



Estimation d'attitude et diagnostic d'une centrale d'attitude par des outils ensemblistes

Hoang Van Nguyen

► To cite this version:

Hoang Van Nguyen. Estimation d'attitude et diagnostic d'une centrale d'attitude par des outils ensemblistes. Automatique / Robotique. Université de Grenoble, 2011. Français. NNT: . tel-00585525v1

HAL Id: tel-00585525

<https://theses.hal.science/tel-00585525v1>

Submitted on 13 Apr 2011 (v1), last revised 28 Apr 2011 (v2)

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **AUTOMATIQUE - PRODUCTIQUE**

Arrêté ministériel : 7 août 2006

Présentée par

NGUYEN Hoang Van

Thèse dirigée par **Suzanne LESECQ** et
codirigée par **Christelle GODIN**

préparée au sein du **laboratoire Gipsa-lab**
dans l'**Ecole Doctorale Electronique, Electrotechnique, Automatique,**
Traitement du Signal (EEATS)

Estimation d'attitude et diagnostic d'une centrale d'attitude par des outils ensemblistes

Thèse soutenue publiquement **le 24 mars 2011**,
devant le jury composé de :

M. Alain BARRAUD

Professeur Émérite, Institut Polytechnique de Grenoble, Président

M. Nacim RAMDANI

Professeur des Universités, Université d'Orléans, Rapporteur

M. Didier THEILLIOL

Professeur des Universités, Université Henri Poincaré, Rapporteur

M. Christophe COMBASTEL

Maître de Conférences, ENSEA, Examineur

Mm. Suzanne LESECQ

Ingénieur-chercheur, CEA-LETI, Directrice de thèse

Mm. Christelle GODIN

Ingénieur-chercheur, CEA-LETI, Co-directrice de thèse



A ma grand-mère,
à mes parents,
à mon épouse,
à mon frère.

’Dành tặng những người thân yêu của tôi, đặc biệt là bà nội tôi’.

Remerciements

Un sincère et chaleureux remerciement pour Madame Suzanne Lesecq, Ingénieur-Chercheur au CEA/LETI pour la qualité de son encadrement, sa disponibilité mais aussi la confiance qu'elle m'a témoigné durant plus de 4 années d'études sous sa direction.

Je tiens à remercier Madame Christelle Godin, Ingénieur-Chercheur au CEA/LETI, pour son co-encadrement, ses remarques et suggestions.

Je désire remercier le Ministère de l'éducation et de la formation professionnelle du Vietnamien de m'avoir octroyé dans le cadre du VietNam Overseas Scholarship Programme (projet 322), la bourse de séjour en France pour mener à bien ma thèse.

Mes remerciements les plus sincères à Monsieur Alain Barraud, Madame Sylviane Gentil et Monsieur Jean-Marc Thiriet pour leurs conseils et leur extrême gentillesse.

Mes sincères remerciements à Marie-Thérèse, Patricia, Virginie, Didier Blanc et Thierry Blanc pour leur aide dans ma vie au quotidien au laboratoire et leur amitié.

Je remercie mon cher ami, Khoat, qui m'a aidé lors de mon arrivée en France et a partagé mes difficultés durant mon séjour en France. Je remercie aussi mes amis Truong, Duc, Hieu, Nam, Tuan qui ont partagé avec moi la vie en France.

Je remercie profondément Mechraoui Amine, Do Trong Hieu, Guerrero José Fermi, Pham Van Thang, Alma Marouane, Do Anh Lam pour l'amitié au sein du laboratoire.

Un grand merci à mon cher ami Cédric Berbra, pour son aide précieuse dans ma vie en France et ses conseils lors de la rédaction de ce manuscrit.

Enfin, je tiens à faire de gros bisous à ma fille, à exprimer toute mon affection à mon épouse, et à remercier mes parents pour leur soutien aimant depuis toujours.

Tableau des matières

1	Introduction	17
2	Etat de l'art et choix préliminaires	21
2.1	Représentation de l'attitude	22
2.1.1	Motivation et définition	22
2.1.2	Représentation de l'attitude	23
2.1.2.1	Propriétés de la matrice de rotation	25
2.1.2.2	Vitesse de rotation	25
2.1.2.3	Angles de Cardan	25
2.1.2.4	Quaternion	27
2.1.3	Erreur d'attitude	30
2.1.4	Commentaire sur la représentation d'attitude	31
2.2	IMU et leur mesures	31
2.2.1	Accéléromètre	32
2.2.2	Gyromètre	34
2.2.3	Magnétomètre	36
2.2.4	Modèle des mesures de la centrale d'attitude	38
2.3	Calcul ensembliste	39
2.3.1	Arithmétique par intervalles	39
2.3.1.1	Calcul par intervalles	39
2.3.1.2	Fonction d'inclusion	40
2.3.1.3	Bissection	42
2.3.2	SIVIA	44
2.3.3	Problème de Satisfaction de Contraintes et contracteurs	45
2.3.3.1	Contracteur C_K (méthode de Krawczyk)	46
2.3.3.2	Contracteur C_N (méthode de Newton)	46
2.3.3.3	Contracteur $C_{\parallel}$ (linéarisation parallèle)	47
2.3.3.4	Contracteur $C_{\uparrow\downarrow}$ "Forward-backward"	48
2.3.4	Phase de contraction intégrée à SIVIA	48
2.4	Calcul du quaternion d'erreur dans le cas d'une représentation par intervalles	49
2.5	Conclusion	50

