l'algorithme ESKF employé en conjonction d'un IEKF pour un couplage lâche VINS.	85
5.1	Schéma-bloc illustrant l'algorithme MSF-EKF employé sur le jeu de données KITTI vision.	93
5.2	Estimées sur la séquence 05 du jeu de données KITTI vision.	94
5.3	Estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence.	95
5.4	Trois premiers termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquence 05 du jeu de données KITTI vision.	95

5.5	Estimées sur la séquence 05 du jeu de données KITTI vision avec indisponibilité du Parallax Beam de 10 secondes	97
5.6	Estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence avec indisponibilité du Parallax Beam de 10 secondes.	97
5.7	Trois premiers termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquence 05 du jeu de données KITTI vision avec indisponibilité du Parallax Beam de 10 secondes.	98
6.1	Histogramme des erreurs relatives moyennes en translation (fois cent mètres par mètre parcouru) sur les séquences 04, 05, 06, 09 et 10 du KITTI vision benchmark.	102
6.2	Histogramme des erreurs relatives moyennes en rotation (en radians par mètre) sur les séquences 04, 05, 06, 09 et 10 du KITTI vision benchmark.	102
6.3	Histogramme des écarts de l'erreur relative moyenne en translation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour le MSF-EKF entre l'évaluation perturbées et non perturbées.	104
6.4	Histogramme des écarts de l'erreur relative moyenne en rotation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour le MSF-EKF entre l'évaluation perturbée et non perturbée.	105
6.5	Histogramme des écarts de l'erreur relative moyenne en translation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour l'ESKF entre l'évaluation perturbée et non perturbée.	105
6.6	Histogramme des temps de calcul requis pour les différentes séquences testées par l'ESKF, le MSF-EKF, le Parallax Beam et l'écart entre les méthodes ESKF et MSF-EKF+Parallax Beam.	106
6.7	Le Kamera und IMU Synchronisiertes System (KISS).	108
B.1	ESKF sur la séquence 04 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	128
B.2	ESKF sur la séquence 06 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	129
B.3	ESKF sur la séquence 9 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	130
B.4	ESKF sur la séquence 10 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	131

B.5	ESKF sur la séquence 04 avec Parallax Beam à 5 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	132
B.6	ESKF sur la séquence 06 avec Parallax Beam à 5 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	133
B.7	ESKF sur la séquence 9 avec Parallax Beam à 5 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	134
B.8	ESKF sur la séquence 10 avec Parallax Beam à 5 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	135
B.9	IEKF sur le jeu de données KITTI vision. A) séquence 04. B) séquence 05. C) séquence 06. D) séquence 9.	136
C.1	MSF-EKF sur la séquence 04 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	138
C.2	MSF-EKF sur la séquence 9 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	139
C.3	MSF-EKF sur la séquence 10 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	140
C.4	MSF-EKF sur la séquence 04 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	141
C.5	MSF-EKF sur la séquence 09 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	142
C.6	MSF-EKF sur la séquence 10 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.	143

Liste des tableaux

2.1	Différents types de cartes et leurs spécificités.[15]	19
2.2	Résultats sur le jeu de données KITTI vision de différentes méthodes de navigation à l'estime par la vision comme l'initialisation d'ORB-SLAM [96], 5-point [85], 8-point [26] et une simple homographie. En vert sont signalées les erreurs les plus faibles.	41
3.1	RMSE (m) après deux minutes de vol. Résultat des essais effectuées dans [108].	49
3.2	Résultat du filtre MSCKF adaptatif comparé au filtre MSCKF standard [35].	62
4.1	RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (ESKF). Les meilleurs résultats sont exprimés en vert.	80
4.2	RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (ESKF). Les meilleurs résultats sont exprimés en vert. Les résultats des tests réalisés avec Parallax Beam à 5 Hz sont ajoutés.	82
4.3	RMSE en translation et rotation obtenues sur les séquences 05, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB) et l'IEKF. Les meilleurs résultats sont exprimés en vert.	85
4.4	RMSE en translation et rotation obtenues sur les séquences 05, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB), l'IEKF et l'ESKF associé à l'IEKF. Les meilleurs résultats sont exprimés en vert.	86

5.1	RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (MSF-EKF). Les meilleurs résultats sont exprimés en vert.	96
5.2	RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l'estime classique (DR), l'algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (MSF-EKF) avec et sans coupure. Les meilleurs résultats sont exprimés en vert.	98
6.1	Tableau comparatif résumant l'étude relative entre l'ESKF et le MSF-EKF en précision, robustesse et rapidité d'exécution.	107

Liste des abréviations

AHRS	A ttitude and H eading R eference S ystem
ARTEMIPS	A utonomous R eal- T ime E xperimental platform of MIPS
ATE	A bsolute T rajectory E rror
AVP	A dvanced V isionics P rocessing
BDS	B ei D ou Navigation Satellite S ystem
CNN	C onvolutional N eural N etwork
EGNOS	E uropean G eostationary Navigation O verlay S ervice
EKF	E xtended K alman F ilter
ESKF	E rror S tate K alman F ilter
EuRoC	E uropean R obotic C hallenge
FOV	F ield O f V iew
GLONASS	G LO B al Navigation Satellite S ystem
GNC	G uidage, N avigation et C ontrôle
GNSS	G lobal Navigation Satellite S ystem
GPS	G lobal P ositioning S ystem
IEKF	I nvariant E xtended K alman F ilter
IMU	I nertial M easurement U nit
INS	I nertial Navigation S ystem
IRIMAS	I nstitut de R echerche en I nformatique, M athématiques, A utomatique et S ignal
ISL	F rench- G erman Research I nstitute of S aint- L ouis
IUKF	I nvariant U nscented K alman F ilter
KF	K alman F ilter
KITTI	K arlschrue I nstitute of T echnology and T oyota I nstitute
LiDAR	L ight D etection A nd R anging
LSTM	L ong S hort- T erm M emory
MAV	M icro A erial V ehicle
MEMS	M icro E lectro M echanical S ystem
MSF-EKF	M ulti S ensor F usion E xtended K alman F ilter
PB	P arallax B eam
RANSAC	R AN d om S A m ple C onsensus
RMSE	R oot M ean S quare E rror
RPE	R elative P ose E rror
SAE	S ociety of A utomotive E ngineers
SLAM	S imultaneous L ocalization A nd M apping
SONAR	S O u nd Navigation A nd R anging
UAV	U nmanned A erial V ehicle
UGV	U nmanned G round V ehicle
UKF	U nscented K alman F ilter
VINS	V isual I nertial Navigation S ystem
VO	V isual O dometry

V-SLAM **V**isual **S**imultaneous **L**ocalization **A**nd **M**apping
WAAS **W**ide **A**rea **A**ugmentation **S**ystem

Liste des symboles

Notations génériques

a	Scalaire	\mathbb{R}
\mathbf{a}	Vecteur	$M_{p,q}(\mathbb{R})$
$\mathbf{a}(i)$	i^{ieme} élément du vecteur \mathbf{a}	\mathbb{R}
\mathbf{A}	Matrice	$M_{p,q}(\mathbb{R})$
$a_{i,j}$	Élément de \mathbf{A} à la i^{ieme} ligne et j^{ieme} colonne	\mathbb{R}
\mathbf{A}_i	i^{ieme} Colonne de la matrice \mathbf{A}	$M_{p,1}(\mathbb{R})$
$\hat{\mathbf{a}}$	Estimée d'un Vecteur	$M_{p,1}(\mathbb{R})$
$\tilde{\mathbf{a}}$	Erreur du vecteur entre valeur estimée et valeur mesurée	$M_{p,1}(\mathbb{R})$
$\mathbf{a}_{n n'}$	Prédiction de a à l'instant n sachant sa valeur à l'instant n'	$M_{p,q}(\mathbb{R})$

Entités

t	temps continu	\mathbb{R}
n	temps discret	\mathbb{N}
\mathbf{x}_n	Vecteur d'état du système à l'instant n	$M_{p,1}(\mathbb{R})$
\mathbf{u}_n	Vecteur de commande du système à l'instant n	$M_{p,1}(\mathbb{R})$
\mathbf{y}_n	Vecteur de mesure d'état du système à l'instant n	$M_{p,1}(\mathbb{R})$
\mathbf{p}_n^{repere}	Vecteur position dans le repère indiqué à l'instant n	$M_{3,1}(\mathbb{R})$
\mathbf{r}_n^{repere}	Vecteur orientation dans le repère indiqué à l'instant n	$M_{3,1}(\mathbb{R})$
\mathbf{q}_n^{repere}	Quaternion d'orientation dans le repère indiqué à l'instant n	$M_{4,1}(\mathbb{R})$
$\boldsymbol{\omega}_n^{repere}$	Vecteur vitesse de rotation dans le repère indiqué à l'instant n	$M_{3,1}(\mathbb{R})$
\mathbf{a}_n^{repere}	Vecteur accélération dans le repère indiqué à l'instant n	$M_{3,1}(\mathbb{R})$
$\mathbf{b}_n^{vecteur}$	Biais appliqué au vecteur vecteur à l'instant n	$M_{3,1}(\mathbb{R})$
$\mathbf{w}_n^{vecteur}$	Bruit blanc gaussien appliqué au vecteur vecteur à l'instant n	$M_{3,1}(\mathbb{R})$
\mathbf{g}	Vecteur de gravité [0,0,g]	$M_{3,1}(\mathbb{R})$
\mathbf{t}_n^{repere}	Vecteur translation exprimé dans le repère indiqué à l'instant n	$M_{3,1}(\mathbb{R})$
\mathbf{R}_n^{repere}	Matrice de rotation instantannée dans le repère indiqué à l'instant n	$SO(3)$
$\mathbf{S}_{(A-B)}$	Matrice de passage du repère B vers le repère A	$M_{3,3}(\mathbb{R})$
\mathbf{M}	Matrice de transformation rigide	$SE(3)$
$\mathbf{C}_{R1_n}^{R2}$	Matrice de rotation entre les repères $R1$ et $R2$ à l'instant n	$M_{3,3}(\mathbb{R})$
\mathbf{I}_N	Matrice identité de taille N	$M_{N,N}(\mathbb{N})$
\mathbf{R}	Matrice de covariance du bruit de mesure	$M_{p,p}(\mathbb{R})$
\mathbf{Q}	Matrice de covariance du bruit de l'état	$M_{p,p}(\mathbb{R})$
\mathbf{F}	Matrice d'évolution du modèle d'état	$M_{p,p}(\mathbb{R})$
\mathbf{G}	Matrice de commande	$M_{p,q}(\mathbb{R})$
\mathbf{H}	Matrice de mesure ou matrice d'observation	$M_{p,q}(\mathbb{R})$

Opérateurs

\mathbf{A}^T	Transposée de la matrice \mathbf{A}
\mathbf{A}^{-1}	Inverse de la matrice \mathbf{A}
$[\mathbf{A} \mid \mathbf{B}]$	Matrice définie par blocs avec les matrices \mathbf{A} et \mathbf{B}
$S(\mathbf{a})$	Matrice antisymétrique construite à partir du vecteur \mathbf{a}
$\text{diag}(\mathbf{A})$	Diagonales de la matrice \mathbf{A}
\cdot	Produit scalaire ou matriciel
\otimes	Produit de quaternion

Dédicace à Fay qui a accepté de
ne pas trop marcher sur le clavier
pendant la rédaction de ce
document.

1 Introduction générale

1.1 Contexte de recherche

Ce document présente les travaux accomplis dans le cadre de la thèse de doctorat que j'ai préparée sur la thématique de la fusion de capteurs visuels et inertiels pour la navigation. Cette thèse est effectuée en collaboration entre l'Université de Haute-Alsace (UHA) au travers de son Institut de Recherches en Informatique, Mathématiques, Automatique et Signal (IRIMAS, UR 7499), Guidage, Navigation et Contrôle (GNC) et Advanced Visionics Processing (AVP) de l'Institut franco-allemand de recherches de Saint-Louis (ISL).

Cette thèse, liée à l'école doctorale 269 (Mathématiques, Sciences de l'Information et de l'Ingénieur) de l'Unistra, est supervisée par l'équipe suivante :

Directeur : Pr. Jean-Philippe Lauffenburger, UHA-IRIMAS.

Co-Directeur : Dr.HDR Sébastien Changey, ISL-GNC

Encadrant : Dr. David Monnin, ISL-AVP

L'IRIMAS comprend trois grands domaines de recherches structurés autour de 3 départements :

- Informatique,
- Mathématiques,
- Automatique, Signal, Image.

La thèse est associée à cette dernière thématique. L'IRIMAS travaille activement l'autonomie des systèmes au travers de la commande, de la perception et de la fusion de données. Dans ce cadre, elle développe et entretient plusieurs plateformes instrumentées pour la navigation et la conduite autonome comme l'ARTEMIPS (cf. figure 1.1).

L'ISL est un centre de recherches bi-national commun à l'Allemagne et la France. Il est co-financé par les ministères de la défense allemand et français. De ce fait, la recherche qui y est menée se concentre sur les applications militaires et de sécurité. Les activités de recherches sont très variées et divisées en quatre grandes divisions :

- **Matériaux de protection avancés et énergétiques.** Ceci inclut la conception et la gestion des matériaux énergétiques qui composent les carburants propulsifs et les explosifs. Leurs activités comprennent l'étude jusqu'à l'échelle nanométrique de ces matériaux, la pyrotechnie et l'élaboration et le test de matériaux de protection avancée.



FIGURE 1.1 – Autonomous Real-Time Experimental platform of MIPS ou ARTEMIPS.

- **Techniques de vols pour projectiles.** Le groupe GNC appartient à cette division. Celle-ci étudie tout les aspects inhérents à la conception d'un projectile opérationnel, comme l'aérodynamique sur terrain d'essais et en soufflerie, la télémétrie, la communication et le guidage, navigation et contrôle. En particulier, le groupe GNC dispose d'une expertise dans la navigation magnéto-inertielle puisque couramment utilisée dans un projectile.
- **Laser et technologies électromagnétiques.** Cette division développe de nouvelles sources laser, des semi-conducteurs de puissance et les technologies d'impulsions électriques de forte puissance. En outre, elle est en charge d'un projet de développement d'un canon à rail.
- **Technologie de protection, sécurité, perception de la situation.** Le groupe AVP appartient à cette division. Celle-ci comprend les technologies de protections

balistiques et explosives, les protections accoustiques, la vision assistée par ordinateur et les interactions rayonnement-matière.

C'est au sein des groupes GNC et AVP que la thèse prend place.

1.2 Contexte applicatif

Les véhicules autonomes terrestres sont un sujet qui concentre l'attention d'une partie de la communauté scientifique, aussi bien pour des applications militaires (déminages, reconnaissance, transport de matériels) que civiles (transports de biens, de personnes, agriculture, etc.). À cet égard, ISL-AVP a développé une solution de navigation par la vision. Équipant la gamme des véhicules autonomes STAMINA-UGV ("Unmanned Ground Vehicle") de l'ISL, la brique technologique de navigation autonome par la vision STAMINA, est capable sans GNSS ("Global Navigation Satellite Systems") et en utilisant seulement une caméra monoculaire de :

- mémoriser en temps réel de nouveaux itinéraires dès qu'ils sont parcourus,
- réduire l'information image d'un itinéraire au minimum nécessaire pour permettre la transmission à d'autres véhicules,
- suivre un itinéraire en autonomie en avant et arrière avec évitement d'obstacles.

Ce système est installé sur plusieurs UGV, notamment le STAMINA-Aurochs, sujet principal des développements et visible sur la figure 1.2.



FIGURE 1.2 – Le STAMINA-Aurochs.

Les objectifs de STAMINA sont réalisés par une fonction de navigation à l'estime

par la vision avancée reposant sur une caméra monoculaire. La forme la plus récente de cette brique technologique est l'algorithme "Parallax Beam" (décrit en détails au chapitre 2). Pour ce faire, il utilise comme entrée deux images successives de la même scène permettant ainsi l'observation d'un léger mouvement relatif, lequel est par la suite identifié par plusieurs procédés de descriptions et de détections.

Parallax Beam propose une navigation à l'estime précise et efficace (de l'ordre de 2% d'erreur quadratique moyenne en translation) avec un champ de vision dégagé. Cependant, en se reposant sur une unique caméra, cette solution de navigation se heurte aux problèmes propres à ce capteur extéroceptif. Ainsi, les problèmes d'indisponibilité de l'image par occultation partielle ou totale du champ de vision (feuillages, obstacles importants et très proches, changements rapides de luminosité, sable, orientation non conforme de la caméra due aux aléas du terrain, etc.), peuvent dégrader la fiabilité du système voire entraîner une incapacité totale à réaliser la tâche de navigation. Sans intention de modifier Parallax Beam, une solution est alors de coupler la caméra à un second capteur. La piste d'un capteur proprioceptif semblant être la plus à même de répondre au besoin exprimé, la collaboration avec ISL-GNC, qui dispose d'une expertise en ce sens pour la navigation de projectiles, apparaît alors comme une évidence. Le choix a donc été fait d'intégrer dans le dispositif une centrale inertielle (IMU, "Inertial Measurement Unit") afin de procéder au couplage d'une navigation visuelle et inertielle.

La fusion Vision/IMU, champ de recherche étudié par l'IRIMAS, constitue ainsi une solution adaptée à la problématique introduite qui est d'améliorer la navigation à l'estime visuelle STAMINA lorsqu'elle est confrontée à des situations d'estimation erronée ou impossible en raison de l'indisponibilité d'images de qualité suffisante. En effet, la nature proprioceptive de l'IMU fondée sur ses accéléromètres et gyromètres permet la navigation à l'estime dans les cas limites pour la vision.

1.3 Objectifs de la thèse

L'objectif du sujet de thèse est donc de compléter la navigation à l'estime visuelle Parallax Beam de STAMINA par une centrale inertielle visant à améliorer la qualité de la navigation et en particulier :

- préserver la précision de Parallax Beam en toutes circonstances. L'erreur quadratique moyenne minimale de la fusion doit être du même ordre de grandeur que celle de Parallax Beam,
- ne pas modifier Parallax Beam. Parallax Beam doit être utilisé comme une "boîte grise" qui reçoit des images en entrée et fournit une estimation du déplacement en sortie,

- garantir la navigation en toute circonstance. Il faut proposer une solution robuste face à un défaut d'un capteur,
- être adaptable à tout type de mobile, de l'UGV au projectile.

1.4 Diffusions scientifiques et publications

Afin de répondre à la problématique posée, la solution du couplage lâche a été privilégiée, celle-ci permettant de disposer des estimations issues d'au moins l'un des capteurs quand l'autre est en défaut.

Une première solution basée sur un filtre "Error-State Kalman Filter" (ESKF) a été proposée et présentée aux 9^{me} (poster), 10^{me} (présentation) éditions du Budding Science de l'ISL ainsi qu'aux Journées STP et Automatiques de la SAGIP de 2021 sous le titre "Couplage lâche Vision/INS pour navigation bas-coût, robuste et précise de véhicules autonomes". Par la suite, une deuxième contribution a concerné une architecture "Multi-Sensor Fusion Extended Kalman Filter" (MSF-EKF) que nous avons présentée et discutée lors des 10^{me} (présentation) et 11^{me} (présentation) éditions du Budding Science de l'ISL ainsi que durant les journées d'Axes 2022 de l'Institut.

Une comparaison directe entre les deux méthodes a été proposée dans une communication pour le GRETSI 2022, à Nancy [51] tandis que l'architecture MSF-EKF employée et évaluée sur KITTI a été présentée à IEEE-ICARCV 2022 à Singapour [48]. Enfin, la plateforme d'acquisition multi-capteurs pour la navigation KISS, développée durant la thèse, a été présentée aux 6^{me} Journées des Démonstrateurs en Automatique, à Angers [52].

Liste des publications :

- Chevrin Gaëtan **and others**. « Visual-Inertial Fusion on KITTI using MSF-EKF ». *in 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*: IEEE. 2022, pages 35–40.
- Chevrin Gaëtan **and others**. « La plateforme KISS (Kamera und IMU Synchronisiertes Systems) ». *in Angers: 6^{me} Journées des Démonstrateurs en Automatique, 2022*.
- Chevrin Gaëtan **and others**. « Couplage Visuel Inertiel pour une navigation précise et robuste ». *in 28ième Colloque sur le traitement du signal et des images*: 001-0175. Nancy: GRETSI - Groupe de Recherche en Traitement du Signal et des Images, 2022, p. 701–704.

1.5 Plan du manuscrit

Le présent manuscrit, intitulé **Couplage de centrales inertielles et de solutions d'odométrie visuelle pour la navigation bas coût, précise et robuste** couvre les sujets précédemment introduits selon la structure suivante.

Tout d'abord, le chapitre 2 introduit la navigation, les différentes tâches qui la constituent, les types de capteurs employés pour mener ces tâches à bien et les manières d'en évaluer la qualité. Cette partie est conclue par une présentation des plateformes et algorithmes employés par l'ISL, lesquels nous permettent d'évaluer les besoins en matière de solution de navigation et notamment la mise en place d'un système visuel-inertiel.

Ceci nous amène donc, dans le chapitre 3, à considérer l'état de l'art des systèmes de navigations visuel-inertiel ou VINS. Découpé en 3 parties, ce chapitre aborde tour à tour les méthodes VINS en fonction de la nature de leur couplage, lâche ou serré, avant d'étudier la question de la robustesse au sein de ces différentes architectures. Cette partie se termine sur la détermination d'un choix de couplage adapté compte tenu de notre problématique : le couplage lâche.

Le chapitre 4 va alors présenter la première architecture étudiée, celle de l'ESKF. Découpé en deux parties, ce chapitre développe les équations nécessaires à l'implémentation du filtre avant d'en montrer les performances sur le jeu de données KITTI [54] et d'en discuter les qualités et limites, notamment l'impact d'une navigation inertielle de faible qualité sur la qualité de la fusion proposée. Ce chapitre se termine par la mise en oeuvre d'un estimateur IEKF pour palier cette limitation. La démarche consiste à réaliser la navigation inertielle en amont de la fusion en lieu et place d'une navigation à l'estime. On montre alors les limites de notre couplage face à l'intégration de la complexité supplémentaire inhérente à un tel filtre.

Afin de disposer d'une solution de fusion plus adaptable et performante, nous avons étudié le MSF-EKF, sujet du chapitre 5. Dans ce dernier, nous présentons dans une première partie les développements théoriques nécessaires à l'implémentation du filtre. Ensuite, nous menons son évaluation sur le jeu de données KITTI dans une seconde partie avec notamment un test de robustesse face à une coupure prolongée d'informations visuelles. Cette partie se termine par une mise en évidence des qualités et limites de notre solution, notamment le besoin d'être en mesure de déterminer quand l'estimation visuelle commet une erreur en amont du couplage.

Ceci nous conduit alors à un dernier chapitre, le chapitre 6, dans lequel nous comparons nos deux solutions présentées sur des critères de précision, robustesse face aux aléas de la vision et rapidité d'exécution dans une perspective de temps réel. Ce chapitre nous permet de mettre en avant les qualités de la solution proposée à l'issue de nos travaux dans le cadre de cette thèse. Nous terminons celui-ci par une discussion sur les différents axes d'amélioration possibles, notamment la nécessité de disposer de jeux de données mieux adaptés à notre problématique mais aussi le besoin d'être en capacité de réaliser des diagnostics d'erreurs pré-couplage en vue d'une amélioration de la robustesse

face à des aléas sur la vision.

Ce document inclut également un appendice, l'appendice **A** qui présente les principes de base des filtres de Kalman continus, discrets et étendus qui servent de bases à la plupart des travaux présentés ici.

2 La navigation

2.1 Introduction

Avant de s'intéresser à un problème de navigation, il convient de comprendre ce qu'est la navigation. Cette partie en présente les grandes lignes et les outils qui sont utilisés pour la mettre en oeuvre, principalement par le biais des moyens de perception nécessaires. Ensuite, après avoir expliqué les moyens utilisés pour évaluer la navigation, cette partie présente le jeu de données qui sera utilisé pour réaliser nos évaluations : KITTI vision. Cela étant fait, nous nous intéresserons au sujet de notre problème de navigation : le système STAMINA et l'algorithme Parallax Beam, afin de montrer l'intérêt de l'étude d'un couplage visuel-inertiel pour STAMINA.

2.2 Concepts de la navigation

Le dictionnaire "le Robert" définit la navigation comme "le repérage et calcul d'un itinéraire après détermination du point où l'on se trouve". En automatique, la navigation représente la capacité d'un système à acquérir avec précision sa position et son orientation dans un repère de référence, afin de réaliser une tâche de planification de trajet. Dans le contexte des véhicules autonomes, où l'on souhaite réduire au maximum la supervision humaine, cette tâche de navigation revient de plus en plus aux systèmes, réduisant d'autant le rôle des pilotes, déjà souvent exécutants selon les indications des systèmes de navigation.

Afin de pouvoir naviguer au sein de son environnement, un véhicule a besoin de (figure 2.1) :

- se localiser dans son repère de référence,
- cartographier son environnement pour mieux l'interpréter,
- planifier son trajet jusqu'à objectif.

La navigation est ensuite associée à la commande du véhicule afin de réaliser le trajet ainsi planifié. Les différentes tâches de navigation seront ensuite réactualisées tout au long du trajet en cours.

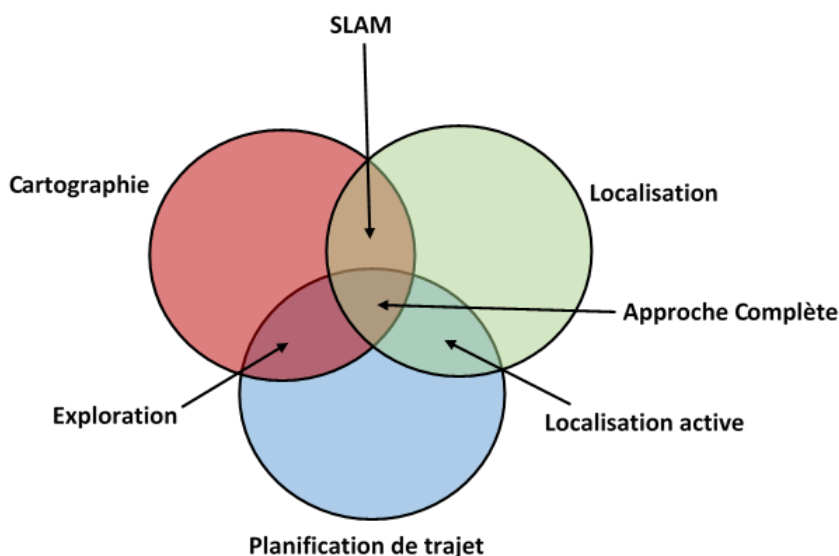


FIGURE 2.1 – La navigation et ses tâches.

2.2.1 Localisation

L'acte de localisation consiste, pour un système, à déterminer sa position absolue \mathbf{p}_t à l'instant t dans un repère de référence \mathbf{R}_{ref} comme dans la figure 2.2.

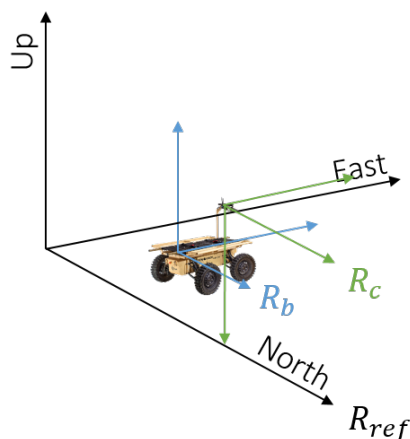


FIGURE 2.2 – L'objectif de la localisation : exprimer la position d'un mobile dans un repère de référence \mathbf{R}_{ref} en disposant d'informations capteur mesurées dans le repère capteur \mathbf{R}_c mouvant avec le repère corps (body) \mathbf{R}_b défini sur le centre de gravité du véhicule.

Une localisation parfaite est atteinte dans l'hypothèse d'une connaissance parfaite de l'environnement du système et des mouvements parfaits des actionneurs vis-à-vis de leurs modèles mécaniques. Ce n'est pas le cas. La connaissance parfaite de notre environnement n'est presque jamais disponible tandis que l'hypothèse d'actionneurs parfaits n'est elle pas plus réaliste.

Pour répondre à cette problématique, la tâche de localisation se décompose en plusieurs

sous-tâches : perception, prédiction et correction tel que représentée sur la figure 2.3.

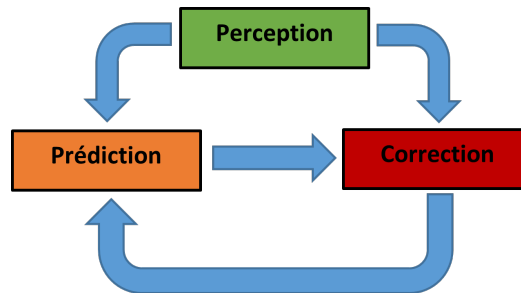


FIGURE 2.3 – Les sous-tâches de la localisation : la perception fournit à la prédiction et à la correction les données nécessaires. La prédiction réalise une première estimation, que la correction met à jour.

La perception est la tâche majeure de la localisation, c'est elle qui fournit à l'opérateur, humain ou machine, les informations nécessaires aux autres fonctions de localisation. Lorsque la tâche de perception incombe à un être humain, celui-ci dispose de plusieurs solutions. L'opérateur peut utiliser des points de repère dans son environnement. Par le passé, l'utilisation des outils comme un sextant ou un spar islandais (figure 2.4) permettait de se réperer avec le soleil et d'autres astres alors que la boussole permettait de prendre connaissance de son orientation par rapport à la direction du pôle nord. Plus récemment, les radars ont permis aux marins de rester informés de la proximité et de la forme des côtes. De nos jours, les méthodes de localisation par Global Navigation Satellite Systems (GNSS), comme le GPS (Global Positioning System) ou Galileo, sont majoritairement utilisées pour la navigation par l'humain, du fait de leurs marges d'erreur très faibles d'environ 20m [36], pour des technologies d'un coût accessible.



FIGURE 2.4 – À gauche, un spar islandais, utilisé pour la navigation via la polarisation de la lumière du soleil. À droite, des caméras polarisées, équivalent moderne du spar islandais.

Un système réalisant la perception de façon autonome dispose bien entendu d'outils similaires dont les fonctions seront détaillées dans la section 2.3 : centrales inertielles, lidar, GNSS et caméras constituent un large panel d'outils pouvant être utilisés afin d'aider un système à percevoir son environnement.

La prédiction est l'étape qui exploite les informations de perception afin d'estimer une

localisation. En fonction des moyens de perception utilisés, cette localisation sera plus ou moins précise. La Marine a longtemps utilisé la navigation à l'estime, ou Dead-Reckoning (DR)[56], afin de se localiser. Cette méthode consiste à prédire sa position à partir uniquement de sa vitesse, mesurée en noeuds (via un loch comme en figure 2.5), et son cap. Des méthodes modernes de navigation à l'estime existent et seront présentées à la section 2.3.



FIGURE 2.5 – Un loch à plateau, premier outil pour réaliser un dead-reckoning. La bobine est pourvue de noeuds utilisés pour mesurer la vitesse d'un navire.

Selon la précision de localisation souhaitée, et les moyens de perception disponibles, une étape de correction peut alors être engagée. Celle-ci utilise soit un nouveau moyen de perception ou une deuxième perception avec le même capteur pour ajuster la prédiction et ainsi assurer une meilleure localisation. L'usage d'une carte pour comparer des points de repères et affiner sa localisation est un exemple de correction. La navigation à l'estime des marins est aujourd'hui largement corrigée pour tenir compte de la dérive, des effets du courant et des erreurs de direction pour fournir une position estimée aussi précise que possible grâce une centrale INS (Inertial Navigation System). La correction est souvent un élément central des techniques de couplages modernes (section 2.3.6). À l'issue de l'ensemble de cette démarche de perception-prédiction-correction, un système est en mesure de proposer une position estimée $\hat{\mathbf{p}}_t$ à l'instant t .

2.2.2 Planification de trajectoire

Le rôle de la planification de trajectoire est de déterminer le trajet à accomplir entre la position actuelle d'un véhicule et celle que l'on souhaite atteindre. Cette tâche inclut une sous-tâche importante qui est l'évitement d'obstacle, raison pour laquelle la planification de trajectoire doit être établie à chaque instant discret de rafraîchissement des capteurs n , ou a minima dès que les capteurs détectent un nouvel obstacle dans l'environnement du véhicule. De plus la planification, qui permet d'établir une trajectoire de A vers B dans un espace restreint et immédiat au mobile, permet aussi d'établir un trajet. En effet, un trajet peut être défini assez simplement comme une succession

continue de trajectoires. Pour cette raison, nous ne distinguerons pas trajet et trajectoire dans le reste de cette partie.

La tâche de planification est la tâche de la navigation la plus difficile à réaliser pour un système autonome, puisqu'elle implique la prise d'une décision et constitue la difficulté majeure pour l'amélioration de l'autonomie des véhicules (comme expliqué dans la figure 2.6). Cela justifie la nécessité pour nos véhicules non guidés sur des rails d'être pilotés par des humains qui réalisent cette tâche de planification avec une difficulté moindre.

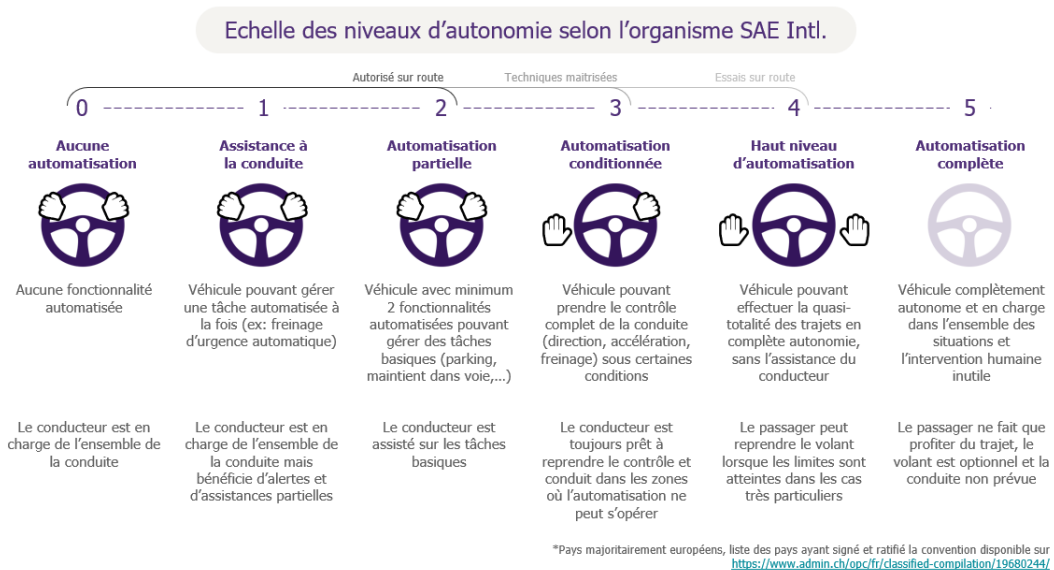


FIGURE 2.6 – Infographie de la société des ingénieurs en automobiles sur les différents niveaux d'autonomies d'un véhicule.

Le premier cas d'application d'une planification de trajectoire fut réalisé sur le robot Shakey¹. Développé par le Stanford Research Institute entre 1966 et 1972, il devait naviguer entre plusieurs salles reliées entre elles par des rampes. Disposant d'une cartographie complète indexant les objets présents dans son environnement, il pouvait traiter des consignes complexes comme "Vas dans la pièce D et pousse le bloc 9 jusqu'à l'entrée 4." via un système d'interprétation appelé STRIPS ("Stanford Research Institute Problem Solver").

Afin de déterminer le chemin à emprunter, Shakey planifiait sa trajectoire sous la forme de points de passages ("way points") tout en évitant les obstacles et en essayant de minimiser la longueur du trajet. L'algorithme A* [58] fut développé à cette fin. Cet algorithme recherche le chemin le plus court parmi un graphe. Établi à partir d'une cartographie, ce graphe comporte différents points (appelés noeuds) qui sont les lieux d'intérêts de l'environnement (la pièce D, le bloc 9 ou l'entrée 4 dans la consigne plus haut). Chacun de ces noeuds est relié aux autres noeuds les plus proches par un segment

1. <https://www.computerhistory.org/revolution/artificial-intelligence-robotics/13/289>

qui indique la distance entre deux noeuds. Cet ensemble noeud+segment est ce qu'on appelle ici un graphe (voir figure 2.7).

L'algorithme A* va effectuer la recherche du trajet le plus court entre son noeud de départ et le noeud d'arrivée. Pour cela, il va essayer de trouver le minimum de la fonction de coût $f(n) = g(n) + h(n)$, avec $g(n)$ la fonction du coût calculée depuis le point de départ jusqu'au noeud n et $h(n)$ la fonction heuristique qui estime le chemin au plus faible coût entre le noeud n et l'arrivée. Le coût exprimé étant pour une application de navigation la distance à parcourir, l'algorithme A* va calculer la fonction de coût $f(n)$ de chaque noeud en choisissant toujours de traiter en priorité les noeuds rattachés au noeud de coût le plus faible, jusqu'à parvenir à déterminer un chemin minimisant le coût total.

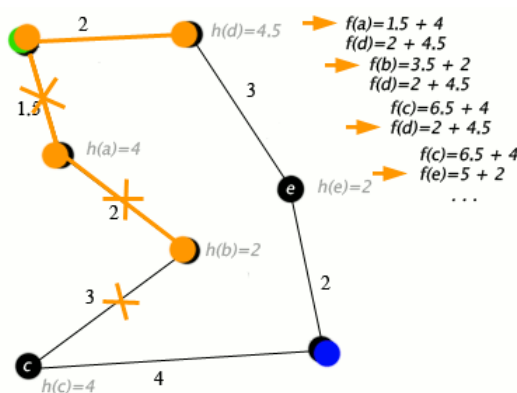


FIGURE 2.7 – Schéma présentant un cas d'application de l'algorithme A*.

Suivant la tendance portée par Shakey et l'algorithme A*, les méthodes basées sur des graphes composent une part importante des méthodes de planification de trajectoire. Les diagrammes de Voronoi forment par exemple une méthode de navigation dans un espace cartographié, permettant l'évitement d'obstacles en divisant l'espace en cellules dont les arêtes sont équidistantes de tout obstacles. Les algorithmes de parcours de graphes comme A* ou Dijkstra [34] permettent alors de déterminer le chemin le plus court au sein d'un tel diagramme. Cette approche par points de passage est aujourd'hui largement utilisée pour les auto-pilotes de véhicules aériens autonomes (ou Unmanned Aerial Vehicles, UAVs) comme sur l'exemple en figure 2.8.

Une dernière méthode classique est l'approche par champ de potentiel artificiel (APF), dont un schéma explique le principe en figure 2.9. Présentée pour la première fois en 1986 [74], elle agit comme si les obstacles et l'arrivée sont des surfaces chargées électriquement, qui forment donc une carte de potentiels induisant une force au robot qui doit naviguer. L'arrivée générant une force de potentielle attractive et les obstacles une force répulsive, le chemin du véhicule se dessine alors en suivant le gradient négatif pour atteindre l'arrivée en évitant les obstacles. Cette technique a été suivie par de nombreux travaux visant à empêcher les minima locaux, permettre la navigation temps réel pour enfin parvenir à des résultats concluants dans des environnements à trois dimensions



FIGURE 2.8 – Exemple d'une navigation par point de passage.

comme l'air [20] ou l'eau.

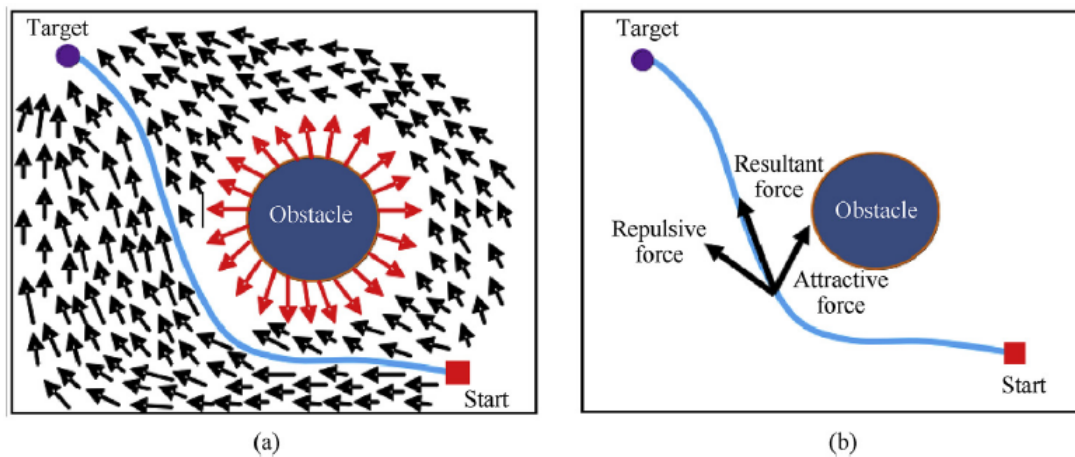


FIGURE 2.9 – Illustration de l'approche par champ de potentiel artificiel. Sur l'image (a), on peut voir la carte des potentiels exprimés. Sur l'image (b), la manière de déterminer la direction que le véhicule doit emprunter.

Cependant, depuis le début des années 90, on constate une augmentation croissante de l'usage de méthodes basées sur l'intelligence artificielle pour réaliser cette tâche de planification du trajet [99]. Outre les applications à base de réseaux de neurones et d'apprentissage profond, plusieurs méthodes présentent un intérêt potentiel, comme les

algorithmes génétiques ou les méthodes basées sur des essaims comme l'optimisation de colonie de fourmis (Ant Colony Optimization (ACO) [125]) ou l'optimisation d'essaim de particules (Particle Swarm Optimization (PSO) [37]). Il s'agit de méthodes bio-inspirées qui utilisent plusieurs agents pour explorer un espace virtuel afin d'en déterminer le trajet le plus court à emprunter pour le robot.

Enfin, on peut citer la logique floue [127], qui utilise la règle du "Si-Alors" et ses équivalents mathématiques afin de caractériser les performances d'un système, ici les performances d'une trajectoire ("Si la trajectoire rencontre un obstacle, alors elle est exclue"). La logique floue a été utilisée pour des méthodes basées perception comme le suivi de mur via sonar [18], pour le renforcement d'algorithmes de localisation afin de réduire les incertitudes ou en combinaison des méthodes mentionnées au paragraphe précédent afin de minimiser les erreurs d'apprentissages au fil des itérations. Cette technique se trouve de plus en plus rarement dans les applications récentes.

La plupart des méthodes de planification de trajet présentées ici ne fonctionnent qu'avec une connaissance a priori de l'environnement du robot. Une cartographie de son environnement est alors nécessaire.

2.2.3 Cartographie

La tâche de cartographie fournit les informations nécessaires à la planification de trajets, tout en permettant à la tâche de localisation d'obtenir des informations a priori utiles pour son accomplissement, comme sur la figure 2.10. Bien que cette tâche soit parfois exclue des applications à cause du coût en calcul pour maintenir et mettre à jour une carte, celui-ci devient de plus en plus faible et rend cette tâche de plus en plus présente dans les applications de navigation.

Il existe un grand nombre de types de cartes différentes, mais elles peuvent être classées en deux familles : les cartes simples d'interprétation pour l'humain mais qui sont alors difficiles à construire et à interpréter via des données capteurs, et au contraire les cartes conçues à partir de données capteurs et donc plus difficiles à comprendre pour un être humain.

Afin d'évaluer un type de carte, on peut définir plusieurs critères :

- Ce que représente la carte,
- si elle est métrique dense (les obstacles sont pleins sur la carte), métrique creuse (seules les arrêtes des obstacles seront visibles) ou non métrique (graphe),
- si elle permet la planification de trajet ou non,
- si elle est facile à construire à partir de données capteurs,
- la difficulté d'une augmentation d'échelle.

L'ensemble de ces critères permet de réaliser un tableau de classification des types de cartes (Tab. 2.1).

De telles cartes peuvent être construites de différentes manières. Pour un graphe des poses, la pose du véhicule mesurée à chaque instant et le déplacement relatif entre



FIGURE 2.10 – Illustration de l'intérêt de la cartographie. Le robot présent sur l'image a) a défini les différents obstacles comme des amers sur sa carte en image b). Ainsi notifiées, leurs positions sont connues du robot qui pourra utiliser cette information pour sa navigation.

chaque pose suffit à sa réalisation. Dans le cas d'un nuage de points, d'une carte d'amers ou d'une grille d'occupation, un scan de l'environnement est réalisé via différents types de capteurs (lidar, caméra, capteur de portée) et la position connue du capteur permet alors d'établir une cartographie. Dans tous les cas, il s'agit d'accumuler des données capteurs traduites sur une carte. La dépendance à la tâche de localisation devient alors évidente puisque conditionnant la bonne réalisation de la carte.

Dans un contexte de navigation autonome, comme sur les UGVs (Unmanned Ground Vehicles, pour "véhicules terrestres sans pilote") de la figure 2.11, la connaissance a priori de l'environnement et l'existence d'une carte appropriée n'est plus toujours garantie. Une solution courante est alors de réaliser la tâche de localisation en parallèle de celle de cartographie. Cette méthode, appelée SLAM (Simultaneous Localization And Mapping pour "cartographie et localisation simultanées") est sûrement la méthode la plus utilisée pour l'exploration d'espace non cartographié [9]. On la trouve par exemple à bord des sondes martiennes ou de nos aspirateurs autonomes.



FIGURE 2.11 – À gauche, la sonde martienne Persévérance et à droite l'aspirateur autonome Rumba i7. Deux systèmes qui utilisent des méthodes de SLAM pour la navigation.

Comme son nom le suggère, cette technique permet à un robot de construire de manière incrémentale une carte de son environnement tout en étant en mesure de se localiser en temps réel au sein de celle-ci. Pour ce faire, la méthode SLAM (illustrée figure 2.12) doit réaliser en simultané l'estimation de plusieurs positions : d'un côté la position du robot, et de l'autre les positions des différents amers disponibles. Autrefois réalisé à l'aide de capteurs de portée infrarouge ou de lidar, les méthodes actuelles de SLAM sont plus souvent réalisées en conjonction d'une caméra. On les trouve sous l'appellation V-SLAM (Visual-SLAM) [67].

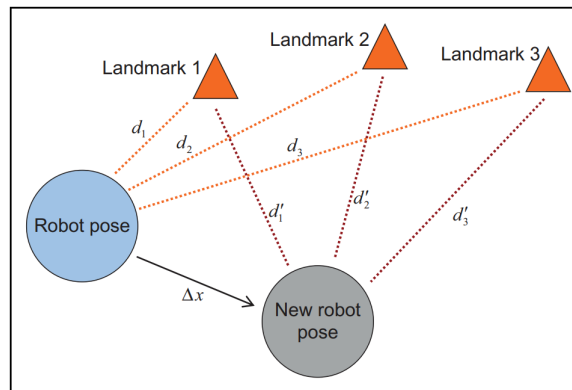


FIGURE 2.12 – Illustration du problème du SLAM : localiser un véhicule et des amers pour la cartographie dont les positions dépendent alors de la bonne localisation du véhicule.

La cartographie est ainsi une tâche difficile à réaliser sans une localisation efficace. Chaque tâche de navigation dépendant de la tâche de localisation, c'est donc cette dernière qui est la part essentielle de la navigation.

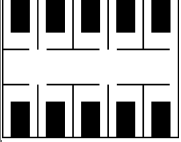

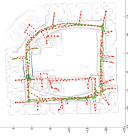
Type de carte	Représentation	Planification de trajet	Construction avec capteurs	Image
Géométrique	Obstacles	Facile	Difficile	
Amers	Éléments saillants	Très Difficile	Facile	
Nuage de points	Surfaces par échantillonnage	Difficile	Facile	
Grille d'occupations	Espace libre, occupé ou inconnu	Facile	Bonne	
Graphe de Pose	Liste des poses des robots	Difficile	Facile	
Topologique	graphe de zones	Moyenne	Difficile	
Élévation	Hauteur du sol	Facile	Bonne	

TABLE 2.1 – Différents types de cartes et leurs spécificités. [15]

2.3 Outils de la navigation

Toutes les tâches de la navigation reposent sur la capacité à disposer d'informations sur le véhicule et son environnement. Ainsi, les capteurs, qui recueillent des données physiques, sont les principaux outils de la navigation. C'est en manipulant les données issues des capteurs que l'on est en mesure de déterminer l'état de notre véhicule. Il existe plusieurs grandes familles de capteurs qui disposent de leurs avantages et inconvénients et de leurs propres méthodes d'exploitation pour réaliser la tâche de navigation. Ces données sont souvent fusionnées ou couplées afin que les complémentarités des domaines d'observation des différents capteurs nous permettent de meilleurs résultats qu'avec un capteur unique.

2.3.1 Représentation d'un système

La modélisation de systèmes dynamiques réels, nécessaire pour la navigation, repose essentiellement sur la notion mathématique de processus stochastique (ou processus aléatoire). En effet, de nombreux phénomènes physiques appliqués à un système ne sont pas ou mal connus (perturbations diverses comme le bruit, vent, une panne, etc.) alors qu'ils peuvent avoir une influence significative sur le comportement du système. Ces phénomènes sont modélisables sous la forme de processus stochastiques. Ceci nous permet de prendre en compte dans la modélisation d'un système dynamique les différentes altérations aléatoires du modèle en agissant sur les propriétés du ou des processus stochastiques concernés. On peut alors définir un modèle général que l'on nomme représentation d'état.

On considère un modèle continu d'un système dynamique d'entrée \mathbf{u} supposée connue, d'état \mathbf{x} et de sortie \mathbf{y} perturbés respectivement par un bruit blanc gaussien \mathbf{v} de matrice de covariance \mathbf{Q} et \mathbf{w} de matrice de covariance \mathbf{R} , tous deux centrés sur zéro. On exprime alors le modèle d'état du système à l'instant t de la manière suivante :

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{F} \cdot \mathbf{x}(t) + \mathbf{G} \cdot \mathbf{u}(t) + \mathbf{v}(t) & \text{modèle d'évolution} \\ \mathbf{y}(t) = \mathbf{H} \cdot \mathbf{x}(t) + \mathbf{w}(t) & \text{modèle de mesure} \end{cases} \quad (2.1)$$

Avec \mathbf{F} la matrice d'évolution du modèle et \mathbf{G} la matrice de commande, issues d'une modélisation du phénomène physique qui régit le système (lois de la mécanique, de la thermodynamique, de la chimie, etc.), tandis que \mathbf{H} est la matrice d'observation qui relie \mathbf{x} à \mathbf{y} .

La matrice \mathbf{H} modélise la manière dont les capteurs d'un système parviennent à restituer son état. Celle-ci nous permet de vérifier l'observabilité de notre système, c'est-à-dire sa capacité à nous transmettre l'ensemble des états du système \mathbf{x} à tout instant t en fonction des entrées \mathbf{u} et sorties \mathbf{y} de celui-ci. Selon le critère de Kalman sur l'observabilité

[73], notre système est pleinement observable si et seulement si :

$$\text{rang} \begin{bmatrix} \mathbf{H} \\ \mathbf{H} \cdot \mathbf{F} \\ \dots \\ \mathbf{H} \cdot \mathbf{F}^{n-1} \end{bmatrix} = p \quad (2.2)$$

Avec p la longueur du vecteur d'état \mathbf{x} . L'observabilité d'un système va donc dépendre du type et du nombre de capteurs choisis.