3	Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle	53
3.1	Définition des cas statique et dynamique et utilisation de mesures intervalles	54
3.2	Estimation de l'attitude à partir des mesures AM : cas statique	57
3.2.1	Estimation de l'attitude AM : un problème d'identification	60
3.2.1.1	Estimation de l'attitude AM : utilisation de SIVIA	60
3.2.1.2	Autre approche dérivée de SIVIA	63
3.2.1.3	Algorithmes LOOP et LoopBorne	64
3.2.2	Application des algorithmes sur des données simulées et réelles	68
3.2.2.1	Génération des données simulées	68
3.2.2.2	Description des données réelles	70
3.2.2.3	Résultats obtenus en simulation, représentation quaternion	70
3.2.2.4	Résultats obtenus en simulation, représentation angles de Cardan	80
3.2.2.5	Résultats obtenus avec des données réelles	84
3.2.3	Conclusion partielle sur l'estimation de l'attitude dans le cas statique	89
3.3	Estimation de l'attitude à partir des mesures AGM : cas dynamique	90
3.3.1	Résumé de différents observateurs par intervalles	93
3.3.1.1	Filtre de Kalman à intervalles	93
3.3.1.2	Algorithme Waltz	94
3.3.1.3	Algorithme max/min	95
3.3.2	Algorithmes d'estimation de l'attitude dans le cas AGM utilisant une intersection ou de la rétropropagation dans la phase de correction	95
3.3.2.1	Problèmes liés à l'intégration numérique	96
3.3.2.2	Algorithme EstimEI : mise en œuvre d'une intersection dans la phase de correction	98
3.3.2.3	Algorithme EstimEC : prise en compte de l'intervalle obtenu via l'équation dynamique lors du traitement des données AM	101
3.3.3	Conclusion partielle sur l'estimation de l'attitude dans le cas dynamique	105
3.4	Conclusion sur l'estimation d'attitude par des techniques ensemblistes	105
4	Diagnostic des capteurs d'une centrale inertielle AGM	107
4.1	Diagnostic et centrales inertielles	108
4.1.1	Définitions et méthodes de diagnostic utilisant une arithmétique "standard"	108
4.1.2	Utilisation d'approches ensemblistes pour le diagnostic	111
4.1.3	Diagnostic d'une centrale inertielle	113
4.2	Description des défauts ajoutés aux mesures simulées	113

4.3	Comportement des algorithmes d'estimation d'attitude en présence d'un défaut dans le jeu de mesures	114
4.4	Diagnostic d'une centrale inertielle, cas des mesures AM	118
4.4.1	Détection/localisation de défauts AM avec 5/6 mesures AM	119
4.4.2	Détection/localisation de défauts AM avec 5/6 mesures AM et un intervalle $[-1; 1]$	120
4.4.3	Détection/localisation/identification de défauts AM avec 5/6 mesures et un intervalle $[-1; 1]$	127
4.5	Diagnostic d'une centrale inertielle, cas des mesures AGM	132
4.5.1	Algorithme de diagnostic dans le cas AGM	133
4.5.2	Application de l'algorithme DiagED	135
4.5.2.1	Scénario 3 : défaut sur m_4	135
4.5.2.2	Scénario 4 : défaut sur m_9	136
4.6	Conclusion	138
5	Conclusions et perspectives	139
	Bibliographie	142

Table des figures

2.1	Repères inertiel N et repère mobile B	23
2.2	Convention des angles de rotation (angles de Cardan)	26
2.3	Composition des trois rotations (angles de Cardan)	27
2.4	Quaternion	28
2.5	Principe du capteur <i>MAG3</i> chez MEMsense [30]	32
2.6	Principe de l'accéléromètre	32
2.7	Exemple de gyromètre MEMS de chez Berkeley Sensors and Actuators Center [70]	35
2.8	Champs magnétique	36
2.9	Exemple d'un magnétomètre en technologie MEMS	37
2.10	Différentes fonctions d'inclusion	41
2.11	Effet des multi-occurrences	42
2.12	Principe d'une fonction d'inclusion par bisection	43
2.13	Contracteur $C_{\uparrow\downarrow}$	48
2.14	Illustration de l'erreur de quaternion dans un espace à 3 dimensions . . .	51
3.1	Donnée expérimentale utile y^* , donnée fournie par le capteur y et incerti- tude de mesure $\pm\delta_y$	55
3.2	Définition des différents pavés de l'ensemble des solutions obtenu par cal- cul ensembliste	56
3.3	Estimation de l'attitude à partir des mesures AM	58