Les capteurs utilisés dans la robotique mobile appartiennent toujours à l'une des deux catégories suivantes [28] :

- **Les capteurs proprioceptifs** délivrent une mesure par rapport à ce qu'ils perçoivent localement du mouvement du véhicule dans l'espace (comme par exemple la vitesse de rotation). Ces mesures sont indépendantes des conditions environnementales dans lesquelles évolue le système étudié. Ces informations revêtent donc un caractère essentiel dans un contexte de navigation autonome. Néanmoins, dans le cadre de capteurs proprioceptifs à faible coût, la précision des informations fournies se dégrade continuellement au cours du temps, les rendant inutilisables sans corrections régulières via une source d'information supplémentaire.
- **Les capteurs extéroceptifs** permettent plutôt de recueillir des informations sur l'environnement dans lequel le véhicule évolue (pression, champ magnétique, etc.). Ces informations correspondent à des observations par rapport à un référentiel absolu attaché à l'environnement (comme la position des satellites). Les informations obtenues via ces capteurs sont donc essentielles pour corriger les estimations issues des capteurs proprioceptifs.

Cette catégorisation de nos capteurs nous permet déjà de voir la complémentarité évidente entre les capteurs proprioceptifs et extéroceptifs : les premiers réalisent des mesures indépendantes de leur environnement et peuvent être corrigés par les seconds, dont les mesures dépendent de leur environnement.

2.3.2 Global Navigation Satellite System

Un système GNSS (Global Navigation Satellite System) permet de déterminer la position absolue d'un objet sur Terre grâce aux signaux émis par une constellation de satellites prévus à cet effet. Une constellation comprend au moins 24 satellites, en orbite à une altitude comprise entre 25000 et 30000 km, afin d'assurer que le signal d'au moins 4 satellites soit disponible n'importe où sur Terre. Chaque satellite émet un signal daté, par le moyen d'une horloge atomique embarquée, sur deux à trois bandes de fréquences comprises entre 1.145 et 1.414 MHz et modulées en phase.

Via son horloge interne, un récepteur GNSS détermine le temps de propagation du signal, entre son émission par un satellite et sa réception, afin d'estimer une distance

à l'émetteur. Chaque mesure de distance permet une localisation du récepteur sur une sphère centrée sur chaque satellite émetteur. Grâce aux localisations déterminées par trois satellites émetteurs, la position absolue du récepteur peut alors être déduite précisément par trilatération. Cependant, cette localisation est limitée si les horloges des trois satellites et du récepteur ne sont pas synchronisées. En effet, un décalage d'un millièmme de seconde engendre une erreur de plus de 300 km [28]. Pour remédier à ce problème, un quatrième satellite (voir figure 2.13) permet au récepteur de calibrer son horloge. Ainsi, on obtient une estimation avec une précision d'une dizaine de mètres pour les systèmes standards, sous la forme d'un triplet latitude, longitude, altitude qui peut facilement être représenté dans l'espace par un système de référence géodésique comme le WGS84 [1]. La disponibilité de satellites supplémentaires peut encore améliorer cette précision.

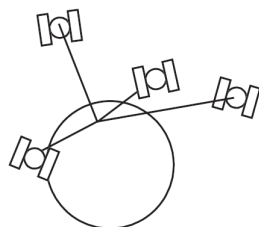


FIGURE 2.13 – Illustration du principe de fonctionnement de la technologie GNSS à 4 satellites.

Chaque système GNSS diffuse une variété de signaux différents sur des fréquences différentes. La plupart des signaux sont accessibles sans coût, via des récepteurs tous publics aux performances réduites, tandis que d'autres sont réservés à des usages militaires ou commerciaux. Dans ces cas là, et avec l'apport d'informations additionnelles via des méthodes de positionnement relatif comme le "Wide Area Augmentation System" (WAAS) ou le "European Geostationary Navigation Overlay Service" (EGNOS) qui proposent un ensemble de stations fixes dont la position est parfaitement connue pour calibrer l'erreur de positionnement, les technologies GNSS obtiennent des erreurs d'estimations comprises entre 1 à 3.9m dans le plan horizontal et entre 1.6 à 6.3m dans le plan vertical [56]. L'usage de balises de localisation au sol pour faciliter la calibration de certaines erreurs peut réduire cette erreur à l'ordre du mètre, tandis que les technologies de mesures de phase de la porteuse de l'onde ("Carrier phase") proposent une précision centimétrique.

On distingue actuellement 4 constellations disponibles pour la technologie GNSS. Le GPS ("Global Positioning System") est sans doute le plus connu. Il est développé et contrôlé par les Etats-Unis depuis 1973 qui le rend accessible à toutes et tous avec un niveau de précision réduit. Sa contrepartie russe, le GLONASS ("GLObal Navigation Satellite System"), a été développé depuis les années 1982. L'Union Européenne

et d'autres acteurs travaillent quant à eux sur la constellation GALILEO afin de garantir une autonomie vis-à-vis des états susmentionnés pour les applications civiles. Disponible depuis 2016, la constellation doit achever son déploiement pour 2024. Enfin, Beidou-3, le projet chinois, est totalement opérationnel depuis juin 2020.

Dans des véhicules aériens, ces technologies offrent une précision très peu altérée. Cependant, dans des environnements urbains, qui sont généralement encombrés, cette précision est plus aléatoire et requiert le support d'algorithmes de fusion ou d'un matériel plus performant et donc plus coûteux.

De part sa quasi-omniprésence dans nos systèmes de navigations et sa faible puissance, la technologie GNSS est une cible privilégiée d'attaques portées sur les fonctions de navigation embarquée dans des systèmes sensibles, comme les projectiles. Le brouillage ("jamming"), qui consiste à émettre un signal de forte puissance sur les plages d'émissions des signaux GNSS, est une méthode simple et accessible limitant l'opérabilité des systèmes de navigation dépendants d'un GNSS. Pour lutter contre ces attaques, des méthodes anti-brouillages sont développées [50] ou bien l'étude de la réception de signaux d'opportunité des satellites de communication de faible orbite pour palier à l'indisponibilité des satellites GNSS [66] est menée.

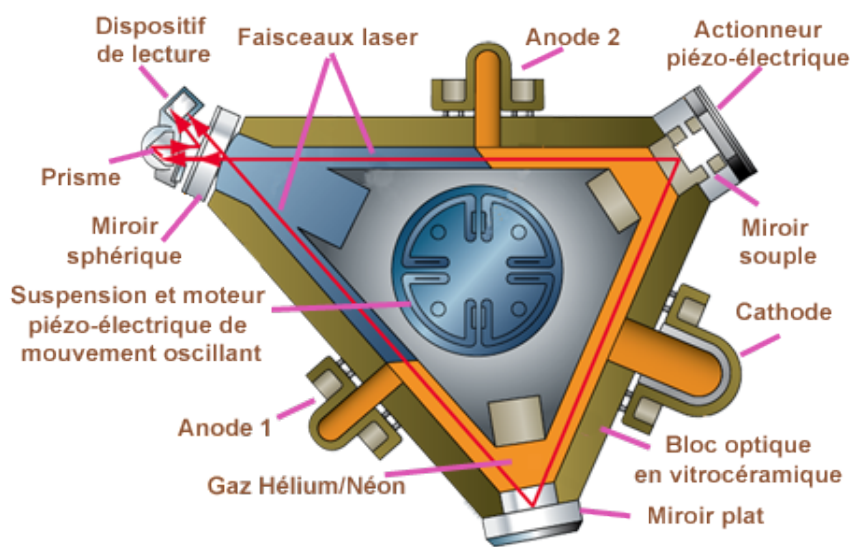
2.3.3 Systèmes magnéto-inertiels

Avant de parler des systèmes magnéto-inertiels, il convient de présenter trois autres capteurs : les accéléromètres, les gyromètres, et les magnétomètres.

Un accéléromètre mesure l'accélération linéaire d'un mobile ou de tout objet sur lequel il est fixé (comme un smartphone). Pour cela, il est constitué d'une masse d'épreuve, dont la masse et la forme sont parfaitement connues, suspendue par une charnière déformable (ressort, lame flexible, membrane) à un cadre rigide. Immobile à vitesse nulle ou constante, la masse va se déplacer sous l'effet d'une variation de vitesse linéaire appliquée au système. Ce déplacement, qui va induire une déformation des charnières, va permettre de déterminer l'accélération subie car directement proportionnelle à la déformation (système masse-ressort) et que l'on peut donc traduire en signal électrique selon diverses méthodes.

Un gyromètre mesure la vitesse angulaire d'un corps en mouvement dans un repère de référence par rapport à un repère inertiel. On distingue deux types de gyromètres : les gyromètres optiques basés sur l'effet Sagnac [82] et les gyromètres mécaniques qui reposent sur l'effet Coriolis [122]. Les premiers sont les gyromètres laser (voir figure 2.14) ou à fibre optique et offrent généralement de très bonnes performances au prix d'un encombrement conséquent. Les seconds sont les gyromètres à oscillateur piézoélectrique dont les mouvements de rotations induisent des différences d'amplitudes entre les différents éléments vibrants proportionnelles à la vitesse de rotation.

Un magnétomètre donne une direction de référence, généralement le nord magnétique, sur le plan horizontal afin de permettre la mesure d'angle dans ce plan par rapport à



Source: www.faa.gov/handbooks_manuals/media

FIGURE 2.14 – Schéma de la structure interne d'un gyromètre laser.

cette direction. Pour ce faire, un magnétomètre mesure la projection du champ magnétique terrestre et son orientation par rapport à ce même champ magnétique à la manière d'une boussole par le biais de différents principes comme l'effet Hall². Il s'agit d'un capteur accessible qui donne une information d'alignement souvent utile.

Les systèmes inertiels se divisent en trois catégories :

- les unités de mesures inertielles ("Inertial Measurement Unit", IMU) qui sont composées de trois accéléromètres et de trois gyromètres desquels il est possible de déterminer la pose et l'attitude d'un mobile relativement à une situation initiale,
- les centrales d'attitudes ("Attitude and HEading Reference System", AHRS) sont des IMUs augmentées d'un calcul en temps réel de l'orientation de l'objet qui supporte le capteur,
- enfin, les centrales inertielles ("Inertial Navigation System", INS, figure 2.15) sont capables de calculer la vitesse et la position de l'objet sur lequel le capteur est équipé.

Les principales différences entre IMU, AHRS et INS sont alors l'existence d'unités de calcul additionnelles. On peut également noter que la plupart des INS actuelles comportent également un magnétomètre afin d'améliorer, via un algorithme de fusion, les performances du bloc IMU. Les avantages principaux de ces systèmes sont leur caractère proprioceptif, dont les bénéfices ont été mentionnés plus haut, et leur très grandes fréquences d'acquisition possibles eu égard à la dynamique des systèmes observés, jusqu'à plusieurs centaines de Hertz.

2. https://fr.wikipedia.org/wiki/Effet_Hall

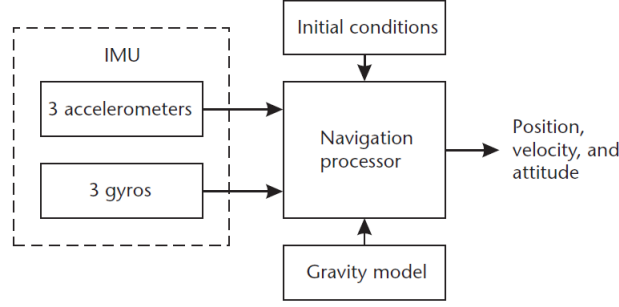


FIGURE 2.15 – Schéma d’une centrale inertielle : on peut constater qu’outre l’IMU composée d’accéléromètres et de gyromètres, l’INS comporte une modélisation du champ de gravité qui permet de faire fonctionner un processeur de navigation pour déduire la position, la vitesse et l’attitude des données IMU.

Trois paramètres vont conditionner le choix des accéléromètres et des gyromètres et, par conséquent, le choix d’un système inertiel. Ces paramètres sont le facteur d’échelle qui est défini comme le rapport entre la sortie du capteur et la grandeur d’entrée, le biais qui traduit une erreur systématique de mesure d’un capteur et la résolution qui correspond au plus petit signal d’entrée mesurable.

Une IMU est ainsi fortement impactée par les défauts des capteurs qui affectent les mesures des accélérations et des vitesses de rotation comme le bruit, un biais ou un facteur d’échelle trop élevé, etc.

Actuellement, les technologies MEMS (“MicroElectroMechanical Systems”), de part leur avantage conséquent de taille et de poids et leur large gamme de coûts et de performances, occupent la large majorité des applications militaires comme civiles.

Une IMU fournit donc une estimation biaisée et bruitée du vecteur de vitesse angulaire $\boldsymbol{\omega}_n^{imu}$ et de l’accélération spécifique³ \mathbf{a}_n^{imu} [76] comme illustré dans les équations 2.3 à 2.9. L’algorithme de navigation à l’estime utilise ces données d’entrées pour déterminer le couple \mathbf{p}_n^{imu} et \mathbf{r}_n^{imu} , respectivement la position et l’orientation du mobile selon les données de l’IMU.

$$\boldsymbol{\omega}_n^{imu} = \boldsymbol{\omega}_n + \mathbf{b}_n^\omega + \mathbf{w}_n^\omega, \quad (2.3)$$

$$\mathbf{a}_n^{imu} = \mathbf{a}_n + \mathbf{b}_n^a + \mathbf{w}_n^a. \quad (2.4)$$

\mathbf{b}_n^ω et \mathbf{b}_n^a sont des biais tandis que \mathbf{w}_n^ω et \mathbf{w}_n^a sont des bruits blancs gaussiens. Les biais suivent une marche aléatoire bruitée par un bruit blanc gaussien elle aussi.

$$\mathbf{b}_{n+1}^\omega = \mathbf{b}_n^\omega + \mathbf{w}_n^{b\omega}, \quad (2.5)$$

$$\mathbf{b}_{n+1}^a = \mathbf{b}_n^a + \mathbf{w}_n^{ba}, \quad (2.6)$$

3. accélération dépourvue des effets de l’accélération gravitationnelle.

avec $w_n^{\mathbf{b}^i}$ et $w_n^{\mathbf{b}^a}$ des bruits blancs gaussiens.

On définit le modèle cinématique entre deux instants discrets échantillonnés à la période d'échantillonnage dt comme suit :

$$\mathbf{q}_n = \exp(1/2 \cdot (\mathbf{\Omega}_n(\boldsymbol{\omega}_n) \cdot dt)) \cdot \mathbf{q}_{n-1} \quad (2.7)$$

$$\mathbf{v}_n^{imu} = \mathbf{v}_{n-1}^{imu} + (\mathbf{q}_n \otimes \{\mathbf{a}_n^{imu}\} \otimes \tilde{\mathbf{q}}_n - \mathbf{g}) dt \quad (2.8)$$

$$\mathbf{p}_n^{imu} = \mathbf{p}_{n-1}^{imu} + \mathbf{v}_n^{imu} dt, \quad (2.9)$$

avec \mathbf{q}_n le quaternion [113] de la rotation duquel on peut extraire \mathbf{r}_n^{imu} [113], $\mathbf{\Omega}_n(\boldsymbol{\omega}_n) = \mathbf{S}(\boldsymbol{\omega}_n)$ la matrice antisymétrique associée au produit croisé avec $\boldsymbol{\omega}_n$, $\mathbf{v}_n^{imu} \in \mathbb{R}^3$ la vitesse calculée avec les données de l'IMU et $\{\mathbf{a}_n^{imu}\}$ le quaternion construit à partir de \mathbf{a}_n^{imu} .

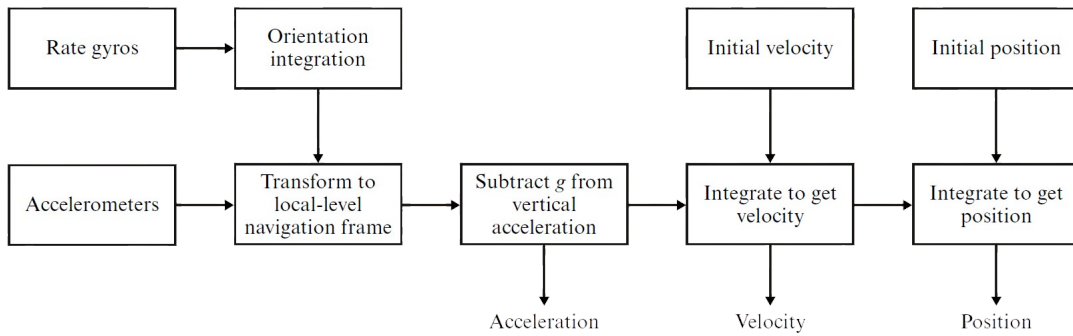


FIGURE 2.16 – Schéma-bloc de l'algorithme de navigation à l'estime inertielle.[36]

Comme l'attestent les équations de la navigation à l'estime inertielle (rappelée par le schéma figure 2.16), l'impact des bruits et des biais, très importants sur les centrales MEMS de coût accessible au public, engendre alors des dérives temporelles importantes qui impactent les estimations de vitesse, position et orientation du mobile. Cela a pour effet de produire une erreur systématique d'estimation croissante au fil du temps par récursivité, d'autant que ces capteurs sont conçus pour fonctionner à plusieurs centaines de Hertz.

Outre la solution simple qui consisterait à s'équiper de centrales plus coûteuses avec des biais plus faibles, des méthodes impliquant un couplage avec un capteur d'appui, généralement extéroceptif, afin de corriger les biais des centrales inertielles, sont employées. L'intérêt d'un tel couplage sera présenté à la fin de cette section et dans le chapitre 3. La méthode ZUPT ("Zero Velocity Potential updaTe", mise à jour de potentiel à vitesse nulle) [116] est une innovation intéressante en navigation inertielle

puisqu'elle profite des instants où le mobile est à l'arrêt (vitesse nulle) afin de corriger la dérive des capteurs inertiels.

D'autres améliorations de la navigation inertielle se reposent sur l'usage de filtres stochastiques comme le filtre de Kalman. Il est donc devenu fréquent de réaliser la navigation à l'estime par le biais de filtres de Kalman étendus (EKF, "Extended Kalman Filter") [114] et étendus invariants (IEKF, "Invariant Extended Kalman Filter") [11], seront détaillées au travers des prochains chapitres.

2.3.4 Télémètres

Les télémètres regroupent l'ensemble des capteurs qui permettent de détecter des objets et de déterminer la distance ou position angulaire par rapport au capteur de ceux-ci. Le principe général consiste à envoyer un signal dans une direction et à collecter les signaux qui lui sont réfléchis. On parle alors de technologie active. La direction de la réception indique la direction de l'objet tandis que la durée entre l'émission et la réception permet d'en déduire la distance. Différents capteurs existent selon le type d'onde employée.

Les sonars ("SOund Navigation And Ranging") utilisent des ultrasons. Populaires en raison de leur accessibilité et de leur simplicité, ils ont généralement une portée de 6 m⁴. Cependant, l'étalement de l'onde sonore dès son émission empêche le récepteur de déterminer correctement la direction de la source de l'onde. En effet, les réverbérations multiples précédant la réception expliquent sa faible portée et rendent difficile son utilisation pour la localisation d'objets plus éloignés ou pour la navigation à une vitesse qui requiert un champ de vision plus profond.

Les capteurs de portée infrarouge (figure 2.17) sont une alternative encore accessible aux sonars. Envoyant un unique faisceau infrarouge devant lui, le capteur réceptionne ensuite le faisceau réfléchi pour, à la manière du sonar, déterminer la distance entre le capteur et l'obstacle le plus proche.

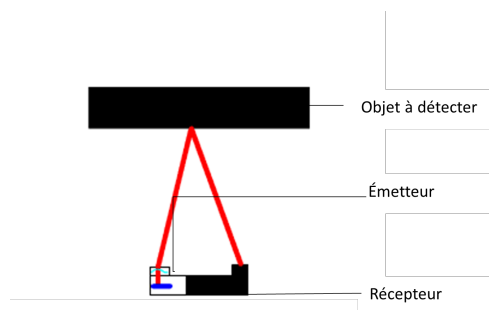


FIGURE 2.17 – Illustration du principe de fonctionnement d'un capteur de portée infrarouge.

Ce capteur donne une information de profondeur très utile pour un coût relativement

4. <https://acroname.com/store/sonar-rangefinder-r241-srf10>

faible. Il présente un complément très utile à une caméra monoculaire pour un coût total inférieure à celui d'une caméra stéréoculaire. De ce fait, il présente un intérêt manifeste dans la navigation autonome à faible coût [109].

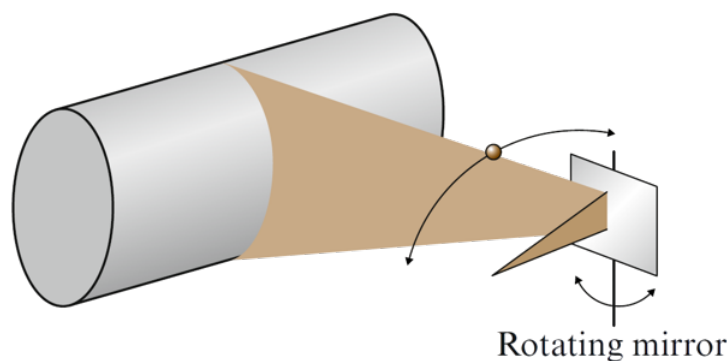


FIGURE 2.18 – Concept du balayage d'un lidar [43]

Les lidars (LIght Detection And Ranging) se présentent comme un généralisation du principe du capteur de portée pour observer l'ensemble de l'environnement du capteur et non plus une unique direction. La technologie lidar la plus simple consiste en un miroir tournant (figure 2.18) qui permet à un faisceau infrarouge de balayer l'environnement du capteur pour, à la manière du capteur de portée, mesurer la distance entre le capteur et tous les obstacles immédiats de l'environnement du capteur. De plus, si l'infrarouge est émis sous la forme d'une onde perpendiculaire à l'axe de rotation du miroir, alors le lidar permet d'observer son environnement en 3D. On peut ainsi utiliser un lidar pour réaliser une cartographie de son environnement sous la forme d'un nuage de points dont la densité dépend de la résolution du lidar. Combinée avec des méthodes d'extractions des points d'intérêts [43], les lidars deviennent des outils très intéressants pour la navigation autonome et particulièrement utilisée pour des applications de SLAM ("Simultaneous Localization And Mapping") [33] bien que pour le moment très coûteux et encombrants et donc peu adaptés à des applications sur UAV ("Unmanned Aerial Vehicle", drones volants) et projectiles. On les trouve plutôt au coeur de nombreux travaux sur les voitures autonomes comme montré en figure 2.19.

2.3.5 Caméras

Les caméras présentent une alternative très intéressante aux lidars en raison de leur faible coût, de leur meilleure compacité grâce aux technologies CMOS, de la densité des données fournies contrairement aux lidars et à la passivité du capteur (une caméra n'émet pas de signaux contrairement à un lidar). On lui connaît deux applications majeures en navigation : l'estimation du déplacement relatif entre deux images [103] ou

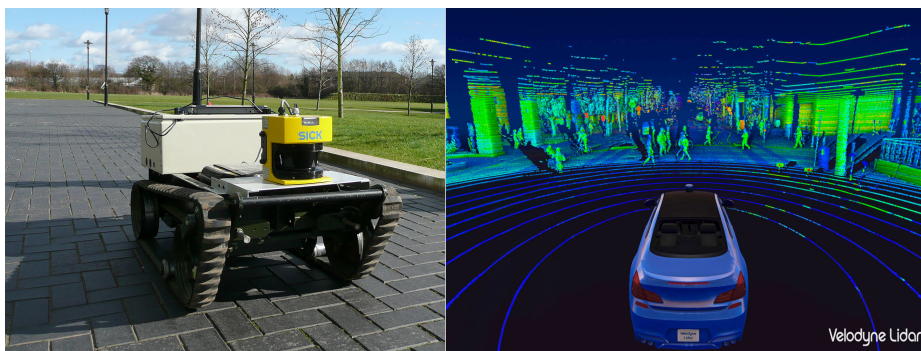


FIGURE 2.19 – À gauche, un UGV équipé d'un lidar SICK et à droite un exemple d'une cartographie lidar selon le constructeur Velodyne, acteur majeur du marché du lidar.

l'estimation de la pose de la caméra à partir de la projection de points 3D connus dans le plan du capteur [46]. La deuxième méthode est celle que l'on retrouve dans la plupart des applications de V-SLAM [67] qui constituent un axe majeur de la recherche actuelle en navigation.

De plus, la large variété de spectres lumineux pour lesquels des caméras sont disponibles permettent le développement d'un grand nombre d'applications comme par exemple l'utilisation de la polarisation de la lumière céleste pour la navigation [29].

Outre leurs usages pour la localisation, les caméras fournissent une large quantité d'informations, qui peuvent certes être difficiles à exploiter, mais qui se révèlent néanmoins très intéressantes pour les différentes tâches de la navigation. Les nombreux travaux de V-SLAM attestent déjà de l'intérêt manifeste de ce capteur pour les applications de cartographie tandis que d'autres travaux emploient des caméras pour obtenir une meilleure interprétation de l'environnement d'un véhicule pour la détection et l'évitement d'obstacles (comme sur la figure 2.20) [81] ou pour la lecture des panneaux de trafic routier [107].



FIGURE 2.20 – Cas exemple d'une application de détection d'obstacle : un piéton est détecté et mis en évidence par une boîte l'encadrant sur l'image issue du flux vidéo.

Les caméras peuvent également servir de moyens de mesure de la vitesse par le biais d'une étude du flux optique [108] [6].

La majorité des applications de navigation utilisant des caméras reposent sur des caméras stéréo [77] [97], soit sous la forme d'un unique capteur ou par l'utilisation simultanée de plusieurs caméras monoculaires. Ce type de dispositif offre l'avantage d'être en mesure de fournir une information de profondeur permettant de déterminer facilement la distance entre le capteur et les objets observés.

Les caméras monoculaires ne sont quant à elles que plus récemment utilisées dans la navigation [19], principalement en raison de leur omniprésence dans les smartphones et de leur coût plus faible adaptés à des usages sur UAVs. La perte de l'information de profondeur doit alors être compensée ou contournée, l'usage d'un capteur de portée à cette fin est alors fréquent [109]. Plusieurs méthodes de navigation très performantes comme ORB-SLAM ("Oriented fast and Rotated Brief-SLAM") [96] ou VINS-mono ("Visual Inertial Navigation System-mono") [100] reposent entre autre sur une caméra monoculaire.

Pour détecter des objets au sein d'une image, plusieurs méthodes existent comme la méthode SIFT (Scale-Invariant Feature Transform) [91] ou bien la méthode de Harris [57] (illustrée figure 2.21), qui consiste à détecter les zones de l'image où le gradient de contraste est le plus élevé en considérant une matrice de points autour du point évalué. La méthode de Harris est encore favorisée lors d'applications où le temps de calcul cherche à être minimisé.

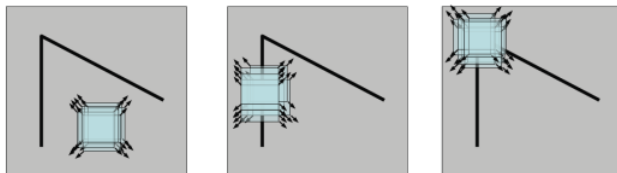


FIGURE 2.21 – Exemple du fonctionnement d'un détecteur de Harris. Si la matrice (carré bleu) autour du point étudié est inchangée dans toutes les directions, rien n'est détecté (zone plate). Si un changement de contraste est exprimé dans une seule direction, une arête est détectée. Si un changement de contraste s'opère dans toutes les directions, alors un coin est détecté.

Ces méthodes permettent alors de définir des zones d'intérêts sur une image, comme sur la figure 2.22, qu'il est alors nécessaire de classifier pour détecter des correspondances entre images successives.

Une méthode courante pour estimer des correspondances entre deux images est la méthode RANSAC ("RANdom SAmple Consensus") [42]. Il s'agit d'une méthode itérative d'estimation des paramètres d'un modèle mathématique à partir de données. Ces données peuvent contenir à la fois des valeurs pertinentes au modèle ("inliers") ou aberrantes ("outliers"). La méthode RANSAC permet de distinguer les valeurs pertinentes des valeurs aberrantes et par ce moyen d'associer des coins détectés sur une image avec des coins détectés sur une autre image, à condition d'un recouvrement partiel entre les deux.



FIGURE 2.22 – Applications du détecteur de Harris à l'image de gauche. On constate la détection de nombreux coins sur l'image. Le nombre de coins détectés dépend du seuil décidé par l'expérimentateur.

L'utilisation successive d'un détecteur de Harris et de RANSAC sur un jeu d'images monoculaires permet de mettre en place un algorithme de navigation à l'estime visuel basique.

En dépit de leurs nombreuses qualités, les caméras souffrent d'une faiblesse : elles doivent être en conditions de voir correctement leur environnement pour permettre la navigation. En effet, les lumières parasites (ou "lens flare"), les forts changements d'intensité lumineuse ou encore l'occultation du champ de vision sont autant de moments où la caméra est indisponible pour une navigation de qualité. La complémentation avec un capteur proprioceptif semble alors indiquée.

2.3.6 De l'intérêt d'associer plusieurs capteurs : le couplage

Dans la plupart des cas d'applications, comme dans [109] par exemple, la navigation n'est pas réalisée à partir d'un unique capteur. En effet, nous avons pu constater précédemment que ceux-ci sont tous sujets à des faiblesses ou des cas limites d'applicabilité (biais, dérive, brouillage, occultation, etc.). En particulier, nous pouvons constater que les principaux capteurs proprioceptifs, les centrales inertielles, souffrent des lourdes dérives tandis que la plupart des capteurs extéroceptifs rencontrent des situations d'indisponibilité de l'information. Par conséquent, la majorité des travaux de navigation se porte sur des systèmes qui couplent au moins deux solutions pour assurer la meilleure fonction de navigation possible. Comme expliqué pour la localisation, la plupart des

techniques multi-capteurs utilisent un premier capteur (souvent proprioceptif) pour réaliser une prédiction de la localisation tandis qu'un second capteur (souvent extéroceptif) sera utilisé pour corriger cette prédiction comme c'est le cas sur la figure 2.23.

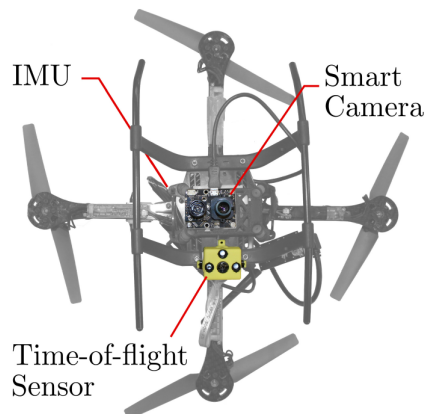


FIGURE 2.23 – Exemple de drone volant équipé de plusieurs capteurs pour la navigation [109]. Ici, on retrouve un capteur proprioceptif (IMU) et deux capteurs extéroceptifs le complétant (une caméra monoculaire, et un capteur de portée infrarouge).

Quand on parle de couplage, ou fusion, de capteurs pour la navigation d'un point de vue algorithmique (i. e. les moyens par lesquels on réalise la fusion), deux catégories principales peuvent être dégagées [23]. D'un côté, nous avons le couplage lâche (figure 2.24) qui revient à fusionner des estimations déjà réalisées à partir des données de chaque capteur. De l'autre côté, le couplage serré (figure 2.25) consiste en l'utilisation des données de tous les capteurs pour fournir directement une estimation unique sans passer par des estimations intermédiaires.. Bien entendu, ces deux catégories constituent des extrêmes en termes de solutions de couplage, et il est par exemple fréquent pour des méthodes de couplage lâche d'exploiter les mesures directes d'une IMU [5] [111].

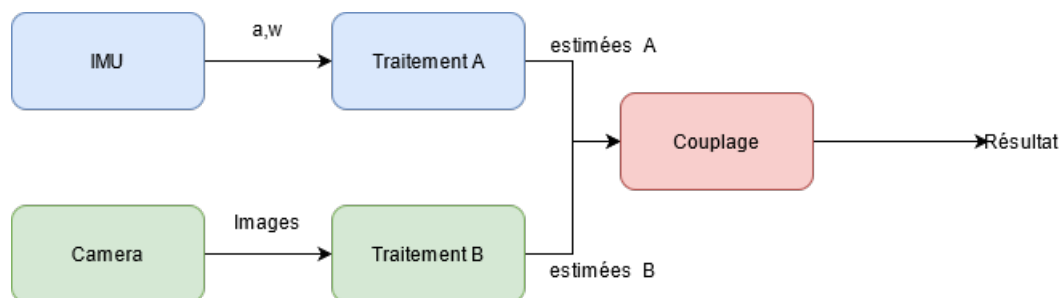


FIGURE 2.24 – Schéma illustrant le principe d'un couplage lâche : chaque capteur est traité séparément pour obtenir plusieurs estimations indépendantes, lesquelles sont ensuite couplées.

L'intérêt du couplage lâche est alors de disposer de résultats intermédiaires, ce qui peut faciliter la calibration des alignements des capteurs entre eux [89], permettre d'assurer la navigation en l'absence d'information de l'un des capteurs ou encore ne pas avoir à

gérer la synchronicité temporelle des capteurs en amont de la fusion [92].

De son côté, le couplage serré propose généralement les résultats les plus précis car disposant d'une large quantité et variété d'informations de différents capteurs [23] [32].

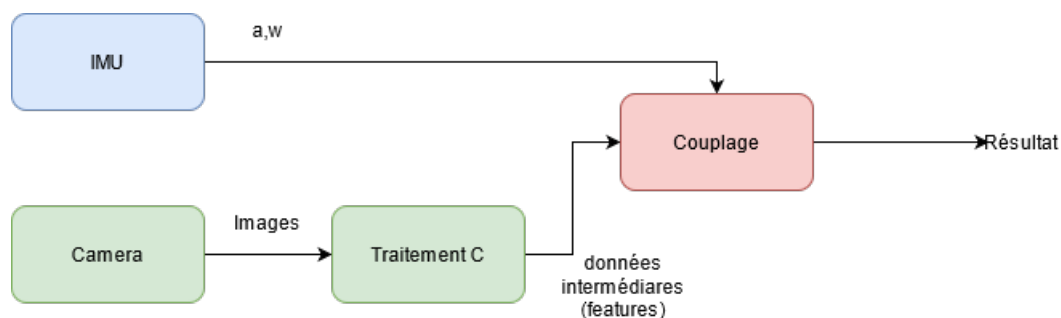


FIGURE 2.25 – Schéma illustrant le principe d'un couplage serré : les données des capteurs sont directement couplées avec peu ou pas d'estimation intermédiaire.

Dans tous les cas, le couplage confère un avantage significatif à une application de navigation puisqu'il améliore l'observabilité du système. L'augmentation du nombre et de la variété de mesures capteurs nous permet en effet de déterminer un plus grand nombre d'états du système dont on veut estimer la pose et l'attitude [121]. Cette amélioration de l'observabilité explique la capacité d'un couplage proprioceptif/extéroceptif à permettre d'estimer la pose du système et les biais du capteur proprioceptif [65].

2.4 Évaluer la navigation

Une fois que l'on a défini ce qu'est la navigation et les moyens par lesquels on la réalise, on peut s'interroger sur les moyens d'évaluer les résultats. En effet, comment peut-on évaluer la précision de nos moyens de localisation, leur robustesse face aux dérives capteurs ou encore leur intégrité? C'est grâce à ces moyens que l'on obtient les informations nécessaires pour nous localiser, la plupart du temps dans un environnement non contrôlé qui rend difficile l'évaluation quantitative de cette localisation.

La solution choisie par la communauté scientifique est double : d'un côté on réalise des jeux de données de navigation [54] [16] qui nous permettent de disposer d'un contrôle maximal des paramètres contextuels de la navigation (trajectoires, capteurs, mobiles, etc.), et de l'autre on se compare à une ou plusieurs méthodes de navigation pré-existantes afin de mettre en évidence les différences, et donc souvent les améliorations, de nos méthodes vis-à-vis de celles-ci.

Ainsi, les jeux de données de navigation, en intégrant souvent le résultat d'une navigation GNSS utilisant des capteurs aux coûts élevés, fournissent un groupe témoin auquel il est possible de comparer les performances des algorithmes de navigation dans des contextes et environnements contrôlés, mais proches de conditions réelles.

L'avantage principal de cette solution réside dans le fait que cette validation s'effectue avec des vrais mobiles équipés de vrais capteurs. Cette base va ainsi permettre aux chercheurs et chercheuses en navigation de comparer leurs développements avec la vérité-terrain GNSS fournie.

2.4.1 Métriques de la navigation

Les performances de la navigation sont assez simples à évaluer une fois que l'on a une référence à laquelle se comparer. Si l'on définit une séquence de poses issues d'une trajectoire estimée $\mathbf{P}_1, \dots, \mathbf{P}_n \in \mathbf{SE}(3)$ qui est synchronisée et avec le même nombre d'échantillons qu'une séquence de poses issues de la vérité terrain $\mathbf{Q}_1, \dots, \mathbf{Q}_n \in \mathbf{SE}(3)$, on peut classiquement chercher à évaluer la précision de notre estimation au regard de notre vérité terrain. Pour ce faire, on dispose de deux manières principales de calculer une erreur : l'erreur relative de pose (ou "Relative Pose Error", RPE) et l'erreur absolue de trajectoire ("Absolute Trajectory Error", ATE) [115].

L'erreur relative de pose mesure la précision locale sur la trajectoire durant un laps de temps constant Δ . De ce fait, cette erreur mesure la dérive de notre estimation, ce qui est particulièrement utile pour évaluer les performances des méthodes de navigation à l'estime visuelle ou inertielle. On peut définir l'erreur relative de pose à l'instant n comme suit :

$$\mathbf{E}_n = (\mathbf{Q}_n^{-1} \cdot \mathbf{Q}_{n+\Delta})^{-1} \cdot (\mathbf{P}_n^{-1} \cdot \mathbf{P}_{n+\Delta}) \quad (2.10)$$

A partir d'une séquence de N poses, on obtient alors $M = N - \Delta$ erreurs individuelles le long de notre séquence. Ces erreurs peuvent ainsi permettre de calculer une erreur quadratique moyenne ("Root Mean Square Error", RMSE) afin de caractériser notre navigation sur l'ensemble de la trajectoire.

$$RMSE(\mathbf{E}_{(i:N)}; \Delta) = \sqrt{\frac{1}{M} \sum_{i=1}^m \|\mathit{trans}(\mathbf{E}_i)\|^2} \quad (2.11)$$

Avec $\mathit{trans}(\mathbf{E}_i)$ la composante translation de la matrice de l'erreur relative de pose \mathbf{E}_n définie comme suivante :

$$\mathbf{E}_n = \begin{pmatrix} \mathit{rot}(\mathbf{E}_n) & \mathit{trans}(\mathbf{E}_n) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

On peut également mentionner l'usage du calcul de l'erreur moyenne au lieu de la RMSE, ceci afin de réduire l'impact des outliers que l'on pourrait attribuer à une faible robustesse de la navigation. L'évaluation de la navigation par le calcul d'une RPE est la plus courante dans la littérature et dans les outils de calculs d'erreurs fournis avec les jeux de données car elle permet également d'investiguer les instants où cette erreur est maximale mais aussi la présence d'outliers pour faire le lien avec le contexte de la

navigation ou les limites de la méthode évaluée.

Dans certains cas où on peut aligner correctement les repères de coordonnées de la vérité terrain et de l'estimation via une matrice de passage \mathbf{S} , on peut implémenter l'erreur de trajectoire absolue comme suit :

$$\mathbf{F}_n = \mathbf{Q}_n^{-1} \cdot \mathbf{S} \cdot \mathbf{P}_n \quad (2.13)$$

Cet ensemble d'erreurs peut également être moyenné par le calcul d'une RMSE comme précédent avec $M = N$ ou d'une moyenne. Cette erreur ne tient compte que des erreurs de translation, là où la RPE tenait compte des erreurs en rotation et en translation. De plus, les erreurs de rotation induisent des erreurs de translations qui sont dès lors capturées par l'ATE. Celle-ci qui se présente alors comme une erreur qui facilite la détection des points de difficultés d'une navigation sans pour autant nous dispenser du calcul d'une RPE. En effet, les résultats des deux méthodes sont souvent très corrélés [115].

2.4.2 Jeux de données courants

KITTI vision

Le jeu de données KITTI vision⁵[54] est un jeu de données réalisé par l'Institut de Technologie de Karlsruhe et l'Institut de Technologie de Toyota à Chicago. Ce jeu de données est une référence de la navigation terrestre par la , et de nombreux algorithmes d'odométrie visuelle ou par lidar l'ont utilisé pour leurs essais⁶ [30] [24].

Ce jeu de données a été réalisé à l'aide d'une voiture (voir figure 2.26) sur laquelle ont été embarqués :

- 2 caméras monoculaires à niveau de gris de 1.4 Megapixels, FL2-14S3M-C, avec objectifs Edmund Optics lenses de 4 mm.
- 2 caméras monoculaires couleurs de 1.4 Megapixels, FL2-14S3C-C , avec objectifs Edmund Optics lenses de 4 mm.
- 1 lidar Velodyne HDL-64E, 64 rayons à 10Hz à 360° horizontal, 26.8° vertical, portée de 120 m, une résolution angulaire de 0.09° et une précision de 2 cm.
- 1 centrale de navigation GPS/IMU OXTS RT3003 6 axes cadencée à 100 Hz avec L1/L2 RTK et une résolution de 0.02m/0.1°.

Le véhicule a ensuite parcouru la ville de Francfort dans des environnements urbains, résidentiels, des routes, sur un campus. Une acquisition à l'arrêt de plusieurs cohortes de personnes est également disponible (jeu de données "personne").

KITTI vision fournit les données sous deux formes : une forme synchronisée et rectifiée dans laquelle les données capteurs sont toutes échantillonnées à 10Hz et une forme brute qui contient les données capteurs à leurs fréquences de fonctionnement nominales.

5. <https://www.cvlibs.net/datasets/kitti/>

6. https://www.cvlibs.net/datasets/kitti/eval_odometry.php

Environ 180 GB de données brutes sont disponibles. Néanmoins, la validation des algorithmes ne s'effectue que sur un ensemble de dix séquences, le reste ayant pour vocation de servir à des applications d'apprentissage. Les matrices de calibration utilisées sont fournies si besoin, permettant la projection des données d'un capteur dans le repère d'un autre capteur.

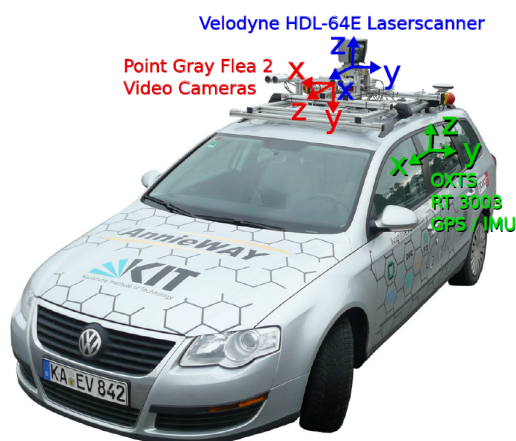


FIGURE 2.26 – AnniWay, le véhicule équipé de capteurs utilisé pour la réalisation du jeu de données jeu de données KITTI vision. On peut également voir les différents repères attachés aux capteurs.

Les données GPS sont fournies selon le protocole WGS84 [1]. Les latitudes, longitudes et altitudes peuvent ainsi facilement être passées dans un référentiel cartésien East-North-Up⁷ qui correspond au repère vert de l'image 2.27 ci-dessus.

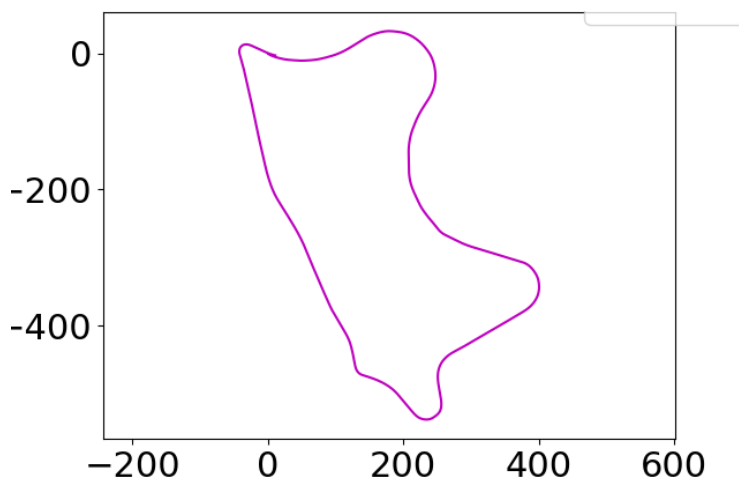


FIGURE 2.27 – Vérité-terrain en 2D de la séquence de validation n° 9 du jeu de données KITTI vision. Les axes sont exprimés en mètre.

7. https://en.wikipedia.org/wiki/Geographic_coordinate_conversion#From_ECEF_to_ENU

Le passage du repère caméra (repère rouge) au repère inertiel (repère vert) se fait simplement via la matrice de passage suivante :

$$\mathbf{S}_{(imu_cam)} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \quad (2.14)$$

Afin de permettre à tous les utilisateurs du jeu de données de disposer d'une base fiable et saine, dix séquences sont explicitement mentionnées comme séquences de validation et leurs instants de départ et de fin sont mentionnées dans un fichier disponible sur le site. De plus, un algorithme d'évaluation est fourni. Cet algorithme prend en entrée une trajectoire de vérité-terrain et une trajectoire de navigation exprimées dans le même repère. Ces trajectoires sont découpées en plusieurs points de départ sur la trajectoire espacés d'une seconde pour recouvrir l'ensemble des trajectoires. Ensuite, des RPE sont calculées sur des segments de trajectoires de longueurs [100, 200, 300, 400, 500, 600, 700, 800] m partant de chaque points de départ précédemment générés. Ceci permet de calculer un nombre de RPE bien plus important que par un calcul classique et renforce l'impact des outliers qui seront souvent parcourus plusieurs fois par cette méthode. De plus, elle permet de mettre en évidence les effets d'une dérive sur la navigation qui peut être constatée en étudiant les moyennes des RPE par taille de segment. L'algorithme de KITTI calcule pour finir les moyennes globales de la RPE en rotation et en translation sur tous les segments, ce sont ces valeurs qui sont généralement diffusées sous la forme de résultats.

Les unités des RPE fournit par cet algorithme d'évaluations sont le mètre/mètres parcourus (m/m) pour la translation et le radian par mètres (rad/m) parcourus pour la rotation. Afin de faciliter la lecture des données, la RPE en translation est généralement multipliée par cent pour donner une RPE en %. Cette modalité de représentation des données se retrouve dans plusieurs travaux utilisant KITTI, comme [11] et [102]. La méthode d'évaluation des résultats proposée par le jeu de données KITTI vision permet de disposer de résultats très précis et fiables sur la précision et la répétitivité des solutions de navigation ainsi testées.

EuRoC

Le jeu de données pour véhicules micro-aériens pour le European Robotic Challenge (EuRoC)⁸ [16] consiste en un ensemble de trajectoires acquises par un drone AscTee "Firefly" équipé de plusieurs capteurs. Les capteurs disponibles sont :

- 2 caméras monoculaires à niveau de gris qui échantillonnent à 20 Hz, MT9V034.
- 1 IMU ADIS16448 échantillonnée à 200 Hz.
- 1 station Leica MS50 échantillonnée à 20 Hz pour la référence de position.
- 1 système Vicon à 100 Hz pour la référence de pose.

8. <http://www.euroc-project.eu/>

Les trajectoires ont été réalisées dans deux environnements différents : une salle des machines qui comporte donc des obstacles (notée MH) et une salle Vicon avec deux configurations d'obstacles différentes (notées V1 et V2). Chaque trajectoire dure entre 99 et 182 secondes pour des distances parcourues d'une centaine de mètres. Chaque trajectoire est notifiée selon sa complexité en matière de dynamique de vol et de conditions d'éclairages (notées easy, medium et difficult). Le jeu de données pourvoit l'ensemble des paramètres de calibration utilisés lors des acquisitions. Onze trajectoires d'environnements et de difficultés variables sont ainsi proposées.

Le jeu de données EuRoC est un jeu de données très régulièrement utilisé pour la validation d'algorithmes de navigation.

2.5 Localisation par la vision à l'ISL

2.5.1 Présentation des systèmes STAMINA-UGV

Les systèmes STAMINA-UGV, montrés sur la figure 2.28, forment un ensemble d'UGVs (Unmanned Ground Vehicle, véhicule terrestre sans pilote) conçus pour la réalisation d'un suivi autonome de trajectoire ainsi que la détection et l'évitement d'obstacles. Au fil des générations, ces systèmes ont gagné en autonomie par l'amélioration de leur fonction de navigation par la vision.

Le Wolf/P4 est le premier véhicule STAMINA. Il s'agit d'un 4x4 militaire allemand, un Mercedes Wolf, équipé d'une caméra monoculaire haute-résolution montée au-dessus du pare-brise avant et utilisée pour la détection de changements en tête des convois militaires. Initialement non autonome, il est aujourd'hui équipé d'un système de pilotage automatique.

Les premiers essais de navigation autonome ont été réalisés sur le STAMINA-UGV μ Scout, un très petit UGV capable d'aller à un maximum de 4km/h et équipé d'une caméra FLIR Blackfly modèle FLY-U3-23S6M-C placée à 0.4 m au dessus du sol et orientée de 24° vers le sol. Cette même caméra sera utilisée ensuite sur le STAMINA-UGC Patrol⁹, où elle est placée à 0.82 mètres de hauteur et orientée de 13.8° vers le sol. Le Patrol est encore capable de réaliser de la navigation en intérieur, mais avec sa vitesse de pointe de 30km/h, il est également adapté à des environnements extérieurs. Enfin, le dernier UGV en date, le STAMINA-Aurochs¹⁰, est capable de transporter une personne couchée et plusieurs dizaines de kilos de matériels. Conçu comme un robot tactique polyvalent à application pour l'Armée de Terre, il peut aller jusqu'à 80km/h. Il est équipé d'une caméra Flir orientée comme sur le Patrol ainsi que de deux lidars Velodyne VLP-16 d'une portée de 100m pour un angle de vue de +/-15°, d'une résolution verticale de 2° et d'une résolution horizontale de 0.1° pour réaliser du suivi

9. https://www.isl.eu/documents/flyers/FR/isl-STAMINA_FR_nm.pdf

10. <https://www.isl.eu/documents/flyers/FR/isl-STAMINA-UGV-Aurochs-FR-nm.pdf>



FIGURE 2.28 – Les différents STAMINA-UGV. Dans le sens de lecture : le Wolf/P4, le μ Scout, l'Aurochs et le Patrol.

de fantassin.

L'ensemble de ces UGVs ont pour fonction de faire avancer la brique technologique de navigation autonome par la vision STAMINA, laquelle est capable sans GNSS de :

- Mémoriser en temps réel de nouveaux itinéraires dès qu'ils sont parcourus.
- Réduire l'information image d'un itinéraire au minimum nécessaire pour transmission à d'autres véhicules.
- Suivre un itinéraire en autonomie en marche avant et arrière avec évitement d'obstacle (figure 2.29).

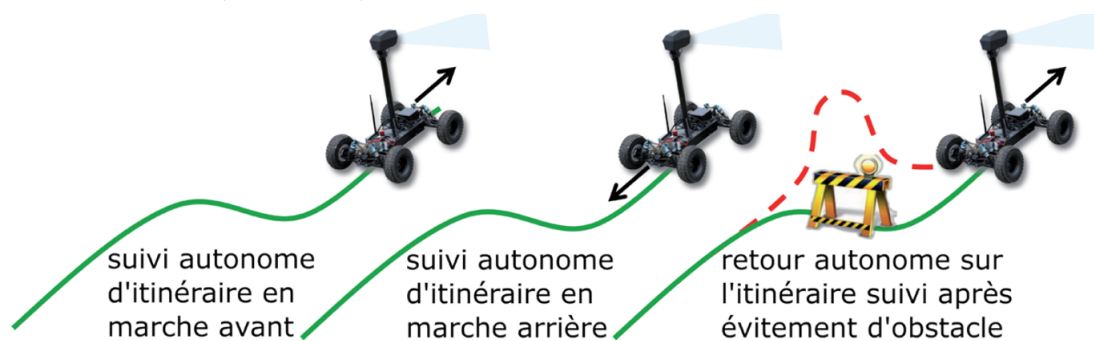


FIGURE 2.29 – Schéma des capacités de navigation de STAMINA : navigation autonome et replanification de trajectoire en cas d'obstacle à éviter.

Les objectifs de STAMINA impliquent la réalisation d'une fonction de navigation par la

vision avancée reposant sur une caméra monoculaire. L'une des dernières améliorations de STAMINA est l'algorithme "Parallax Beam".

2.5.2 Parallax Beam

Parallax Beam est une méthode de navigation à l'estime visuelle qui calcule les paramètres du mouvement à partir d'images successives. Pour cela, il utilise deux images successives de la même scène et exploite le léger mouvement relatif entre les deux. Les points-clés de l'image sont ensuite repérés et décrits. La détection utilise un détecteur de coin de Harris modifié [55], qui assure une répartition homogène sur l'ensemble de l'image des points-clés révélés, tandis que la description est effectuée en utilisant le descripteur BRIEF [17]. Une homographie est ensuite estimée de manière robuste en utilisant une procédure RANSAC modifiée [94]. Le résultat de cette procédure est le mouvement de la caméra stocké dans une matrice d'homographie notée \mathbf{M}_n^{pb} , avec l'hypothèse que le plan principal est vu par la caméra.

$$\mathbf{M}_n^{pb} = \begin{bmatrix} \mathbf{R}_n^{pb} & \mathbf{t}_n^{pb} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.15)$$

Avec \mathbf{R}_n^{pb} la matrice de rotation 3x3 du mouvement entre les deux images et \mathbf{t}_n^{pb} le vecteur translation du mouvement entre les deux images.

Dans le cas d'une scène non plane, le modèle de l'homographie est insuffisant pour tenir compte de toutes les informations fournies par les points-clés. L'algorithme utilise alors le modèle Parallax Beam, expliqué plus en détail dans [103] et illustré figure 2.30, pour estimer la matrice fondamentale et récupérer le mouvement de la caméra \mathbf{M}_n^{pb} .

Comme montré dans la table 2.2, cet algorithme obtient la plus faible ou seconde plus faible erreur par mètre sur le jeu de données KITTI vision si on le compare avec des méthodes connues équivalentes comme l'initialisation de ORB-SLAM [96]. Ces résultats sont atteints alors même que le jeu de données KITTI vision n'est pas adapté à cet algorithme car conçu pour des caméras orientées en direction du sol comme sur les systèmes STAMINA, ce qui entraîne une importante dérive selon l'axe vertical sur certaines séquences, visible sur la figure 2.31.

Cependant, cet algorithme a besoin d'images successives avec des points-clés communs pour permettre une navigation précise avec une dérive minimale. De ce fait, il est en difficulté en cas de coupures d'images.

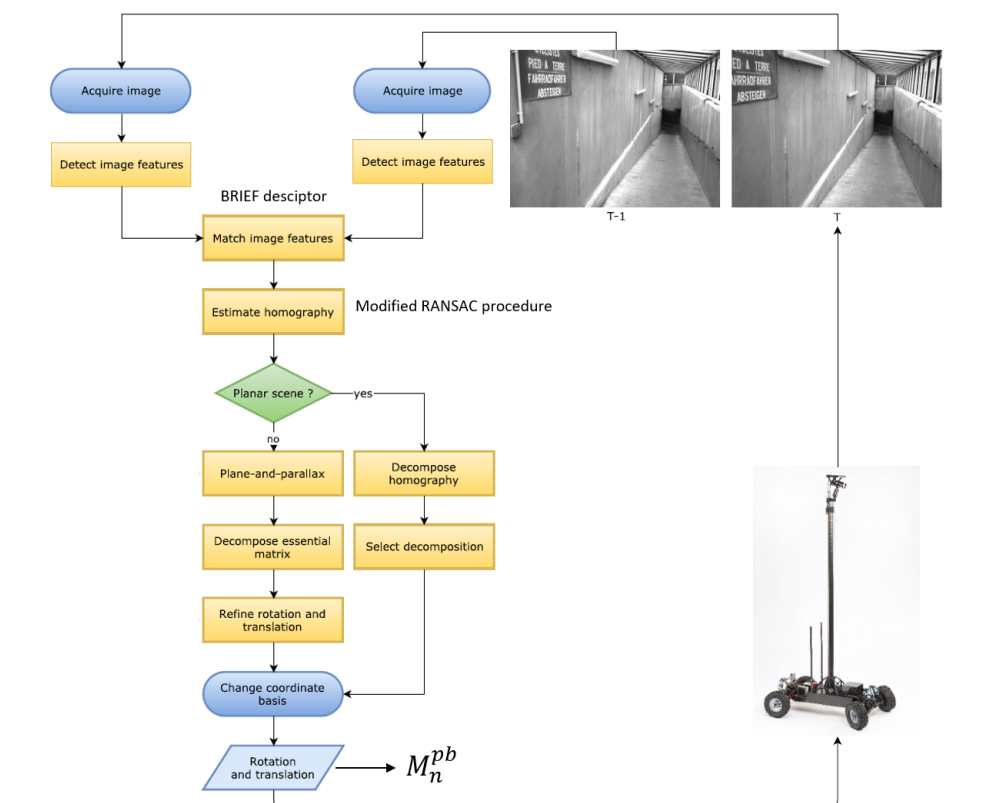


FIGURE 2.30 – Schéma bloc du fonctionnement de l’algorithme Parallax Beam, extrait de [103]

	Paral. Beam	5-point	8-point	Homogr.	ORB-SLAM Ini.
Sequence 00					
Translation [%]	0.92	3.77	1.04	17.28	1.65
Rotation [°/m]	0.0041	0.0200	0.0049	0.0708	0.0197
Sequence 02					
Translation [%]	1.12	3.80	0.95	24.33	7.95
Rotation [°/m]	0.0049	0.0170	0.0038	0.0718	0.0552
Sequence 03					
Translation [%]	0.77	3.77	0.69	16.39	2.185
Rotation [°/m]	0.0035	0.0217	0.0026	0.0705	0.0154
Sequence 10					
Translation [%]	1.06	2.79	1.32	31.12	1.72
Rotation [°/m]	0.0081	0.0189	0.0059	0.1053	0.0129

TABLE 2.2 – Résultats sur le jeu de données KITTI vision de différentes méthodes de navigation à l’estime par la vision comme l’initialisation d’ORB-SLAM [96], 5-point [85], 8-point [26] et une simple homographie. En vert sont signalées les erreurs les plus faibles.

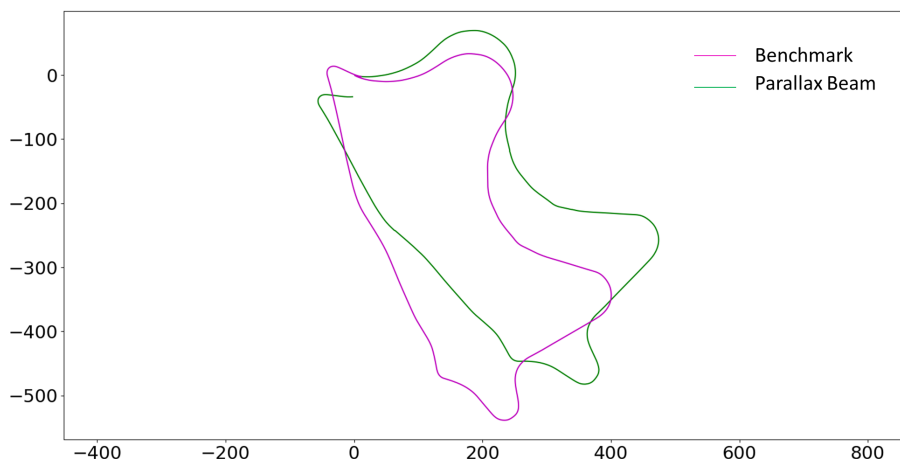


FIGURE 2.31 – Trajectoire estimée par Parallax Beam, en vert, sur la séquence de validation n° 9 du jeu de données KITTI vision. Axes en mètre.

2.5.3 Limitations de STAMINA

STAMINA, via des briques de navigation performantes comme celle du Parallax Beam, propose une navigation à l'estime précise et efficace dans des conditions idéales de fonctionnement, lorsque le plan principal perçu dans les images correspond au plan du sol. Cependant, en se reposant sur une unique caméra, cette solution de navigation se heurte aux problèmes propres à ce capteur extéroceptif. Ainsi, les problèmes d'indisponibilité de l'image par occultation partielle ou totale du champ de vision (feuillages, obstacles importants et très proches, changements rapides de luminosité, sable, orientation non conforme de la caméra du aux aléas du terrain, etc.), comme illustré sur la figure 2.32, peuvent dégrader la fiabilité du système voire entraîner une incapacité totale à réaliser la tâche de navigation.

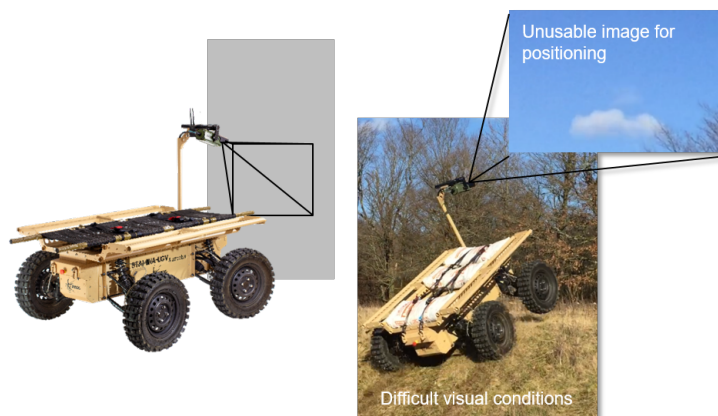


FIGURE 2.32 – Illustration des cas limites d'un algorithme de navigation à l'estime visuelle seule : l'occultation du champ de vision ou la mauvaise orientation de la caméra à cause du terrain à naviguer peuvent entraîner une indisponibilité de la fonction de navigation.

Pour s'en convaincre, on peut s'intéresser au comportement du Parallax Beam sur le jeu de données KITTI vision lorsque l'on coupe temporairement le flux d'images. Au bout de vingt secondes sur la séquence 9 du jeu de données, on coupe le flux d'images pendant 2, 5 et 10 secondes pour simuler une indisponibilité du flux d'images. Les résultats visibles ci-dessous (figure 2.33) attestent de la dégradation de la fonction de navigation par la vision.

Sans intention de modifier Parallax Beam, une solution semble alors de coupler la caméra à un second capteur. Pour limiter des risques additionnels de coupures d'informations, et toujours dans un contexte sans GNSS, la solution s'est portée sur un capteur proprioceptif. Une fusion Vision/IMU paraît alors être une solution satisfaisante au problème soulevé qui est d'améliorer la navigation à l'estime visuelle STAMINA face aux situations d'estimations mauvaises ou absentes liées à des indisponibilités d'images de qualités suffisantes. En effet, la nature proprioceptive de l'IMU ainsi que les deux informations qu'elle apporte grâce aux accéléromètres et aux gyromètres devrait être en mesure de permettre la navigation à l'estime dans les cas limites de la vision. De même, la précision de la navigation visuelle devrait être en mesure de corriger efficacement les dérives inertielles.

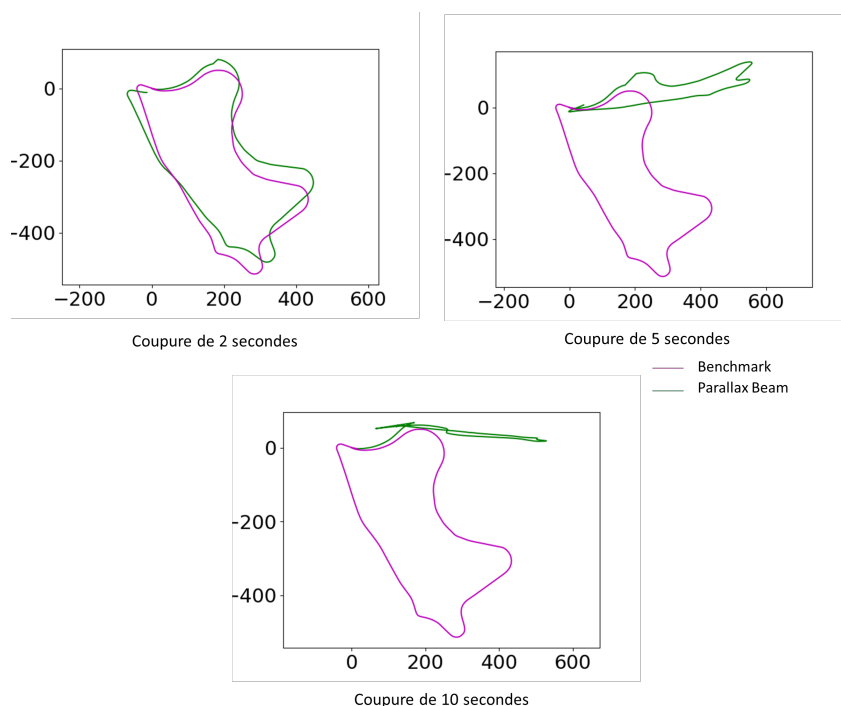


FIGURE 2.33 – Séquence 9 du jeu de données KITTI vision, Parallax Beam (en vert) subit une coupure du flux d'image au bout de 20 secondes pendant respectivement 2, 5 et 10 secondes. Axes en mètres.

En raison des activités de l'ISL et des perspectives de développement de STAMINA vers les UAVs et les projectiles, la solution proposée, en plus d'apporter une robustesse, actuellement limitée, doit être agnostique aux systèmes sur lesquels elle pourrait être

embarquée, comme illustré sur la figure 2.34.

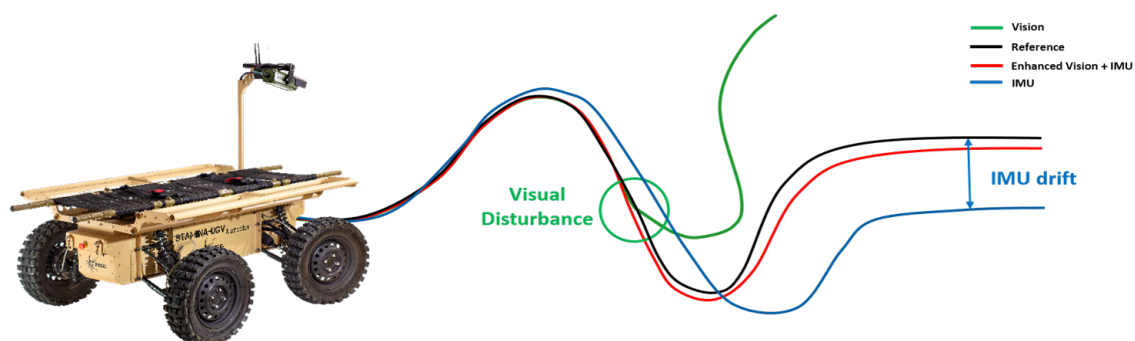


FIGURE 2.34 – Illustration de l’objectif d’un couplage visuel-inertiel : être capable de compenser la dérive (drift) inertielle et assurer la navigation en cas d’estimation aberrante de la navigation visuelle.

Les solutions de navigation visuelle-inertielle peuvent être trouvées dans la littérature sous le nom de systèmes VINS, une étude de ces systèmes doit alors être menée.

2.6 Conclusion

Après avoir défini ce qu’est la navigation, les différentes tâches qui la composent ainsi que les capteurs utilisés pour la réaliser, nous avons présenté les moyens d’évaluer une navigation et le jeu de données KITTI vision qui sera utilisé pour évaluer les travaux présentés dans les chapitres suivants. Nous avons pu aborder le fonctionnement de la plateforme robotique de l’ISL STAMINA, et en particulier son algorithme Parallax Beam, un algorithme de navigation à l’estime visuelle qui permet une précision élevée (environ 2% d’erreur moyenne en translation pour une trajectoire de 800 m) mais souffre des défauts inhérents aux caméras et autres capteurs proprioceptifs quand ils sont privés de la capacité de réaliser des acquisitions satisfaisantes. Forts de notre compréhension des outils de la navigation présentés dans cette partie, nous avons émis l’hypothèse qu’un couplage visuel-inertiel, grâce à l’ajout d’une centrale inertielle au système STAMINA, nous permettrait de résoudre la faiblesse majeure de la navigation des systèmes STAMINA. La solution proposée faisant partie de la famille des systèmes VINS (“Visual Inertial Navigation Systems”), elle nécessite donc une étude approfondie des méthodes de navigation VINS disponibles dans l’état de l’art, ce qui sera le sujet de la prochaine partie.

3 Etat de l'art des systèmes VINS

3.1 Introduction

Dans le chapitre précédent, nous avons choisi d'utiliser une IMU pour compléter la navigation à l'estime visuelle Parallax Beam. Cette solution se place parmi les solutions de couplage d'odométrie inertielle et visuelle dites VINS. Ainsi, ce chapitre est donc consacré à l'étude des systèmes VINS. Dans un premier temps, nous considérerons les solutions VINS selon la dichotomie couplage lâche/couplage serré, puis nous nous intéresserons au critère de robustesse au sein des systèmes VINS. Enfin, nous discuterons de notre préférence pour une solution de couplage lâche au vu des différentes méthodes précédemment exposées.

3.2 Méthodes de couplage lâche

Le couplage lâche regroupe des méthodes pour la plupart assez anciennes au sein des systèmes VINS. Néanmoins, la nature même du couplage lâche, introduit au chapitre 2, nous autorise à nous intéresser à des méthodes de couplage IMU + capteur extéroceptif puisqu'elles peuvent ensuite être adaptées au cas VINS. La majorité des méthodes repose sur des filtres stochastiques, en particulier bayésiens comme le filtre de Kalman (dont l'annexe A, page 121, détaille les principes de base) et ses variantes, mais on trouve également des filtres qui, contrairement au filtre de Kalman, considèrent que les variables d'états suivent des distributions non gaussiennes. Enfin, on peut également noter que le couplage lâche est sujet à quelques applications d'intelligence artificielle et d'apprentissage machine qui, bien que peu éprouvées à l'heure actuelle, présentent un axe intéressant d'étude.

3.2.1 Couplage lâche par filtrage de Kalman

La structure d'un couplage lâche par filtrage de Kalman est généralement assez simple, comme l'illustre la figure 3.1 qui montre un couplage GNSS/INS par filtrage de Kalman expliqué en détails dans [56]. En effet, le couplage GNSS/INS est très courant et pourvoit des ressources théoriques simples et claires pour mieux comprendre comment intégrer un couplage lâche avec un capteur inertielle et un capteur de correction extéroceptif (défini au chapitre 2).

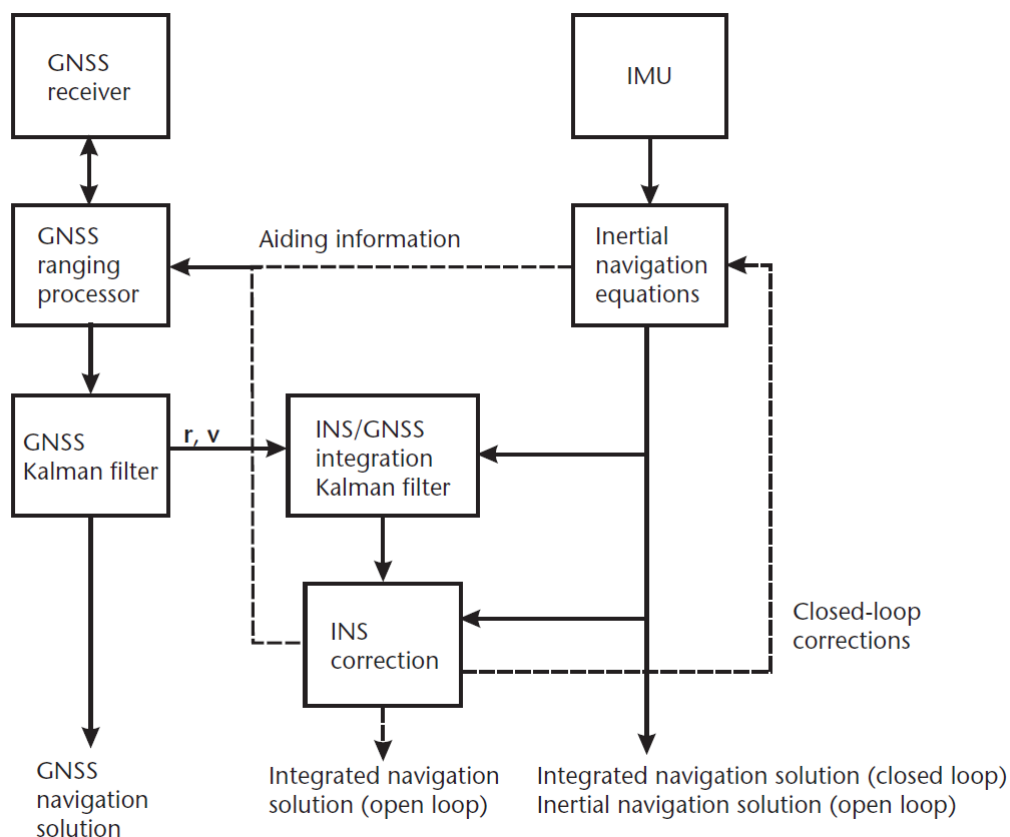


FIGURE 3.1 – Schéma-bloc typique d'un couplage lâche GNSS/INS, dont la structure en couplage lâche est adaptable à un couplage VINS [56].

On retrouve ici l'architecture attendue d'un couplage lâche avec, en premier lieu, un pré-traitement inertiel et un pré-traitement GNSS associés, dans un deuxième lieu par le biais du filtre de Kalman, afin d'appliquer une correction sur le résultat de la navigation inertielle pour fournir une estimée. L'avantage de ce couplage est qu'il est toujours possible de disposer des résultats de traitement avant l'étape d'intégration comme solution de navigation à part entière. L'indépendance de traitement des données des deux sources garantit la fonction de navigation en cas de coupure d'information de l'un ou l'autre des capteurs. Une fermeture de boucle, via un retour de l'estimée finale aux blocs de pré-traitement, peut être ajoutée pour apporter des corrections de dérives et améliorer la précision globale du système.

Le couplage EKF et ESKF

Dans le cadre du couplage GNSS/INS, la majorité des techniques se base sur le "Extended Kalman Filter" (EKF, estimateur pour systèmes non-linéaires présenté en annexe A, page 121) qui va alors évaluer l'erreur d'estimation du bloc INS grâce au bloc GNSS pour la compenser. Cette méthode, que l'on peut trouver plus en détails dans [5], [65],

[112] ou encore [93], n'estime pas l'état \mathbf{x} du véhicule mais un état-erreur $\delta\mathbf{x}$. Cette formulation sous forme d'état-erreur requiert certes la capacité d'exprimer les équations de modèle et de mesure du système sous une forme adaptée à ce nouvel état, mais elle permet de résoudre les problèmes d'alignement des repères des capteurs qui seront alors contenus dans l'erreur estimée. Cette structure de filtre est souvent nommée ESKF ("Error-State Kalman Filter").

Cette formulation d'un EKF pour la fusion multi-capteurs a également été employée pour la fusion VINS, dans les travaux de [105] par exemple avec une architecture de filtre nommée "Indirect Kalman Filter". Celui-ci utilise une copie de l'état-erreur pour n'effectuer l'étape de propagation, i.e. l'exploitation des données inertielles et du modèle du système, basée sur les informations inertielles que sur cette copie. Ainsi, l'état non copié n'évolue que sous l'effet de la mise à jour obtenue par les informations issues de l'algorithme d'estimation de pose à partir des images. Ceci a pour avantage d'augmenter la robustesse du système d'estimation à une erreur inertielle tout en améliorant la précision via un couplage. Ces travaux obtiennent ainsi, dans le cas de la navigation d'un robot à cinq degrés de liberté, des résultats considérés comme satisfaisants dans un environnement très contraint et maîtrisé lors de la réalisation de trapèzes de vitesses pré-définis.

Le couplage lâche multi-capteurs par le biais d'un ESKF peut également être utilisé afin de coupler plus de 2 capteurs, comme dans les travaux de [25]. Ici, une caméra monoculaire, un GNSS et une IMU sont couplés par le biais d'un filtre ESKF classique (tel que celui présenté au chapitre 4). Les données de navigation obtenues par le GNSS et la caméra (via un algorithme de navigation à l'estime visuelle 8-points [62]) sont centralisées par une première étape de couplage afin d'obtenir une estimation de la matrice de rotation du système \mathbf{C}_b^n et une covariance de l'erreur \mathbf{P}_C de cette estimation. Ce sont ces données qui sont alors utilisées pour la mise à jour du filtre, comme illustré sur la figure 3.2.

Cette structure a été testée sur une portion de route sujette à un signal GNSS perturbé. Elle obtient une erreur de position moyenne de 2.70 m et une erreur d'orientation moyenne de 0.64° sur les 300 s d'acquisition avec un Xsens Mti (IMU milieu de gamme) et une caméra à 7.5 Hz.

En parallèle des solutions basées ESKF et assimilés, plusieurs travaux se sont intéressés à des estimateurs EKF classiques pour la fusion VINS. A cet égard, les travaux [108] et [109] sont particulièrement intéressants. Ces travaux mènent une étude d'un EKF et d'un ESKF au travers de plusieurs variantes sur un UAV quadrirotors équipé d'une caméra monoculaire, d'un capteur de portée infrarouge et d'une IMU. Trois paramètres sont étudiés dans ces travaux : le calcul de l'erreur d'orientation par usage des quaternions, l'intégration des quaternions pour la prédiction à l'aide du modèle et l'ordre de la série de Taylor utilisée pour le calcul de la jacobienne de la matrice \mathbf{F} de modèle du système.

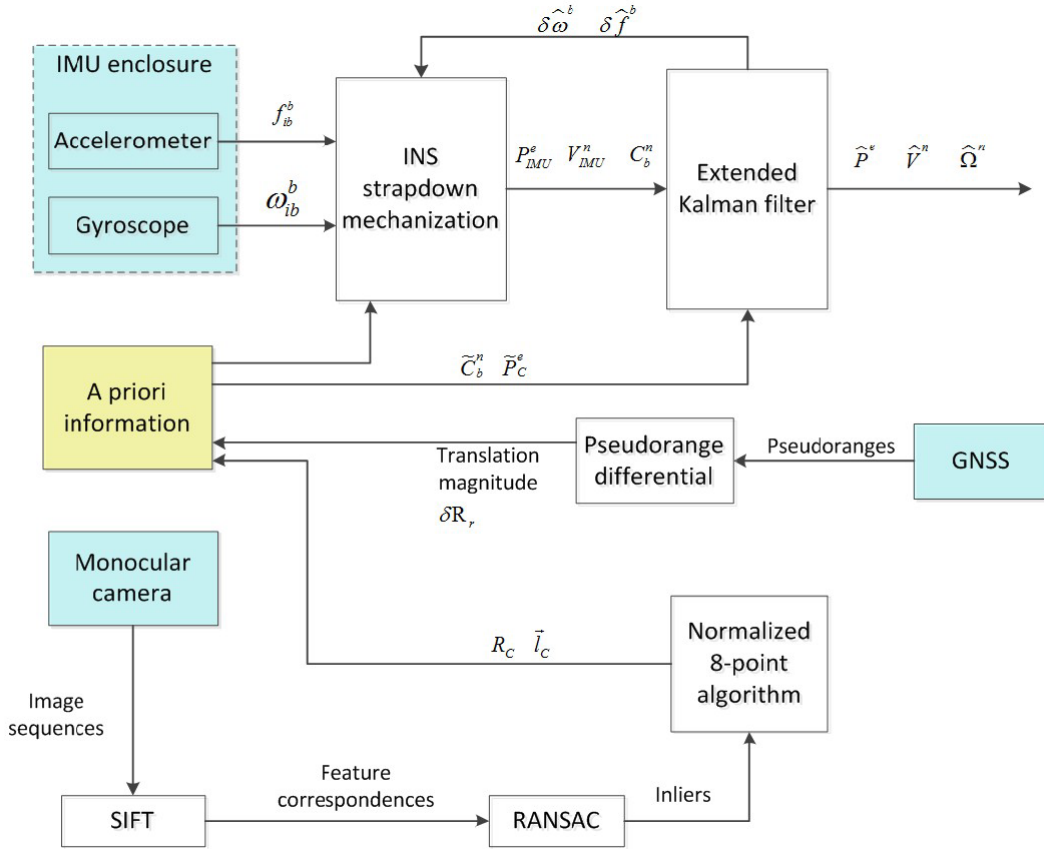


FIGURE 3.2 – Schéma-bloc d'un couplage lâche entre une IMU, une caméra monoculaire et un GNSS [25].

L'erreur d'orientation minimale peut être définie dans le repère local (LE, Local Error) ou global (GE, Global Error), selon que le produit soit effectué à droite ou à gauche du quaternion nominal :

$$\text{global error}(GE) : \mathbf{q}_t = \delta \mathbf{q} \otimes \mathbf{q} \quad (3.1)$$

$$\text{local error}(LE) : \mathbf{q}_t = \mathbf{q} \otimes \delta \mathbf{q} \quad (3.2)$$

Avec $\delta \mathbf{q}$, l'erreur d'orientation exprimée sous forme de quaternion.

Pour l'intégration des quaternions, trois manières de procéder sont testées et sont notées Q0F, Q0B et Q1 qui sont l'intégration avec la vitesse de rotation instantanée à respectivement l'instant k , à l'instant $k - 1$ ou une moyenne des deux.

Enfin, le dernier paramètre, noté \mathbf{F}_N , exprime l'ordre auquel on a tronqué la linéarisation en série de Taylor de la fonction d'évolution du modèle \mathbf{F} lors de la discrétisation :

$$\mathbf{F}_N \approx \sum_{n=0}^{N < \infty} \frac{1}{n!} \mathbf{F}^n \Delta t^n. \quad (3.3)$$

Des essais avec des filtres EKF et ESKF faisant varier ces différents paramètres ont ensuite été testés sur une même trajectoire de deux minutes et soixante-dix mètres

répétée quinze fois par le drone afin de calculer une RMSE. Les résultats, disponibles sur la table 3.1, donnent une information intéressante : pour le système à trois capteurs étudié, un EKF dans sa forme la plus simple (Taylor ordre 2, Q0F, LE) suffit à exploiter avec des bonnes performances (RMSE de 0.1984m sur 2 minutes de vol) l'ensemble des données de mesures.

Quat. error	GE			LE		
Quat. int.	Q0F	Q0B	Q1	Q0F	Q0B	Q1
EKF F_2	-	-	-	0.1984	0.2003	0.1993
ESKF F_1	0.1986	0.2007	0.1996	0.1986	0.2005	0.1995
ESKF F_2	0.1987	0.2007	0.1997	0.1987	0.2005	0.1996
ESKF F_3	0.1987	0.2007	0.1997	0.1987	0.2005	0.1996

TABLE 3.1 – RMSE (m) après deux minutes de vol. Résultat des essais effectués dans [108].

Apport des groupes de Lie

Cette tendance à favoriser un filtre EKF, plus simple et efficace dans la plupart des cas que le plus intuitif ESKF, se retrouve largement dans les algorithmes de SLAM qui seront traités dans la partie page 59. Une innovation importante pour le couplage lâche par filtrage de Kalman va plutôt être l'introduction des filtres de Kalman invariants [8]. Le filtre de Kalman invariant essaye de répondre au principal problème des filtres EKF. En effet, la matrice d'évolution est une linéarisation autour d'un point de fonctionnement et n'est donc valable que localement et non pas sur l'ensemble de la trajectoire. Le modèle de l'EKF peut par conséquent ne plus être valide dans certains cas, détériorant d'autant la qualité de la navigation. Une solution antérieure aux filtres invariants face à ce problème était l'usage de filtres contenant des modèles différents et discriminés par l'usage de modèles de Markovs comme dans [88] et [126] ou bien l'usage de l'UKF (Unscented Kalman Filter, filtre de Kalman sans parfum) comme détaillé dans [28].

La solution préférée par l'IEKF (Invariant Extended Kalman Filter, filtre de Kalman étendu et invariant) est plutôt de recourir aux propriétés des groupes de Lie $SO(3)$ sur les systèmes possédant des symétries ou des invariances (qui ne seront pas détaillées ici mais dont une explication peut-être trouvée dans [8] ou en annexe de [28]) pour définir un terme correcteur basé sur une erreur de sortie non pas linéaire, mais invariante. Avec le même procédé, la matrice de gain de Kalman n'est plus mise à jour à l'aide d'une erreur d'état linéaire mais une erreur d'état invariante. Pour que cela fasse sens d'un point de vue stochastique, on émet l'hypothèse que les bruits d'état et de mesure sont invariants.

Le principal bénéfice d'un IEKF est alors que les matrices \mathbf{F} et \mathbf{H} sont constantes

sur une portion bien plus importante de la trajectoire que simplement au point de linéarisation. On peut constater l'apport de cette solution dans [12], [13], [59] ou [28], où les méthodes basées sur le groupe de Lie font preuve de performances supérieures à la méthode UKF tout en ayant un coût de calcul plus faible, comme on peut le constater sur la figure 3.3. On constate que les RMSE confirment le net gain de performance apporté par les IEKF pour un temps d'exécution faible, en particulier le RIEKF qui ne requiert que 4.3 secondes de temps de calculs pour 10 secondes de temps de vol en moyenne, là où ses concurrents ont besoins de dix fois plus de temps (cf. figure 3.3).

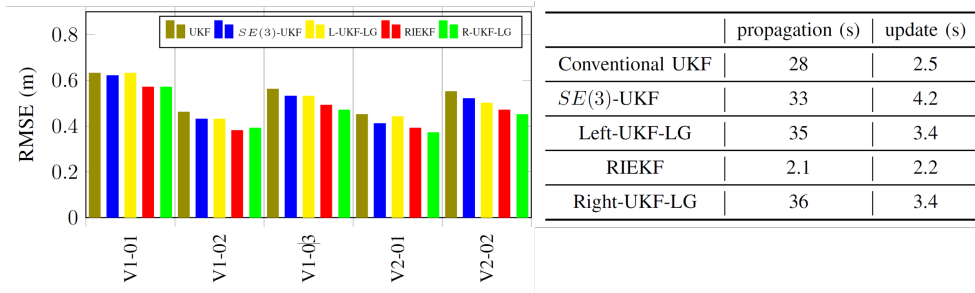


FIGURE 3.3 – Résultats obtenus par des méthodes basées sur les groupes de Lie et un UKF testés sur six trajectoires de EuRoC [16]. À gauche les RMSE, à droite les temps de calculs moyens pour 10 s de vol pour les étapes de prédictions et de mise à jour des filtres [12].

On peut également noter l'efficacité des filtres invariants vis-à-vis de solutions de couplage serré comme c'est le cas dans [13] où l'IEKF rivalise avec comme VINS-Mono ou le MSCKF ("Multi State Constraint Kalman Filter"). La figure 3.4 illustre les bons résultats des méthodes basées sur les groupes de Lie qui performant mieux ou aussi bien que leurs concurrentes tout en maintenant des charges processeurs faibles.

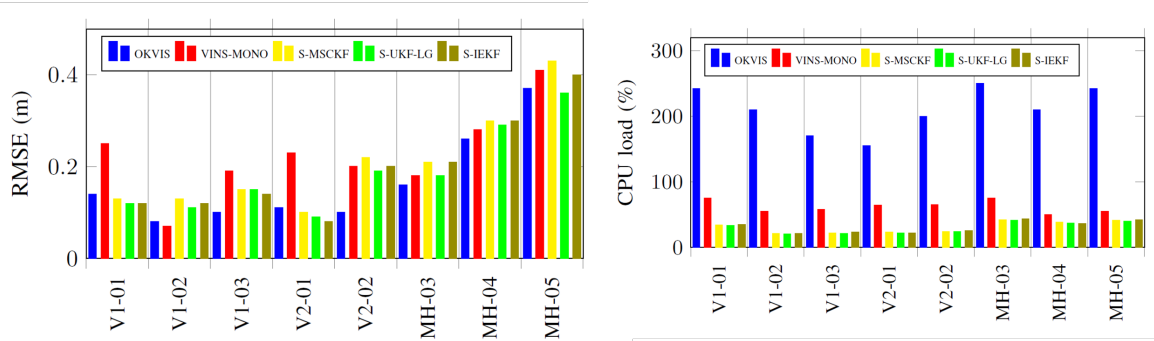


FIGURE 3.4 – Résultats obtenus par des filtres basés sur les groupes de Lie (notées S-UKF-LG et S-IEKF) sur différentes séquences de EuRoC [16]. À gauche les RMSE, à droite les charges processeur maximales [13].

Fusion multi capteurs

En parallèle de ces travaux, d'autres études ont été poursuivies sur les filtres EKF et ESKF, avec une attention particulière sur le rejet de valeurs aberrantes. C'est le cas dans [119] qui utilise un filtre médian sur le quaternion d'estimation d'attitude du système. Ceci permet de rejeter les valeurs aberrantes afin de déterminer le facteur d'échelle d'un système VINS avec une bonne précision et rapidement.

Ces outils vont permettre la mise en place du filtre MSF-EKF [92] (Multi Sensor Fusion EKF). Son objectif est d'assurer une navigation robuste et précise en associant plusieurs capteurs, en gérant leur calibration, l'asynchronisme ainsi que les indisponibilités et délais. Faisant suite aux travaux de [119], le MSF-EKF est capable de traiter des mesures absolues et relatives non synchrones à partir d'un nombre important de sources (les travaux ne présentent que des exemples à trois capteurs). Capable de gérer l'ajout en ligne de capteurs, il est aussi apte à fonctionner si un capteur fait défaut grâce à un tampon circulaire à état dont une illustration est donnée en figure 3.5. Ce tampon assure une propagation de l'état fourni par chaque capteur pour être au plus proche du point de linéarisation optimal de la trajectoire suivie. Le tampon circulaire fonctionne ainsi :

- a) L'état courant sert de référence et on garde en mémoire les prédictions antérieures réalisées par propagation du modèle inertiel. La covariance de l'erreur n'est calculée qu'au premier point du buffer.
- b) Une mesure retardée arrive et va se voir attribuer un emplacement adapté dans le buffer par correspondance avec les états présents. La covariance prédite est propagée à cette mesure afin qu'elle soit corrigée.
- c) L'état nouvellement mis à jour va être propagé jusqu'à l'instant courant par plusieurs prédictions successives. La covariance ne sera de nouveau propagée que lors d'une nouvelle mesure.

De plus, les paramètres de calibration entre les capteurs peuvent être estimés par le filtre pour permettre une automatisation de cette étape.

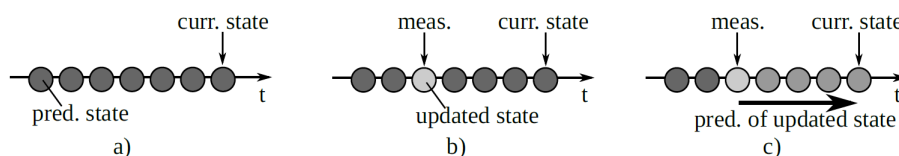


FIGURE 3.5 – Illustration du ring buffer utilisé dans une MSF-EKF pour compenser les délais de mesures [92].

Cette technique est utilisée sur des UAVs dans les travaux [120] et [39], à chaque fois pour des applications de SLAM pour lesquelles le MSF-EKF sera préféré à un EKF standard pour réaliser la tâche de localisation du véhicule. La figure 3.6 détaille une version simplifiée de l'architecture employée pour la localisation du mobile (un UAV hexacoptère "Firefly"). On constate la grande différence des fréquences d'acquisition des différents capteurs qui permet d'exploiter la pleine cadence de chacun. Un MSF-EKF (notée EKF sensor fusion ici) assure l'ensemble de la tâche de localisation à partir des mesures inertielles et de poses estimées à partir d'images d'une caméra monoculaire. Dans ces deux travaux, le MSF-EKF permet d'obtenir des performances de localisation en situation réelle et complexe (terrains accidentés, ruines) assez précise et robuste pour réaliser des cartographies des sites à explorer pour des applications de recherche et sauvetage.

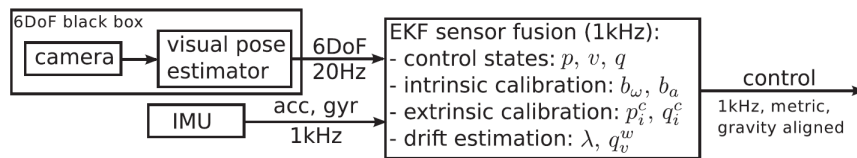


FIGURE 3.6 – Schéma de principe de l'organisation de la tâche de localisation du mobile [120].

3.2.2 Couplage tâche par filtrage particulière

Les filtres de Kalman émettent l'hypothèse que les variables d'état stochastiques sont des variables aléatoires gaussiennes. Bien que cette hypothèse soit valable dans la plupart des cas et supportée par le théorème central limite si l'on dispose d'un nombre suffisamment important d'échantillons, elle n'est pas forcément la plus adaptée pour tous les problèmes. Les filtres particuliers sont une famille de filtres à la structure similaire à celle d'un filtre de Kalman (en particulier un UKF) mais qui peuvent être conçus pour des distributions aléatoires non gaussiennes.

On peut trouver un très bon exemple de ce type de filtre sont les filtres t de Student [2],[3], qui utilisent la distribution t afin de réaliser une tâche de navigation multi-capteurs qui utilise un GNSS, des odomètres de roues, un gyromètre pour le lacet et une caméra monoculaire. Un filtre de Student a une structure en 2 étapes similaires à celles d'un filtre de Kalman (prédiction, mise à jour), comme expliqué dans [3]. Le grand nombre de capteurs employés rend nécessaire l'usage de méthode de détections de valeurs aberrantes en amont de l'étape de mise à jour.

Les travaux présentés dans [21] et [22] montrent que, pour les filtres particuliers, la considération des groupes de Lie amène aussi des résultats intéressants. [22] présente un usage d'un filtre particulier de Laplace. Ce type de filtre requiert que l'état suive un processus de Markov et réalise son étape de mise à jour à partir de l'évaluation

d'un maximum a posteriori pour ré-échantillonner l'ensemble des particules obtenues par l'étape de prédiction afin de fournir une estimation. Les travaux présentés utilisent les groupes de Lie comme un cadre mieux adapté au traitement des grands angles que les filtres standards euclidiens qui réalisent souvent l'approximation des petits angles. Les filtres basés sur des structures différentes du filtre de Kalman présentent ainsi des évolutions intéressantes mais pour le moment encore un peu moins performantes en terme de minimisation de la RMSE que les filtres standards.

3.2.3 Apports des méthodes d'apprentissage machine

En parallèle des développements sur les méthodes de filtrages, des techniques d'apprentissage machine se sont développées. Celles-ci présentent l'avantage d'être adaptables au problème qu'elles entendent résoudre, généralement moyennant la nécessité de disposer d'une grande quantité de données pour correctement dimensionner ou entraîner l'algorithme.

Les premiers développements d'apprentissage machine pour la navigation ont débuté avec la logique floue comme dans [70], [71], [72] ou encore [90]. [70] par exemple présente une architecture intéressante pour du couplage INS/GPS qui utilise un "Strong Tracking Kalman Filter" (STKF) [129], un filtre héritant de l'EKF. Ce filtre règle la problématique des changements de dynamique appliqués aux systèmes en ajoutant un facteur d'échelle variant avec le temps pour réduire le poids de la matrice de covariance de l'état prédit. Ainsi, dans les situations où le modèle n'est plus en cohérence avec les mesures à cause d'un changement de dynamique important appliquée au véhicule, comme dans un virage, le filtre STKF augmentera temporairement sa confiance dans sa mesure. Ce paramètre, normalement défini par des conditions de seuil est modifié dynamiquement par un système adaptatif à logique floue (ou FLAS, Fuzzy Logic Adaptive System) à partir de l'innovation du filtre comme sur l'image 3.7.

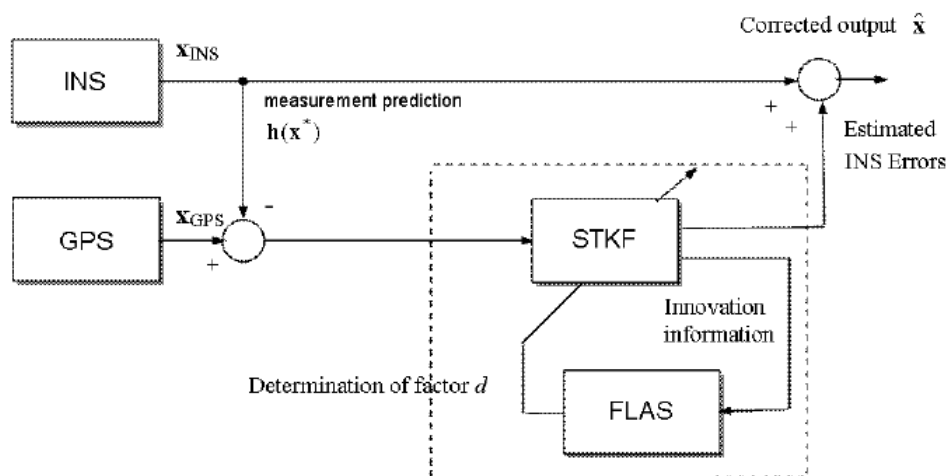


FIGURE 3.7 – Schéma bloc de l'Adaptative Fuzzy Strong Tracking Kalman Filter [70].

Ces travaux ne donnent pas de performances quantifiables mais montrent que le STKF augmenté par la logique floue est en mesure de mieux répondre aux changements de dynamique importants.

Il faut néanmoins noter que la logique floue est une technique assez âgée qui a perdu beaucoup en visibilité au profit d'autres méthodes d'apprentissage machine comme les réseaux de neurones.

Une application particulièrement intéressante est d'utiliser un réseau de neurones à convolution, ou CNN, réseau dont chaque couche opère des opérations de convolution sur les données. Par exemple, [11] propose un CNN afin d'automatiser un réglage dynamique d'une des deux matrices de covariances des bruits d'un filtre IEKF pour la navigation inertielle, comme illustrée sur la figure 3.8. Testée sur KITTI, cette technique affiche des performances très intéressantes avec une RMSE moyenne de 0.97 % en translation et $2.3^\circ/\text{km}$ en rotation via une navigation purement inertielle. Cette technique a également fait ses preuves dans les travaux de [106] pour la navigation de projectiles.

Des travaux visant à utiliser des réseaux de neurones comme seuls moyens de réaliser la navigation et le couplage ont aussi été développés. La plupart repose sur des réseaux de type "Long-Short Term Memory" (LSTM) [63], réseaux de neurones récurrents dont la particularité réside dans l'existence de rétroactions entre ces différentes couches, les rendant particulièrement adaptés au traitement des séries temporelles. La plupart des méthodes utilisant des LSTM pour la navigation utilisent ces réseaux de neurones comme des unités de traitement des données directement extraites des capteurs. Pour cette raison, elles constituent des applications de couplage lâche récentes permettant d'étudier des couplages IMU + capteurs.

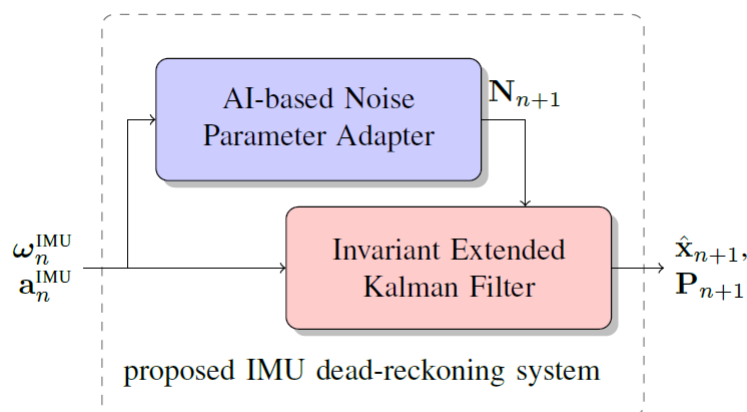


FIGURE 3.8 – Schéma bloc d'un IEKF augmenté par un réseau de neurones pour estimer une des matrices de covariances des bruits (notées N) [11].

Les travaux dans [27] présentent une première approche intéressante de l'utilisation de LSTM pour la navigation. L'intention première est de réduire le nombre de paramètres de réglages de la solution de navigation en plus de l'automatiser via l'apprentissage du réseau comme illustré sur la figure 3.9. Celle-ci met en avant, en bleu, l'ensemble des paramètres qui vont demander un réglage à l'opérateur. On peut toutefois noter que les liaisons du réseau de neurones vont être apprises en autonomie par le réseau, sans action de l'utilisateur.

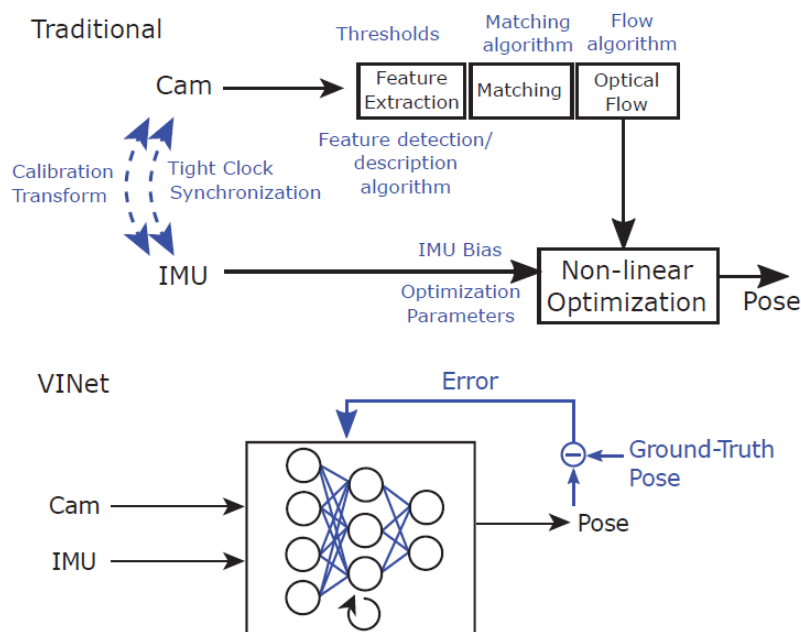


FIGURE 3.9 – Schéma illustrant les avantages d'une solution d'apprentissage profond pour la fusion, ici VINet (Visual-Inertial Net) [27]. En bleu, on trouve tous les paramètres qui requièrent une action de l'utilisateur du système.

La solution proposée consiste en deux réseaux qui vont chacun traiter les données d'un seul capteur, soit un CNN pour les images issues de la caméra et un LSTM pour l'IMU, à la cadence dudit capteur comme on peut le voir sur la figure 3.10. Un dernier LSTM central va alors réaliser la fusion. Un avantage important de cette méthode est la garantie de disposer du maximum d'informations délivrable par chaque capteur, avec l'inconvénient de devoir entraîner plusieurs réseaux simultanément.

Évalué sur le jeu de données EuRoC [16], une première qualité de cette approche est la robustesse à des erreurs de calibration dépassant des méthodes de SLAM performantes comme OKVIS [84] qui sera présenté dans la section 3.3. En effet, en entraînant les réseaux avec des données volontairement mal calibrées, ceux-ci parviennent alors à devenir robustes à des erreurs de calibration jusqu'à 15° avec une perte minimale de précision (la RMSE passe de 0.1842 m à 0.5178 m) là où OKVIS échoue.

Évalué sur KITTI [54], ce système parvient à montrer des performances de précision

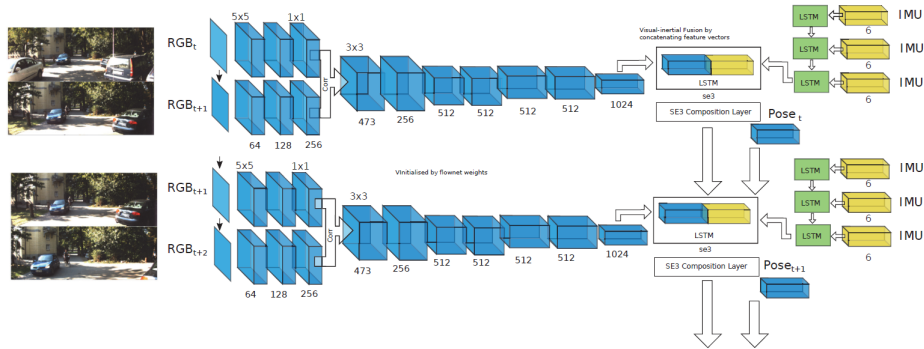


FIGURE 3.10 – Architecture de VINet [27] qui réalise une navigation VINS uniquement à l'aide de réseaux de neurones.

en translation meilleures qu'un MSF-EKF [121] utilisant LIBVISO2 [53] pour son traitement visuel sans parvenir à le dépasser pour ce qui est de la précision en orientation comme illustré par la figure 3.11.

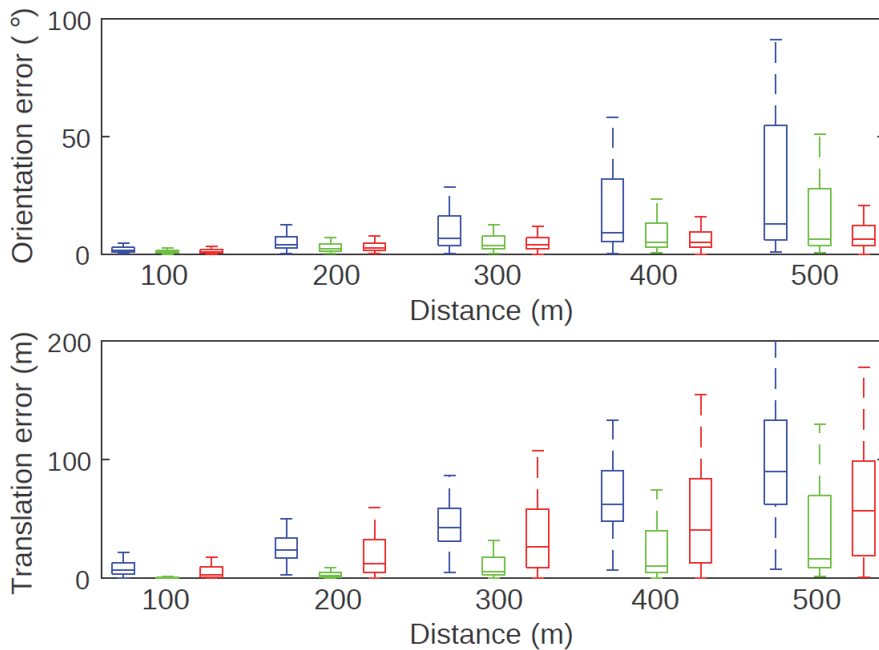


FIGURE 3.11 – Résultats de VINet [27] sur une séquence de KITTI. Sont comparés VINet sans apport inertiel en bleu, VINet complet en vert et un MSF-EKF en rouge.

Une partie intéressante des applications de navigation à base de réseaux LSTM essaye d'utiliser ceux-ci comme un substitut au capteur de soutien de l'IMU en cas d'aléas, en particulier dans le cas d'un couplage IMU et GNSS [128] [117] [40]. Dans ce cas, le concept est simple avec un algorithme en deux temps. Quand le signal GNSS est disponible, celui-ci est utilisé conjointement avec l'IMU pour la navigation et l'entraînement d'un réseau LSTM. Ce dernier est alors utilisé comme substitut au GNSS en

cas d'indisponibilité en ne se basant que sur des entrées inertielles comme illustré sur la figure 3.12.

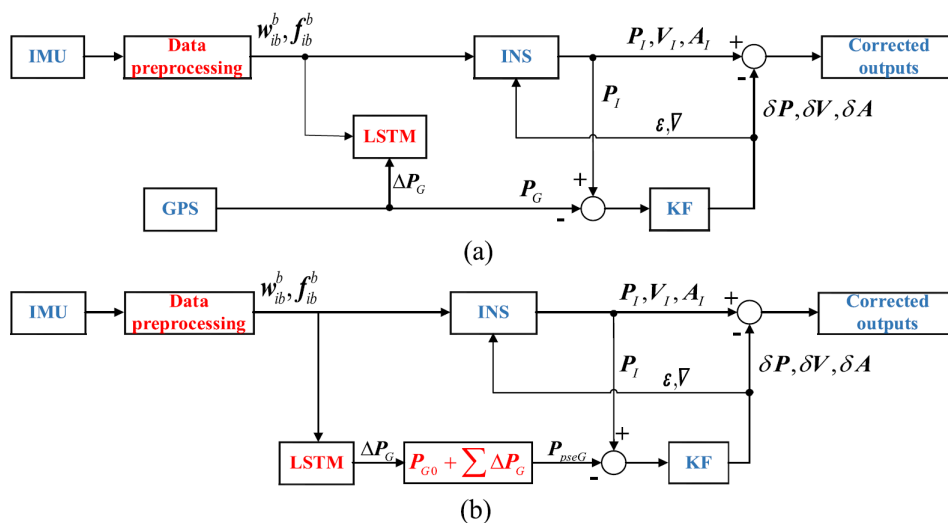


FIGURE 3.12 – Schémas blocs (le premier pour l'apprentissage, le second pour l'utilisation en cas de perte) d'un système de couplage INS/GNSS qui utilise un LSTM pour apprendre le comportement du GNSS et le remplacer lors de coupures du signal [128].

Les simulations effectuées dans [128], dans des situations de coupure GNSS (GPS ici) réalisées sur route, montrent que cette solution est performante avec une RMSE en longitude et latitude de [0.3373, 1.1699] m, durant une période de coupure GPS de 30 secondes, là où l'IMU laissée seule affiche une RMSE de [150.0310, 70.7885] m. Ces bons résultats peuvent se retrouver dans les travaux [117] et [40] qui appliquent ce type de structure pour des véhicules à sustentation magnétique et pour des automobiles sous des conditions de coupures GNSS plus importantes (180 s) avec des résultats satisfaisants.

Une implémentation intéressante des réseaux de neurones vise à reproduire le comportement prédiction/mise à jour d'un filtre de Kalman à l'aide d'un réseau LSTM [64]. Ici, le réseau est utilisé pour apprendre le fonctionnement d'un EKF et la modélisation de l'erreur de l'IMU quand le GNSS est disponible. Entraîné et testé sur KITTI, on peut constater sur la figure 3.13 que ce filtre de Kalman dit profond est plus précis selon la RMSE qu'un EKF classique.

Enfin, un élément important de la navigation est le contexte dans lequel celle-ci s'effectue. C'est pour cela que de nombreux travaux sur la détection de contextes environnementaux pour le couplage multi-capteurs ont été entrepris. On retrouve une synthèse dans [41] qui propose une structure pour la navigation augmentée par de la détection de contexte, décrite sur la figure 3.14, en plus d'une revue des multiples techniques employables pour cela. Ces méthodes de détection de changements de contextes sont décrites pour des signaux GNSS et issus de caméras. Certaines de ces techniques reposent sur des méthodes de segmentation classique comme la détection de coins ou de

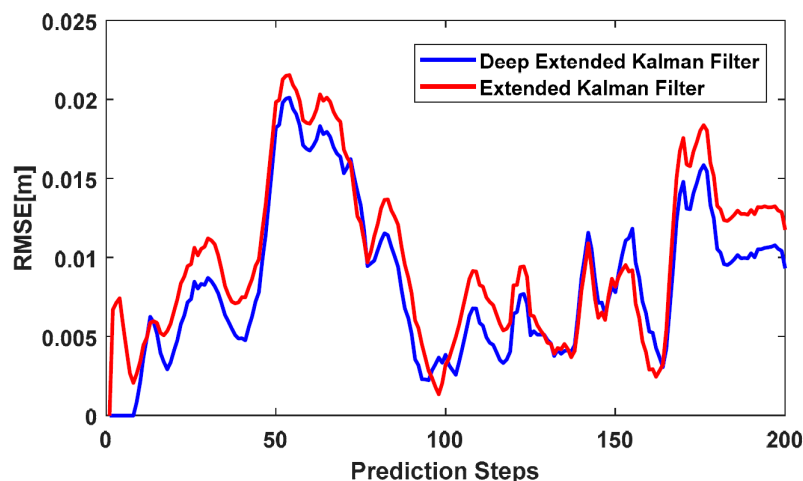


FIGURE 3.13 – Résultats d'un filtre Deep EKF [64] sur la séquence #34 de KITTI comparée à un EKF standard.

bords tandis que d'autres proposent d'utiliser des méthodes d'apprentissage profond, en particulier des réseaux CNN, pour réaliser ces détections. Cette piste d'étude est particulièrement bien développée pour les capteurs GNSS, mais semble néanmoins encore manquer de maturité pour les caméras et ne sera donc pas développée ici.

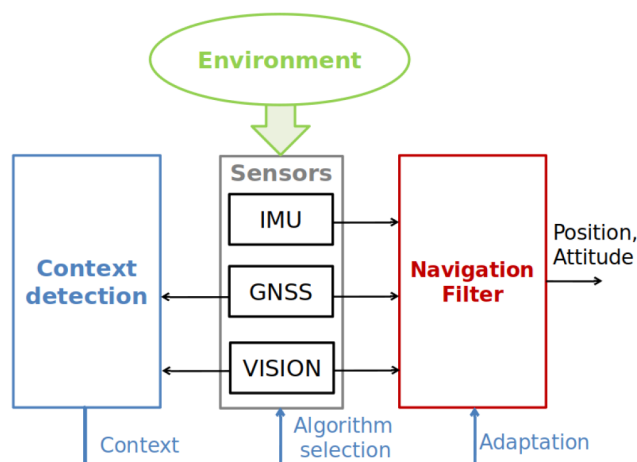


FIGURE 3.14 – Exemple d'une structure de navigation incluant des outils d'évaluation du contexte afin d'améliorer les fonctions de navigation [41].

Dans leur ensemble, les méthodes basées sur des formes d'intelligence machine proposent des performances et des atouts intéressants, en particulier dans le cas des méthodes utilisant l'apprentissage profond. Ces dernières nécessitent alors un large ensemble de données disponibles pour l'entraînement. Néanmoins, ces techniques, performantes dans des situations similaires à celles pour lesquelles elles ont été entraînées, sont très vulnérables à des changements de contexte comme le véhicule utilisé par exemple.

3.3 Méthodes de couplage serré

Les méthodes de couplage serrés sont les plus récentes et performantes au sein des systèmes VINS selon des critères de fiabilité et de précision. Souvent plus contraignantes à implémenter, elles sont néanmoins très intéressantes car régulièrement utilisées pour les applications de SLAM. Ici également, les méthodes de filtrages optimales comme le filtre de Kalman ont une forte présence.

3.3.1 Omniprésence du SLAM

Les méthodes de couplage reposent généralement sur l'appariement d'amers visibles à l'image (i.e. points de repères identifiables d'une image à une autre) avec les données inertielles acquises au même instant. Pour cette raison, elles sont parfaitement compatibles à la résolution du problème de SLAM, illustré par la figure 2.12, au chapitre précédent. Étudier les différentes méthodes de couplage serré revient alors à étudier les méthodes employées afin de réaliser un SLAM. Ici, nous ne nous intéresserons qu'aux méthodes qui utilisent une caméra malgré le fait que le lidar soit un capteur aussi exploité pour le SLAM (comme dans [33] par exemple).

Le V-SLAM et le Visual-Inertial-SLAM (VI-SLAM) sont des sujets majeurs de la recherche en navigation dont l'étude débute dans les années 1980 grâce à l'introduction de l'EKF. Comme présenté dans [67] dont la figure 3.15 est extraite, il existe une large variété de techniques qui contribuent soit à un gain pour la précision de l'estimation, soit à une amélioration de la robustesse de la navigation face à un défaut d'information via des méthodes comme la fermeture de boucle [4].

Il faut toutefois noter que, malgré l'avancement affiché des techniques de V-SLAM, les solutions comme l'EKF-SLAM font toujours l'objet de recherches et d'améliorations. La possibilité de réaliser un SLAM avec des méthodes de couplage lâche comme le MSF-EKF [39] ayant déjà été abordée dans la partie précédente, on peut ici encore remarquer l'inclusion des groupes de Lie pour venir renforcer le SLAM qui a souvent des degrés de liberté non observables [10]. Ces méthodes utilisent le calcul de l'erreur invariante à droite pour proposer un IEKF dont les performances, présentées dans [12] sur 100 simulations de Monte Carlo, proposent des RMSE inférieurs de 2.5 cm en translation et de 0.2° en attitude à celles d'un UKF dans la même situation. Ce même UKF avait été testé sur un UAV avec des résultats concluants dans [14].

Outre ces développements très prometteurs avec les groupes de Lie, une méthode basée EKF très intéressante est ROVIO ("RObust Visual Inertial Odometry") [7] qui illustre très bien la méthode du SLAM. Construit de prime abord comme un filtre EKF standard, il estime l'état :

$$\mathbf{x} = (\mathbf{r}, \mathbf{v}, \mathbf{q}, \mathbf{b}_a, \mathbf{b}_\omega, \mathbf{c}, \mathbf{z}, \mu_0, \dots, \mu_N, \rho_0, \dots, \rho_N) \quad (3.4)$$

Avec :

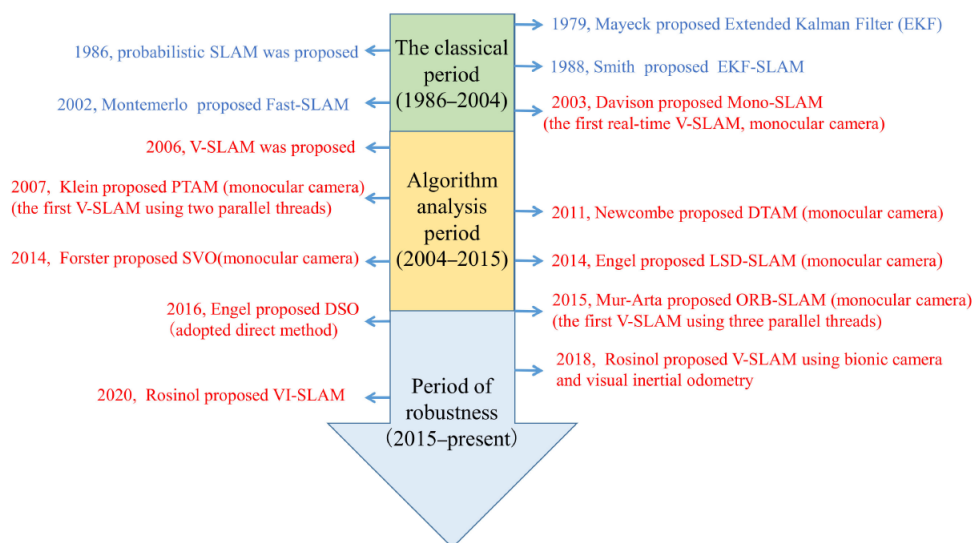


FIGURE 3.15 – Historique des technologies V-SLAM divisée en 3 périodes : une époque classique des technologies SLAM, une époque de développement du V-SLAM à proprement parler (en rouge) et enfin une époque d'amélioration de la robustesse du V-SLAM et de l'introduction du VI-SLAM [67].

- \mathbf{r} : la position du véhicule dans le repère inertiel,
- \mathbf{v} : la vitesse du véhicule dans le repère inertiel,
- \mathbf{q} : l'orientation du véhicule dans le repère inertiel (par rapport au repère monde ou repère de référence) exprimé sous la forme d'un quaternion,
- \mathbf{b}_a : le biais de l'accéléromètre exprimé dans le repère inertiel,
- \mathbf{b}_ω : le biais du gyromètre exprimé dans le repère inertiel,
- \mathbf{c} : le décalage en translation entre le repère inertiel et le repère caméra,
- \mathbf{z} : le décalage en orientation entre le repère inertiel et le repère caméra, exprimé sous la forme d'un quaternion,
- $\boldsymbol{\mu}_i$: le vecteur de direction pointant vers l'amer i , exprimé dans le repère caméra,
- ρ_i : la distance du véhicule à l'amer i .

Ici, le vecteur d'état va donc changer en taille au fur et à mesure de la détection et de la création de nouveaux amers que le filtre va traquer en permanence, ce que l'on va aussi retrouver dans OKVIS ("Optimized Keyframe Visual Inertial SLAM") [84] par exemple.

Tandis que la partie inertielle est propagée suivant les équations déjà présentées dans la partie 2.3.3, les directions des amers sont prédites à partir des vitesses linéaires et vitesses de rotation de la caméra et la projection des directions 3D dans l'espace 2D tangent au vecteur de direction à l'instant précédent. De la même manière, la mise à jour des informations associées aux amers nécessite une caméra correctement calibrée dans le but d'associer les vecteurs de direction des amers à des coordonnées en pixels sur l'image pour la comparaison avec la mesure. A chaque nouvel amer détecté,

l'ensemble de la routine de ROVIO, qui inclut plusieurs moyens de rejeter les amers non intéressants, va ainsi s'exécuter. Elle a été validée en réalisant la stabilisation en vol d'un UAV du décollage à l'atterrissage avec succès. Ainsi, la majorité des couplages serrés VINS vise une application de SLAM.

Le MSCKF [95] est une méthode qui repose sur un fonctionnement similaire à celui de ROVIO et des EKF-SLAM qui augmentent l'état et la matrice de covariance de l'état au fur et à mesure des acquisitions de la caméra. Cependant le MSCKF augmente son état non pas avec des informations sur des amers sélectionnés mais enregistre la position et l'orientation de la caméra dans le repère inertiel à chacune de ses acquisitions. De plus, ce filtre va plutôt s'intéresser à l'état-erreur plutôt qu'à un état standard. Le modèle de prédiction est alors un modèle de prédiction inertiel standard comme on peut le trouver dans un ESKF assorti des opérations de translations et d'orientations adéquates pour maintenir les poses caméras dans le repère inertiel courant. Ici, les amers sont suivis et associés à l'ensemble des poses caméras où ils apparaissent. Dès qu'un amer cesse d'être observable en quittant le champ de vision de la caméra, le MSCKF va utiliser l'ensemble des poses associées à l'amer en question afin de réaliser la mise à jour du filtre comme illustré par la figure 3.16.

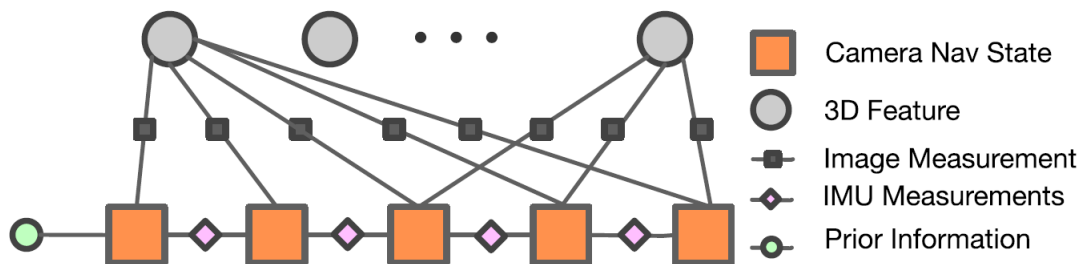


FIGURE 3.16 – Schéma descriptif du principe du MSCKF. On voit clairement l'appariement entre les amers 3D ("3D feature") et les poses caméras ("Camera Nav State")[60].

En effet, l'ensemble des mesures du même amer dans un espace 3D permet de définir une équation de contrainte associant l'ensemble des poses caméra. Cette fenêtre d'observation d'un amer est maintenue à une taille assez faible (3 poses) qui réduit fortement le coût en calcul mais aussi la précision potentiellement achievable par ce filtre. Bien que prometteur, il affiche toutefois dans ces essais réels une erreur terminale de 10 m pour un trajet de 3.2 km illustré en figure 3.17. Cette expérimentation est réalisée avec une caméra Pointgrey FireFly et une IMU Inertial Science ISIS.

Du fait de ces résultats encourageants, le MSCKF a été l'objet de nombreux développements : étude d'observabilité menant à une amélioration du calcul de la Jacobienne du modèle d'observation [86] [87], optimisation non-linéaire de la fenêtre glissante des états associés aux amers rencontrés afin de minimiser la perte d'information due à la taille de

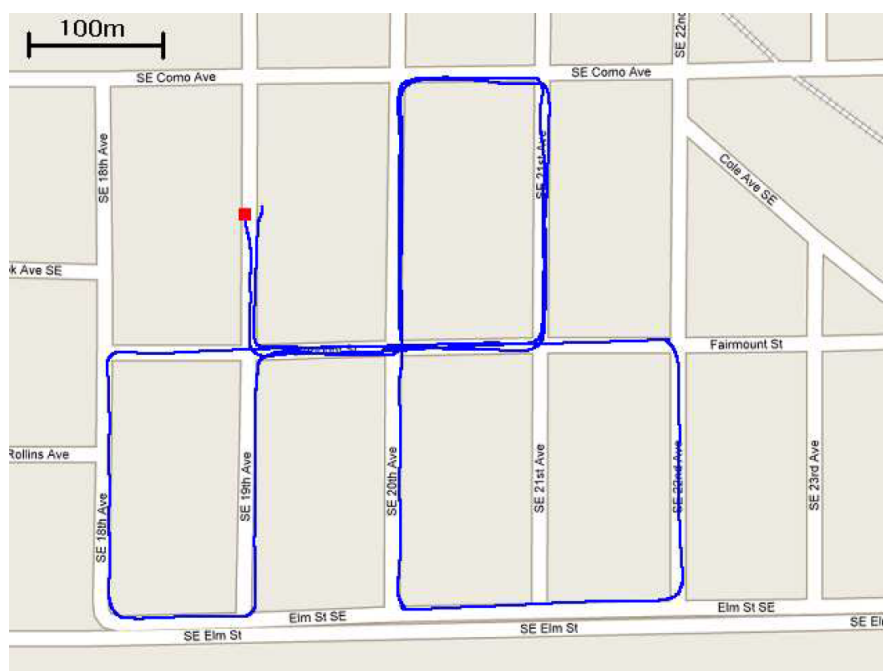


FIGURE 3.17 – Estimée du trajet réalisé dans [95] afin valider le MSCKF. On constate l'écart de l'estimée à l'arrivée (le carré rouge, symbolisant aussi le départ)[95].

la fenêtre [60], amélioration via les groupes de Lie [61], réalisation sous forme adaptative pour modifier dynamiquement les matrices de covariance des bruits de mesure afin de les adapter aux informations contenues dans la fenêtre utilisée pour la mise à jour [35]. A chaque fois, les améliorations proposées ont permis de réduire significativement la RMSE du MSCKF, comme par exemple avec le Adaptive MSCKF [35] comme montré dans le tableau 3.2.

		Adaptive MSCKF	Conventional MSCKF
Position (m)	North	0.0492	0.1069
	East	0.0646	0.0991
	Down	0.1696	0.3915
Velocity (m/s)	North	0.0119	0.0166
	East	0.0216	0.0243
	Down	0.0325	0.0572
Orientation (deg)	Roll	0.3843	0.2726
	Pitch	0.2921	0.3271
	Yaw	0.3772	0.2893

TABLE 3.2 – Résultat du filtre MSCKF adaptatif comparé au filtre MSCKF standard [35].

Le MSCKF est une méthode de navigation VINS performante qui se distingue du SLAM au sein d'un environnement de techniques VINS majoritairement conçues pour du SLAM.

En parallèle de ces développements autour de l'EKF pour le SLAM ou la navigation VINS, plusieurs méthodes proposent du V-SLAM qui se passe alors d'IMU pour réaliser la navigation, comme ORB-SLAM [96]. Ces systèmes V-SLAM sont un axe de recherche encore très étudié qui ne sera pas développé ici car dépourvu de navigation inertielle. Néanmoins, il est intéressant de voir qu'ils ont servi de base au développement de nouveaux systèmes VI-SLAM.

3.3.2 Méthodes les plus performantes

Un nombre important des contributions du V-SLAM sur la sélection et le traitement des amers, la cartographie ou en particulier la fermeture de boucle se retrouve dans les méthodes de navigation VI-SLAM ou VINS. Comme on peut le voir dans le benchmark issu de [32], les méthodes ROVIO, OKVIS, MSF-EKF associées à l'odométrie visuelle SVO ("Semi-direct Visual Odometry") [46] ou encore MSCKF sont comparées à des méthodes plus performantes en termes de précision ou de fiabilité, comme par exemple VINS-Mono [100]. La figure 3.18 donne un aperçu des résultats de ces algorithmes dont les performances ont été évaluées sur différents types d'ordinateur aux puissances de calculs et compacités variées. On y constate une nette surperformance des méthodes VINS-Mono qui minimisent leur RMSE sur toutes les machines à l'essai. On peut également voir que OKVIS et ROVIO, proposant des algorithmes similaires, ont des performances proches même si ROVIO ne fonctionne pas sur la Up-Board. Enfin, parmi les moins performants, le MSCKF propose des performances légèrement meilleures que le MSF-EKF combiné avec un algorithme de cartographie. Il faut toutefois noter que la version du MSCKF utilisée dans ce benchmark est celle de [95] qui n'est pas la plus récente comme présentée dans la partie précédente.

VINS-Mono

VINS-Mono [100] est un estimateur à fenêtre glissante, à la manière d'OKVIS, qui réalise un suivi robuste des amers via des processus d'optimisation non linéaires. En plus de cela, VINS-Mono introduit plusieurs innovations majeures pour ce genre de systèmes de navigation, dont une synthèse est visible en figure 3.19.

En premier lieu, un couplage lâche est effectué (à l'issue du "Measurement Preprocessing") entre une méthode de navigation à l'estime visuelle comme 8-point [62] ou 5-point [98] et l'IMU. Le but de ce couplage est alors d'aligner l'IMU et la caméra comme illustré en figure 3.20, ce que le couplage lâche fait souvent bien (cf. le MSF-EKF qui l'estime dans son vecteur d'état) afin de récupérer des informations comme le facteur d'échelle, la gravité, la vitesse initiale ou encore les biais gyrométriques.

Ensuite, VINS-Mono réalise un pré-traitement des données inertielles afin de simplifier l'appariement avec des images et réalise une fermeture de boucle contrainte pour garantir sa robustesse en recalant les ensembles d'images reconnus comme identiques.

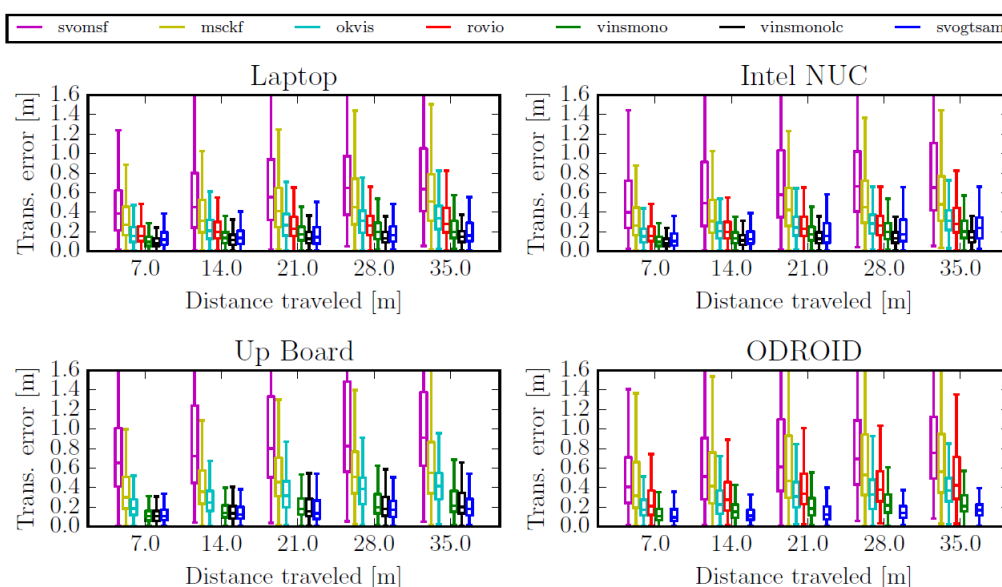


FIGURE 3.18 – Benchmark sur différentes plate-formes comparant plusieurs méthodes de navigation VINS dont ROVIO [7], OKVIS [84], MSCKF [95] ou encore VINS-Mono [100] par un calcul de RMSE sur KITTI [32].

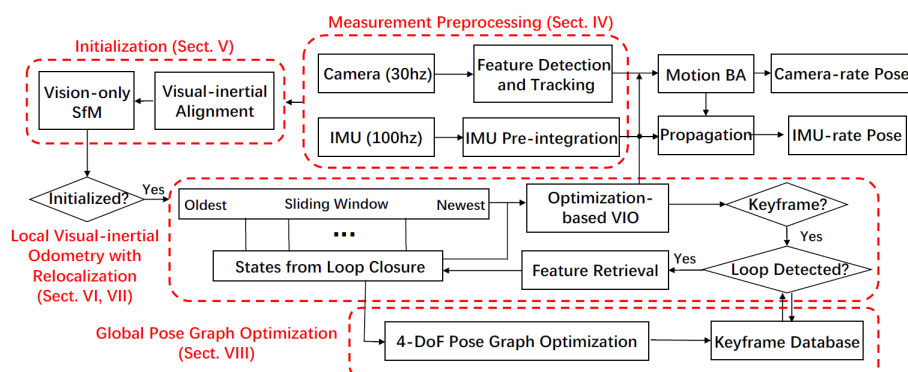


FIGURE 3.19 – Schéma bloc de VINS-Mono [100].

Pour finir, il introduit plusieurs outils destinés à garantir la robustesse de la navigation dans des conditions adverses (changement violent d'illumination, mouvements violents du système, etc.). VINS-mono performe extrêmement bien comme l'atteste le benchmark précédent. Des comparaisons avec OKVIS permettent en particulier de constater l'impact de la fermeture de boucle sur la qualité de la navigation, comme en figure 3.21.

LOVINS

L'algorithme LOVINS ("Lightweight Omnidirectional Visual-Inertial Navigation System") [49] est un autre algorithme VINS très performant. Conçu dans le but d'exploiter

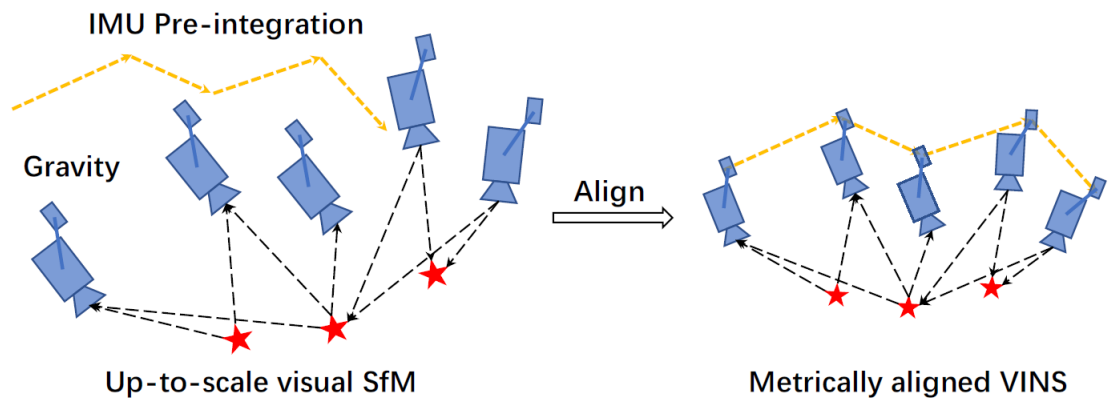


FIGURE 3.20 – Schéma illustrant l’alignement métrique entre la caméra et l’IMU (trajectoire en jaune) [100].

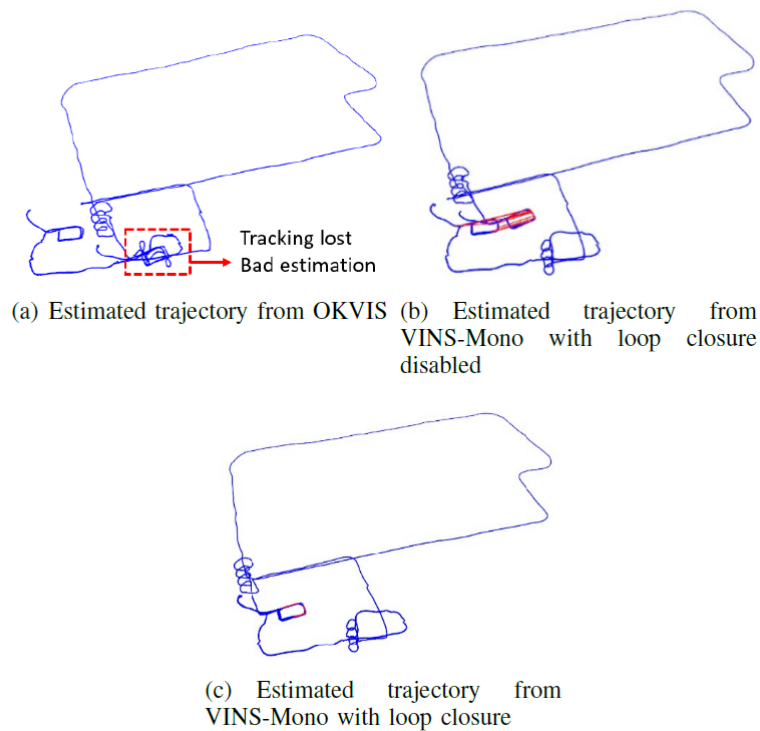


FIGURE 3.21 – Trajectoires réalisées par OKVIS et VINS-mono. On constate en particulier l’utilité de la fermeture de boucle qui réalise un recalage nécessaire à la qualité de l’estimation [100].

des capteurs et moyens de calculs de faible coût, il se veut léger d’un point de vue computationnel. Comme la plupart des algorithmes VINS de cette partie, il comporte une stratégie de sélection des images pertinentes à conserver, un processus d’optimisation

en sortie d'estimation et une détection de fermeture de boucle. Testé sur le jeu de données TUM VI [75] face à ORB-SLAM et VINS-Mono, il affiche de meilleurs résultats en termes de RMSE que les deux autres méthodes comme on peut le voir sur la figure 3.22.

Sequence	VINS-Mono	Omni ORB-SLAM	Ours	Length(m)
corridor1	1.1489	0.1252	0.0903	305
corridor2	0.7030	0.1349	0.1125	322
corridor3	1.3963	0.1360	0.1173	300
corridor4	0.1201	0.1196	0.0911	114
corridor5	0.6019	0.1159	0.1005	270
room1	0.0425	0.0423	0.0401	146
room2	0.0602	0.0466	0.0399	142
room3	0.1675	0.0446	0.0378	135

FIGURE 3.22 – Résultats (RMSE) de LOVINS sur le jeu de données TUM VI [75] et comparé à ORB-SLAM et VINS-mono [49].

Kimera

Enfin, la librairie Kimera [104] propose quatre modules, dont un d'odométrie visuelle-inertielle basée sur le Georgia Tech Smoothing and Mapping GTSAM qui est un optimiseur de graphes [31] appairé avec les données inertielles selon les techniques présentées dans [45]. Un autre module est un optimiseur de graphe de pose robuste utilisé afin d'exclure les problèmes liés à l'aliasing (par crénelage ou moiré) par exemple. Enfin, les deux autres modules sont utilisés pour la création de maillages en 3D. L'ensemble des modules forme une architecture complexe capable de réaliser l'ensemble des tâches de la navigation en local et en global en couplant une caméra stéréo et une IMU. Kimera constitue la méthode la plus complète à ce jour qui fait preuve d'excellentes performances sur EuRoC, en comparaison à d'autres méthodes de pointes, comme on peut le voir sur la figure 3.23.

Résumé

On constate que les méthodes les plus performantes en couplages serrés s'accompagnent systématiquement de routines de sélections des informations fournies par les capteurs, que ce soit en excluant certaines de ces informations comme dans le cas de LOVINS, ou en utilisant des routines de pré-traitement destinées à nettoyer les informations indésirables pour l'estimation (comme le pré-traitement inertiel de VINS-Mono). De plus, toutes ont recours à des routines d'optimisation non linéaires pour associer les données capteurs à des poses du véhicule. La précision de ces algorithmes est ainsi étroitement liée à une grande complexité calculatoire et nécessite des ordinateurs puissants pour garantir leurs performances.

Seq.	RMSE ATE [m]									
	Fixed-lag Smoothing					Full Smoothing			Loop Closure	
	OKVIS	MSCKF	ROVIO	VINS-Mono	Kimera-VIO	SVO-GTSAM	Kimera-VIO	VINS-LC	Kimera-RPGO	
MH_1	0.16	0.42	0.21	0.15	0.11	0.05	0.04	0.12	0.08	
MH_2	0.22	0.45	0.25	0.15	0.10	0.03	0.07	0.12	0.09	
MH_3	0.24	0.23	0.25	0.22	0.16	0.12	0.12	0.13	0.11	
MH_4	0.34	0.37	0.49	0.32	0.24	0.13	0.27	0.18	0.15	
MH_5	0.47	0.48	0.52	0.30	0.35	0.16	0.20	0.21	0.24	
V1_1	0.09	0.34	0.10	0.08	0.05	0.07	0.06	0.06	0.05	
V1_2	0.20	0.20	0.10	0.11	0.08	0.11	0.07	0.08	0.11	
V1_3	0.24	0.67	0.14	0.18	0.07	×	0.09	0.19	0.12	
V2_1	0.13	0.10	0.12	0.08	0.08	0.07	0.07	0.08	0.07	
V2_2	0.16	0.16	0.14	0.16	0.10	×	0.09	0.16	0.10	
V2_3	0.29	1.13	0.14	0.27	0.21	×	0.19	1.39	0.19	

FIGURE 3.23 – Résultats (RMSE) de Kimera sur le jeu de données EuRoC [16] et comparé à OKVIS, MSCKF, ROVIO et VINS-mono. On constate clairement la performance de Kimera [104].

3.4 Couplage et robustesse

La question de la robustesse en navigation n'a pas de réponse unique. En effet, à la manière de la robustesse en commande, c'est le but final qui détermine les moyens à mettre en oeuvre. Dans la présente partie, nous nous concentrerons donc sur les notions de robustesse face à trois problèmes :

- les perturbations des signaux,
- les erreurs d'estimations,
- l'indisponibilité d'un capteur.

La robustesse est rarement et peu quantifiée dans les différents travaux décrits dans les sections précédentes. La RMSE contient une information partielle sur la robustesse mais elle ne suffit pas. L'erreur d'estimation étant une variable stochastique, la représentation sous la forme de diagrammes de quartiles comme dans [32] permet déjà de fournir une meilleure idée de la capacité d'une navigation à conserver une erreur faible sur l'ensemble de la trajectoire à estimer malgré les différentes perturbations possibles.

Robustesse contre les perturbations des signaux

La qualité des signaux issus des capteurs est essentielle pour garantir la bonne qualité d'une navigation. Ainsi, une bonne robustesse face à leur dégradation est essentielle. Problématique bien connue pour la commande, elle est aussi très étudiée pour le traitement des signaux GNSS comme par exemple dans [123]. Ce problème est traité de deux manières, en abordant d'un côté les bruits qui sont sources d'erreurs permanentes

et de l'autre les perturbations qui vont être sources d'estimations aberrantes. Au regard des capteurs inertiels, l'analyse de la variance d'Allan (décrite par la figure 3.24) ou de la densité spectrale de puissance du capteur permet de caractériser les bruits et de les filtrer, par exemple avec un filtre à ondelettes. C'est le cas dans [101] qui utilise ce type de filtre afin de débruiter une centrale inertielle lors d'un couplage lâche EKF avec un GNSS pour garantir une navigation inertielle de qualité lors de coupure GPS.

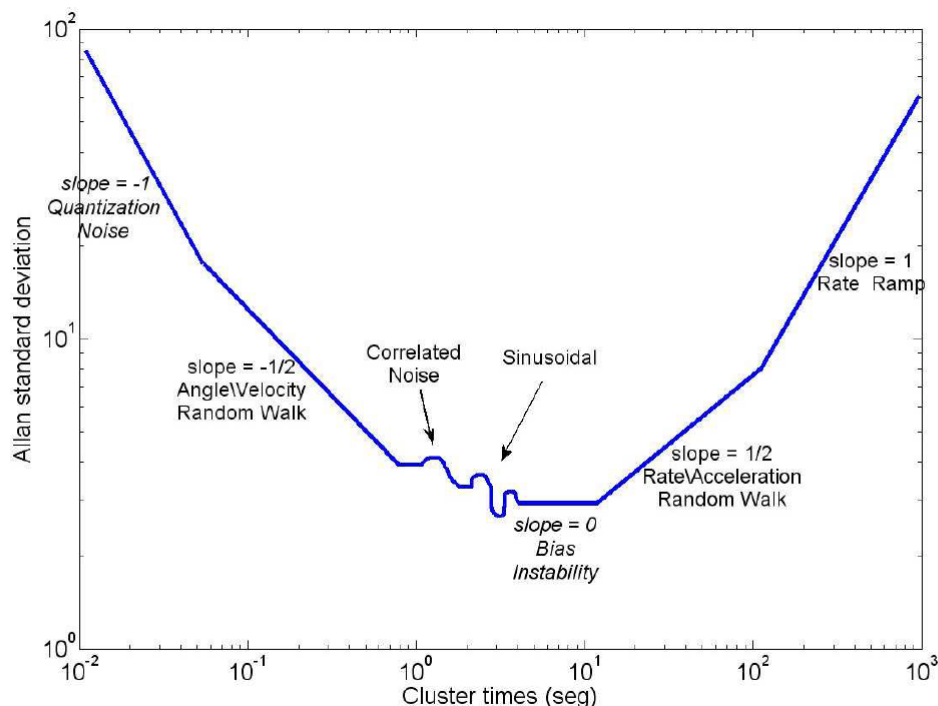


FIGURE 3.24 – Représentation de la variance d'Allan [101].

Les résultats d'une telle démarche, présentés en figure 3.25, montre la réduction importante de l'erreur générée par les bruits sur le capteur lorsque le filtrage par ondelette (courbe bleue foncée notée "with WD") est appliqué sur les données inertielles lors d'une coupure GPS. En particulier, on constate que cette modalité de filtrage permet de mieux minimiser l'erreur par rapport à une éventuelle augmentation de la taille du vecteur d'état.

Du côté de la vision, la robustesse face à des problèmes similaires à du bruit ("lens flare", obstacles temporaires pris comme amers, etc.) peut trouver une solution dans la multiplication des méthodes de détectons et qualifications d'amers comme dans [19], suivie par une méthode de synthèse pour ne conserver que les amers pertinents. Afin d'améliorer sa robustesse, VINS-mono par exemple limite le nombre d'amers qui peuvent être simultanément traqués sur une même image et réinitialise sa navigation lorsque le

seuil fixé est franchi. Les routines d'élimination d'amers sont par ailleurs couramment utilisées pour améliorer la précision et la robustesse dans ce genre d'applications.

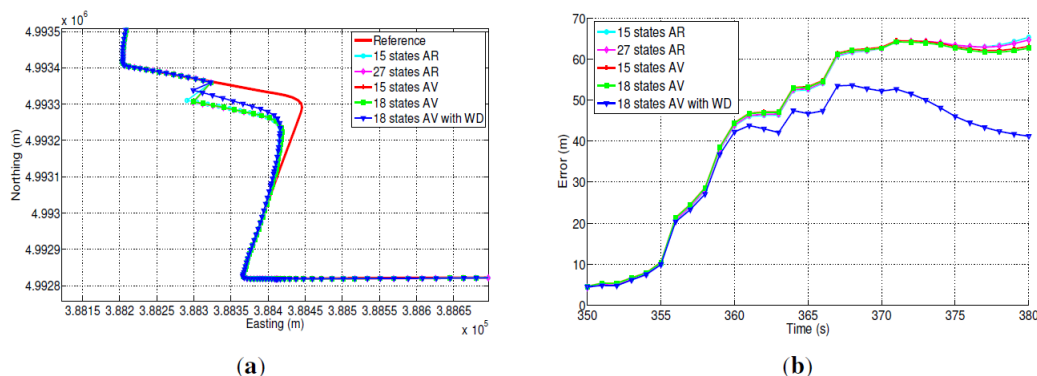


FIGURE 3.25 – Position dans le plan Est-Nord lors d'une coupure GPS (gauche). Erreur de position dans ce même plan (droite) [101].

Robustesse contre les erreurs d'estimations

Les perturbations de signaux, si elles ne sont pas contrôlées, peuvent impacter la navigation et entraîner des erreurs d'estimations (comme on a pu le voir sur la figure 3.25 par exemple). Une robustesse à de tels problèmes ou à une dérive de l'estimation est alors souhaitée pour préserver la qualité de l'estimation. Ici, plusieurs outils existent. VINS-Mono monitor les discontinuités trop importantes en position et orientation entre deux estimations afin de déterminer si le système doit être réinitialisé ou non. D'autres méthodes plus efficaces peuvent reposer sur l'usage de la distance de Mahalanobis¹ qui permet de déterminer la similarité entre deux séquences temporelles et que l'on voit employée dans [7], [49], [60] ou encore [86] pour ne citer qu'eux. On peut également citer l'usage d'un filtre médian pour détecter les discontinuités des angles et rejeter les mauvaises estimations comme dans [119] et illustré à la figure 3.26.

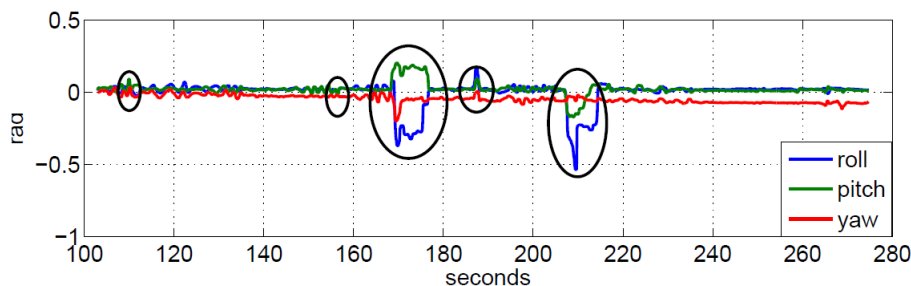


FIGURE 3.26 – Détection de valeurs aberrantes (cerclée en noir) d'angles par filtre médian [119].

1. Une mesure de similarité entre deux séries de données (souvent une connue et une inconnue) qui utilise la variance et la corrélation des séries de données comparées.

La fermeture de boucle, comme dans VINS-Mono ou LOVINS, est également un excellent moyen de contrôle pour augmenter la robustesse d'une navigation, les estimations précédentes constituant alors autant de références à même de corriger une dérive.

Enfin, si l'estimation erronée est due à des changements de dynamique appliqués au mobile, des méthodes adaptatives peuvent répondre à cette problématique comme les filtres adaptatifs présentés dans [70] qui adaptent dynamiquement certains de leurs paramètres en fonction de la dynamique du mobile. On peut également mentionner les multiples modèles interactifs comme dans [69], [88] ou encore [126] qui utilisent un modèle de Markov caché pour sélectionner une estimation parmi celles de différents filtres de Kalman ou réaliser une moyenne pondérée de ces estimations. Pour ce faire, un module d'interaction utilise les covariances de sortie de plusieurs filtres de Kalman contenant des modélisations différentes des mouvements du mobile. Associé à un modèle de Markov caché, le module d'interaction va déterminer quelle estimation doit être sélectionnée ou bien réaliser une moyenne pondérée de ses estimations selon les travaux les plus récents. L'architecture d'un tel système est montrée en figure 3.27.

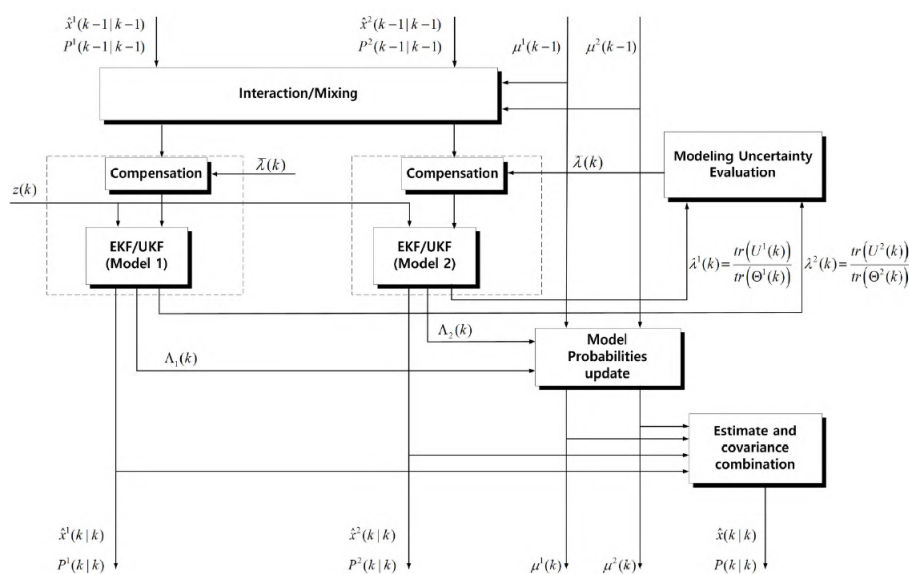


FIGURE 3.27 – Exemple d'architecture à multiples modèles interactifs à deux modèles [126].

Indisponibilité d'un capteur

L'absence de signaux due à la défaillance majeure (même temporaire) d'un capteur est critique pour une application de navigation et doit faire l'objet d'une surveillance également. VINS-mono monitorise pour cela l'évolution des paramètres intrinsèques des capteurs comme leurs biais et leurs dérives, ceci afin de ré-initialiser le système en cas

de variation trop importante. On a également pu constater que la robustification d'un capteur face au bruit comme dans [123] permet d'assurer une navigation, au moins temporairement, en cas de défaillance du capteur extéroceptif. Il faut toutefois noter que ceci n'est valable que pour une application de couplage lâche.

Puisqu'un capteur défaillant peut perturber une estimation de bien des façons (bruit important, biais amplifié, pic d'amplitude, offset, absence d'information, etc.), de nombreux moyens de lutte ont déjà été couverts via les outils de robustesses face aux perturbations des signaux et aux erreurs d'estimation. Toutefois, l'étude de méthodes de détection de défauts de capteurs via l'analyse de la consistance de leurs informations comme dans [38] est intéressante. En effet, elle fournit des outils statistiques afin de détecter les défaillances d'un capteur au sein d'un système multi-capteurs et parvient à montrer l'efficacité de ces outils dans des conditions normales de fonctionnement et dans une situation où les capteurs sont soumis à d'importants flux d'informations. Enfin, l'analyse de contextes environnementaux de [41], et expliquée plus haut, présente également de nombreux atouts pour les configurations où les défauts des capteurs peuvent être anticipés comme avec les cas limites de la vision par exemple.

Conclusion

En raison de la multiplicité des aléas aléas pour lesquels il est nécessaire de mettre en place des mesures de robustesse, nous avons décidé de nous concentrer sur la gestion de l'indisponibilité d'un capteur et la lutte contre les erreurs d'estimation. En effet, les systèmes de la gamme STAMINA-UGV, en raison des applications de défense auxquels ils sont destinés, peuvent rencontrer des situations pouvant provoquer une indisponibilité d'un capteur, en particulier la caméra. De plus, du fait des applications tout terrain à fortes dynamiques, les différents aléas pouvant affecter une caméra (décrits au chapitre précédent) et donc une navigation à l'estime visuelle ont une forte probabilité de survenir, risquant alors d'entraîner des erreurs d'estimations dommageables à la qualité de la navigation.

3.5 Conclusion : choix d'un couplage lâche

Après avoir présenté les méthodes de navigation par couplage visuel-inertiel lâche et serré ainsi que les différents outils d'amélioration de la robustesse d'un système de navigation, il est nécessaire de rappeler les objectifs et contraintes des travaux présentés dans ce document afin de justifier le choix d'un couplage lâche.

Nous proposons de compléter la navigation à l'estime visuelle Parallax Beam de STAMINA d'une centrale inertielle visant à améliorer la qualité de la navigation et en particulier :

- **Préserver la précision de Parallax Beam.** La RMSE minimale en translation doit être du même ordre de grandeur que celle de Parallax Beam,

- **Ne pas modifier Parallax Beam.** Parallax Beam ne fait pas l'objet de ces travaux et est donc utilisé comme une "boîte grise" qui reçoit des images en entrée et fournit une estimation de pose en sortie,
- **Garantir la navigation en dépit de défaut de la vision.** Il faut proposer une solution capable de continuer la navigation malgré un défaut d'un capteur, que ce soit une erreur d'estimation ou une indisponibilité dudit capteur,
- **Être adaptable à tout type de mobiles, de l'UGV au projectile.** En vue des développements envisagés pour la gamme STAMINA (UAV, projectile) dans le futur, la solution proposée devra nécessiter le moins de modifications possibles pour être applicable à ces autres mobiles.

La considération du second objectif nous conduit à privilégier un couplage ne nécessitant aucune information autre que la pose fournie par Parallax Beam, ce qui exclut plusieurs méthodes de couplages serrés qui utilisent les informations intrasèques de la caméra. Ensuite, plusieurs couplages ayant la capacité à continuer la navigation malgré un capteur absent ou défectueux existent. On peut par exemple citer [123] dont les travaux étudient la situation de coupure d'informations extéroceptives dans le cadre d'une fusion IMU/GNSS ou encore avec le MSF-EKF dont le tampon circulaire permet de manipuler des données issues de plusieurs capteurs malgré un asynchronisme ou une indisponibilité temporaire. Tout ceci vient renforcer d'avantage l'intérêt d'un couplage lâche. Les RMSE du Parallax Beam sur KITTI ou EuRoC indiquent que la condition de précision peut être satisfaite avec des méthodes de couplages lâches à condition que la navigation inertielle soit de qualité suffisante. Enfin, les méthodes de couplage lâche reposant majoritairement sur des méthodes de filtrage, la condition d'agnosticité limite la complexité que l'on peut donner à notre matrice de modèle du filtre afin de ne pas modéliser des dynamiques singulières à un type de véhicule comme le frottement des roues ou les frottements aérodynamiques.

Pour ces raisons, deux méthodes de couplage lâche seront présentées dans les prochains chapitres : un couplage à l'aide d'un Error-State Kalman Filter (ESKF), chapitre 4, et un couplage à l'aide d'un MSF-EKF, chapitre 5.

4 Fusion Inertielle-Visuelle par filtrage de Kalman à état-erreur

4.1 Introduction

Un Error State Kalman Filter (ESKF) est une solution de couplage lâche simple et efficace utilisée dans de nombreux travaux avec des résultats pertinents [5] [65] [68] [108]. Dans ce chapitre est présenté un ESKF pour coupler la navigation par la vision Parallax Beam avec une navigation à l'estime inertielle classique tel que décrite au chapitre 2. Ce filtre, en estimant l'erreur du système plutôt que son état, doit permettre de se soustraire du modèle, en respect de notre objectif système-agnostique. De plus, cette solution doit offrir une navigation sans dégradation de la précision comparée au Parallax Beam seul, tout en étant en mesure d'assurer la navigation en cas de défaut de celui-ci.

Dans un premier temps, nous allons présenter les équations du filtre ESKF afin d'en expliquer l'intérêt. Puis, nous montrerons et analyserons les résultats de ce filtre obtenu sur le jeu de données KITTI vision. Enfin nous étudierons certaines pistes d'amélioration de ce filtre et leurs limites.

4.2 Filtre ESKF

4.2.1 Équations de mécanisation de l'INS

Avant de présenter l'ESKF, il est nécessaire de rappeler les équations utilisées pour modéliser le mouvement d'un mobile à partir des données inertielles que sont les accélérations et les vitesses de rotation.

Le vecteur d'état-erreur de l'ESKF est le suivant : $\delta\mathbf{x}_n = [\delta\mathbf{r}_n, \mathbf{b}_n^\omega, \delta\mathbf{p}_n, \delta\mathbf{v}_n, \mathbf{b}_n^a]$. Ce vecteur de 15 éléments contient les biais des accéléromètres \mathbf{b}_n^a et des gyromètres \mathbf{b}_n^ω , ainsi que les erreurs d'attitude selon le formalisme d'Euler (roulis, tangage, lacet) $\delta\mathbf{r}_n$, mais aussi les erreurs en position $\delta\mathbf{p}_n$ et en vitesse $\delta\mathbf{v}_n$. Chacune de ces cinq composantes du vecteur contient trois éléments correspondants à une estimation en 3D.

L'ESKF est ici utilisé pour corriger l'estimation inertielle obtenue par navigation à l'estime via une mesure de l'écart d'estimation entre l'algorithme Parallax Beam et la

navigation à l'estime inertielle. Le modèle de propagation du filtre suit donc le processus de mécanisation que l'on trouve classiquement dans certains travaux sur les INS [47] [68] [110].

Tout d'abord, l'étape de compensation des biais utilise les biais des accéléromètres et des gyromètres estimés par l'ESKF pour corriger les données inertielles brutes \mathbf{a}_n^{imu} et $\boldsymbol{\omega}_n^{imu}$:

$$\begin{cases} \boldsymbol{\omega}'_n &= \boldsymbol{\omega}_n^{imu} - \mathbf{b}_n^\omega \\ \mathbf{a}'_n &= \mathbf{a}_n^{imu} - \mathbf{b}_n^a \end{cases} \quad (4.1)$$

avec \mathbf{a}'_n et $\boldsymbol{\omega}'_n$ les mesures des accéléromètres et gyromètres avec les biais compensés.

Du point de vue du modèle de propagation, les biais sont constants, par conséquent :

$$\begin{cases} \mathbf{b}_{n+1}^\omega &= \mathbf{I}_3 \cdot \mathbf{b}_n^\omega \\ \mathbf{b}_{n+1}^a &= \mathbf{I}_3 \cdot \mathbf{b}_n^a. \end{cases} \quad (4.2)$$

L'étape suivante consiste à mettre à jour l'orientation de l'IMU par rapport au repère de navigation Est-Nord-Haut choisi. Pour ce faire, on définit la matrice 4x4 antisymétrique des vitesses de rotations Ω_n comme suit :

$$\Omega_n = \frac{1}{2} \cdot \begin{bmatrix} 0 & -\boldsymbol{\omega}'_n(1) & -\boldsymbol{\omega}'_n(2) & -\boldsymbol{\omega}'_n(3) \\ \boldsymbol{\omega}'_n(1) & 0 & \boldsymbol{\omega}'_n(3) & -\boldsymbol{\omega}'_n(2) \\ \boldsymbol{\omega}'_n(2) & -\boldsymbol{\omega}'_n(3) & 0 & \boldsymbol{\omega}'_n(1) \\ \boldsymbol{\omega}'_n(3) & \boldsymbol{\omega}'_n(2) & -\boldsymbol{\omega}'_n(1) & 0 \end{bmatrix}. \quad (4.3)$$

On utilise alors l'approximation de Padé à l'ordre 1 de la fonction exponentielle pour réaliser la mise à jour du quaternion associé à l'orientation :

$$\mathbf{q}_n = \frac{2 \cdot \mathbf{I}_3 + \Omega_n \cdot \Delta t}{2 \cdot \mathbf{I}_3 - \Omega_n \cdot \Delta t} \cdot \mathbf{q}_{n-1}, \quad (4.4)$$

avec Δt la période d'échantillonnage et \mathbf{q}_n le quaternion associé à l'orientation de l'IMU. Via ce dernier, on peut ensuite retrouver la matrice de rotation du système entre le repère inertiel et le repère de navigation $\mathbf{C}_{imu_n}^N$ ou encore l'orientation du système selon le formalisme d'Euler r_n en utilisant les propriétés classiques des quaternions [113].

Pour le modèle de propagation, l'erreur d'orientation se propage et est amplifiée par le biais sur les gyromètres exprimé dans le repère inertiel courant :

$$\delta \mathbf{r}_{n+1} = \mathbf{I}_3 \cdot \delta \mathbf{r}_n + \Delta t \cdot \mathbf{C}_{imu_n}^N \cdot \mathbf{b}_n^\omega. \quad (4.5)$$

L'étape suivante consiste à transformer l'accélération du repère inertiel vers le repère de navigation (cf. Chapitre 2) pour ensuite y enlever l'accélération de la gravité \mathbf{g} :

$$\mathbf{a}_n^{libre} = \mathbf{C}_{imu_n}^N \cdot \mathbf{a}'_n - \mathbf{g}. \quad (4.6)$$

L'accélération libre de la gravité est ensuite utilisée dans la quatrième étape pour être intégrée afin d'obtenir la vitesse du mobile à l'instant n :

$$\mathbf{v}_{n|n-1} = \mathbf{v}_{n-1} + \mathbf{a}_n^{libre} \cdot \Delta t. \quad (4.7)$$

Du point de vue du modèle de propagation, l'erreur en vitesse se propage et est amplifiée par l'erreur en orientation qui sert à projeter l'accélération du repère inertiel vers le repère de navigation :

$$\delta \mathbf{v}_{n+1} = \mathbf{I}_3 \cdot \delta \mathbf{v}_n + \Delta t \cdot (\mathbf{C}_{imu_n}^N \cdot \mathbf{b}_n^a - \mathbf{S}(\mathbf{a}_n^{libre}) \cdot \delta \mathbf{r}_n), \quad (4.8)$$

Avec $\mathbf{S}(\mathbf{a}_n^{libre})$ la matrice antisymétrique de l'accélération libre :

$$\mathbf{S}(\mathbf{a}_n^{libre}) = \begin{bmatrix} 0 & -\mathbf{a}_n^{libre}(3) & \mathbf{a}_n^{libre}(2) \\ \mathbf{a}_n^{libre}(3) & 0 & -\mathbf{a}_n^{libre}(1) \\ -\mathbf{a}_n^{libre}(2) & \mathbf{a}_n^{libre}(1) & 0 \end{bmatrix}. \quad (4.9)$$

La vitesse est ensuite elle-même intégrée pour obtenir la position du mobile à l'instant n :

$$\mathbf{p}_{n|n-1} = \mathbf{p}_{n-1} + \mathbf{v}_{n|n-1} \cdot \Delta t. \quad (4.10)$$

Enfin, la dernière étape consiste à corriger les vitesses et positions calculées à l'aide de l'estimation de l'ESKF à jour avec la dernière mesure à l'instant n :

$$\begin{cases} \mathbf{p}_n = \mathbf{p}_{n|n-1} - \delta \mathbf{p}_n \\ \mathbf{v}_n = \mathbf{v}_{n|n-1} - \delta \mathbf{v}_n. \end{cases} \quad (4.11)$$

4.2.2 Principe du filtre ESKF

Avec notre vecteur d'état de dimension quinze à l'instant n , $\delta \mathbf{x}_n = [\delta \mathbf{r}_n, \mathbf{b}_n^\omega, \delta \mathbf{p}_n, \delta \mathbf{v}_n, \mathbf{b}_n^a]$, il est possible de définir un modèle de propagation de l'état linéarisé [47] [110] tel que :

$$\delta \mathbf{x}_{n|n-1} = \mathbf{F}_n \cdot \delta \mathbf{x}_{n-1|n-1} + \mathbf{w}_{n-1}, \quad (4.12)$$

Avec $\delta \mathbf{x}_{n|n-1}$ l'état-erreur prédit, $\delta \mathbf{x}_{n-1|n-1}$ le dernier état-erreur estimé à l'instant $n-1$, \mathbf{w}_{n-1} le bruit de propagation de matrice de covariance $\mathbf{Q}_n = E\{\mathbf{w}_n \mathbf{w}_n^T\}$ et \mathbf{F}_n la matrice de transition d'état de taille 15x15 :

$$\mathbf{F}_n = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{C}_{imu_n}^N & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 & \mathbf{0}_3 \\ -\Delta t \cdot \mathbf{S}(\mathbf{a}_n^{libre}) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \Delta t \cdot \mathbf{C}_{imu_n}^N \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}. \quad (4.13)$$

Tandis que l'équation de mesure ou d'observation est exprimée comme suit :

$$\mathbf{z}_n = \mathbf{H} \cdot \delta \mathbf{x}_n + \mathbf{v}_n, \quad (4.14)$$

Avec \mathbf{z}_n l'erreur mesurée, \mathbf{H} la matrice de mesure définie ci-dessous :

$$\mathbf{H} = \begin{bmatrix} 0_3 & 0_3 & \mathbf{I}_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \mathbf{I}_3 & 0_3 \end{bmatrix}, \quad (4.15)$$

Et \mathbf{v}_n le bruit de mesure dont la matrice de covariance est $\mathbf{R}_n = E\{\mathbf{v}_n \mathbf{v}_n^T\}$. La structure de l'ESKF reprend ensuite celle d'un EKF classique tel que présenté en Annexe A avec notre état-erreur $\delta \mathbf{x}_n$ utilisé à la place de l'état standard du système. On retrouve alors l'équation de mise à jour du filtre comme ci-dessous :

$$\delta \mathbf{x}_n = \delta \mathbf{x}_{n|n-1} + \mathbf{K}_n \cdot [\mathbf{e}_n] - \mathbf{H} \delta \mathbf{x}_{n|n-1}. \quad (4.16)$$

Par conséquent, la mesure que l'on fournit à notre ESKF pour réaliser la mise à jour du filtre doit aussi être une erreur. L'erreur mesurée \mathbf{e}_n est calculée à partir de l'estimation de la navigation à l'estime inertielle non corrigée par l'ESKF présentée précédemment et d'une estimation de position et de vitesse déterminée par Parallax Beam, notées \mathbf{p}_n^{pb} et \mathbf{v}_n^{pb} respectivement. Ces estimations sont projetées du repère caméra au repère de navigation par le biais de la matrice de passage $\mathbf{R}_{(imu_cam)}$ présentée au chapitre 2.

$$\mathbf{e}_n = \begin{bmatrix} \mathbf{p}_{n|n-1} - \mathbf{p}_n^{pb} \\ \mathbf{v}_{n|n-1} - \mathbf{v}_n^{pb} \end{bmatrix}. \quad (4.17)$$

L'ensemble de la méthode peut-être expliquée par le schéma-bloc figure 4.1. Sur celui-ci, on retrouve la structure décrite par les équations précédentes :

- un ensemble inertiel constitué d'une IMU et du Bloc de navigation à l'estime inertielle,
- un ensemble visuel constitué d'une caméra, du Parallax Beam, et de la matrice de changement de repère qui nous permet de situer les données issues de Parallax Beam dans le repère inertiel,
- l'ESKF, qui prend en entrée les données de l'IMU pour sa prédiction et l'erreur entre l'ensemble inertiel et l'ensemble visuel pour sa mise à jour (ou correction),
- l'erreur estimée par l'ESKF va ensuite servir à corriger la navigation à l'estime inertielle pour produire l'estimation finale du couplage choisi.

La méthode de couplage lâche ainsi proposée entend utiliser la navigation visuelle afin de corriger et d'estimer les biais inertiels et ainsi corriger la dérive de la navigation inertielle, ceci afin de garantir une navigation inertielle précise en cas d'absence d'informations nouvelles de la part de la partie vision de l'algorithme.

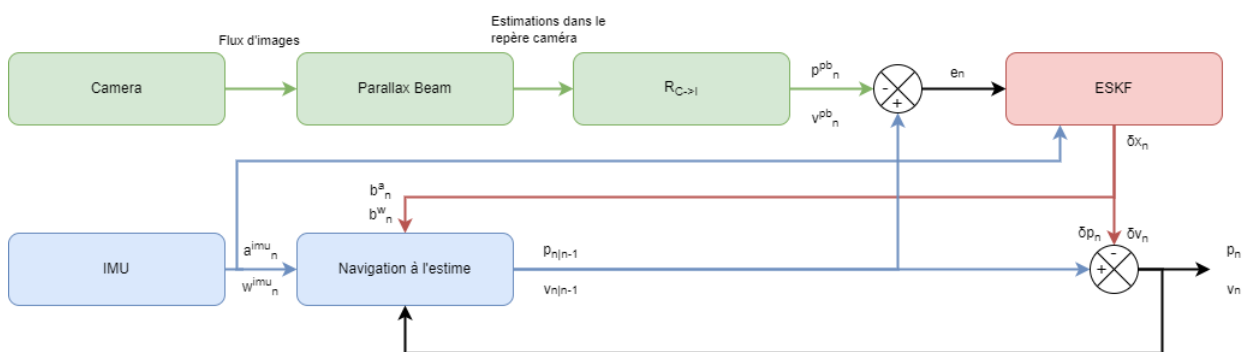


FIGURE 4.1 – Schéma-bloc illustrant l'algorithme ESKF employé pour un couplage lâche VINS.

4.3 Résultats sur le jeu de données KITTI vision

4.3.1 Présentation et analyse des résultats

L'algorithme ESKF a été testé sur le jeu de données KITTI vision [54] (introduit dans le chapitre 2) en utilisant les données synchronisées et rectifiées à 10 Hz. Le vecteur d'état erreur est initialisé à zéro tandis que la covariance de l'erreur d'estimation initiale \mathbf{P}_0 est fixée à :

$$\mathbf{P}_0 = \begin{bmatrix} \left(\frac{2\pi}{3}\right)^2 \cdot \mathbf{I}_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \mathbf{I}_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 25 \cdot \mathbf{I}_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix}, \quad (4.18)$$

Ce choix de coefficients pour la matrice \mathbf{P}_0 exprime la recherche d'une estimation d'angles initiaux dans l'intervalle $[0; 2\pi] rad/s$ (orientation quelconque) et une estimation de vitesse initiale dans l'intervalle $[-15, 15] m/s$ (soit un maximum de 50 km/h). Pour l'estimation en position, comme le repère de navigation est centré sur la position initiale supposée du véhicule, on fait l'hypothèse d'une estimation d'erreur en position égale à zéro sur chaque axe à ± 3 m. Ces coefficients sont adaptés pour l'initialisation du filtre sur KITTI mais devront être redéfinis pour chaque nouveau jeu de données et/ou dynamique de véhicule.

Pour définir le bruit de mesure, des mesures de la variance de l'erreur du Parallax Beam sur KITTI ont été faites. Pour le bruit de modèle, on s'appuie sur les valeurs d'autres travaux ayant été réalisés sur le même jeu de données [11]. Ainsi, les matrices \mathbf{R} et \mathbf{Q} sont définies ainsi :

$$\mathbf{R} = \begin{bmatrix} 10^{-4} \cdot \mathbf{I}_3 & 0_3 \\ 0_3 & 10^{-4} \cdot \mathbf{I}_3 \end{bmatrix}, \quad (4.19)$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & 1.5 \cdot 10^{-3} \cdot \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & 10^{-3} \cdot \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \quad (4.20)$$

Les tests ont été réalisés sur les séquences 04, 05 (figures 4.3, 4.2, 4.4), 06, 09 et 10 de KITTI vision qui sont des séquences de validation pour lesquelles la vérité terrain est disponible. Ces séquences représentent des environnements urbains complexes avec des circulations d'autres usagers. Elles ont été sélectionnées plutôt que d'autres séquences de validations en raison de leur complexité ou longueur de trajet (05 06 et 09) ou bien à cause des comportements des capteurs sur celles-ci. En effet, les séquence 04 et 10 présentent des pics d'accélération selon l'axe vertical qui impactent visiblement la fusion (voir Annexes). Afin de vérifier la convergence du filtre, les coefficients diagonaux de la matrice de covariance de l'erreur \mathbf{P} sont monitorés pour s'assurer de leur convergence vers zéro. Seules les figures sur la séquence 05 sont montrées dans le corps du chapitre. Les autres figures peuvent être trouvées en fin de chapitre et sont utilisées pour l'analyse de nos résultats.

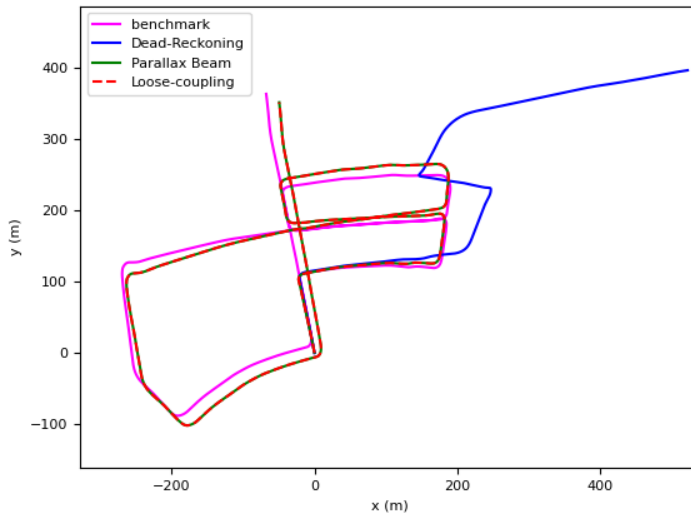


FIGURE 4.2 – Trajectoire estimées sur la séquence 05 du jeu de données KITTI vision

Sur ces figures, on constate que l'estimation de l'ESKF parvient à une précision équivalente à celle de Parallax Beam en dépit d'une très importante dérive inertielle. Ceci est confirmé par des variances d'erreurs d'estimation du filtre qui tendent vers zéro en quelques millisecondes sur chacune des trajectoires évaluées (cf. figure 4.4). Sur les figures 4.2 et 4.3, on peut constater la dérive très importante de la navigation à l'estime inertielle. Cette dérive ne semble cependant pas impacter le couplage qui parvient à

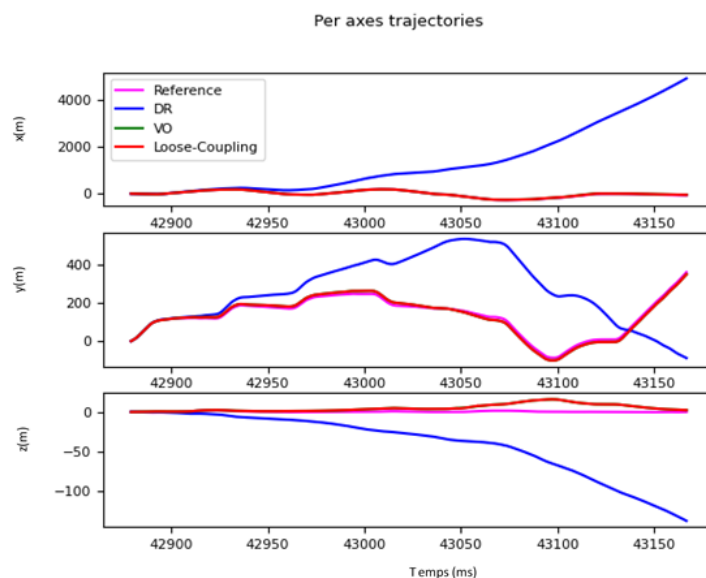


FIGURE 4.3 – Trajectoire estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axes du repère de référence.

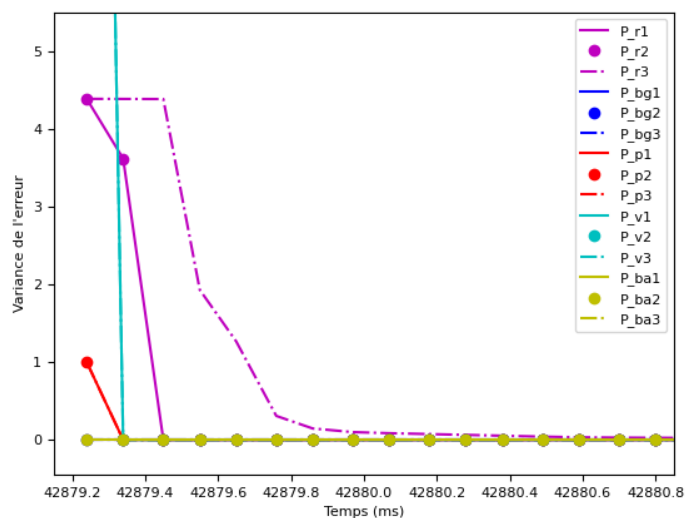


FIGURE 4.4 – Termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquences 05 du jeu de données KITTI vision.

suivre les estimations de Parallax Beam, reproduisant également ses erreurs d'estimation sur l'axe vertical.

Afin de confirmer la qualité de nos estimations, la RMSE calculée avec l'algorithme de validation du jeu de données KITTI vision est présentée en table 4.1.

Les RMSE attestent tout d'abord de la dérive importante de la navigation inertielle en translation que l'on pouvait déjà facilement visualiser tandis qu'elle est très précise en rotation. De même, la précision de la navigation avec Parallax Beam en translation est

	DR	PB(10 Hz)	ESKF(10 Hz)
Sequence 04			
Translation [%]	24.3	1.63	23.9
Rotation [$^{\circ}/m$]	0.000022	0.000049	0.00094
Sequence 05			
Translation [%]	275	2.14	3.76
Rotation [$^{\circ}/m$]	0.0056	0.0083	0.0056
Sequence 06			
Translation [%]	76	3.37	8.58
Rotation [$^{\circ}/m$]	0.000079	0.000216	0.000079
Sequence 09			
Translation [%]	37.5	2.7	7.9
Rotation [$^{\circ}/m$]	0.0040	0.0074	0.0040
Sequence 10			
Translation [%]	27.5	1.54	3.03
Rotation [$^{\circ}/m$]	0.0041	0.0096	0.0041

TABLE 4.1 – RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (ESKF). Les meilleurs résultats sont exprimés en vert.

aussi performante que ce que l’on pouvait observer sur la figure 4.3. Cependant, l’ESKF n’affiche pas une précision en translation aussi bonne que Parallax Beam et on constate alors que son usage dégrade un peu la précision en translation. Cette dégradation peut être attribuée aux mauvaises performances en translation de la navigation à l’estime qui ne sont pas totalement corrigées par le filtre ESKF. En particulier sur la séquence 04, le filtre ne parvient pas à réaliser une estimation correcte des biais sur la centrale qui entraîne une dégradation de la précision en rotation. En général, le filtre ESKF permet de conserver la précision en rotation de la navigation à l’estime inertielle. De plus, il corrige assez largement l’importante erreur en translation de la navigation inertielle à l’aide de la navigation visuelle.

Afin de tester la capacité de notre fusion à garantir la navigation en cas de coupure, des tests similaires ont été réalisés en sous-échantillonnant à 5 Hz le flux d’images fourni au Parallax Beam. Ce choix a été pris en raison de la mauvaise qualité de la navigation inertielle employée, laquelle assure une divergence du filtre lorsqu’il n’est pas maintenu à jour par des nouvelles données de Parallax Beam. Ainsi, l’ESKF ne peut corriger la navigation à l’estime qu’à l’aide de sa prédiction quand Parallax Beam ne transmet pas de mise à jour. De plus, ceci peut induire en erreur la navigation visuelle, comme on peut le voir sur les figures 4.5, 4.6 et 4.7 qui montrent un filtre plus lent à converger, et sujet à plusieurs divergences le long de sa trajectoire, particulièrement visibles sur l’axe vertical. A l’issue de ces essais, nous pouvons augmenter la table précédente de nouvelles données pour constituer la table 4.2.

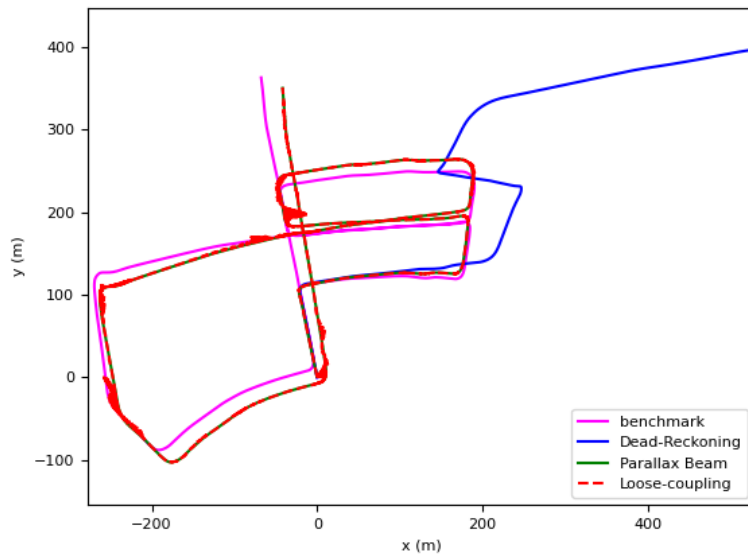


FIGURE 4.5 – Trajectoire estimée sur la séquence 5 du jeu de données KITTI vision. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.

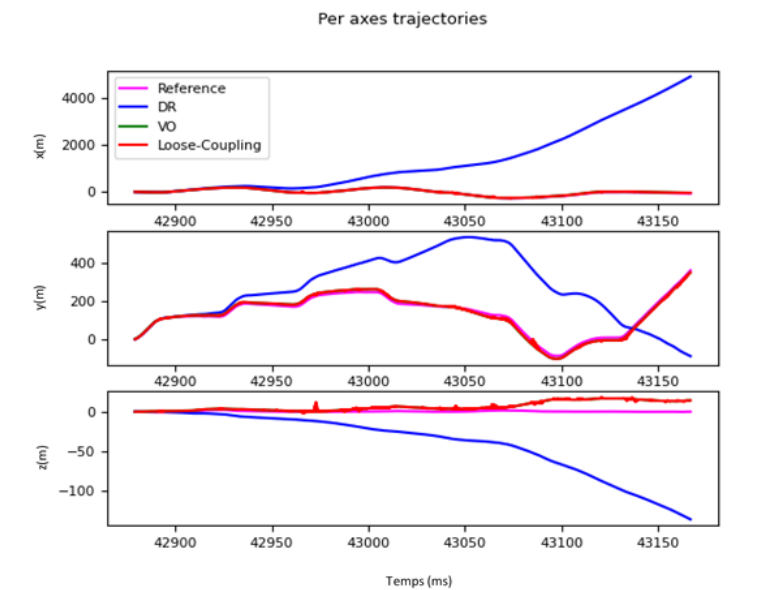


FIGURE 4.6 – Trajectoire estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.

On constate que le sous-échantillonnage à 5 Hz a eu un impact important sur la navigation visuelle dont la RMSE en translation augmente entre 1 et 20% selon les trajectoires. Ceci a un impact important sur l'ESKF qui va alors avoir plus de difficultés à corriger efficacement la navigation inertielle et qui est mis en échec sur les séquences

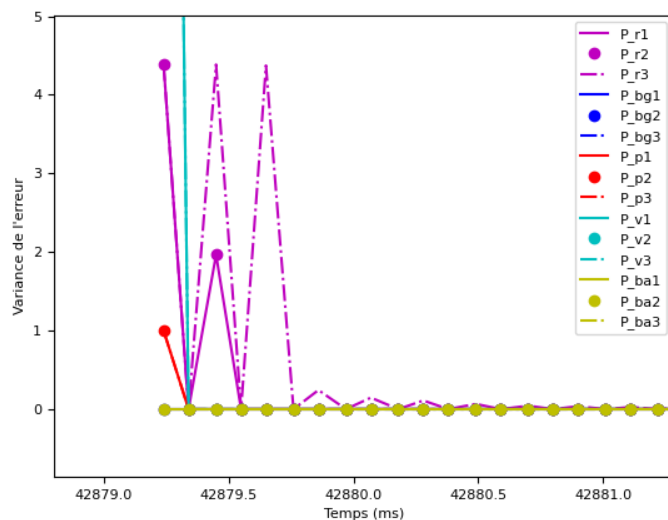


FIGURE 4.7 – Termes diagonaux de la matrice de covariance de l’erreur d’estimation de l’ESKF sur la séquence 05 du jeu de données KITTI vision. Ici, Parallax Beam fonctionne avec un flux image à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz.

	DR	PB(10 Hz)	PB(5 Hz)	ESKF(10 Hz)	ESKF(5 Hz)
Sequence 04					
Translation [%]	24.3	1.63	23.6	23.9	26.7
Rotation [°/m]	0.000022	0.000049	0.000145	0.00094	0.00094
Sequence 05					
Translation [%]	275	2.14	3.8	3.76	7.6
Rotation [°/m]	0.0056	0.0083	0.0207	0.0056	0.0056
Sequence 06					
Translation [%]	76	3.37	2.79	8.58	4.25
Rotation [°/m]	0.000079	0.000216	0.000189	0.000079	0.000079
Sequence 09					
Translation [%]	37.5	2.7	15.8	7.9	51
Rotation [°/m]	0.0040	0.0074	0.0764	0.0040	0.0040
Sequence 10					
Translation [%]	27.5	1.54	8.7	3.03	21.2
Rotation [°/m]	0.0041	0.0096	0.0337	0.0041	0.0041

TABLE 4.2 – RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (ESKF). Les meilleurs résultats sont exprimés en vert. Les résultats des tests réalisés avec Parallax Beam à 5 Hz sont ajoutés.

04, 09 et 10 en particulier avec une dégradation des performances due à la navigation inertielle. On peut toutefois noter que, sur les séquences 05 et 06, l’ESKF parvient à maintenir la navigation inertielle à un niveau de précision satisfaisant car Parallax

Beam y est moins mis en échec. La figure 4.5 nous permet également de constater que l'estimation effectuée traque Parallax Beam et ne le corrige jamais. Ce comportement est une conséquence directe du choix de structure qui se concentre sur l'amélioration de la navigation inertielle et qui utilise Parallax Beam comme une référence.

4.3.2 Bilan et pistes d'améliorations

Dans la section précédente, nous avons vu les performances du filtre ESKF pour la fusion visuelle-inertielle sur le jeu de données KITTI vision. Dans des situations normales de fonctionnement sur KITTI vision (tous les capteurs à 10 Hz), l'ESKF propose un suivi efficace du Parallax Beam qui converge rapidement sur toutes les trajectoires au prix d'une perte modérée de précision. Dans un scénario de légère dégradation de la navigation visuelle (sous-échantillonnée à 5 Hz), on constate que la perte de précision est grandement accentuée et que la rapidité de convergence n'est plus assurée. Ceci confirme l'incapacité du filtre à être autonome de Parallax Beam, même de manière temporaire. Les résultats présentés peuvent être améliorés en agissant sur deux paramètres principaux.

Dans un premier temps, il est nécessaire d'améliorer la navigation inertielle en amont de la fusion. Les tests réalisés à 10 Hz montrent en effet que la dérive importante de la navigation inertielle est un facteur limitant du couplage lâche qui a été réalisé. Pour résoudre ce problème, la piste d'une navigation inertielle renforcée à l'aide d'un filtre de Kalman invariant [11] a été étudiée.

Ensuite, il paraît nécessaire d'être en mesure de diagnostiquer la navigation en faute quand cela survient, ceci afin d'être en mesure de corriger la navigation comme on peut le voir lors des essais à 5 Hz. Des méthodes de rejections de valeurs aberrantes comme dans [79] [44] [78] [38] devront donc être investiguées.

4.4 Navigation inertielle par IEKF

4.4.1 L'IEKF

Le filtre de Kalman étendu invariant (IEKF) est une version du filtre de Kalman étendu (EKF) pour les systèmes non linéaires possédant des symétries (aussi appelées invariances). C'est un observateur non linéaire conçu pour estimer des systèmes sur des groupes de Lie. Tandis que le terme de correction de l'EKF est basé sur une erreur de sortie linéaire, l'IEKF utilise un terme de correction qui est géométriquement adapté et basé sur une erreur de sortie invariante. De plus, la matrice de gain n'est pas mise à jour à partir d'une erreur d'état linéaire mais à partir d'une erreur d'état invariante. Plus de détails sur l'IEKF et son application peuvent être trouvés dans les travaux [8] [11] [12] [106] entre autres.

L'IEKF peut être utilisé pour la navigation inertielle à la manière d'un EKF standard.

Néanmoins, il propose de meilleures performances pour les applications de navigation terrestre [10] et présente un intérêt évident pour la navigation de projectile dont les invariances sont plus nombreuses que pour des mobiles terrestres et dont l'exploitation offre des résultats très intéressants [106].

Pour se convaincre des performances d'un IEKF, cet estimateur a été implémenté selon les travaux de [11] sans le CNN associé. Ce filtre a ensuite été testé sur les mêmes séquences du jeu de données KITTI vision que précédemment comme on peut le voir sur la figure 4.8. Il convient de dire ici qu'un EKF simple peut également proposer une amélioration de performance suffisante en comparaison des performances de la navigation inertielle en place avec l'ESKF. Le choix de l'IEKF étant surtout un choix pour proposer une méthode au niveau de l'état d'art dans la discipline.

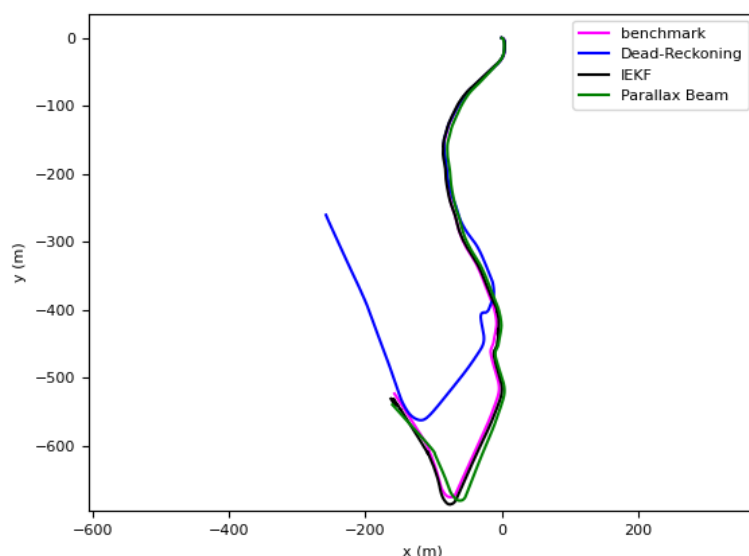


FIGURE 4.8 – Trajectoires estimées sur la séquence 10 du jeu de données KITTI vision avec un IEKF. La navigation à l'estime inertielle (Dead-Reckoning), visuelle (Parallax Beam) et la référence (benchmark) sont également affichées pour comparaison.

Sur cette figure, on peut déjà constater que l'IEKF semble offrir une précision conséquente et bien supérieure à celle d'une navigation à l'estime inertielle. Ceci se confirme par les calculs de RMSE effectués sur les trajectoires 05, 09 et 10 présentés dans le tableau 4.3. Avec ces quelques résultats, on constate que non seulement l'IEKF offre une précision en translation compétitive avec Parallax Beam et donc bien meilleure qu'une simple navigation à l'estime inertielle mais aussi que sa précision en rotation est la meilleure ou deuxième meilleure, surpassant Parallax Beam sur ce point.

Il apparaît donc utile pour améliorer la navigation inertielle d'utiliser l'IEKF proposé ici.

	DR	PB(10 Hz)	IEKF
Sequence 05			
Translation [%]	275	2.14	4.06
Rotation [°/m]	0.0056	0.0083	0.0057
Sequence 09			
Translation [%]	37.5	2.7	2.33
Rotation [°/m]	0.0040	0.0074	0.0030
Sequence 10			
Translation [%]	27.5	1.54	3.33
Rotation [°/m]	0.0041	0.0096	0.0049

TABLE 4.3 – RMSE en translation et rotation obtenues sur les séquences 05, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB) et l’IEKF. Les meilleurs résultats sont exprimés en vert.

4.4.2 Intégration d’un IEKF pour la fusion INS/VISION

Afin de tester l’ESKF avec un IEKF, on réalise l’algorithme tel qu’en figure 4.9 où l’on reprend l’architecture VINS proposée initialement en remplaçant notre navigation à l’estime inertielle par l’IEKF présenté plus tôt. L’IEKF étant en mesure de corriger les biais inertiels, il n’a plus besoin de la correction de biais de l’ESKF. Les résultats obtenus avec cette méthode sont disponibles dans la table 4.4.

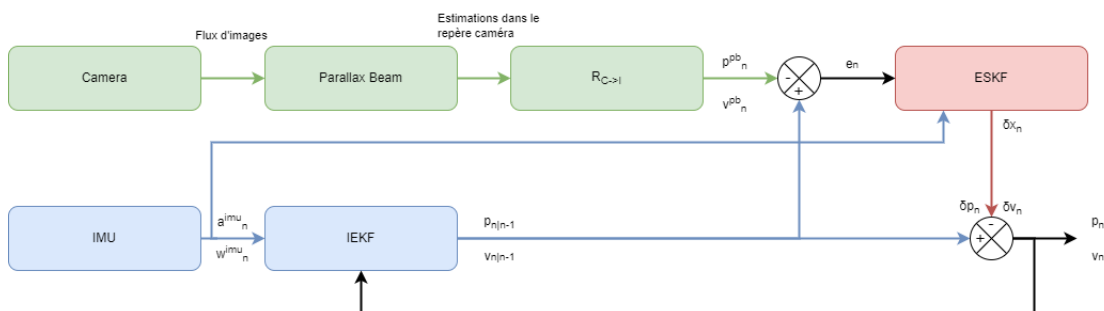


FIGURE 4.9 – Schéma-bloc illustrant l’algorithme ESKF employé en conjonction d’un IEKF pour un couplage lâche VINS.

Comme on peut le voir dans ce tableau, l’introduction de l’IEKF dans notre structure ESKF donne des résultats très dégradés par rapport à un IEKF seul. Ceci s’explique par la nécessité de conserver le modèle d’évolution d’une navigation à l’estime inertielle dans notre ESKF, ce qui rend la mesure fournie à l’ESKF inadéquate. En effet, le modèle d’évolution de notre ESKF est un modèle qui reprend la navigation à l’estime inertielle dont la qualité est bien différente de celle de l’IEKF. Ainsi, l’estimation en sortie de l’ESKF est mise à jour par une erreur plus faible que celle nécessaire pour corriger sa prédiction. En retour, cela a pour effet de fausser grandement la navigation

	DR	PB(10 Hz)	IEKF	ESKF+IEKF
Sequence 05				
Translation [%]	275	2.14	4.06	52
Rotation [$^{\circ}/m$]	0.0056	0.0083	0.0057	0.34
Sequence 09				
Translation [%]	37.5	2.7	2.33	57.5
Rotation [$^{\circ}/m$]	0.0040	0.0074	0.0030	0.36
Sequence 10				
Translation [%]	27.5	1.54	3.33	87
Rotation [$^{\circ}/m$]	0.0041	0.0096	0.0049	0.57

TABLE 4.4 – RMSE en translation et rotation obtenues sur les séquences 05, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB), l’IEKF et l’ESKF associé à l’IEKF.

Les meilleurs résultats sont exprimés en vert.

corrigée par l’estimation de l’ESKF, constant que l’on retrouve dans notre tableau. Par conséquent, pour bénéficier des apports sur la navigation inertielle d’un IEKF, il est nécessaire de proposer une structure de filtre capable d’exploiter correctement la sortie d’un IEKF.

4.4.3 Conclusion

L’ESKF est une solution intéressante pour réaliser un suivi efficace de l’estimation réalisée par Parallax Beam. En revanche, en raison d’une contribution inertielle faible et handicapante, cette méthode n’est pas en mesure de proposer un suivi robuste de Parallax Beam lors d’absence d’informations de sa part. De plus, cette solution n’est pas en mesure de corriger Parallax Beam en cas d’erreur de ce dernier. Afin de résoudre ces problèmes, un IEKF en lieu d’une navigation à l’estime a été envisagé pour renforcer la contribution inertielle. Néanmoins, on constate que cette solution dégrade finalement les performances de l’architecture ESKF proposée. Ceci est principalement lié à la structure de filtre ESKF et à son modèle de prédiction inertiel peu précis. Par conséquent, il paraît nécessaire de proposer une structure capable d’exploiter correctement notre contribution inertielle renforcée. A cette fin, l’étude d’une architecture de couplage adaptative multi-capteurs, telle que le MSF-EKF, va être conduite.

5 Multi-Sensor Fusion Extended Kalman Filter

5.1 Introduction

Dans le chapitre précédent, nous avons pu étudier un filtre de Kalman étendu à entrée erreur pour le couplage lâche visuel-inertiel. Dans une première approche, la navigation inertielle est obtenue par une simple méthode d'intégration (navigation à l'estime), remplacée dans un second temps par un filtre de Kalman étendu invariant. Bien que cette solution soit capable de suivre avec efficacité les estimations visuelles utilisées en tant qu'observation de l'ESKF, il se révèle inapte à apporter une correction significative à la navigation et maintient difficilement une estimation correcte en cas de coupures régulières du flux d'images. La cause principale étant une navigation à l'estime inertielle trop impactante, que nous avons alors proposé d'améliorer par un IEKF. Cependant, nous avons montré que la mise en cascade d'un IEKF et filtre ESKF n'est pas optimale au regard de la précision finale de la navigation VINS. Par conséquent, nous nous sommes tournés vers un autre type de couplage lâche : le Multi Sensor Fusion Extended Kalman Filter ou MSF-EKF [92]. Dans un premier temps, nous présenterons les équations de ce filtre et nous expliquerons son intérêt dans la résolution de notre problématique. Ensuite, nous montrerons et analyserons les résultats des tests effectués avec notre MSF-EKF sur le jeu de données KITTI vision. Enfin, nous conclurons sur les améliorations envisageables à apporter à l'ensemble par de futurs travaux.

5.2 MSF-EKF : fusion multi-capteurs par filtrage de Kalman Etendu

Le filtre MSF-EKF [92] (Multi Sensor Fusion EKF) a pour objectif d'assurer une navigation robuste et précise en associant plusieurs capteurs en gérant leur calibration, asynchronisme ainsi que les indisponibilités et délais. Faisant suite aux travaux de [119], le MSF-EKF est capable de traiter des mesures absolues et relatives non synchrones à partir d'un nombre important de sources (jusqu'à trois dans les travaux qui le présentent). Capables de gérer l'ajout en ligne de capteurs, il est aussi apte à fonctionner si un capteur fait défaut grâce à un tampon circulaire à état. Ce tampon assure une

propagation de l'état fourni par chaque capteur pour être au plus proche du point de linéarisation optimal de la trajectoire suivie. Contrairement à un ESKF, il prend en observation des mesures directes issues de différentes sources et peut donc fonctionner avec Parallax Beam et un IEKF inertiel comme mesures. De plus, cette solution permet de travailler aux fréquences nominales de fonctionnement de chaque capteur sans avoir à se soucier de leur synchronicité, celle-ci étant gérée par le tampon circulaire.

5.2.1 Principe du MSF-EKF

Représentation d'états

Tout comme l'ESKF, il est nécessaire de disposer des équations de la mécanique inertielle afin de définir les équations d'évolution du modèle de notre MSF-EKF. Ces équations ayant été définies pleinement dans les chapitres 2 et 4, on les considère connues. L'état \mathbf{x}_n que l'on souhaite estimer est ici plus complexe que l'ESKF. Tout d'abord, on souhaite estimer les états du véhicule dans le repère de navigation que sont sa position \mathbf{p}_n , sa vitesse \mathbf{v}_n et son orientation sous la forme d'un quaternion \mathbf{q}_n . De plus, on veut estimer les paramètres associés à nos différents systèmes de navigation, à savoir les biais \mathbf{b}_n^ω et \mathbf{b}_n^a (respectivement biais sur les gyromètres et les accéléromètres) de notre centrale inertielle et le facteur d'échelle λ_n de Parallax Beam qui définit le rapport entre la valeur réelle et la valeur estimée par Parallax Beam (qui varie faiblement autour de 1). Pour finir, on cherche à déterminer les paramètres d'alignement des différents repères de navigation, à savoir l'orientation entre le repère de navigation et le repère caméra \mathbf{q}_n^{wv} ainsi que l'orientation et le décalage entre le repère caméra et le repère inertiel, respectivement \mathbf{q}_n^{ci} et \mathbf{p}_n^{ci} . On obtient donc le vecteur d'état suivant :

$$\mathbf{x}_n = [\mathbf{p}_n, \mathbf{v}_n, \mathbf{q}_n, \mathbf{b}_n^\omega, \mathbf{b}_n^a, \lambda_n, \mathbf{q}_n^{wv}, \mathbf{q}_n^{ci}, \mathbf{p}_n^{ci}]. \quad (5.1)$$

Cet état obéit aux équations de la mécanique inertielle et on peut donc facilement les relier aux équations de la partie 2.3.3 :

$$\hat{\mathbf{p}}_t = \hat{\mathbf{v}}_t \quad (5.2)$$

$$\hat{\mathbf{v}}_t = \mathbf{C}_{(\hat{\mathbf{q}}_t)}^T * (\mathbf{a}_t^{imu} - \hat{\mathbf{b}}_t^a) - \mathbf{g} \quad (5.3)$$

$$\hat{\mathbf{q}}_t = \frac{1}{2} \mathbf{\Omega}_t (\boldsymbol{\omega}_t^{imu} - \hat{\mathbf{b}}_t^\omega) \hat{\mathbf{q}}_t \quad (5.4)$$

$$\hat{\mathbf{b}}_t^\omega = 0; \hat{\mathbf{b}}_t^a = 0; \hat{\lambda}_t = 0 \quad (5.5)$$

$$\hat{\mathbf{q}}_t^{wv} = 0; \hat{\mathbf{q}}_t^{ci} = 0; \hat{\mathbf{p}}_t^{ci} = 0, \quad (5.6)$$

avec \mathbf{g} le vecteur de gravitation exprimé dans le repère de navigation et \mathbf{C}_q la matrice de rotation directe associée au quaternion \mathbf{q} . Ici, $\mathbf{C}_{(\hat{\mathbf{q}}_t)}$ équivaut à $\mathbf{C}_{imu_t}^N$ de la section 4.2.1.

Dans le cas présent, on fait usage d'un quaternion pour représenter les différentes orientations que l'on souhaite estimer. Dans ce cas, afin de comparer différentes orientations, il est recommandé d'avoir recours à un quaternion-erreur [118] plutôt qu'à une simple différence mathématique pour des raisons de stabilité, en évitant le maintien d'un quaternion unitaire qui exige une matrice de covariance singulière [83]. On définit le quaternion-erreur entre le quaternion réel \mathbf{q}_n^x et le quaternion estimé $\hat{\mathbf{q}}_n^x$ par une approximation des petits angles comme suit :

$$\delta \mathbf{q}_n^x = \mathbf{q}_n^x \otimes \hat{\mathbf{q}}_n^{x^{-1}} \approx \left[\frac{1}{2} \delta \theta_n^{xT} \mathbf{1} \right]^T. \quad (5.7)$$

Le vecteur d'erreur angulaire résultant $\delta \theta_n^{xT}$ est alors la quantité que l'on va évaluer via un état-erreur.

Par conséquent, on va privilégier l'estimation d'un état-erreur par notre filtre, les autres éléments de celui-ci pouvant être définis par une simple soustraction telle que $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n$. On peut alors définir l'état-erreur suivant :

$$\tilde{\mathbf{x}}_n = [\Delta \mathbf{p}_n, \Delta \mathbf{v}_n, \delta \theta_n, \Delta \mathbf{b}_n^\omega, \Delta \mathbf{b}_n^a, \Delta \lambda_n, \delta \theta_n^{wv}, \delta \theta_n^{ci}, \Delta \mathbf{p}_n^{ci}]. \quad (5.8)$$

Cependant, comme on dispose des informations nécessaires qui lient l'état et son état-erreur, le filtre prend en entrée des mesures d'état et fournit en sortie des estimations d'état. Les équations différentielles associées à notre état-erreur sont alors :

$$\Delta \dot{\mathbf{p}}_t = \Delta \mathbf{v}_t \quad (5.9)$$

$$\Delta \dot{\mathbf{v}}_t = -\mathbf{C}_{(\hat{\mathbf{q}}_t)}^T * (S(\hat{\mathbf{a}}_t) \cdot \delta \theta_t - \Delta \mathbf{b}_t^a - \mathbf{n}_a) \quad (5.10)$$

$$\delta \dot{\theta}_t = -S(\hat{\omega}_t) \cdot \delta \theta_t - \Delta \mathbf{b}_t^\omega - \mathbf{n}_\omega \quad (5.11)$$

$$\Delta \dot{\mathbf{b}}_t^\omega = \mathbf{n}_{b^\omega}; \Delta \dot{\mathbf{b}}_t^a = \mathbf{n}_{b^a}; \Delta \dot{\lambda}_t = 0 \quad (5.12)$$

$$\delta \dot{\theta}_t^{wv} = 0; \delta \dot{\theta}_t^{ci} = 0; \Delta \dot{\mathbf{p}}_t^{ci} = 0, \quad (5.13)$$

avec $\hat{\omega} = \omega^{imu} - \hat{\mathbf{b}}^\omega$ et $\hat{\mathbf{a}} = \mathbf{a}^{imu} - \hat{\mathbf{b}}^a$. Ces équations peuvent être résumées par l'équation d'état continue linéarisée suivante :

$$\dot{\tilde{\mathbf{x}}} = \mathbf{F}_c \cdot \tilde{\mathbf{x}} + \mathbf{G}_c \cdot N \quad (5.14)$$

Avec N le vecteur des bruits $N = [\mathbf{n}_a^T, \mathbf{n}_{b^a}^T, \mathbf{n}_\omega^T, \mathbf{n}_{b^\omega}^T]^T$ qui sont les bruits blancs supposés gaussiens qui perturbent les accélérations, vitesses de rotation et leurs biais.

La solution choisie, comme dans [119], essaye de maximiser la vitesse d'exécution de l'algorithme. Pour cela, on fait l'hypothèse que les matrices d'état continues du système \mathbf{F}_c et \mathbf{G}_c restent constantes sur la durée d'un intervalle de temps d'intégration noté Δt [118]. Par conséquent, on peut réaliser la discrétisation suivante (approximation en

séries de Taylor) :

$$\mathbf{F}_d = \exp(\mathbf{F}_c \cdot \Delta t) = \mathbf{I}_{25} + \mathbf{F}_c \cdot \Delta t + \frac{1}{2!} \cdot \mathbf{F}_c^2 \cdot \Delta t^2 + \dots \quad (5.15)$$

De plus, on fait l'approximation des petits angles pour laquelle $|\omega| \rightarrow 0$ et, par application de la règle de Bernouilli, on obtient la matrice d'état discrète suivante, comme précisé dans [118] et [119] :

$$\mathbf{F}_d = \begin{bmatrix} \mathbf{I}_3 & \Delta t & \mathbf{A} & \mathbf{B} & -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \frac{\Delta t^2}{2} & \mathbf{0}_{3 \times 10} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{C} & \mathbf{D} & -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \Delta t & \mathbf{0}_{3 \times 10} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{E} & \mathbf{F} & \mathbf{0}_3 & \mathbf{0}_{3 \times 10} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 10} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3 \times 10} \\ \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{I}_{10} \end{bmatrix}, \quad (5.16)$$

avec :

$$\mathbf{A} = -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T S(\mathbf{a}_n^{imu}) \left(\frac{\Delta t^2}{2} - \frac{\Delta t^3}{3!} S(\omega_n^{imu}) + \frac{\Delta t^4}{4!} S(\omega_n^{imu})^2 \right), \quad (5.17)$$

$$\mathbf{B} = -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T S(\mathbf{a}_n^{imu}) \left(-\frac{\Delta t^3}{3!} + \frac{\Delta t^4}{4!} S(\omega_n^{imu}) - \frac{\Delta t^5}{5!} S(\omega_n^{imu})^2 \right), \quad (5.18)$$

$$\mathbf{C} = -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T S(\mathbf{a}_n^{imu}) \left(\Delta t - \frac{\Delta t^2}{2!} S(\omega_n^{imu}) + \frac{\Delta t^3}{3!} S(\omega_n^{imu})^2 \right), \quad (5.19)$$

$$\mathbf{D} = -\mathbf{A} \quad (5.20)$$

$$\mathbf{E} = \mathbf{I}_3 - \Delta t S(\omega_n^{imu}) + \frac{\Delta t^2}{2!} S(\omega_n^{imu})^2, \quad (5.21)$$

$$\mathbf{F} = -\Delta t + \frac{\Delta t^2}{2!} S(\omega_n^{imu}) + \frac{\Delta t^3}{3!} S(\omega_n^{imu})^2, \quad (5.22)$$

Quant à \mathbf{G}_c , elle est définie comme suit :

$$\mathbf{G}_c = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}_{(\hat{\mathbf{q}}_n)}^T & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 3} \end{bmatrix}. \quad (5.23)$$

Ceci permet alors de calculer la matrice discrète de covariance des bruits sur l'état \mathbf{Q}_d :

$$\mathbf{Q}_d = \int_{\Delta t} \mathbf{F}_d(\tau) \cdot \mathbf{G}_c \cdot \mathbf{Q}_c \cdot \mathbf{G}_c^T \cdot \mathbf{F}_d(\tau)^T \cdot d\tau \quad (5.24)$$

Avec la matrice continue de covariance des bruits sur l'état \mathbf{Q}_c telle que $\text{diag}(\mathbf{Q}_c) = [\sigma_{\mathbf{n}_a}^2, \sigma_{\mathbf{n}_{b^a}}^2, \sigma_{\mathbf{n}_\omega}^2, \sigma_{\mathbf{n}_{b^\omega}}^2]$.

L'état est alors propagé comme un EKF classique à l'exception des quaternions qui sont propagés via une intégration du premier ordre détaillée dans [118].

Modèle de mesure

Afin d'exploiter l'information du mouvement extraite du Parallax Beam qui est la matrice homographique \mathbf{M}_n^{pb} , on définit une matrice homographique \mathbf{P}_n décrivant le mouvement du véhicule dans le repère caméra d'après Parallax Beam.

$$\mathbf{P}_n = \begin{bmatrix} \mathbf{R}_n^{pbv} & \mathbf{p}_n^{pbv} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.25)$$

avec \mathbf{R}_n^{pbv} et \mathbf{p}_n^{pbv} la matrice d'orientation et le vecteur position du véhicule dans le repère vision d'après Parallax Beam. \mathbf{P}_n est initialisée aux positions et orientations initiales du véhicule au début du mouvement \mathbf{R}_0^{pbv} et \mathbf{p}_0^{pbv} . Dès que Parallax Beam calcule une nouvelle matrice homographique, \mathbf{P}_n est calculée comme suit :

$$\mathbf{P}_n = \mathbf{P}_{n-1} \cdot \mathbf{M}_n^{pb}. \quad (5.26)$$

Ceci nous permet de définir notre mesure de position :

$$\mathbf{z}_n^p = \mathbf{p}_n^{pbv}. \quad (5.27)$$

On fait l'hypothèse que les différents repères sont correctement alignés au préalable d'une acquisition en utilisant la matrice $\mathbf{R}_{(imu_cam)}$ (voir chapitre 2). De fait les quaternions \mathbf{q}_n^{wv} et \mathbf{q}_n^{ci} ont pour valeurs $\mathbf{q}^0 = [1, 0, 0, 0]^T$. Ceci nous permet de définir notre vecteur de mesure comme ci-dessous :

$$\mathbf{z}_n = \begin{bmatrix} \mathbf{p}_n^{pbv} \\ \mathbf{q}^0 \\ \mathbf{q}^0 \end{bmatrix}. \quad (5.28)$$

Or, on peut aussi définir le modèle de mesure suivant, qui associe la mesure Parallax Beam dans le repère caméra à cette mesure dans le repère inertiel, comme la composition suivante :

$$\mathbf{z}_n^p = \mathbf{p}_n^{pbv} = \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot (\mathbf{p}_n^{iw} + \mathbf{C}_{(\hat{\mathbf{q}}_n^{iw})}^T \cdot \mathbf{p}_n^{pbi}) \cdot \lambda_n + \mathbf{n}_p \quad (5.29)$$

Avec $\mathbf{C}_{(\hat{\mathbf{q}}_n^{iw})}^T$ l'orientation de l'IMU, $\mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T$ la rotation entre le repère visuel et le repère de navigation, et \mathbf{n}_p le bruit de mesure associé à la mesure de position. L'erreur de position est alors :

$$\tilde{\mathbf{z}}_n^p = \mathbf{z}_n^p - \hat{\mathbf{z}}_n^p = \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot (\mathbf{p}_n^{iw} + \mathbf{C}_{(\hat{\mathbf{q}}_n^{iw})}^T \cdot \mathbf{p}_n^{pbi}) \cdot \lambda_n + \mathbf{n}_p - \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot (\hat{\mathbf{p}}_n^{iw} + \mathbf{C}_{(\hat{\mathbf{q}}_n^{iw})}^T \cdot \hat{\mathbf{p}}_n^{pbi}) \cdot \hat{\lambda}_n \quad (5.30)$$

Cette expression peut être linéarisée en :

$$\tilde{\mathbf{z}}_n^p = \mathbf{H}_p \cdot \tilde{\mathbf{x}}_n \quad (5.31)$$

Avec :

$$\mathbf{H}_p^T = \begin{bmatrix} \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \hat{\lambda}_n \\ 0_3 \\ -\mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \cdot \mathcal{S}(\hat{\mathbf{p}}_n^{pbi}) \cdot \hat{\lambda}_n \\ 0_3 \\ 0_3 \\ \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \cdot \hat{\mathbf{p}}_n^{pbi} + \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \hat{\mathbf{p}}_n \\ -\mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \mathcal{S}(\hat{\mathbf{p}}_n + \mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \cdot \hat{\mathbf{p}}_n^{pbi}) \cdot \hat{\lambda}_n \\ 0_3 \\ \mathbf{C}_{(\hat{\mathbf{q}}_n^{wv})}^T \cdot \mathbf{C}_{(\hat{\mathbf{q}}_n)}^T \cdot \hat{\lambda}_n \end{bmatrix} \quad (5.32)$$

Par conséquent :

$$\tilde{\mathbf{z}}_n = \mathbf{H}\tilde{\mathbf{x}} = \begin{bmatrix} & & & & \mathbf{H}_p & & & & & \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0 & \mathbf{I}_3 & 0_3 & 0_3 & \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0 & 0_3 & \mathbf{I}_3 & 0_3 & \end{bmatrix} \tilde{\mathbf{x}} \quad (5.33)$$

Cette matrice de mesure \mathbf{H} est ensuite utilisée pour réaliser la routine de mise à jour classique d'un EKF :

- calculer le résidu : $\tilde{\mathbf{z}}_n = \mathbf{z}_n - \hat{\mathbf{z}}_n$,
- calculer le gain de Kalman : $\mathbf{K} = \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}$ avec \mathbf{R} la matrice de covariance de la mesure,
- calculer la correction : $\hat{\mathbf{x}}_n = \mathbf{K}\tilde{\mathbf{z}}_n$.

La covariance de l'erreur du filtre \mathbf{P} est ensuite propagée selon la routine usuelle et la correction appliquée à l'état \mathbf{x} dès lors que la correction valide un test de Mahalanobis [79].

En plus de cette structure EKF, le MSF-EKF comprend une routine nommée tampon circulaire qui permet de prendre en compte plusieurs mesures asynchrones et de propager a posteriori une covariance de l'erreur modifiée par ces nouvelles informations. Bien que très intéressant, cet outil est pour le moment exploité au minimum avec des mesures venant uniquement de Parallax Beam en gérant les données-capteurs selon leur chronologie d'arrivée. Une exploitation plus approfondie de cet outil requiert de disposer d'au moins deux sources de mesures différentes.

L'architecture proposée ici, issue des travaux de [118], [119] et [92], est à l'image de l'ESKF basée sur un état-erreur dont la structure présente l'avantage de rendre l'estimateur plus robuste face aux mauvaises performances de centrales inertielles à faible coût grâce à sa matrice d'état discrète qui contient plus d'informations que celle de

Ceci exprime notre intervalle attendu pour les valeurs initiales de notre état, soit $[-0.3;0.3]$ m pour la position, $[-15;15]$ m/s pour la vitesse et le quaternion unitaire pour l'orientation. Pour définir le bruit de mesure, il a été décidé de réduire la confiance que l'on a dans l'estimation en position de Parallax Beam en fixant une confiance élevée dans l'alignement des repères puisque celui-ci est effectué manuellement en amont du filtre. Pour le bruit de modèle, nous avons utilisé les spécifications de la centrale OXTS3003 utilisée pour le jeu de données KITTI vision afin de définir les variances des bruits $\sigma_{n_a}^2$ et $\sigma_{n_g}^2$. La matrice \mathbf{Q} est alors calculée comme présentée dans [118] et [119] à partir de ces variances.

$$\text{diag}(\mathbf{R}) = [0.01, 0.01, 0.01, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}] \quad (5.35)$$

$$\sigma_{n_a}^2 = 2.7 \cdot 10^{-6} \quad m.s^{-2} \quad (5.36)$$

$$\sigma_{n_g}^2 = 0.0044 \quad rad.s^{-1} \quad (5.37)$$

Comme pour l'ESKF, les tests ont été réalisés sur les séquences 04, 05 (figures 5.2, 5.3 et 5.4), 06, 09 et 10 du jeu de données KITTI vision qui sont des séquences de validation pour lesquelles la vérité terrain est disponible. Afin de vérifier la convergence du filtre, les trois premiers coefficients diagonaux de la matrice de covariance de l'erreur \mathbf{P} sont analysés pour s'assurer de leur convergence vers zéro. Seules les figures sur la séquence 05 sont montrées dans le corps du chapitre. Les autres séquences peuvent être trouvées en fin de chapitre, page 137, mais servent tout de même à l'étude qui suit.

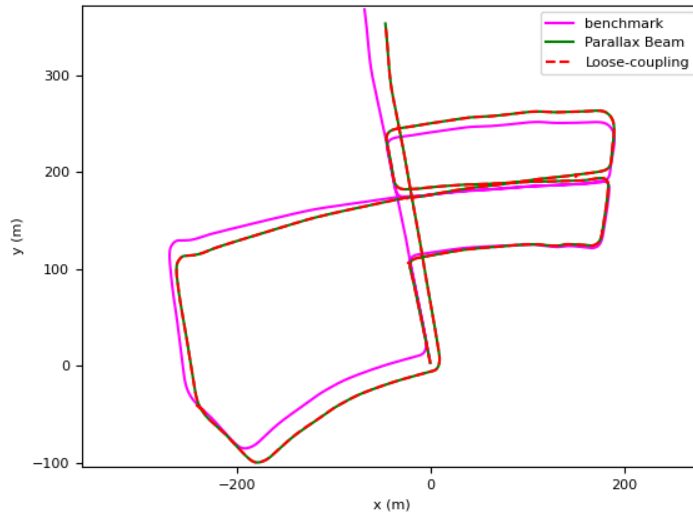


FIGURE 5.2 – Estimées sur la séquence 05 du jeu de données KITTI vision.

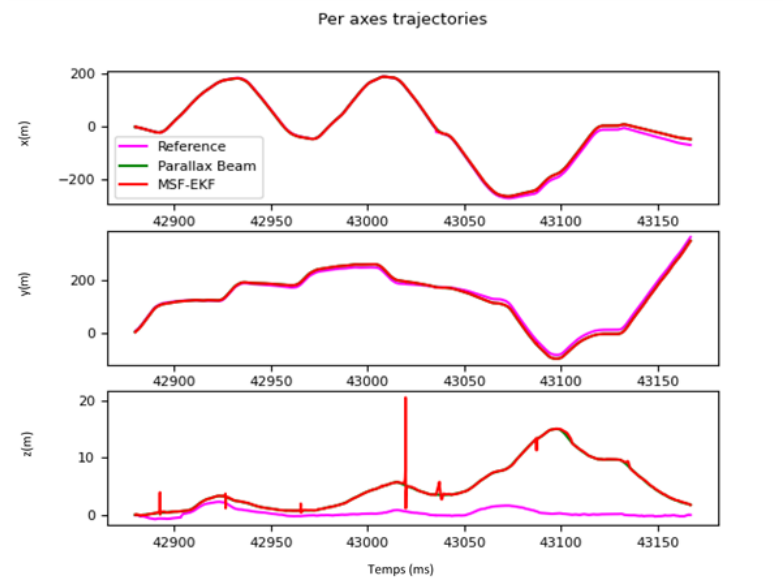


FIGURE 5.3 – Estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence.

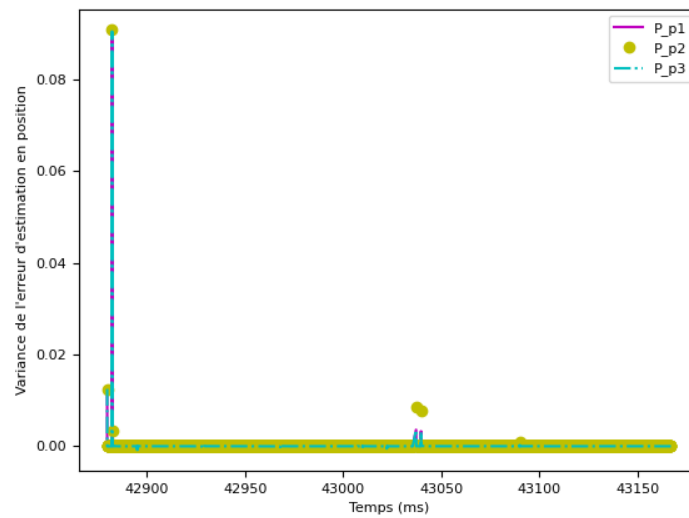


FIGURE 5.4 – Trois premiers termes diagonaux de la matrice de covariance de l'erreur d'estimation de l'ESKF sur la séquence 05 du jeu de données KITTI vision.

Sur ces figures, on constate que l'estimation du MSF-EKF parvient correctement à suivre le Parallax Beam en dépit de dérives sur l'axe vertical, qui est celui pour lequel Parallax Beam est le moins précis sur KITTI en raison d'une orientation de la caméra différente de celle pour laquelle il a été conçu. Ceci est confirmé par les variances de l'erreur d'estimation qui tendent rapidement vers zéro sauf lors d'un outlier important qui est alors rapidement corrigé.

Cette tendance est identique sur toutes les trajectoires étudiées. Afin de confirmer la qualité de ces estimations, la RMSE calculée avec l’algorithme de validation du jeu de données KITTI vision est disponible en table 5.1. Afin d’obtenir ces RMSE, la sortie du MSF-EKF a été sous-échantillonnée à 10 Hz pour pouvoir être comparée à la vérité terrain, également à 10 Hz.

	DR(100 Hz)	PB(10 Hz)	MSF-EKF(100 Hz)
Sequence 04			
Translation [%]	24.3	1.63	1.48
Rotation [°/m]	0.000022	0.000049	0.0009
Sequence 05			
Translation [%]	275	2.14	0.61
Rotation [°/m]	0.0056	0.0083	0.0053
Sequence 06			
Translation [%]	76	3.37	0.75
Rotation [°/m]	0.000079	0.000216	0.0063
Sequence 09			
Translation [%]	37.5	2.7	0.89
Rotation [°/m]	0.0040	0.0074	0.006
Sequence 10			
Translation [%]	27.5	1.54	1.08
Rotation [°/m]	0.0041	0.0096	0.0079

TABLE 5.1 – RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (MSF-EKF). Les meilleurs résultats sont exprimés en vert.

Les RMSE permettent de constater une amélioration de la précision en translation de la fusion par rapport à Parallax Beam seul. La réduction de la confiance accordée à Parallax Beam ainsi que la plus grande cadence d’échantillonnage du filtre et la méthode de réjection de valeurs aberrantes en sortie de mise à jour du filtre contribuent à un gain significatif des performances de navigation en translation et en rotation. On peut toutefois noter qu’en rotation, la navigation inertielle seule reste en générale la plus performante.

5.3.2 Test de coupure

Afin de tester la robustesse de la fusion face à un défaut de Parallax Beam, les mêmes trajectoires ont été évaluées cette fois avec une coupure totale des informations issues du Parallax Beam comme on peut le voir sur les figures 5.5, 5.6 et 5.7. Ainsi, les points de mesures 400 à 500 ne sont pas fournis au MSF-EKF, qui ne peut donc pas réaliser sa mise à jour et doit alors reposer son estimation sur sa seule prédiction. Ceci permet de simuler une interruption du Parallax Beam de dix secondes à 10 Hz au bout d’un nombre suffisant de mises à jours pour garantir la convergence du filtre, en particulier

sur l'estimation des biais inertiels. On peut noter que la séquence 04, ne contenant que 270 points de mesures Parallax Beam, n'a pas été testée selon ce principe.

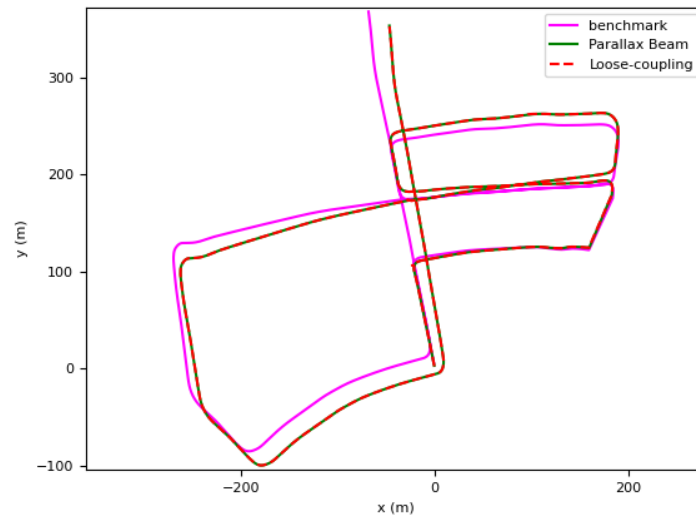


FIGURE 5.5 – Estimées sur la séquence 05 du jeu de données KITTI vision avec indisponibilité du Parallax Beam de 10 secondes

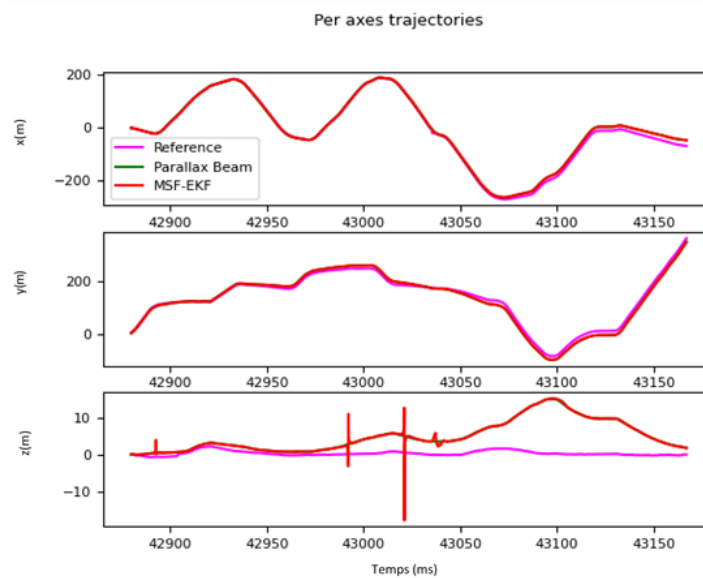


FIGURE 5.6 – Estimée sur la séquence 05 du jeu de données KITTI vision sur chaque axe du repère de référence avec indisponibilité du Parallax Beam de 10 secondes.

En observant les variances de l'erreur 5.7, on constate que la coupure et l'erreur qu'elle provoque est détectée et absorbée par le filtre. Ceci explique l'absence apparente d'erreur

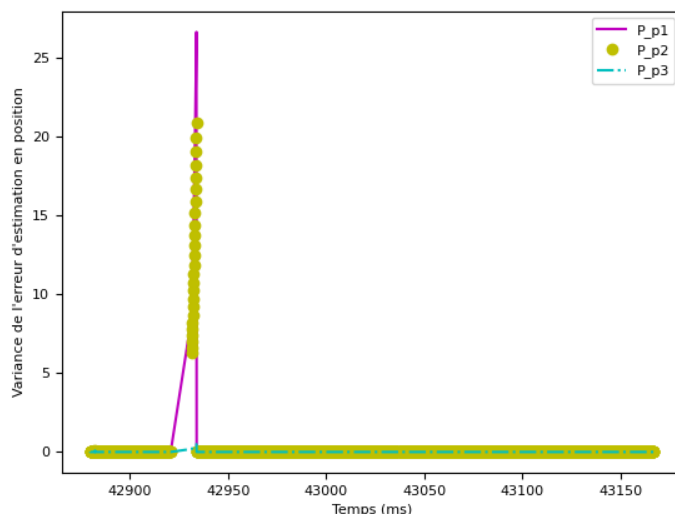


FIGURE 5.7 – Trois premiers termes diagonaux de la matrice de covariance de l’erreur d’estimation de l’ESKF sur la séquence 05 du jeu de données KITTI vision avec indisponibilité du Parallax Beam de 10 secondes.

importante d’estimation sur nos trajectoires. On peut confirmer la faible dégradation de nos résultats à cause de la coupure en consultant la table 5.2.

	DR	PB(10 Hz)	MSF-EKF sans coupure	MSF-EKF avec coupure
Sequence 04				
Translation [%]	24.3	1.63	1.48	-
Rotation [°/m]	0.000022	0.000049	0.0009	-
Sequence 05				
Translation [%]	275	2.14	0.61	0.69
Rotation [°/m]	0.0056	0.0083	0.0053	0.0054
Sequence 06				
Translation [%]	76	3.37	0.75	0.72
Rotation [°/m]	0.000079	0.000216	0.0063	0.0043
Sequence 09				
Translation [%]	37.5	2.7	0.89	0.89
Rotation [°/m]	0.0040	0.0074	0.006	0.006
Sequence 10				
Translation [%]	27.5	1.54	1.08	1.61
Rotation [°/m]	0.0041	0.0096	0.0079	0.005

TABLE 5.2 – RMSE en translation et rotation obtenues sur les séquences 04, 05, 06, 09 et 10 du jeu de données KITTI vision avec une navigation à l’estime classique (DR), l’algorithme Parallax Beam (PB) et la méthode de couplage lâche proposée (MSF-EKF) avec et sans coupure. Les meilleurs résultats sont exprimés en vert.

On constate ici que les coupures ont un impact très faible sur la précision en navigation avec une augmentation maximale de 0.53%. De plus, on observe que la séquence 06

présente même une amélioration de sa précision lors de la coupure, qui survient à un instant où Parallax Beam fait montre d'une perte de précision. Ceci nous permet de conclure sur la robustesse de la solution de navigation proposée vis-à-vis d'une coupure de 10 secondes.

Cependant, on peut tout de même remarquer qu'ici aussi, la fusion proposée ne corrigera pas Parallax Beam en cas de fausse estimation suite à un aléa, celui-ci étant l'unique observation et donc référence du filtre. Ceci peut être solutionné par l'ajout d'autres sources d'observations.

5.4 Conclusion

Afin de répondre au besoin d'une meilleure prédiction inertielle et d'être compatible avec des solutions de navigation modernes comme l'IEKF, le MSF-EKF s'est présenté comme une solution intéressante à étudier. Après avoir présenté les équations d'états et d'observations, nous avons testé ce filtre sur le jeu de données KITTI vision. Le MSF-EKF parvient à réaliser une fusion qui propose une navigation au moins aussi précise que Parallax Beam tout en assurant une robustesse à des coupures d'environ 10 secondes sur KITTI. Cette fusion, déjà performante, peut encore être améliorée de plusieurs manières. Tout d'abord, il serait intéressant d'intégrer une mesure d'attitude pour la mise à jour du filtre, à condition que l'observabilité du système soit suffisante. Ensuite, profiter de la qualité première du Multi Sensor Fusion EKF pour adjoindre au Parallax Beam un IEKF inertielle ou un autre capteur déjà embarqué pour augmenter l'observabilité du système et la redondance des informations en cas de coupure d'un capteur. Il pourrait également être utile d'être en mesure de prévenir une mauvaise estimation servant à la mise à jour du filtre en l'identifiant en amont. Pour ce faire, les techniques de détections de contextes présentées dans [41] et combinées à des méthodes de rejets de valeurs aberrantes ("outliers") permettrait de simplement couper les mauvaises mesures quand elles sont détectées. Enfin, un réglage adaptatif des paramètres de bruits de modèle et de mesure à la manière de [11] semble également proposer une piste intéressante d'amélioration de la solution proposée.

6 Conclusion et Perspectives

6.1 Conclusion

Notre objectif est de renforcer la navigation des systèmes STAMINA d'une centrale inertielle avec, comme cible, le maintien d'une bonne précision, de robustesse face à un défaut de capteur, et le tout dans une logique bas coût.

Dans ce but, nous avons présenté dans un premier temps les acteurs, enjeux et problématiques du sujet et son positionnement dans le contexte plus large de la navigation au travers des chapitres 1 et 2.

Ensuite, nous avons vu dans le chapitre 3 un état de l'art des systèmes de navigation VINS ("Visual-Inertial Navigation Systems") qui nous a permis de nous positionner en faveur d'un couplage lâche tout en fixant les exigences que nous souhaitons satisfaire.

Dans les chapitres 4 et 5, nous avons vu deux méthodes répondant à la problématique de fusion visuelle-inertielle afin d'apporter de la robustesse et de la précision à un algorithme de navigation visuelle complété par une centrale inertielle. Ces deux méthodes, le "Error-State Kalman Filter" (ESKF) et le "Multi Sensor Fusion Extended Kalman Filter" (MSF-EKF) proposent des performances sensiblement différentes en précision, robustesse et rapidité que nous allons comparer dans le présent chapitre en guise de conclusion. À la suite de ces comparaisons, nous nous intéresserons aux pistes d'améliorations et aux futurs travaux qui pourront venir renforcer la qualité de la navigation proposée.

6.2 Étude comparative

Les algorithmes ESKF (chapitre 4) et MSF-EKF (chapitre 5) sont comparés par rapport à leurs performances sur le jeu de données KITTI vision benchmark sur la base de trois paramètres : la précision (minimisation de l'erreur quadratique moyenne de l'estimation), la robustesse (minimisation de l'erreur quadratique moyenne dans des conditions dégradées) et la rapidité (temps de calcul d'une trajectoire).

Pour ce faire, les données présentées dans les chapitres précédents vont être réutilisées.

Précision

Afin de comparer la précision de nos deux méthodes, nous exploitons leurs résultats sur les trajectoires 04, 05, 06, 09 et 10 de KITTI et nous les avons aussi comparés aux résultats de la navigation visuelle Parallax Beam seule. Les résultats de ces comparaisons sont disponibles figures 6.1 pour la translation et 6.2 pour la rotation.

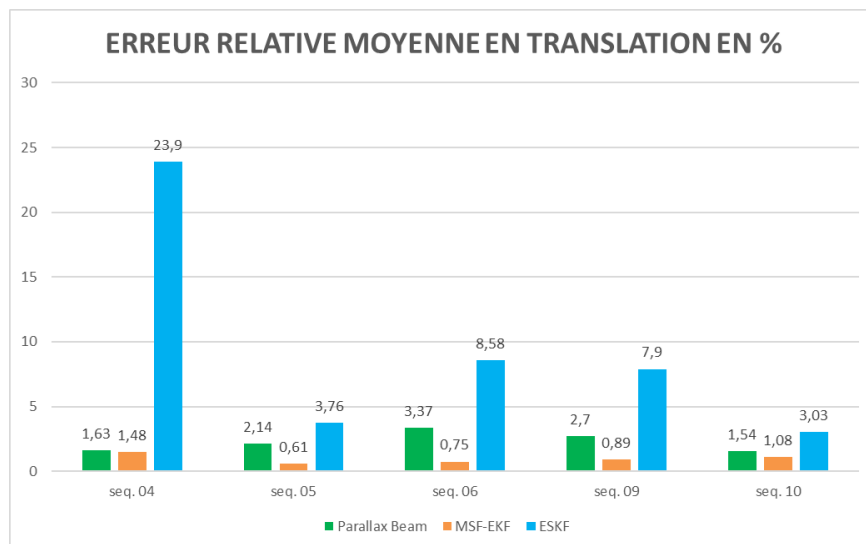


FIGURE 6.1 – Histogramme des erreurs relatives moyennes en translation (fois cent mètres par mètre parcouru) sur les séquences 04, 05, 06, 09 et 10 du KITTI vision benchmark.

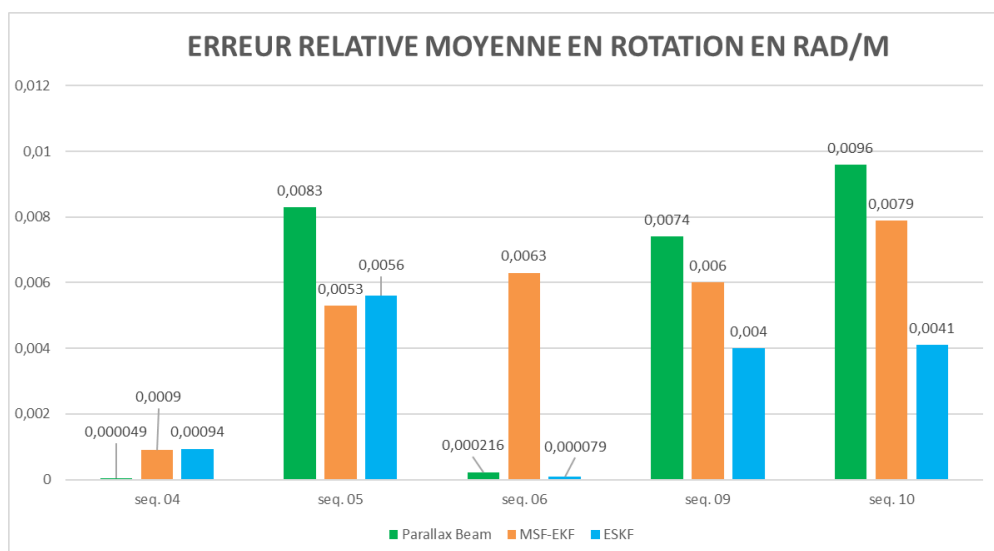


FIGURE 6.2 – Histogramme des erreurs relatives moyennes en rotation (en radians par mètre) sur les séquences 04, 05, 06, 09 et 10 du KITTI vision benchmark.

Sur ces histogrammes, on constate que le MSF-EKF propose des précisions en translation bien supérieures à celles de l'ESKF (environ un ordre de grandeur d'écart) qui peuvent s'expliquer par le choix d'effectuer la décomposition en série de Taylor à un ordre plus important (jusqu'à l'ordre 5) pour la matrice de propagation discrète de l'ESKF, là où nous contentions du premier ordre. Cette constatation se retrouve également dans les travaux théoriques de [86], et en particulier dans [109] qui évalue de manière expérimentale ce phénomène. De plus, le MSF-EKF est capable de rejeter une mauvaise estimation via un calcul de la distance de Mahalanobis que l'ESKF ne réalise pas. Le MSF-EKF propose aussi une meilleure précision en translation que la navigation visuelle seule.

En revanche, le MSF-EKF ne performe pas de manière aussi flagrante en rotation et propose même plutôt des performances moins bonnes que l'ESKF. Ceci peut être dû au fait que l'orientation n'est pas directement mesurée par le filtre et donc les choix des coefficients diagonaux de la matrice de covariance R accordent une confiance peut-être trop importante en faveur d'un bon alignement entre nos différents repères. L'ESKF, en réutilisant simplement un modèle de navigation à l'estime inertielle, conserve la bonne précision en rotation de la navigation inertielle.

Néanmoins, la perte de précision en rotation est légère comparée aux gains importants de précision en translation. On conclue donc que le MSF-EKF est la meilleure solution en matière de précision.

Robustesse

Comparer directement la robustesse de nos deux méthodes est difficile car l'ESKF, de par son bon tracking du Parallax Beam et la faible qualité de la navigation inertielle qui lui est associée, diverge très rapidement lorsque le Parallax Beam diverge ou est indisponible.

L'ESKF, en raison d'une navigation inertielle de mauvaise qualité en entrée, diverge très rapidement lorsqu'il est privé d'informations issues de la navigation visuelle durant plusieurs prédictions successives. Pour cela, la robustesse de l'ESKF a été évaluée face à une diminution de la fréquence d'échantillonnage de la navigation visuelle en sous-échantillonnant cette dernière à 5 Hz tout en conservant des données inertielles à 10 Hz (voir chapitre 4).

Le MSF-EKF fonctionne par défaut avec une centrale inertielle non synchronisée en fréquence avec l'odométrie visuelle (100 Hz pour l'IMU et 10 Hz pour la vision sur KITTI). Afin de vérifier la robustesse de notre méthode face à un aléa de la navigation visuelle, nous avons décidé de simuler une coupure. Ainsi, pendant dix secondes, l'odométrie visuelle ne fournit plus de mesure au MSF-EKF pour sa mise à jour. Ceci permet de vérifier la capacité de la méthode proposée à conserver une précision correcte en dépit d'une absence de mise à jour pendant une durée importante.

Afin de mesurer l'impact des perturbations que nous avons introduites, nous avons

tracé les histogrammes exprimant l'écart de l'erreur relative moyenne entre les évaluations perturbées et non perturbée. Ceci produit les trois figures 6.3, 6.4 et 6.5, la perturbation imposée à l'ESKF n'affectant pas sa précision en rotation. Il est important néanmoins de retenir que ses erreurs relatives sont obtenues en mettant en échec nos solutions de navigation différemment du fait des limitations de l'ESKF.

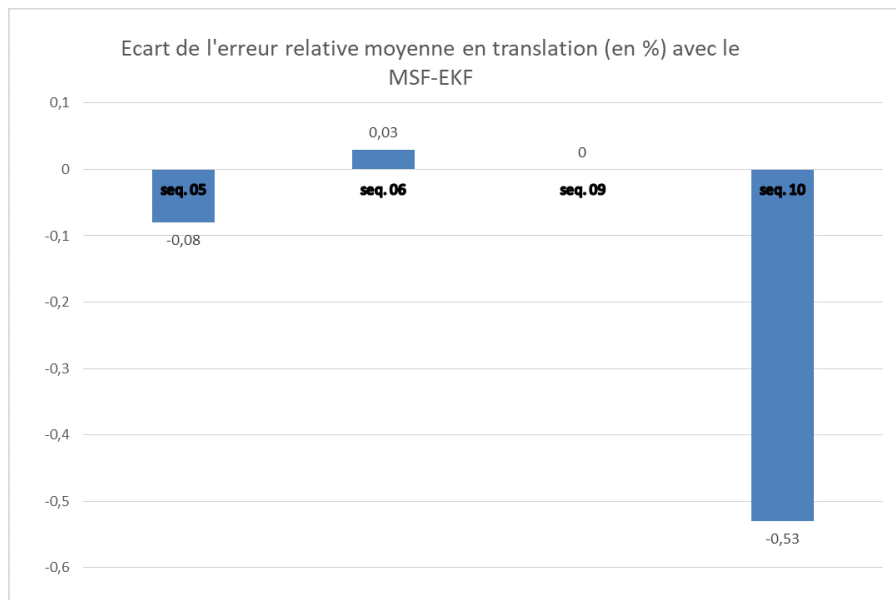


FIGURE 6.3 – Histogramme des écarts de l'erreur relative moyenne en translation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour le MSF-EKF entre l'évaluation perturbées et non perturbées.

Ces figures nous permettent de tirer des informations supplémentaires sur nos filtres. Tout d'abord, on constate que sur la séquence 06, le MSF-EKF et l'ESKF affichent tous deux une amélioration de leur précision dans un contexte de réduction de l'information issue de la navigation visuelle. On peut en conclure que, sur cette séquence, la navigation visuelle est en erreur sur certaines portions de la séquence. Comme cela peut être vu sur les figures en fin du chapitre 4, la navigation visuelle affiche sur cette séquence une dérive plus importante sur son axe longitudinal et vertical que sur les autres séquences. De plus, la figure 6.4 nous permet de constater une amélioration de la précision en rotation sur plusieurs séquences. Ceci nous permet de confirmer l'impact négatif des mesures visuelles sur l'estimation de l'orientation du mobile.

Enfin, on peut constater sur ces histogrammes que la perturbation imposée au MSF-EKF a peu affectée sa précision. Ceci conjugué au fait que l'ESKF n'est pas en mesure d'être robuste à une coupure de plusieurs secondes contrairement au MSF-EKF, nous permet de conclure que le MSF-EKF est bien une méthode plus robuste face à des interruptions pouvant affecter la navigation visuelle. Néanmoins, aucune des deux

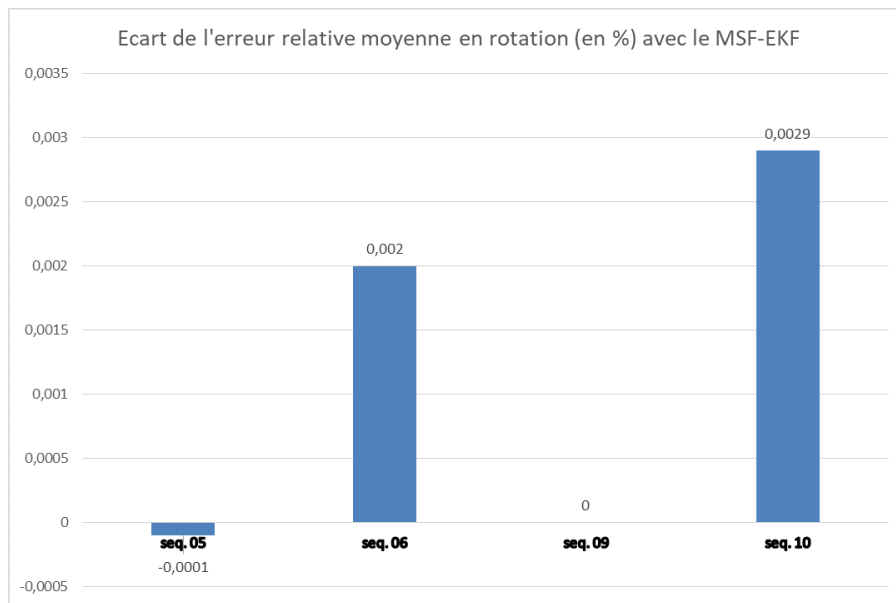


FIGURE 6.4 – Histogramme des écarts de l'erreur relative moyenne en rotation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour le MSF-EKF entre l'évaluation perturbée et non perturbée.

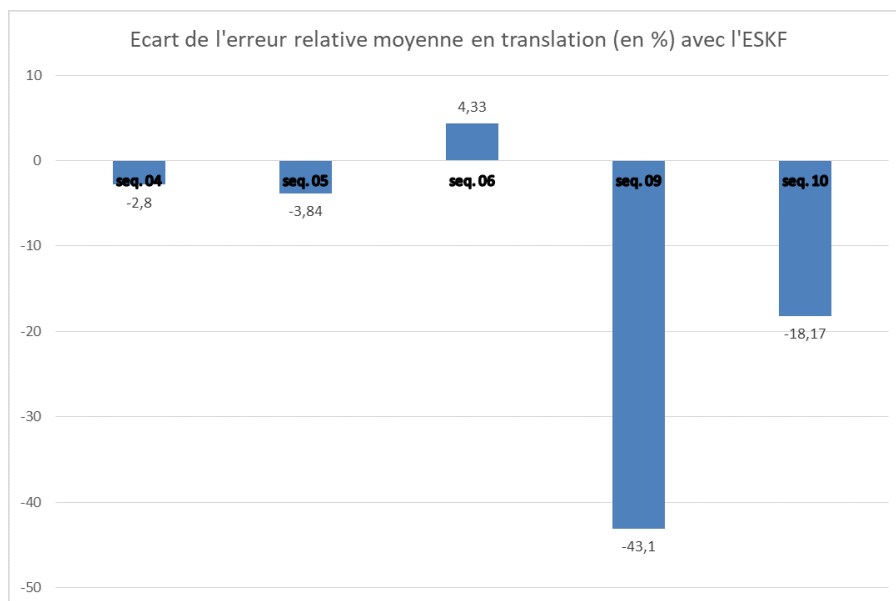


FIGURE 6.5 – Histogramme des écarts de l'erreur relative moyenne en translation sur les séquences 05, 06, 09 et 10 du KITTI vision benchmark pour l'ESKF entre l'évaluation perturbée et non perturbée.

méthodes proposées ne permet de corriger une navigation visuelle en faute.

Complexité combinatoire

Nous définissons ici la rapidité d'un algorithme comme la durée qu'il met pour estimer la trajectoire d'une séquence. Ici encore, il est difficile de faire une comparaison directe en raison de différences majeures entre les deux algorithmes proposés.

Pour l'ESKF, le Parallax Beam précède le filtre et son temps de calcul est inclus dans le temps de calcul de l'ESKF. Pour le MSF-EKF, le Parallax Beam a été pré-calculé et son temps de calcul n'est pas inclus dans celui du MSF-EKF qui n'est pas encore conçu pour du temps réel. De plus, la version du Parallax Beam utilisée n'est pas une version en temps réel et tous les algorithmes s'exécutent sur un seul thread. L'impact de la rapidité d'exécution du Parallax Beam est donc non négligeable sur les résultats présentés ici. Afin de proposer une comparaison pertinente, le temps de précalcul du Parallax Beam est également évalué.

La rapidité des algorithmes a été mesurée sur Python 3 avec la librairie *time* sur un Dell Precision 5730 équipé de 31.1 GiB de RAM et d'un Intel i7-8850H@2.66 Hz x12. Le résultat de ces mesures est visible sur la figure 6.6. Sur cette figure, la partie violette donne l'écart en temps de calcul entre l'ESKF (qui inclut les calculs du Parallax Beam, en bleu) et le MSF-EKF (en rouge) additionné au temps de prétraitement des séquences par le Parallax Beam (en vert).

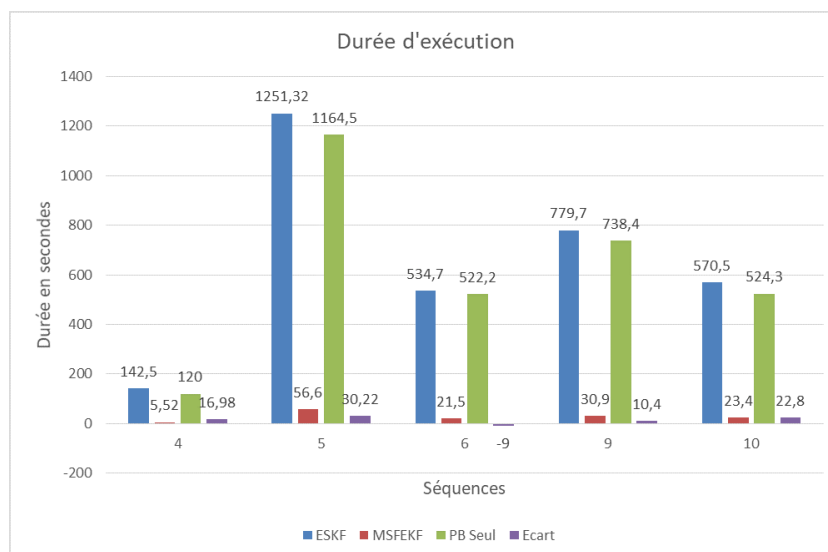


FIGURE 6.6 – Histogramme des temps de calcul requis pour les différentes séquences testées par l'ESKF, le MSF-EKF, le Parallax Beam et l'écart entre les méthodes ESKF et MSF-EKF+Parallax Beam.

On constate ainsi que pour quatre séquences sur cinq, le MSF-EKF + Parallax Beam est plus rapide que l'ESKF. L'écart n'est néanmoins pas si important comparé au temps de calcul du Parallax Beam, les deux méthodes présentées étant basées sur un filtre de Kalman Étendu.

Synthèse

	ESKF	MSF-EKF
Précision	Faible précision en translation à cause de la navigation inertielle de faible qualité (voir Chapitre 4).	Précision supérieure ou égale à celle du Parallax Beam.
Robustesse	Faible dégradation de la précision face à un sous-échantillonnage du Parallax Beam.	Dégradation très faible face à une coupure de 10 secondes consécutives du Parallax Beam.
Rapidité	Plus lent	Plus rapide

TABLE 6.1 – Tableau comparatif résumant l’étude relative entre l’ESKF et le MSF-EKF en précision, robustesse et rapidité d’exécution.

Que ce soit en précision, en robustesse avec une capacité à poursuivre la navigation en dépit d’une coupure d’une dizaine de secondes, ou en rapidité, le MSF-EKF affiche de meilleures performances (résumé dans la table 6.1) que l’ESKF et apparaît donc comme une solution technique adéquate à notre problématique de couplage visuel-inertiel.

6.3 Perspectives

Les solutions que nous avons proposées peuvent encore être améliorées. Tout d’abord, elles ont été testées sur KITTI et n’ont donc pas été éprouvées dans des environnements difficiles comme du tout-terrain, des forts changements de luminosité ou de l’encombrement visuel. De plus, les solutions proposées ne sont pas en mesure de détecter et corriger une valeur aberrante.

6.3.1 De nouveaux moyens d’essais

Afin de disposer de conditions d’essais plus difficiles pour nos algorithmes, nous souhaitons créer un jeu de données personnalisé conçu spécifiquement dans le but de mettre en échec les capteurs. À cet effet, nous avons créé notre propre plateforme de navigation, le Kamera und IMU Synchronisiertes System (KISS), visible figure 6.7.

Adapté à partir d’un véhicule radiocommandé à l’échelle 1 : 5 et piloté par une télécommande Futaba T18SZ, le véhicule est monté d’un bâti en Aluminium + PLA solidaire avec son châssis pour minimiser les vibrations parasites sur les IMUs. Le KISS comprend une caméra FIIR BlackFly USB3 23S6M équipé d’un objectif Edmund optics 5mm et monté sur un goniomètre Misumi à un axe. Il est également équipé d’une IMU Xsens Mti 630 9 DoF et d’une IMU Xsens Mti 710 9 DoF comprenant un module GNSS. Ce sont ces capteurs qui sont utilisés pour la navigation de la plateforme.

Les capteurs sont pilotés par une carte Zynq Te0715 équipée d’un system on chip Zynq-7000 comprenant un CPU 2 coeurs ARM Cortex A9 et d’un FPGA 28nm Xilinx.



FIGURE 6.7 – Le Kamera und IMU Synchronisiertes System (KISS).

L'ensemble est monté sur une carte de support TE0706 et dispose également d'un shield sur-mesure apportant un module Xbee pour s'interfacer avec PC de contrôle, ainsi qu'un module GNSS μ blox NEO M-8 qui servira à générer les signaux de déclenchement permettant une utilisation synchrone des capteurs.

L'ensemble des données est stockée sur un micro-ordinateur Odroid-XU4 qui pilote aussi la caméra selon les déclenchements de la carte Zynq.

Les capteurs sont synchronisés via un signal PPS à 5 Hz fourni par le GNSS qui sert à générer les signaux de déclenchement pour une fréquence minimale d'acquisition de 10 Hz et maximale de 30 Hz pour la caméra et 120 Hz pour les centrales inertielles. Il est possible de stocker jusqu'à 1.2 MB d'enregistrement vidéo et inertiel selon ce procédé. De plus, une modélisation du KISS sous Gazebo a été réalisée afin de disposer d'un moyen d'essai supplémentaire.

Le KISS a fait l'objet d'une publication aux 6^{ème} journées des Démonstrateurs en Automatique, à Angers [52] et a été utilisé dans d'autres travaux de recherche de l'ISL. De futurs travaux avec ce moyen d'expérimentation pourront être mis en oeuvre pour réaliser un jeu de données conçu pour mettre nos algorithmes à l'épreuve.

6.3.2 Diagnostiquer les erreurs en amont de la fusion

Une limitation de notre solution est son incapacité à corriger les dérives de l'odométrie visuelle. Ainsi, si l'odométrie visuelle est perturbée et diverge fortement (comme sur la figure B.7 par exemple), notre solution n'est pas en mesure de corriger ce phénomène. La solution la plus simple serait alors de détecter cette tendance quand elle survient et de réinitialiser la navigation visuelle sur la nouvelle prédiction du filtre, celui-ci ayant prouvé sa capacité à assurer une navigation correcte pendant une coupure de la navigation visuelle. Pour cela, plusieurs méthodes sont à explorer.

En partie déjà discuté dans la partie 3.4, des méthodes de détection de discontinuités par l'usage de la distance de Mahalanobis comme dans [7], [49], [86] ou encore [60] ou bien par un filtre médian comme dans [119] existent et peuvent être intégrées à notre architecture actuelle afin d'être en mesure d'identifier une estimation aberrante. Ces solutions, qui présentent pour seul inconvénient des calculs supplémentaires mais légers par la machine, auraient pour avantage de permettre la qualification des valeurs aberrantes et donc de procéder à la réinitialisation de la navigation à l'origine de l'aberration.

Une autre solution serait de s'intéresser aux méthodes de détection de contextes environnementaux comme celles présentées dans [41]. Ceci permettrait d'identifier les situations susceptibles de provoquer un écart de l'odométrie visuelle (caméra mal orientée ou obstruée), de couper préventivement le flux de la navigation visuelle vers la fusion et ainsi naviguer seulement à l'aide de l'IMU et du MSF-EKF comme dans le scénario de coupure en attendant un contexte plus adéquat pour une réinitialisation.

Enfin, une dernière méthode, qui consiste également en de la détection d'erreurs, serait d'utiliser un auto-encodeur afin de réaliser un diagnostic automatisé de fautes comme dans [124]. L'avantage principal de cette solution serait son automatisme à condition de disposer de la capacité de réaliser un jeu de données adéquat, chose possible grâce au KISS.

Quelque soit la méthode choisie, être en mesure de corriger la navigation visuelle lorsqu'elle est en erreur doit être la prochaine étape pour améliorer les travaux présentés ici.

6.4 Conclusion

Afin de répondre à la problématique du couplage entre une solution d'odométrie visuelle et une centrale inertielle, nous avons proposée deux architectures de couplage lâche visuel-inertiel : le "Error-State Kalman Filter" et le "Multi Sensor Fusion Extended Kalman Filter". Ces deux solutions répondent à la problématique posée et permettent d'assurer une navigation précise même en l'absence d'information de la part de la navigation visuelle. Cependant, le MSF-EKF offre de meilleures performances de précision,

robustesse et rapidité et se présente donc comme la solution à privilégier afin de répondre de manière efficace à notre problématique. Les travaux présentés ne sont qu'un premier pas vers la conception d'une solution de navigation plus performante grâce à des moyens d'essais adaptés aux contraintes auxquelles on souhaite soumettre nos mobiles et nos algorithmes et à l'implémentation de mesures additionnelles de robustesses afin de corriger des divergences d'estimations.

Bibliographie

- [1] United States. Defense Mapping Agency. *Department of Defense World Geodetic System 1984 : its definition and relationships with local geodetic systems*. **volume** 8350. Defense Mapping Agency, 1987.
- [2] Joelle Al Hage, Philippe Xu **and** Philippe Bonnifait. « Bounding localization errors with Student distribution for road vehicles ». **in***International Technical Symposium on Navigation and Timing*: 2018.
- [3] Joelle Al Hage, Philippe Xu **and** Philippe Bonnifait. « Student's t information filter with adaptive degree of freedom for multi-sensor fusion ». **in***2019 22th International Conference on Information Fusion (FUSION)*: IEEE. 2019, **pages** 1–8.
- [4] Adrien Angeli **and others**. « Real-time visual loop-closure detection ». **in***2008 IEEE International Conference on Robotics and Automation*: 2008, **pages** 1842–1847. DOI: [10.1109/ROBOT.2008.4543475](https://doi.org/10.1109/ROBOT.2008.4543475).
- [5] Antonio Angrisano **and others**. « GNSS/INS integration methods ». **in***Dottorato di ricerca (PhD) in Scienze Geodetiche e Topografiche Thesis, Universita'degli Studi di Napoli PARTHENOPE*, Naples: 21 (2010).
- [6] Michael Bloesch **and others**. « Fusion of optical flow and inertial measurements for robust egomotion estimation ». **in***2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*: IEEE. 2014, **pages** 3102–3107.
- [7] Michael Bloesch **and others**. « Robust visual inertial odometry using a direct EKF-based approach ». **in***2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*: IEEE. 2015, **pages** 298–304.
- [8] Silvere Bonnabel, Philippe Martin **and** Erwan Salaün. « Invariant extended Kalman filter : theory and application to a velocity-aided attitude estimation problem ». **in***Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*: IEEE. 2009, **pages** 1297–1304.
- [9] Guillaume Bresson **and others**. « Simultaneous localization and mapping : A survey of current trends in autonomous driving ». **in***IEEE Transactions on Intelligent Vehicles*: 2.3 (2017), **pages** 194–220.
- [10] Martin Brossard, Axel Barrau **and** Silvère Bonnabel. « Exploiting symmetries to design EKFs with consistency properties for navigation and SLAM ». **in***IEEE sensors journal*: 19.4 (2018), **pages** 1572–1579.

- [11] Martin Brossard, Axel Barrau **and** Silvère Bonnabel. « AI-IMU Dead-Reckoning ». **in***IEEE Transactions on Intelligent Vehicles: 5.4* (2020), **pages** 585–595. DOI: [10.1109/TIV.2020.2980758](https://doi.org/10.1109/TIV.2020.2980758).
- [12] Martin Brossard, Silvere Bonnabel **and** Axel Barrau. « Invariant kalman filtering for visual inertial slam ». **in***2018 21st International Conference on Information Fusion (FUSION)*: IEEE. 2018, **pages** 2021–2028.
- [13] Martin Brossard, Silvere Bonnabel **and** Axel Barrau. « Unscented Kalman filter on Lie groups for visual inertial odometry ». **in***2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE. 2018, **pages** 649–655.
- [14] Martin Brossard, Jean-Philippe Condomines **and** Silvère Bonnabel. « Tightly coupled navigation and wind estimation for mini UAVs ». **in***2018 AIAA Guidance, Navigation, and Control Conference*: 2018, **page** 1843.
- [15] Wolfram Burgard, Martial Hebert **and** Maren Bennewitz. « World modeling ». **in***Springer handbook of robotics*: Springer, 2016, **pages** 1135–1152.
- [16] Michael Burri **and** others. « The EuRoC micro aerial vehicle datasets ». **in***The International Journal of Robotics Research*: 35.10 (2016), **pages** 1157–1163.
- [17] Michael Calonder **and** others. « Brief : Binary robust independent elementary features ». **in***European conference on computer vision*: Springer. 2010, **pages** 778–792.
- [18] Ricardo Carelli **and** Eduardo Oliveira Freire. « Corridor navigation and wall-following stable control for sonar-based mobile robots ». **in***Robotics and Autonomous Systems*: 45.3-4 (2003), **pages** 235–247.
- [19] Lorenzo Pasqualetto Cassinis, Robert Fonod **and** Eberhard Gill. « Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft ». **in***Progress in Aerospace Sciences*: 110 (2019), **page** 100548.
- [20] Omer Cetin, Ibrahim Zagli **and** Guray Yilmaz. « Establishing obstacle and collision free communication relay for UAVs with artificial potential fields ». **in***Journal of Intelligent & Robotic Systems*: 69.1 (2013), **pages** 361–372.
- [21] Clément Chahbazian **and** others. « Improved Kalman-Particle Kernel Filter on Lie Groups Applied to Angles-Only UAV Navigation ». **in***2022 International Conference on Robotics and Automation (ICRA)*: 2022, **pages** 1689–1694. DOI: [10.1109/ICRA46639.2022.9812192](https://doi.org/10.1109/ICRA46639.2022.9812192).
- [22] Clément CHAHBAZIAN **and** others. « Laplace Particle Filter on Lie Groups Applied to Angles-Only Navigation ». **in***2021 IEEE 24th International Conference on Information Fusion (FUSION)*: 2021, **pages** 1–8. DOI: [10.23919/FUSION49465.2021.9626926](https://doi.org/10.23919/FUSION49465.2021.9626926).

- [23] Zhimin Chen **and others**. « Study on GPS/INS loose and tight coupling ». **in** *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*: **volume** 1. IEEE. 2015, **pages** 34–37.
- [24] Chih-Chung Chou **and** Cheng-Fu Chou. « Efficient and Accurate Tightly-Coupled Visual-Lidar SLAM ». **in** *IEEE Transactions on Intelligent Transportation Systems*: (2021).
- [25] Tianxing Chu **and others**. « Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments ». **in** *Sensors*: 12.3 (2012), **pages** 3162–3185.
- [26] Renato J Cintra, Fábio M Bayer **and** CJ Tablada. « Low-complexity 8-point DCT approximations based on integer functions ». **in** *Signal Processing*: 99 (2014), **pages** 201–214.
- [27] Ronald Clark **and others**. « Vinet : Visual-inertial odometry as a sequence-to-sequence learning problem ». **in** *Proceedings of the AAAI Conference on Artificial Intelligence*: **volume** 31. 1. 2017.
- [28] Jean-Philippe Condomines. *Filtrage de Kalman non linéaire pour la navigation multiscapteurs d'un mini drone : Application au pilotage-guidage robuste en milieu complexe*. ISTE Group, 2018.
- [29] Guillaume Courtier **and others**. « Image-based navigation system using skylight polarization for an unmanned ground vehicle ». **in** *Unmanned Systems Technology XXIV*: **volume** 12124. SPIE. 2022, **pages** 173–182.
- [30] Igor Cvišić, Ivan Marković **and** Ivan Petrović. « SOFT2 : Stereo Visual Odometry for Road Vehicles Based on a Point-to-Epipolar-Line Metric ». **in** *IEEE Transactions on Robotics*: (2022).
- [31] F Dellaert **and** M Kaess. *Georgia Tech Smoothing And Mapping (GTSAM)*. 2019.
- [32] Jeffrey Delmerico **and** Davide Scaramuzza. « A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots ». **in** *2018 IEEE international conference on robotics and automation (ICRA)*: IEEE. 2018, **pages** 2502–2509.
- [33] Jean-Emmanuel Deschaud. « IMLS-SLAM : Scan-to-model matching based on 3D data ». **in** *2018 IEEE International Conference on Robotics and Automation (ICRA)*: IEEE. 2018, **pages** 2480–2485.
- [34] Edsger Wybe Dijkstra. « A note on two problems in connexion with graphs :(Numerische Mathematik, 1 (1959), p 269-271) ». **in** (1959).
- [35] Hoang Viet Do **and others**. « An Adaptive Approach based on Multi-State Constraint Kalman Filter for UAVs ». **in** *2021 21st International Conference on Control, Automation and Systems (ICCAS)*: IEEE. 2021, **pages** 481–485.
- [36] Gregory Dudek **and** Michael Jenkin. « Inertial sensing, GPS and odometry ». **in** *Springer Handbook of Robotics*: Springer, 2016, **pages** 737–752.

- [37] Russell Eberhart **and** James Kennedy. « A new optimizer using particle swarm theory ». *inMHS'95. Proceedings of the sixth international symposium on micro machine and human science*: Ieee. 1995, **pages** 39–43.
- [38] Jan-Friedrich Ehlenbröcker, Uwe Mönks **and** Volker Lohweg. « Sensor defect detection in multisensor information fusion ». *inJournal of Sensors and Sensor Systems*: 5.2 (2016), **pages** 337–353.
- [39] Matthias Faessler **and**others. « Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle ». *inJournal of Field Robotics*: 33.4 (2016), **pages** 431–450.
- [40] Wei Fang **and**others. « A LSTM algorithm estimating pseudo measurements for aiding INS during GNSS signal outages ». *inRemote sensing*: 12.2 (2020), **page** 256.
- [41] Florent Feriol, Damien Vivet **and** Yoko Watanabe. « A review of environmental context detection for navigation based on multiple sensors ». *inSensors*: 20.16 (2020), **page** 4532.
- [42] Martin A Fischler **and** Robert C Bolles. « Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography ». *inCommunications of the ACM*: 24.6 (1981), **pages** 381–395.
- [43] Robert B Fisher **and** Kurt Konolige. « Range sensors ». *inSpringer Handbook of Robotics*: Springer, 2008, **pages** 521–542.
- [44] Robert Fonod. « Model-based Fault Diagnosis and Fault Accommodation for Space Missions : Application to the Rendezvous Phase of the MSR Mission ». phdthesis. Bordeaux, 2014.
- [45] Christian Forster **and**others. « On-manifold preintegration for real-time visual-inertial odometry ». *inIEEE Transactions on Robotics*: 33.1 (2016), **pages** 1–21.
- [46] Christian Forster **and**others. « SVO : Semidirect visual odometry for monocular and multicamera systems ». *inIEEE Transactions on Robotics*: 33.2 (2016), **pages** 249–265.
- [47] Eric Foxlin. « Pedestrian tracking with shoe-mounted inertial sensors ». *inIEEE Computer graphics and applications*: 25.6 (2005), **pages** 38–46.
- [48] Chevrin Gaëtan **and**others. « Visual-Inertial Fusion on KITTI using MSF-EKF ». *in2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*: IEEE. 2022, **pages** 35–40.
- [49] Bo Gao **and**others. « LOVINS : Lightweight omnidirectional visual-inertial navigation system ». *in2021 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*: IEEE. 2021, **pages** 1–6.
- [50] Grace Xingxin Gao **and**others. « Protecting GNSS receivers from jamming and interference ». *inProceedings of the IEEE*: 104.6 (2016), **pages** 1327–1338.
- [51] Chevrin Gaëtan **and**others. « Couplage Visuel Inertiel pour une navigation précise et robuste ». *in28ième Colloque sur le traitement du signal et des images*:

- 001-0175. Nancy: GRETSI - Groupe de Recherche en Traitement du Signal et des Images, 2022, p. 701–704.
- [52] Chevrin Gaëtan **and others**. « La plateforme KISS (Kamera und IMU Synchronisiertes Systems) ». **in** Angers: 6^{ème} Journées des Démonstrateurs en Automatique, 2022.
- [53] Andreas Geiger, Julius Ziegler **and** Christoph Stiller. « StereoScan : Dense 3D Reconstruction in Real-time ». **in** *Intelligent Vehicles Symposium (IV)*: 2011.
- [54] Andreas Geiger **and others**. « Vision meets robotics : The kitti dataset ». **in** *The International Journal of Robotics Research*: 32.11 (2013), **pages** 1231–1237.
- [55] Laetitia Gond, David Monnin **and** Armin Schneider. « Optimized feature-detection for on-board vision-based surveillance ». **in** *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*: **volume** 8357. SPIE. 2012, **pages** 519–530.
- [56] Paul D Groves. « Principles of GNSS, inertial, and multisensor integrated navigation systems, [Book review] ». **in** *IEEE Aerospace and Electronic Systems Magazine*: 30.2 (2015), **pages** 26–27.
- [57] Chris Harris, Mike Stephens **and others**. « A combined corner and edge detector ». **in** *Alvey vision conference*: **volume** 15. 50. Citeseer. 1988, **pages** 10–5244.
- [58] P Hart, N Nilsson **and** B Raphael. « Systems Science and Cybernetics ». **in** *IEEE Transactions on*: 4.2 (1968), **page** 100.
- [59] Ross Hartley **and others**. « Contact-aided invariant extended Kalman filtering for legged robot state estimation ». **in** *arXiv preprint arXiv :1805.10410*: (2018).
- [60] Sejong Heo, Jaehyuck Cha **and** Chan Gook Park. « EKF-based visual inertial navigation using sliding window nonlinear optimization ». **in** *IEEE Transactions on Intelligent Transportation Systems*: 20.7 (2018), **pages** 2470–2479.
- [61] Sejong Heo **and** Chan Gook Park. « Consistent EKF-based visual-inertial odometry on matrix Lie group ». **in** *IEEE Sensors Journal*: 18.9 (2018), **pages** 3780–3788.
- [62] Anders Heyden **and** Marc Pollefeys. « Multiple view geometry ». **in** *Emerging topics in computer vision*: 3 (2005), **pages** 45–108.
- [63] Sepp Hochreiter **and** Jürgen Schmidhuber. « Long short-term memory ». **in** *Neural computation*: 9.8 (1997), **pages** 1735–1780.
- [64] Siavash Hosseinyalamdary. « Deep Kalman filter : Simultaneous multi-sensor integration and modelling; A GNSS/IMU case study ». **in** *Sensors*: 18.5 (2018), **page** 1316.
- [65] Eduardo Infante. « Development and assessment of loosely-coupled ins using smartphone sensors ». phdthesis. University of New Brunswick., 2016.
- [66] Nabil Jardak **and** Quentin Jault. « The Potential of LEO Satellite-Based Opportunistic Navigation for High Dynamic Applications ». **in** *Sensors*: 22.7 (2022), **page** 2541.

- [67] Guanwei Jia **and others**. « Visual-SLAM Classical framework and key Techniques : a review ». *in Sensors*: 22.12 (2022), **page** 4582.
- [68] Antonio Ramón Jiménez **and others**. « Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU ». *in 2010 7th workshop on positioning, navigation and communication*: IEEE. 2010, **pages** 135–143.
- [69] Kichun Jo, Keounyup Chu **and** Myoungcho Sunwoo. « Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning ». *in IEEE Transactions on Intelligent Transportation Systems*: 13.1 (2011), **pages** 329–343.
- [70] Dah-Jing Jwo **and** Cheng-Min Huang. « An adaptive fuzzy strong tracking Kalman filter for GPS/INS navigation ». *in IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society*: IEEE. 2007, **pages** 2266–2271.
- [71] Dah-Jing Jwo **and** Shih-Yao Lai. « Navigation integration using the fuzzy strong tracking unscented Kalman filter ». *in The Journal of Navigation*: 62.2 (2009), **pages** 303–322.
- [72] Dah-Jing Jwo **and** Chien-Hao Tseng. « Fuzzy adaptive interacting multiple model unscented Kalman filter for integrated navigation ». *in 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC)*: IEEE. 2009, **pages** 1684–1689.
- [73] Rudolf E Kalman. « On the general theory of control systems ». *in Proceedings First International Conference on Automatic Control, Moscow, USSR*: 1960, **pages** 481–492.
- [74] Oussama Khatib. « Real-time obstacle avoidance for manipulators and mobile robots ». *in Proceedings. 1985 IEEE International Conference on Robotics and Automation*: **volume** 2. IEEE. 1985, **pages** 500–505.
- [75] Simon Klenk **and others**. « TUM-VIE : The TUM Stereo Visual-Inertial Event Dataset ». *in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: 2021, **pages** 8601–8608. DOI: [10.1109/IROS51168.2021.9636728](https://doi.org/10.1109/IROS51168.2021.9636728).
- [76] Manon Kok, Jeroen D Hol **and** Thomas B Schön. « Using inertial sensors for position and orientation estimation ». *in arXiv preprint arXiv :1704.06053*: (2017).
- [77] Xianglong Kong **and others**. « Tightly-coupled stereo visual-inertial navigation using point and line features ». *in Sensors*: 15.6 (2015), **pages** 12816–12833.
- [78] Ioannis Kostavelis **and others**. « Visual odometry for autonomous robot navigation through efficient outlier rejection ». *in 2013 IEEE International Conference on Imaging Systems and Techniques (IST)*: IEEE. 2013, **pages** 45–50.
- [79] Hans-Peter Kriegel, Peer Kröger **and** Arthur Zimek. « Outlier detection techniques ». *in Tutorial at KDD*: 10 (2010), **pages** 1–76.
- [80] Marc Labarrère, Jean-Pierre Krief **and** Bernard Gimonet. *Le filtrage et ses applications*. Cepadues-Ed., 1988.

-
- [81] Hind Laghmara **and others**. « Obstacle avoidance, path planning and control for autonomous vehicles ». **in** *2019 IEEE Intelligent Vehicles Symposium (IV)*: IEEE. 2019, **pages** 529–534.
- [82] Anthony Lawrence. *Modern inertial technology : navigation, guidance, and control*. Springer Science & Business Media, 2001.
- [83] Ern J Lefferts, F Landis Markley **and** Malcolm D Shuster. « Kalman filtering for spacecraft attitude estimation ». **in** *Journal of Guidance, Control, and Dynamics*: 5.5 (1982), **pages** 417–429.
- [84] Stefan Leutenegger **and others**. « Keyframe-based visual-inertial slam using non-linear optimization ». **in** *Proceedings of Robotics Science and Systems (RSS) 2013*: (2013).
- [85] Hongdong Li **and** Richard Hartley. « Five-point motion estimation made easy ». **in** *18th International Conference on Pattern Recognition (ICPR'06)*: **volume** 1. IEEE. 2006, **pages** 630–633.
- [86] Mingyang Li **and** Anastasios I Mourikis. « High-precision, consistent EKF-based visual-inertial odometry ». **in** *The International Journal of Robotics Research*: 32.6 (2013), **pages** 690–711.
- [87] Mingyang Li **and** Anastasios I Mourikis. « Improving the accuracy of EKF-based visual-inertial odometry ». **in** *2012 IEEE International Conference on Robotics and Automation*: IEEE. 2012, **pages** 828–835.
- [88] Hua Liu **and** Wen Wu. « Interacting multiple model (IMM) fifth-degree spherical simplex-radial cubature Kalman filter for maneuvering target tracking ». **in** *Sensors*: 17.6 (2017), **page** 1374.
- [89] Jorge Lobo **and** Jorge Dias. « Relative pose calibration between visual and inertial sensors ». **in** *The International Journal of Robotics Research*: 26.6 (2007), **pages** 561–575.
- [90] D Loebis **and others**. « Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system ». **in** *Control engineering practice*: 12.12 (2004), **pages** 1531–1539.
- [91] David G Lowe. « Object recognition from local scale-invariant features ». **in** *Proceedings of the seventh IEEE international conference on computer vision*: **volume** 2. Ieee. 1999, **pages** 1150–1157.
- [92] Simon Lynen **and others**. « A robust and modular multi-sensor fusion approach applied to mav navigation ». **in** *2013 IEEE/RSJ international conference on intelligent robots and systems*: IEEE. 2013, **pages** 3923–3929.
- [93] Anas Mahmoud, Aboelmagd Noureldin **and** Hossam S Hassanein. « Integrated positioning for connected vehicles ». **in** *IEEE Transactions on Intelligent Transportation Systems*: 21.1 (2019), **pages** 397–409.

- [94] David Monnin **and others**. « An effective rigidity constraint for improving RAN-SAC in homography estimation ». **in***International Conference on Advanced Concepts for Intelligent Vision Systems*: Springer. 2010, **pages** 203–214.
- [95] Anastasios I Mourikis, Stergios I Roumeliotis **and others**. « A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. » **in***ICRA: volume 2*. 2007, **page** 6.
- [96] Raul Mur-Artal, Jose Maria Martinez Montiel **and** Juan D Tardos. « ORB-SLAM : a versatile and accurate monocular SLAM system ». **in***IEEE transactions on robotics*: 31.5 (2015), **pages** 1147–1163.
- [97] Dinh Van Nam **and** Kim Gon-Woo. « Robust stereo visual inertial navigation system based on multi-stage outlier removal in dynamic environments ». **in***Sensors*: 20.10 (2020), **page** 2922.
- [98] David Nistér. « An efficient solution to the five-point relative pose problem ». **in***IEEE transactions on pattern analysis and machine intelligence*: 26.6 (2004), **pages** 756–770.
- [99] BK Patle **and others**. « A review : On path planning strategies for navigation of mobile robot ». **in***Defence Technology*: 15.4 (2019), **pages** 582–606.
- [100] Tong Qin, Peiliang Li **and** Shaojie Shen. « Vins-mono : A robust and versatile monocular visual-inertial state estimator ». **in***IEEE Transactions on Robotics*: 34.4 (2018), **pages** 1004–1020.
- [101] Alex G Quinchia **and others**. « A comparison between different error modeling of MEMS applied to GPS/INS integrated systems ». **in***Sensors*: 13.8 (2013), **pages** 9549–9588.
- [102] Martin Rebert. « Two views camera motion in the presence of planar degeneracy ». phdthesis. Université de Haute Alsace-Mulhouse, 2020.
- [103] Martin Rebert **and others**. « Parallax beam : a vision-based motion estimation method robust to nearly planar scenes ». **in***Journal of Electronic Imaging*: 28.2 (2019), **page** 023030.
- [104] Antoni Rosinol **and others**. « Kimera : an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping ». **in***2020 IEEE International Conference on Robotics and Automation (ICRA)*: 2020, **pages** 1689–1696. DOI: [10.1109/ICRA40945.2020.9196885](https://doi.org/10.1109/ICRA40945.2020.9196885).
- [105] Stergios I Roumeliotis, Andrew E Johnson **and** James F Montgomery. « Augmenting inertial navigation with image-based motion estimation ». **in***Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*: **volume** 4. IEEE. 2002, **pages** 4326–4333.
- [106] Alicia Roux **and others**. « CNN-based Invariant Extended Kalman Filter for projectile trajectory estimation using IMU only ». **in***2021 International Conference on Control, Automation and Diagnosis (ICCAD)*: 2021, **pages** 1–6. DOI: [10.1109/ICCAD52417.2021.9638770](https://doi.org/10.1109/ICCAD52417.2021.9638770).

- [107] Andrzej Ruta **and others**. « In-vehicle camera traffic sign detection and recognition ». **in** *Machine Vision and Applications: 22.2* (2011), **pages** 359–375.
- [108] Angel Santamaria-Navarro, Joan Sola **and** Juan Andrade-Cetto. « High-frequency MAV state estimation using low-cost inertial and optical flow measurement units ». **in** *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE. 2015, **pages** 1864–1871.
- [109] Angel Santamaria-Navarro **and others**. « Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors ». **in** *Autonomous robots: 42.6* (2018), **pages** 1263–1280.
- [110] Adrian Schumacher. « Integration of a gps aided strapdown inertial navigation system for land vehicles ». **in** *Master of Science Thesis, KTH Electrical Engineering*: (2006).
- [111] Shaojie Shen **and others**. « Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV ». **in** *2014 IEEE International Conference on Robotics and Automation (ICRA)*: IEEE. 2014, **pages** 4974–4981.
- [112] Eun-Hwan Shin **and** Naser El-Sheimy. « Accuracy improvement of low cost INS/GPS for land applications ». **in** *Proceedings of the 2002 national technical meeting of the institute of navigation*: 2002, **pages** 146–157.
- [113] Joan Sola. « Quaternion kinematics for the error-state Kalman filter ». **in** *arXiv preprint arXiv :1711.02508*: (2017).
- [114] Arno Solin **and others**. « Inertial odometry on handheld smartphones ». **in** *2018 21st International Conference on Information Fusion (FUSION)*: IEEE. 2018, **pages** 1–5.
- [115] Jürgen Sturm **and others**. « A benchmark for the evaluation of RGB-D SLAM systems ». **in** *2012 IEEE/RSJ international conference on intelligent robots and systems*: IEEE. 2012, **pages** 573–580.
- [116] Rahul P Suresh **and others**. « Zero velocity potential update (ZUPT) as a correction technique ». **in** *2018 3rd International Conference On Internet of Things : Smart Innovation and Usages (IoT-SIU)*: IEEE. 2018, **pages** 1–8.
- [117] Qianqian Tong **and others**. « Cascade-LSTM-based visual-inertial navigation for magnetic levitation haptic interaction ». **in** *IEEE Network: 33.3* (2019), **pages** 74–80.
- [118] Nikolas Trawny **and** Stergios I Roumeliotis. « Indirect Kalman filter for 3D attitude estimation ». **in** *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep: 2* (2005), **page** 2005.
- [119] Stephan Weiss **and** Roland Siegwart. « Real-time metric state estimation for modular vision-inertial systems ». **in** *2011 IEEE international conference on robotics and automation*: IEEE. 2011, **pages** 4531–4537.

- [120] Stephan Weiss **and others**. « Monocular vision for long-term micro aerial vehicle state estimation : A compendium ». **in***Journal of Field Robotics*: 30.5 (2013), **pages** 803–831.
- [121] Stephan M Weiss. « Vision based navigation for micro helicopters ». phdthesis. ETH Zurich, 2012.
- [122] JL Weston **and** DH Titterton. « Strapdown inertial navigation technology ». **in***Institution of Electrical Engineers and American Institute of Aeronautics and Astronautics*: (2004).
- [123] Hailiang Xiong, Zhenzhen Mai **and** Juan Tang. « Robust Navigation and Positioning Algorithm and Performance Evaluation in Harsh Scenarios ». **in***2019 IEEE International Conference on Communications Workshops (ICC Workshops)*: IEEE. 2019, **pages** 1–6.
- [124] Zheng Yang **and others**. « Autoencoder-based representation learning and its application in intelligent fault diagnosis : A review ». **in***Measurement*: 189 (2022), **page** 110460. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2021.110460>. URL: <https://www.sciencedirect.com/science/article/pii/S0263224121013464>.
- [125] Chih-Ta Yen **and** Ming-Feng Cheng. « A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance ». **in***Microsystem Technologies*: 24.1 (2018), **pages** 125–135.
- [126] Wonkeun Youn **and** Hyun Myung. « Robust interacting multiple model with modeling uncertainties for maneuvering target tracking ». **in***IEEE Access*: 7 (2019), **pages** 65427–65443.
- [127] Lotfi A Zadeh. « Fuzzy sets ». **in***Information and control*: 8.3 (1965), **pages** 338–353.
- [128] Yuexin Zhang. « A fusion methodology to bridge GPS outages for INS/GPS integrated navigation system ». **in***IEEE access*: 7 (2019), **pages** 61296–61306.
- [129] DH Zhou **and** PM Frank. « Strong tracking filtering of nonlinear time-varying stochastic systems with coloured noise : Application to parameter estimation and empirical robustness analysis ». **in***International Journal of Control*: 65.2 (1996), **pages** 295–307.

A Les filtres de Kalman

Les filtres de Kalman formant une famille d'outils qui va être fréquemment citée, cette partie en détaille le fonctionnement à partir des ressources disponibles dans [80] et [28]. Le filtre de Kalman-Bucy est une solution couramment utilisée pour résoudre le problème du filtrage linéaire et non-linéaire dans une multitude de cas d'applications, de la sismologie à l'aérospatial. Bien adapté au traitement numérique en ligne, il fournit une estimée optimale ainsi que la variance de l'erreur d'estimation.

L'estimation d'un état $\mathbf{x}(t)$ d'un système dynamique soumis à des entrées déterministes et aléatoires à partir de mesures $\mathbf{z}(\tau)$ bruitées peut être divisée en trois classes distinctes en fonction de l'intervalle d'observation $[t_0; t_1]$:

- la prédiction si $t > t_1$
- le filtrage si $t = t_1$
- le lissage si $t_0 < t < t_1$

Le filtre de Kalman-Bucy permet de résoudre les problèmes de prédiction et de filtrage et sert de base à la théorie du lissage. Le filtre classique s'applique aux systèmes dynamiques linéaires, continus ou discrets dont le bruit de mesure est blanc.

A.1 Filtre de Kalman Continu

L'évolution de l'état d'un système continu est décrite par l'équation différentielle suivante :

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t) \cdot \mathbf{x}(t) + \mathbf{u}(t) + \mathbf{v}(t) \quad (\text{A.1})$$

avec \mathbf{x} le vecteur d'état de dimension n , $\mathbf{F}(t)$ une matrice fonction de t de dimension $n \times n$ appelée matrice d'évolution, \mathbf{u} le vecteur d'entrée fonction de t et connu et \mathbf{v} est un bruit blanc gaussien à N dimensions de moyenne nulle. $E[\mathbf{v}(t)] = \mathbf{0}, \forall t$, de covariance :

$$E[\mathbf{v}(t) \cdot \mathbf{v}^T(\tau)] = \mathbf{Q}(t)\delta(t - \tau) \quad (\text{A.2})$$

avec $\mathbf{Q}(t)$ une matrice définie positive et $\delta(t)$ la fonction de Dirac en t .

L'état initial est lui-même aléatoire, de statistique connue, gaussien, de moyenne $E[\mathbf{x}(t_0)] = \mathbf{m}_0$ et de covariance $E[(\mathbf{x}(t_0) - \mathbf{m}_0)(\mathbf{x}(t_0) - \mathbf{m}_0)^T] = \Sigma_0$. Il est indépendant du bruit \mathbf{v} .

L'état de ce système est observé par m mesures $\mathbf{z}(t)$ liées à l'état $\mathbf{x}(t)$ par l'équation

d'observation ou de mesure :

$$\mathbf{z}(t) = \mathbf{H}(t) \cdot \mathbf{x}(t) + \mathbf{w}(t) \quad (\text{A.3})$$

Avec $\mathbf{H}(t)$ une matrice fonction de t de dimension $m \times n$ et w un bruit blanc gaussien à m dimensions, indépendant de $\mathbf{v}(t)$ et de $\mathbf{x}(t_0)$, de moyenne nulle et de covariance :

$$E[\mathbf{w}(t) \cdot \mathbf{w}^T(\tau)] = \mathbf{R}(t)\delta(t - \tau) \quad (\text{A.4})$$

avec $\mathbf{R}(t)$ une matrice définie positive.

Les équations du filtre de Kalman continu peuvent être obtenues simplement en recherchant un filtre linéaire non biaisé à variance minimale dont la formulation est la suivante :

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(t) \cdot \hat{\mathbf{x}}(t) + \mathbf{G}(t) \cdot \mathbf{u}(t) + \mathbf{K}(t) \cdot \mathbf{z}(t) \quad (\text{A.5})$$

Comme l'estimée est non biaisée, on peut dire que :

$$E[\dot{\hat{\mathbf{x}}}(t)] = E[\dot{\mathbf{x}}(t)]E[\hat{\mathbf{x}}(t)] = E[\mathbf{x}(t)] \quad (\text{A.6})$$

Donc, en moyennant les expressions A.1 et A.5, on obtient :

$$\mathbf{A}(t) \cdot E[\mathbf{x}(t)] + \mathbf{G}(t) \cdot \mathbf{u}(t) + \mathbf{K}(t) \cdot \mathbf{H}(t) \cdot E[\mathbf{x}(t)] = \mathbf{F}(t) \cdot E[\mathbf{x}(t)] + \mathbf{u}(t) \quad (\text{A.7})$$

D'où les relations :

$$\mathbf{A}(t) = \mathbf{F}(t) - \mathbf{K}(t) \cdot \mathbf{H}(t) \quad (\text{A.8})$$

$$\mathbf{B}(t) = \mathbf{I} \quad (\text{A.9})$$

Avec \mathbf{I} la matrice identité.

On obtient alors l'expression de notre filtre :

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(t) \cdot \hat{\mathbf{x}}(t) + \mathbf{u}(t) + \mathbf{K}(t) \cdot [\mathbf{z}(t) - \mathbf{H}(t) \cdot \hat{\mathbf{x}}(t)] \quad (\text{A.10})$$

La matrice de gain $\mathbf{K}(t)$ doit minimiser la variance de l'erreur $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$. Cette erreur $\tilde{\mathbf{x}}$ vérifie l'équation différentielle obtenue à partir des relations précédentes :

$$\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{F}(t) - \mathbf{K}(t) \cdot \mathbf{H}(t)) \cdot \tilde{\mathbf{x}}(t) + \mathbf{v}(t) - \mathbf{K}(t) \cdot \mathbf{w}(t) \quad (\text{A.11})$$

$\tilde{\mathbf{x}}(t)$ est alors la sortie d'un système linéaire ayant pour entrée les bruits blancs $\mathbf{v}(t) - \mathbf{K}(t) \cdot \mathbf{w}(t)$ et qui a comme covariance $\mathbf{Q} + \mathbf{K} \cdot \mathbf{R} \cdot \mathbf{K}^T$.

La variance \mathbf{P} de l'erreur $\tilde{\mathbf{x}}$ est solution de l'équation différentielle :

$$\dot{\mathbf{P}}(t) = (\mathbf{F}(t) - \mathbf{K}(t) \cdot \mathbf{H}(t)) \cdot \mathbf{P}(t) + \mathbf{P}(t) \cdot (\mathbf{F}(t) - \mathbf{K}(t) \cdot \mathbf{H}(t))^T + \mathbf{Q} + \mathbf{K} \cdot \mathbf{R} \cdot \mathbf{K}^T \quad (\text{A.12})$$

$$\dot{\mathbf{P}}(t) = \mathbf{F} \cdot \mathbf{P} + \mathbf{P} \cdot \mathbf{F}^T - \mathbf{P} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \cdot \mathbf{P} + \mathbf{Q} + (\mathbf{K} - \mathbf{P} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1}) \cdot \mathbf{R} \cdot (\mathbf{K} - \mathbf{P} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1})^T \quad (\text{A.13})$$

Pour minimiser la variance de l'erreur, il suffit alors de minimiser $\dot{\mathbf{P}}$, c'est à dire choisir $\mathbf{K}(t)$ tel que :

$$\mathbf{K}(t) = \mathbf{P}(t) \cdot \mathbf{H}^T(t) \cdot \mathbf{R}^{-1}(t) \quad (\text{A.14})$$

On peut en déduire les équations du filtre de Kalman continu :

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(t) \cdot \hat{\mathbf{x}}(t) + \mathbf{u}(t) + \mathbf{K}(t) \cdot [\mathbf{z}(t) - \mathbf{H}(t) \cdot \hat{\mathbf{x}}(t)] \quad (\text{A.15})$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t) \cdot \mathbf{P}(t) + \mathbf{P}(t) \cdot \mathbf{F}^T(t) - \mathbf{P}(t) \cdot \mathbf{H}^T(t) \cdot \mathbf{R}^{-1}(t) \cdot \mathbf{H}(t) \cdot \mathbf{P}(t) + \mathbf{Q} \quad (\text{A.16})$$

$$\mathbf{K}(t) = \mathbf{P}(t) \cdot \mathbf{H}^T(t) \cdot \mathbf{R}^{-1}(t) \quad (\text{A.17})$$

A.2 Filtre de Kalman Discret

En raison de la numérisation importante des moyens de traitement de l'information, le filtre de Kalman-Bucy continu est rarement encore implémenté au profit de sa version discrète.

Dans le cas discret, on notera $\mathbf{x}_{k|l}$ la meilleure estimée de \mathbf{x}_k à partir des mesures $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_l$. $k > l$ indiquant une prédiction, $k = l$ une estimation et $k < l$ un lissage.

Les équations d'évolution et de mesure prennent alors la forme suivante :

$$\mathbf{x}_{k+1} = \mathbf{F}_k \cdot \mathbf{x}_k + \mathbf{u}_k + \mathbf{v}_k \quad (\text{A.18})$$

$$\mathbf{z}_k = \mathbf{H}_k \cdot \mathbf{x}_k + \mathbf{w}_k \quad (\text{A.19})$$

avec \mathbf{v}_k et \mathbf{w}_k des bruits pseudo-blancs gaussiens de moyenne nulles tels que, $\forall k, j$:

$$E[\mathbf{v}_k \cdot \mathbf{v}_j^T] = \mathbf{Q}_k \cdot \delta_{kj} \quad (\text{A.20})$$

$$E[\mathbf{w}_k \cdot \mathbf{w}_j^T] = \mathbf{R}_k \cdot \delta_{kj} \quad (\text{A.21})$$

$$E[\mathbf{v}_k \cdot \mathbf{w}_j^T] = 0 \quad (\text{A.22})$$

avec δ_{kj} le symbole de Kronecker :

$$\delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \quad (\text{A.23})$$

L'expression du filtre de Kalman peut s'obtenir simplement et de manière itérative en calculant la moyenne conditionnelle $\hat{\mathbf{x}}_{k+1|k+1} = E[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k]$. Comme l'état et les mesures sont gaussiennes, l'estimée est une fonction linéaire des mesures, indépendante de l'erreur d'estimation $\tilde{\mathbf{x}}_{k+1|k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1}$.

Soit $\hat{\mathbf{x}}_{k|k}$ le meilleur estimé de \mathbf{x}_k à partir des mesures $\mathbf{z}_1 \dots \mathbf{z}_k$ et $\mathbf{P}_{k|k}$ la variance de l'erreur

d'estimation à l'instant k .

$$\hat{\mathbf{x}}_{k|k} = E[\mathbf{x}_k | \mathbf{z}_1 \dots \mathbf{z}_k] \quad (\text{A.24})$$

$$\mathbf{P}_{k|k} = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) \cdot (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T | \mathbf{z}_1 \dots \mathbf{z}_k] \quad (\text{A.25})$$

À l'instant $k + 1$, l'estimée de \mathbf{x}_{k+1} à partir des observations $\mathbf{z}_1 \dots \mathbf{z}_k$ est la meilleure prédiction de \mathbf{x}_{k+1} , notée $\hat{\mathbf{x}}_{k+1|k}$:

$$\hat{\mathbf{x}}_{k+1|k} = E[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k] \quad (\text{A.26})$$

Elle s'obtient en prenant la moyenne conditionnelle de l'équation d'évolution (A.18) et en se rappelant que \mathbf{v}_k est un bruit blanc de moyenne nulle et indépendant des mesures :

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k \cdot \hat{\mathbf{x}}_{k|k} + \mathbf{u}_k \quad (\text{A.27})$$

La variance de l'erreur de prédiction est alors directement :

$$\mathbf{P}_{k+1|k} = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}) \cdot (\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})^T | \mathbf{z}_1 \dots \mathbf{z}_k] \quad (\text{A.28})$$

Que l'on peut réécrire comme :

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \cdot \mathbf{P}_{k|k} \cdot \mathbf{F}_k^T + \mathbf{Q}_k \quad (\text{A.29})$$

L'équation de mesure nous permet alors de déduire la mesure prédite ainsi que la variance de l'erreur de cette prédiction :

$$\hat{\mathbf{z}}_{k+1|k} = E[\mathbf{z}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k] = \mathbf{H}_{k+1} \cdot \hat{\mathbf{x}}_{k+1|k} \quad (\text{A.30})$$

$$E[(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) \cdot (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})^T | \mathbf{z}_1 \dots \mathbf{z}_k] = \mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1} \cdot \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1} \quad (\text{A.31})$$

Car le bruit \mathbf{w}_{k+1} est blanc, de moyenne nulle.

Lorsqu'on fait une nouvelle observation \mathbf{z}_{k+1} , la meilleure estimée est :

$$\hat{\mathbf{x}}_{k+1|k+1} = E[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_{k+1}] \quad (\text{A.32})$$

Il s'obtient à partir des variables gaussiennes \mathbf{x}_{k+1} et \mathbf{z}_{k+1} étant donné les mesures $\mathbf{z}_1 \dots \mathbf{z}_k$. En effet, via la loi de Bayes :

$$\mathbf{P}[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_{k+1}] = \mathbf{P}[(\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k) | (\mathbf{z}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k)] \quad (\text{A.33})$$

On peut définir la probabilité conditionnelle de deux variables gaussiennes de covariances Σ_{XX} , Σ_{XY} et Σ_{YY} comme suivantes :

$$E[X|Y] = E[X] + \Sigma_{XY} \cdot \Sigma_{YY}^{-1} \cdot (Y - E[Y]) \quad (\text{A.34})$$

$$\text{Var}[X|Y] = \Sigma_{XX} - \Sigma_{XY} \cdot \Sigma_{YY}^{-1} \cdot \Sigma_{YX} \quad (\text{A.35})$$

Alors :

$$\hat{\mathbf{x}}_{k+1|k+1} = E[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_{k+1}] = E[\mathbf{x}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k] + \mathbf{K}_{k+1} \cdot (\mathbf{z}_{k+1} - E[\mathbf{z}_{k+1} | \mathbf{z}_1 \dots \mathbf{z}_k]) \quad (\text{A.36})$$

On en déduit :

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \cdot (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) \quad (\text{A.37})$$

Avec :

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T \cdot (\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \quad (\text{A.38})$$

Et la variance de l'erreur d'estimation :

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1|k} \quad (\text{A.39})$$

On obtient ainsi l'ensemble des équations du filtre de Kalman discret. Tout d'abord, les équations de prédictions :

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k \cdot \hat{\mathbf{x}}_{k|k} + \mathbf{u}_k \quad (\text{A.40})$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \cdot \mathbf{P}_{k|k} \cdot \mathbf{F}_k^T + \mathbf{Q}_k \quad (\text{A.41})$$

Puis les équations d'estimation ou de mise à jour :

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T \cdot (\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1} \quad (\text{A.42})$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \cdot (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) \quad (\text{A.43})$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1|k} \quad (\text{A.44})$$

L'ensemble de ces équations permettent d'implémenter un filtre de Kalman discret.

A.3 Filtre de Kalman Etendu

Dans le cas non linéaire, le filtre de Kalman précédemment présenté n'est plus adapté au problème.

Notre équation d'évolution du modèle dans le cas non linéaire discret est en effet :

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \quad (\text{A.45})$$

Tandis que l'équation de mesure :

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \quad (\text{A.46})$$

Avec \mathbf{f} et \mathbf{h} des applications non linéaires.

Afin de définir un filtre de Kalman adapté, on effectue une linéarisation autour du point de fonctionnement $\hat{\mathbf{x}}_{k|k}$, celui-ci étant corrigé et remis à jour à chaque itération. Pour cela on utilise une décomposition en série de Taylor tronquée au premier ou second ordre.

Au premier ordre, notre erreur de prédiction :

$$\epsilon_{k+1|k} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) - \mathbf{f}_k(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \quad (\text{A.47})$$

Devient :

$$\epsilon_{k+1|k} = (\Delta \mathbf{f}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \cdot \epsilon_{k|k} \quad (\text{A.48})$$

Avec $(\Delta \mathbf{f}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}}$ la matrice jacobienne de \mathbf{f} calculée en $\mathbf{x} = \hat{\mathbf{x}}_{k|k}$. C'est-à-dire :

$$[(\Delta \mathbf{f}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}}]_{i,j} = \left. \frac{\delta \mathbf{f}_{k,i}}{\delta \mathbf{x}_j} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \quad (\text{A.49})$$

On peut ainsi exprimer les équations du filtre de Kalman Étendu. Les équations de prédiction sont :

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}, k) \quad (\text{A.50})$$

$$\mathbf{P}_{k+1|k} = (\Delta \mathbf{f}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \cdot \mathbf{P}_{k|k} \cdot (\Delta \mathbf{f}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}^T + \mathbf{Q}_{k-1} \quad (\text{A.51})$$

Et les équations de mise à jour sont :

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot (\Delta \mathbf{h}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}^T \cdot ((\Delta \mathbf{h}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \cdot \mathbf{P}_{k+1|k} \cdot (\Delta \mathbf{h}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}^T + \mathbf{R}_k)^{-1} \quad (\text{A.52})$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \cdot (\mathbf{z}_{k+1} - \mathbf{h}_k(\hat{\mathbf{x}}_{k+1|k})) \quad (\text{A.53})$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_k \cdot (\Delta \mathbf{h}_k)_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}) \cdot \mathbf{P}_{k+1|k} \quad (\text{A.54})$$

Ce filtre simple d'implémentation pour traiter les cas non linéaires reste néanmoins limité aux cas non linéaires faibles pour lesquels la validité du modèle linéaire tangent est garanti et les perturbations sont de variances limitées. De plus, le calcul des matrices jacobiennes peut induire des erreurs numériques qui s'ajoutent aux erreurs de modélisation et aux erreurs d'hypothèses.

Une solution à ces problèmes, au prix d'un coût en calcul plus élevé, est l'Unscented Kalman Filter (UKF) dont une explication approfondie peut-être trouvée dans [28].

B ESKF sur KITTI - figures supplémentaires

Dans cette partie sont présentés les essais réalisés sur les autres trajectoires du jeu de données KITTI vision qui n'ont pas été montrées dans le corps du chapitre 4 mais qui y ont contribué malgré tout. On retrouve donc ici les séquences estimées par l'ESKF avec Parallax Beam à 10 Hz, celles avec Parallax Beam à 5 Hz et enfin les séquences avec l'IEKF. Les synthèses des figures de cette partie ont été présentées dans les tables [4.1](#), [4.2](#), [4.3](#) et [4.4](#).

ESKF sur la séquence 04 avec Parallax Beam à 10 Hz

Sur la séquence 04 à 10 Hz, le filtre est plus long à converger et commet une erreur de suivi du Parallax Beam qui est due à une perturbation sur les données inertielles. Cette erreur est rapidement corrigée. La fusion proposée ne corrige pas le biais important sur l'axe vertical du Parallax Beam.

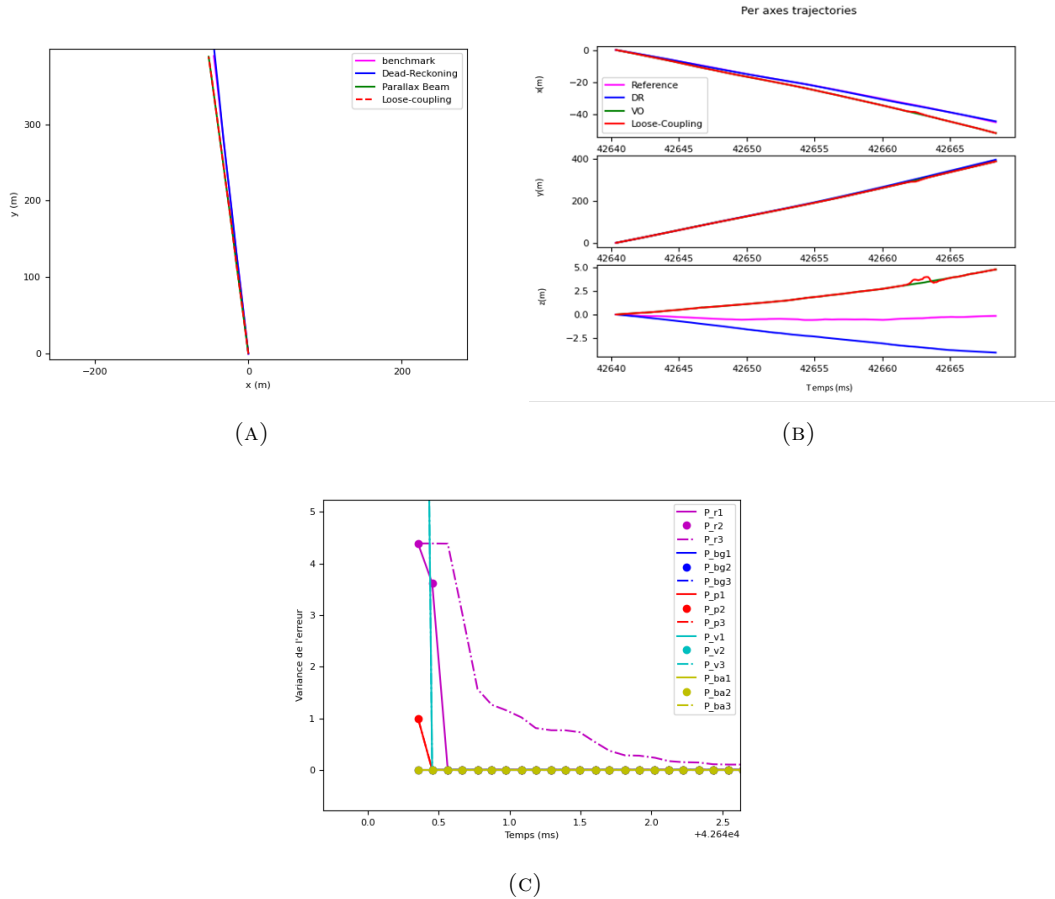


FIGURE B.1 – ESKF sur la séquence 04 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 06 avec Parallax Beam à 10 Hz

Sur la séquence 06 à 10 Hz, le filtre converge rapidement et parvient efficacement à suivre Parallax Beam. Néanmoins, la fusion proposée ne corrige pas Parallax Beam qui se met à dériver avec plus d'importance sur l'axe vertical à la fin de l'acquisition.

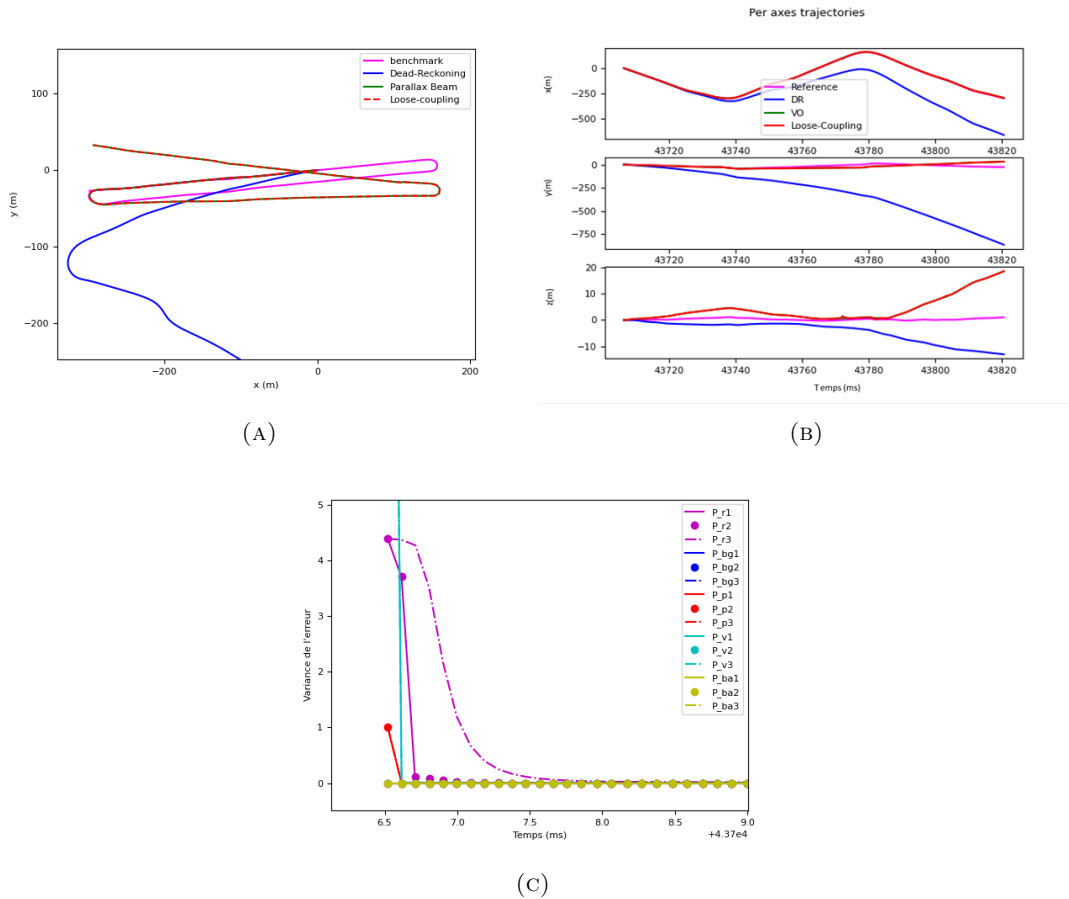


FIGURE B.2 – ESKF sur la séquence 06 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 09 avec Parallax Beam à 10 Hz

Sur la séquence 09 à 10 Hz, le filtre converge rapidement et parvient efficacement à suivre Parallax Beam. Néanmoins, la fusion proposée ne corrige pas Parallax Beam qui se met à dériver avec plus d'importance sur l'axe vertical à la fin de l'acquisition.

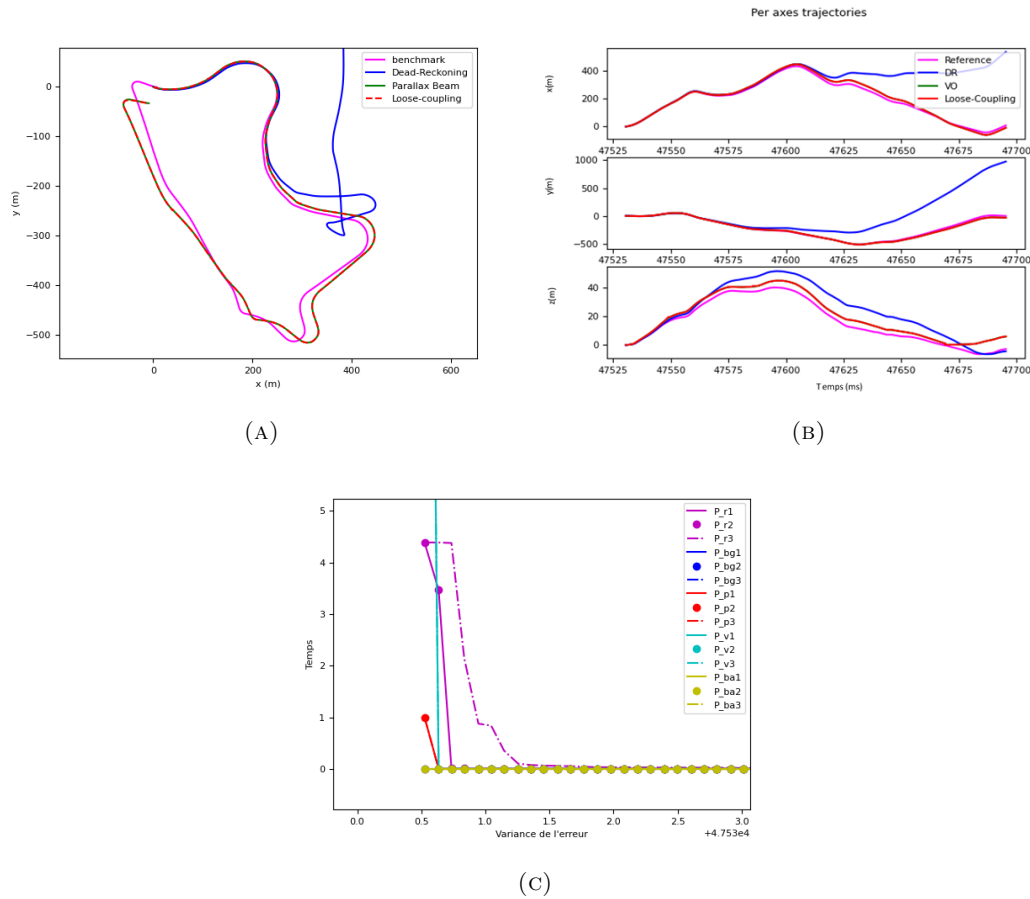


FIGURE B.3 – ESKF sur la séquence 9 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 10 avec Parallax Beam à 10 Hz

Sur la séquence 10 à 10 Hz, le filtre converge rapidement et parvient efficacement à suivre Parallax Beam. Ici, la dérive du Parallax Beam est faible mais toujours non corrigée.

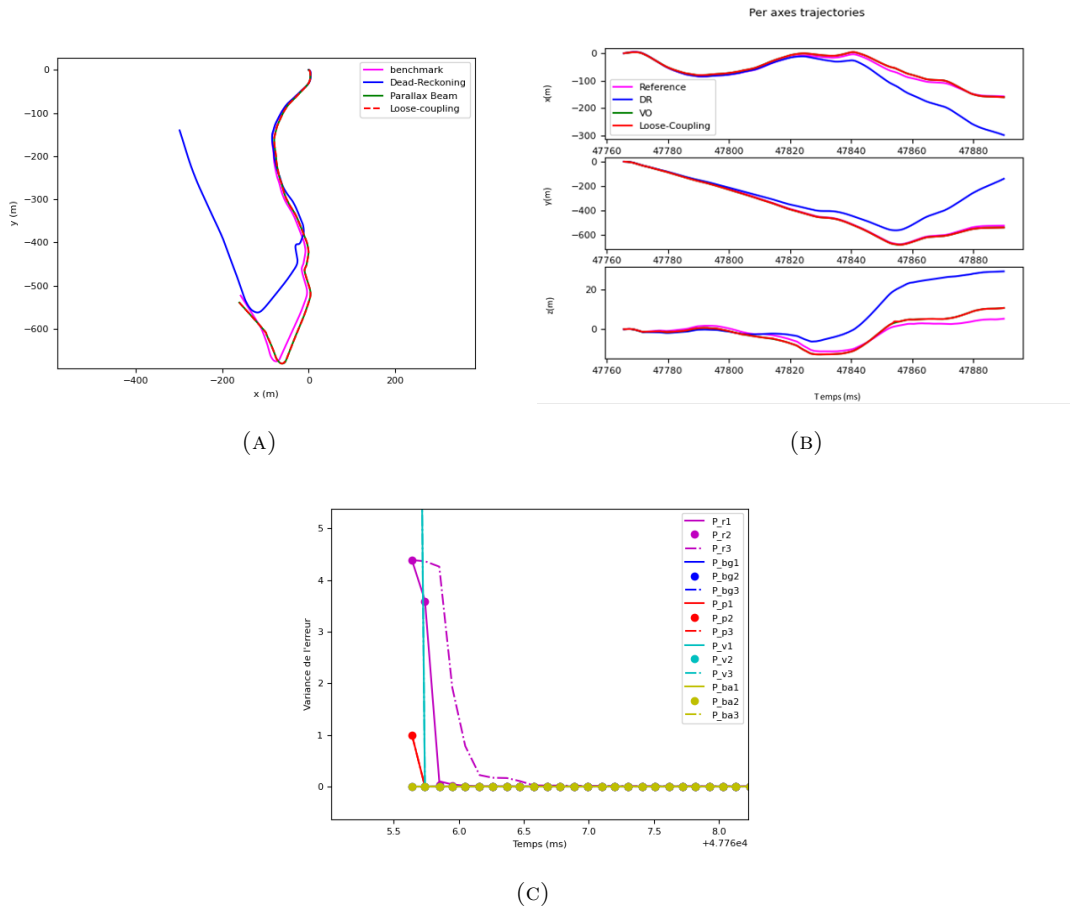


FIGURE B.4 – ESKF sur la séquence 10 avec Parallax Beam à 10 Hz. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 04 avec Parallax Beam à 5 Hz

Ici, Parallax Beam fonctionne avec un flux d'images à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz. Sur la séquence 04, le filtre a des difficultés à converger et devient instable à cause des perturbations sur les données inertielles en fin de trajectoire. La fusion proposée ne corrige pas le biais important sur l'axe vertical du Parallax Beam et propose un résultat dégradé sans le soutien constant du Parallax Beam qui n'est plus qu'à 5 Hz.

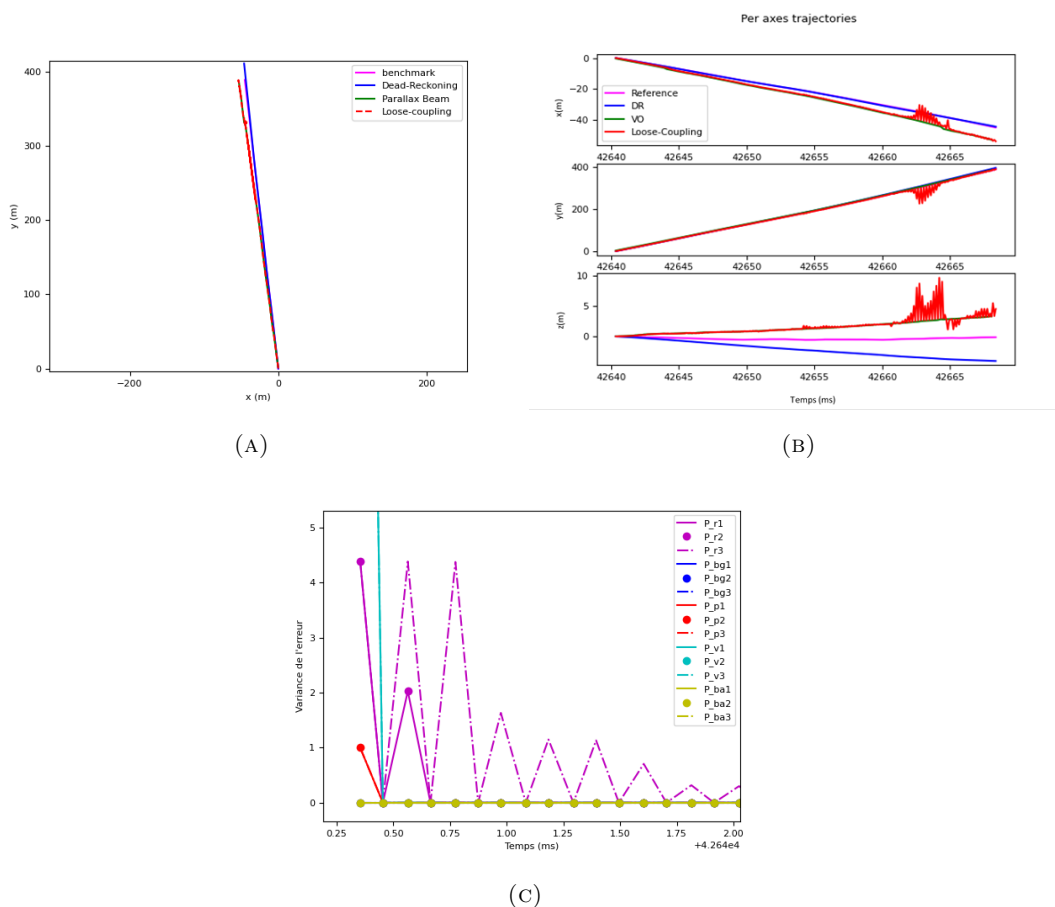


FIGURE B.5 – ESKF sur la séquence 04 avec Parallax Beam à 5 Hz. A) Trajectories estimées dans le plan du déplacement. B) Trajectories estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 06 avec Parallax Beam à 5 Hz

Ici, Parallax Beam fonctionne avec un flux d'images à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz. Sur la séquence 06, le filtre converge plus lentement qu'avec un Parallax Beam à 10 Hz. De plus, il devient beaucoup plus sensible aux valeurs aberrantes des données inertielles qui conduisent à de fortes instabilités qui parviennent à être corrigées en quelques itérations. Néanmoins, la fusion proposée ne corrige pas Parallax Beam qui se met à dériver avec plus d'importance sur l'axe vertical à la fin de l'acquisition.

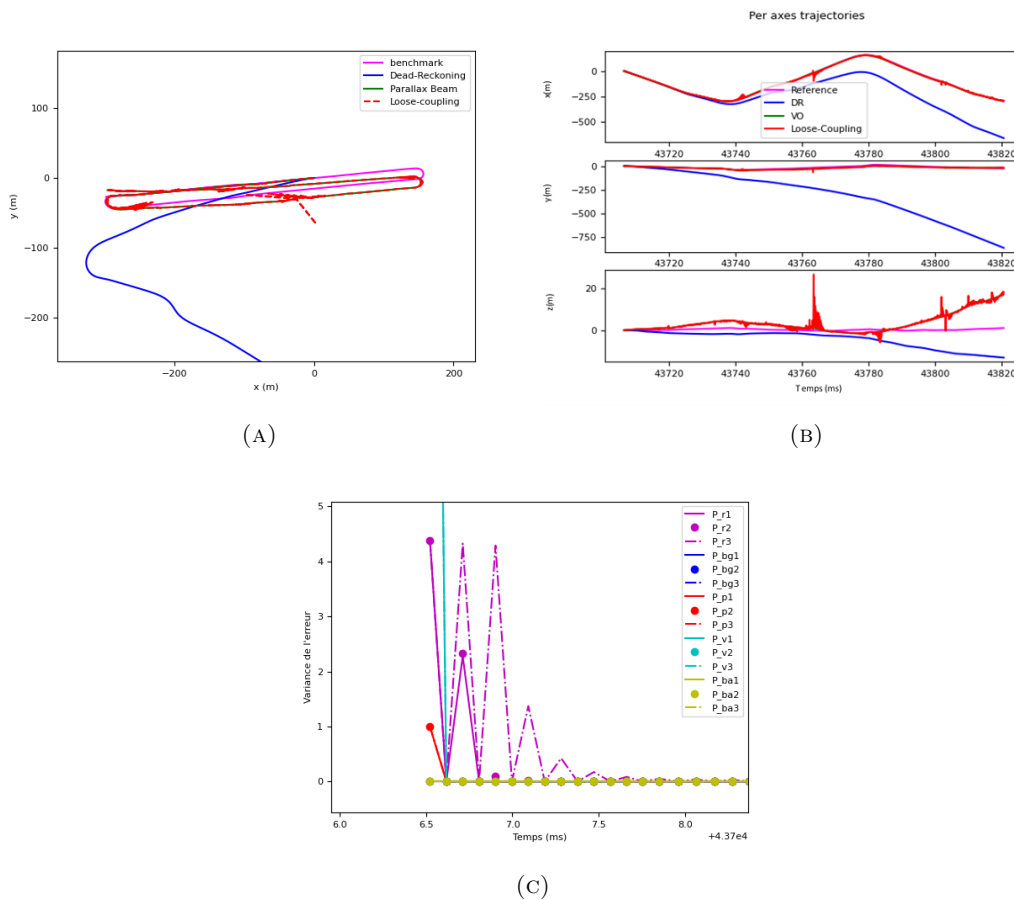


FIGURE B.6 – ESKF sur la séquence 06 avec Parallax Beam à 5 Hz. A) Trajectories estimées dans le plan du déplacement. B) Trajectories estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

ESKF sur la séquence 09 avec Parallax Beam à 5 Hz

Ici, Parallax Beam fonctionne avec un flux d'images à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz. Sur la séquence 09, Parallax Beam diverge totalement sans être corrigé car la coupure d'image survient au moment d'un virage. On constate un suivi correct de cette trajectoire fautive avec un filtre qui met plus de temps à converger qu'avec un Parallax Beam à 10 Hz et malgré quelques instabilités rapidement corrigées.

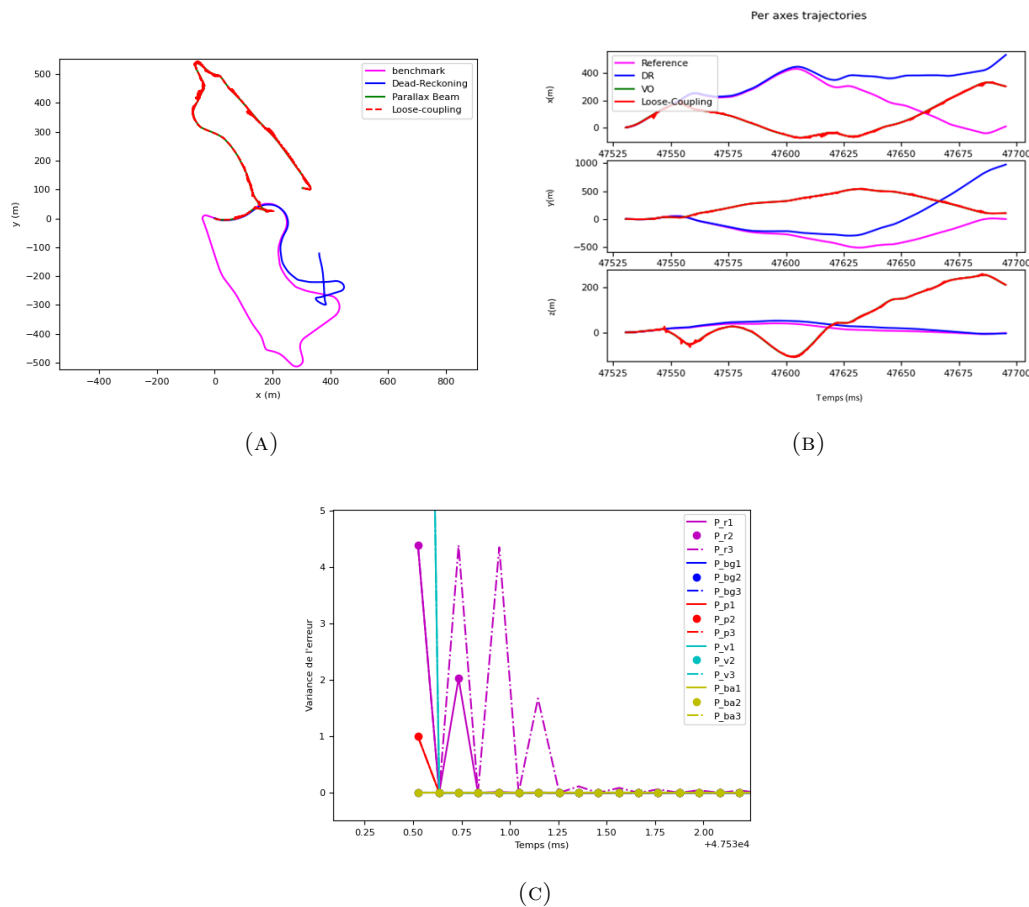


FIGURE B.7 – ESKF sur la séquence 9 avec Parallax Beam à 5 Hz. A) Trajectories estimated in the displacement plane. B) Trajectories estimated per axes. C) Diagonal terms of the error covariance matrix.

ESKF sur la séquence 10 avec Parallax Beam à 5 Hz

Ici, Parallax Beam fonctionne avec un flux d'images à 5 Hz tandis que la navigation inertielle est toujours à 10 Hz. Sur la séquence 10, le filtre converge plus lentement qu'à 10 Hz et parvient efficacement à suivre Parallax Beam malgré quelques nouvelles instabilités. La dérive du Parallax Beam est faible mais toujours non corrigée.

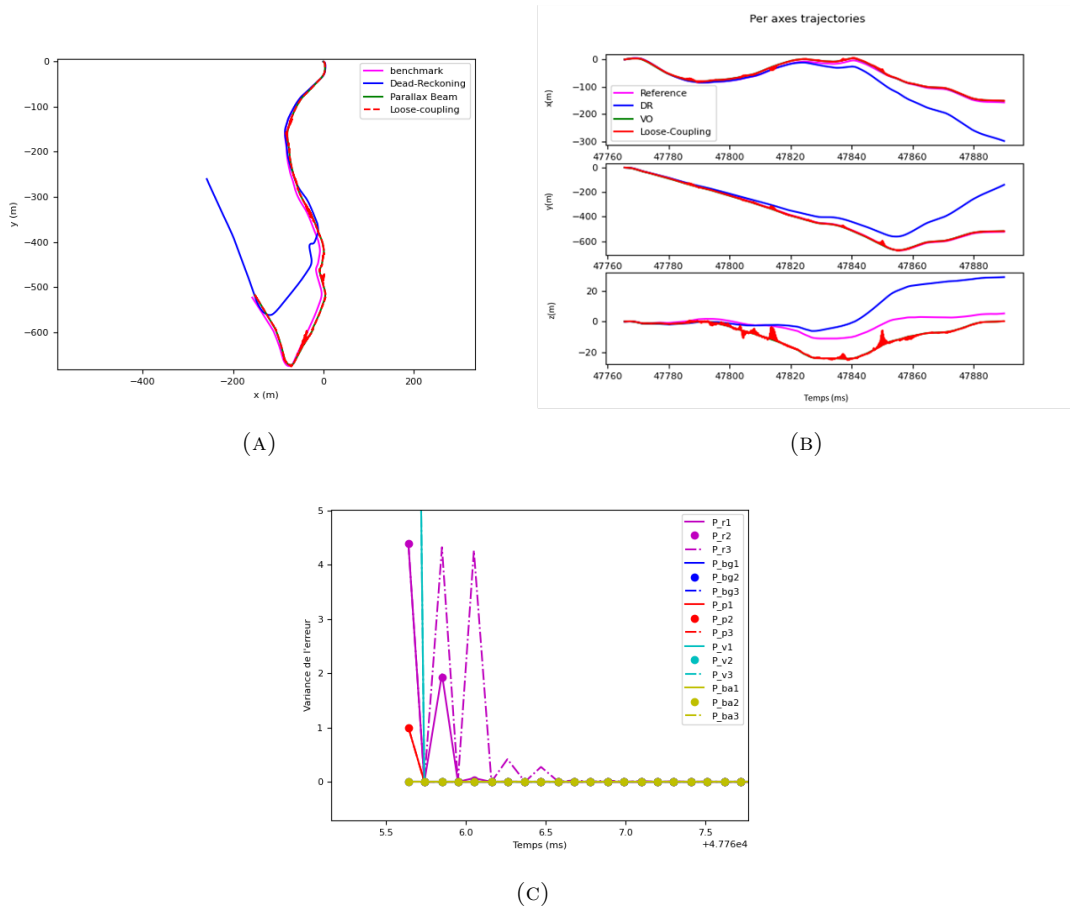


FIGURE B.8 – ESKF sur la séquence 10 avec Parallax Beam à 5 Hz. A) Trajectories estimées dans le plan du déplacement. B) Trajectories estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

IEKF sur le jeu de données KITTI vision

On voit les résultats de l'IEKF sur les séquences 04, 05, 06 et 09. En particulier, on peut constater sur la séquence 05 que l'IEKF (en noir) a une dérive supérieure à celle du Parallax Beam (en vert), sûrement à cause des données inertielles de faible qualité (dont la navigation à l'estime résultante est visible en bleu). Cependant, on peut aussi constater sa plus faible dérive sur les séquences 06 et 09.

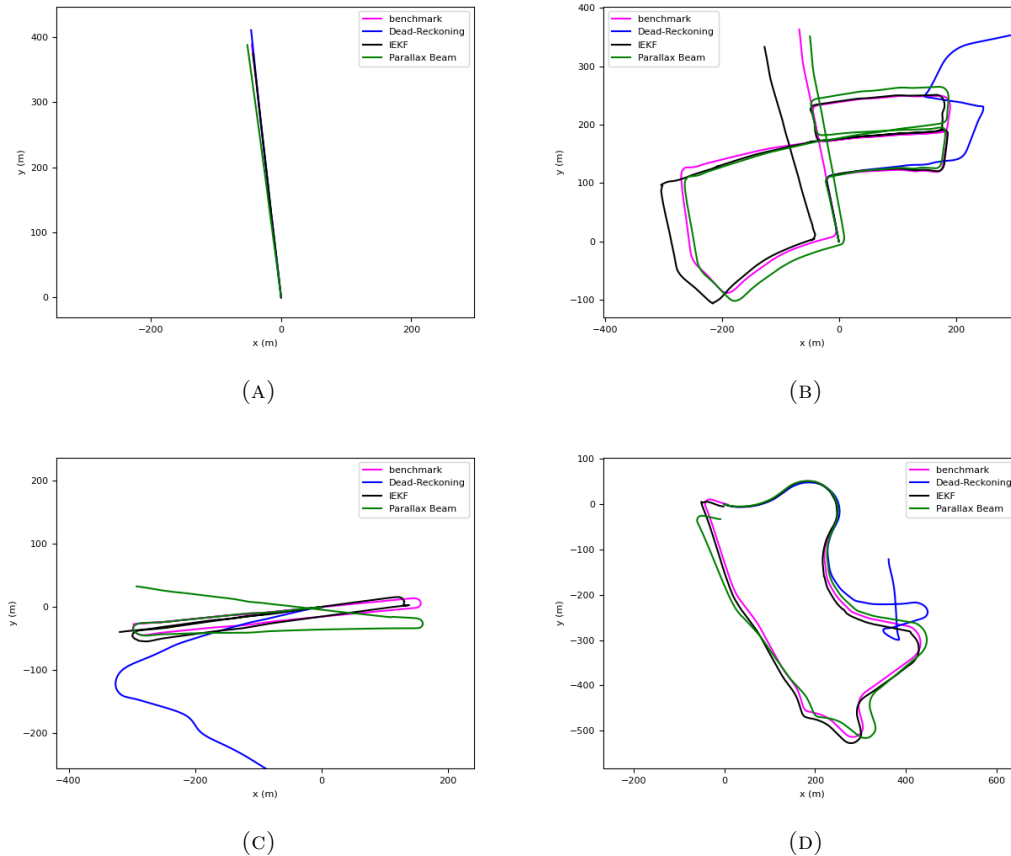


FIGURE B.9 – IEKF sur le jeu de données KITTI vision. A) séquence 04. B) séquence 05. C) séquence 06. D) séquence 9.

C MSF-EKF sur KITTI - figures supplémentaires

Dans cette partie sont présentés les essais réalisés sur les autres trajectoires du jeu de données KITTI vision qui n'ont pas été utilisées dans le corps du chapitre 5 mais qui ont participé à l'analyse développée. On retrouve donc ici les séquences estimées par le MSF-EKF avec et sans coupure du Parallax Beam. Les synthèses des figures de cette partie ont été présentées dans les tables [5.1](#) et [5.2](#).

MSF-EKF sur la séquence 04 sans coupure

Sur la séquence 04, le filtre a un régime transitoire particulièrement visible sur l'axe vertical avant de converger au bout de quelques millisecondes. La fusion proposée ne corrige pas le biais important sur l'axe vertical du Parallax Beam.

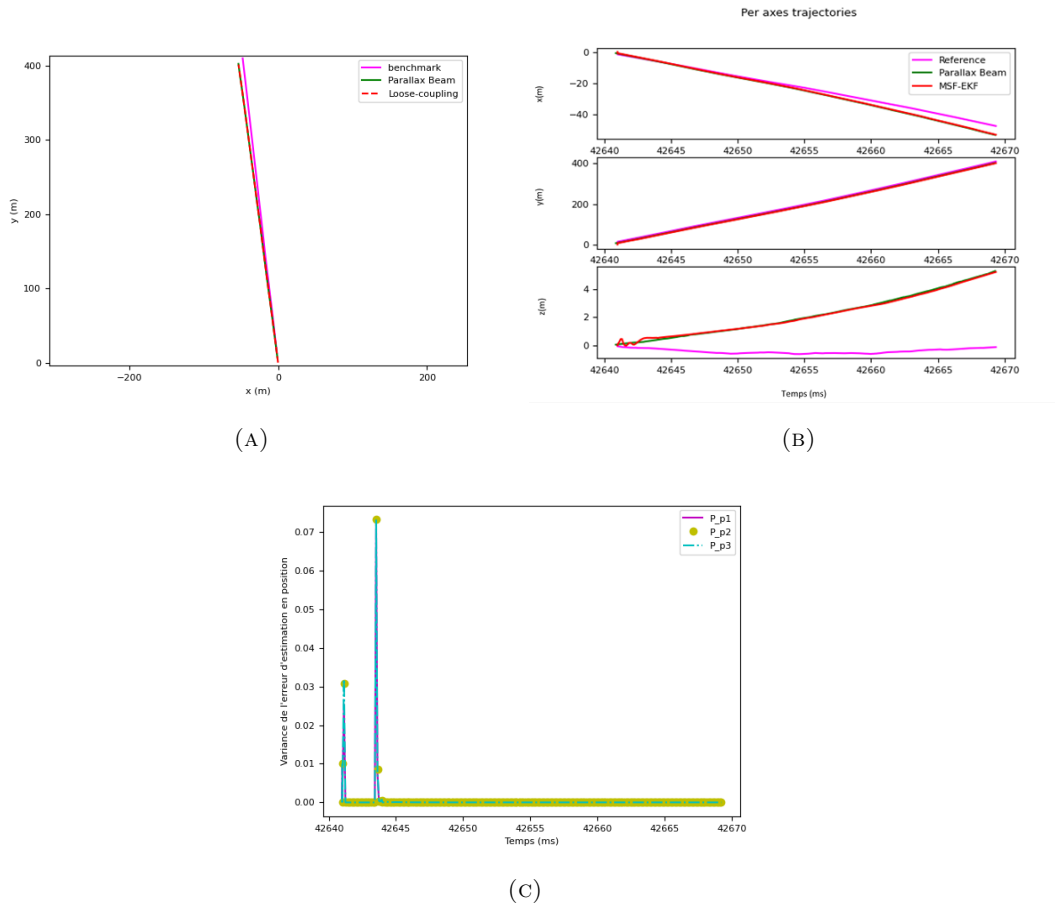


FIGURE C.1 – MSF-EKF sur la séquence 04 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

MSF-EKF sur la séquence 09 sans coupure

Sur la séquence 09, le filtre converge rapidement malgré plusieurs outliers sur l'axe vertical. La fusion proposée ne corrige pas les dérives du Parallax Beam.

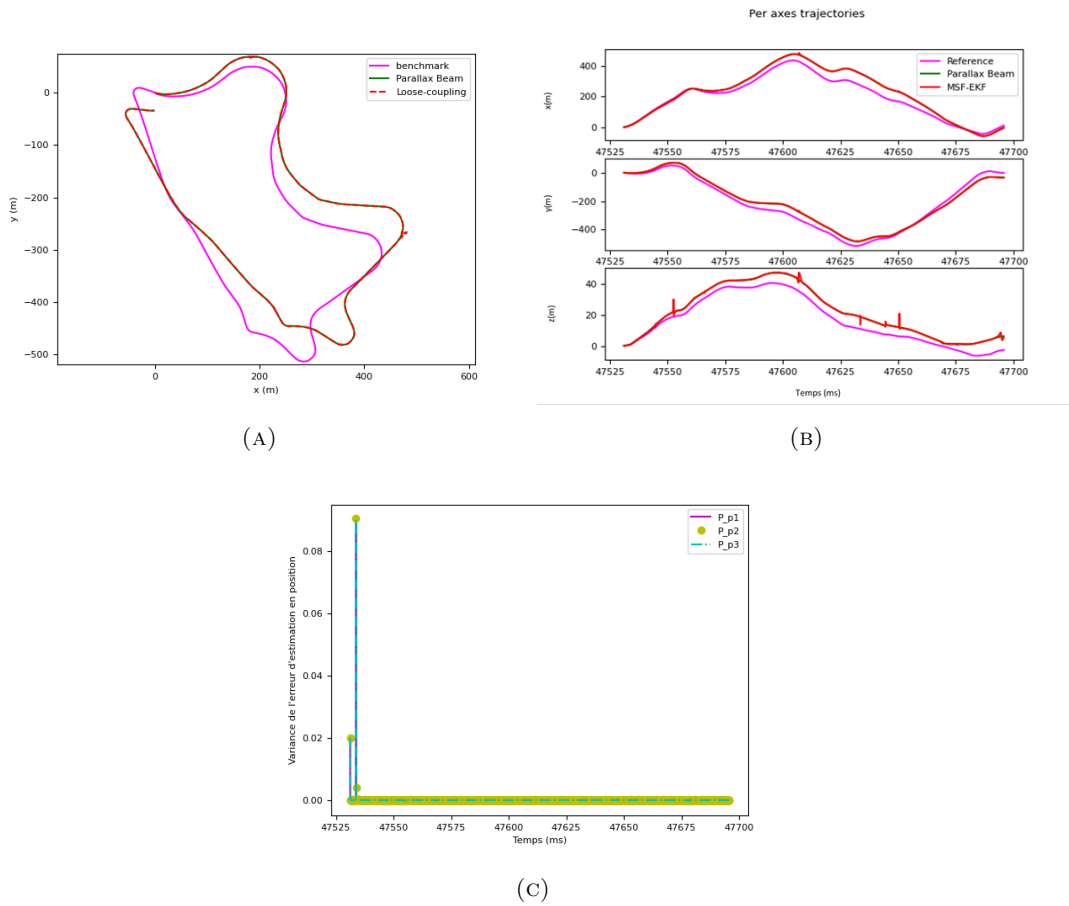


FIGURE C.2 – MSF-EKF sur la séquence 9 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

MSF-EKF sur la séquence 10 sans coupure

Sur la séquence 10, le filtre converge rapidement malgré plusieurs outliers et une dérive sur l'axe vertical. La fusion proposée ne corrige pas les dérives du Parallax Beam.

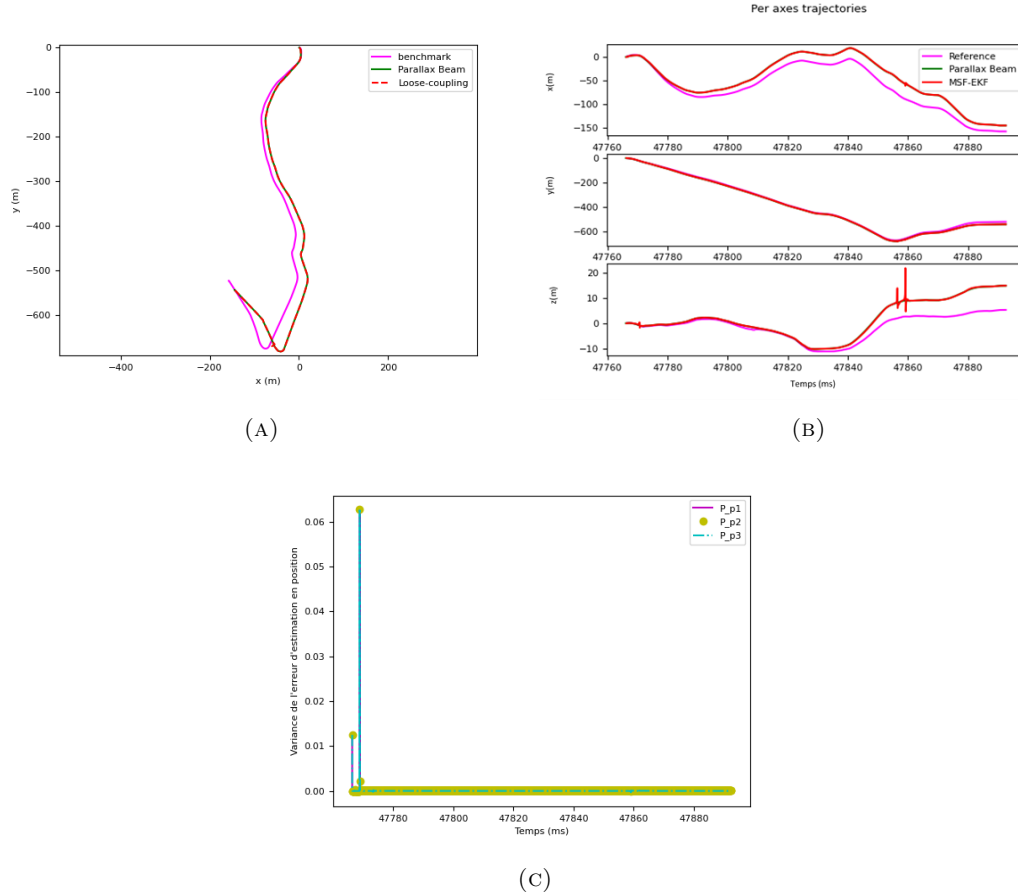


FIGURE C.3 – MSF-EKF sur la séquence 10 sans coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

MSF-EKF sur la séquence 04 avec coupure

Sur la séquence 04, le filtre a un régime transitoire particulièrement visible sur l'axe vertical avant de converger au bout de quelques millisecondes. La fusion proposée ne corrige pas le biais important sur l'axe vertical du Parallax Beam. On peut observer l'effet de la coupure sur la covariance du filtre, celle-ci est rapidement corrigée.

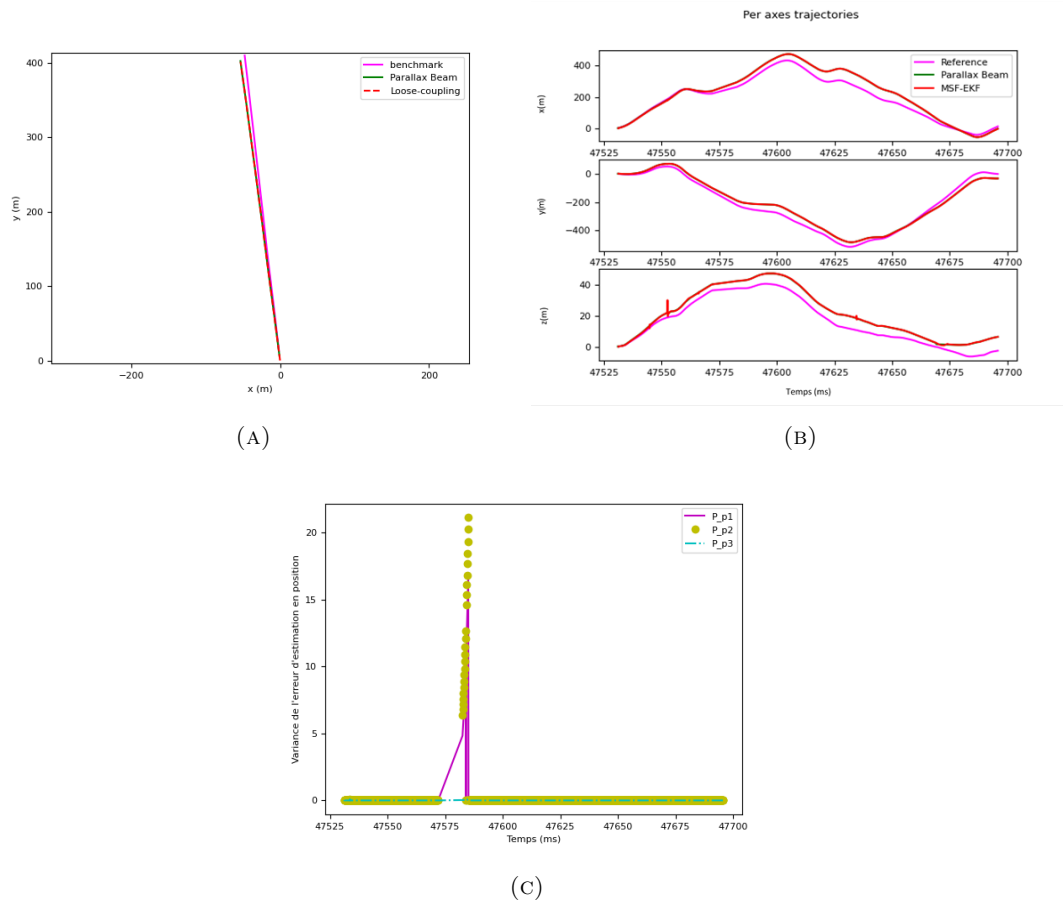


FIGURE C.4 – MSF-EKF sur la séquence 04 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

MSF-EKF sur la séquence 09 avec coupure

Sur la séquence 09, le filtre converge rapidement malgré plusieurs outliers sur l'axe vertical. La fusion proposée ne corrige pas les dérives du Parallax Beam. On peut observer l'effet de la coupure sur la covariance du filtre, celle-ci est rapidement corrigée.

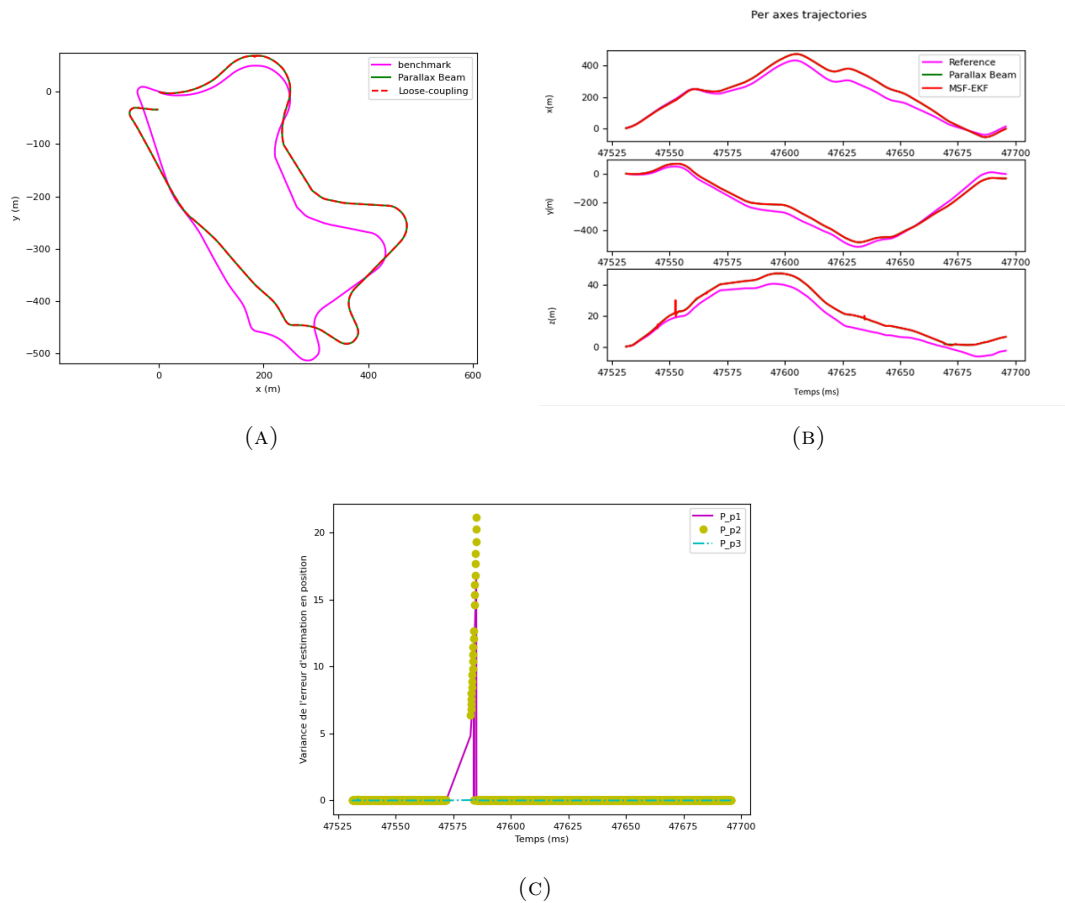


FIGURE C.5 – MSF-EKF sur la séquence 09 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

MSF-EKF sur la séquence 10 avec coupure

Sur la séquence 10, le filtre converge rapidement malgré plusieurs outliers et une dérive sur l'axe vertical. La fusion proposée ne corrige pas les dérives du Parallax Beam. On peut observer l'effet de la coupure sur la covariance du filtre, celle-ci est rapidement corrigée.

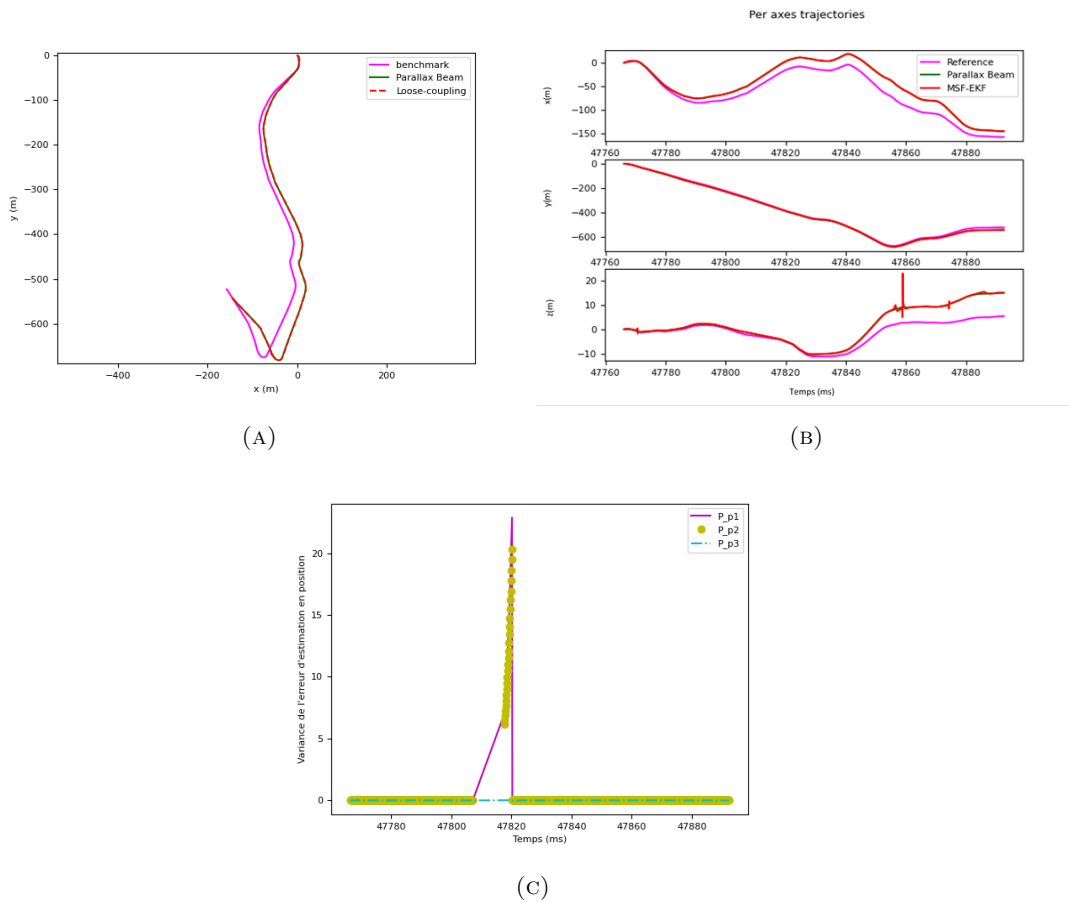


FIGURE C.6 – MSF-EKF sur la séquence 10 avec coupure. A) Trajectoires estimées dans le plan du déplacement. B) Trajectoires estimées par axes. C) Termes diagonaux de la matrice de covariance de l'erreur d'estimation.