Table des figures

3.4	Graphe des contraintes utilisés pour l'estimation d'attitude lorsque cette dernière est modélisée par le quaternion unitaire	59
3.5	Graphe des contraintes lorsque l'attitude est modélisée par les angles de Cardan	60
3.6	Génération des données test et estimation de l'attitude ensembliste . . .	68
3.7	Composantes du quaternion théorique en simulation	69
3.8	Angles de Cardan théoriques en simulation	69
3.9	Mesures du triaxe magnétomètres et du triaxe accéléromètres (données simulées)	69
3.10	Composantes de la vitesse angulaire (données simulées)	70
3.11	Projection de l'ensemble des solutions obtenues par SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$ pour les mesures à $t = 0 \text{ s}$.	71
3.12	Résultats obtenus avec SIVIA, LOOP et LoopBorne après modification de la fonction $[f_7]$ de l'équation (3.2.4). Les pavés jaunes (SIVIA) ne contiennent pas de quaternion de norme unité.	73
3.13	(gauche) Composantes du domaine solution $[q]$ représenté sur la sphère unité (le quaternion est exprimé en termes de paramètres d'Euler α et $\vec{u}$), (droite) coordonnées polaires $(\alpha, \gamma, \kappa)^T$ permettant de définir q	73
3.14	Angles $([\alpha], [\gamma], [\kappa])$ à partir du quaternion $[q]$ obtenu en utilisant SIVIA, LOOP et LoopBorne, mesures à $t = 0 \text{ s}$	75
3.15	Composantes du quaternion obtenues par SIVIA, LOOP et LoopBorne, mesures à $t = 0.09 \text{ s}$	76
3.16	Ensemble des solutions obtenues par LOOP, représentation quaternion de l'attitude	77
3.17	Largeur des intervalles pour chaque composante des quaternions calculés par LOOP	77
3.18	$[\beta_e]$ (2.4.2) par LOOP pour la représentation quaternion. Données simulées	78
3.19	Temps calcul de LOOP en représentation quaternion sur PC Intel i7 1.6GHz	78
3.20	Ensemble solution obtenu par LoopBorne, représentation quaternion de l'attitude	79
3.21	Largeur des pavés solution pour les différentes composantes du quaternion, utilisation de LoopBorne	79

Table des figures

3.22 $[\beta_e]$ (2.4.2) par LoopBorne en représentation quaternion. Données simulées	79
3.23 Temps calcul pour déterminer l'ensemble solution par LoopBorne en représentation quaternion sur PC Intel i7 1.6GHz. Données simulées	80
3.24 Angles de Cadrans obtenus par SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$, mesures à $t = 0$ s	81
3.25 Angles de Cardan obtenus par LOOP	81
3.26 Largeur des intervalles pour chacun des angles de Cardan obtenus par LOOP	82
3.27 Temps de calcul par LOOP en représentation des angles de Cardan sur PC Intel i7 1.6GHz	82
3.28 Angles de Cardan par LoopBorne	83
3.29 Largeur des pavés des angles de Cardan par LoopBorne	83
3.30 Temps de calcul par LoopBorne en représentation angles de Cardan sur PC Intel i7 1.6GHz	84
3.31 Mesures réelles des triaxes magnétomètre et accéléromètre	85
3.32 Estimation des composantes du quaternion par l'algorithme LOOP au cours du temps. Données réelles. q_{est} en bleu et $q_{capteur}$ fourni par l'algorithme interne de l'IMU en rouge	86
3.33 Valeur de $[\beta_e]$ (2.4.2) au cours du temps. Utilisation de l'algorithme LOOP. Données réelles	86
3.34 Temps de calcul par LOOP à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation quaternion de l'attitude	86
3.35 Estimation des composantes du quaternion par l'algorithme LoopBorne au cours du temps. Données réelles	87
3.36 Valeur de $[\beta_e]$ (2.4.2) au cours du temps. Utilisation de l'algorithme LoopBorne. Données réelles	87
3.37 Temps de calcul par LoopBorne à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation quaternion de l'attitude	87
3.38 Estimation des composantes des angles de Cardan par l'algorithme LoopBorne au cours du temps. Données réelles	88

Table des figures

3.39	Temps de calcul par LoopBorne à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation angles de Cardan de l'attitude	88
3.40	Principe d'un observateur ensembliste d'attitude. Les données d'entrée sont fournies par une centrale inertielle	91
3.41	Exemple d'une trajectoire $[y]$ obtenue par un observateur ensembliste . .	93
3.42	Principe de l'algorithme EstimEI pour une paramétrisation de l'attitude avec un quaternion	99
3.43	Domaine solution obtenu avec l'algorithme EstimEI pour la paramétrisation avec le quaternion de l'attitude. Données simulées	100
3.44	Largeur du domaine solution suivant les 4 composantes du quaternion lorsque l'on utilise l'algorithme EstimEI. Données simulées	100
3.45	$[\beta_e]$ (2.4.2) pour l'algorithme EstimEI. Données simulées	101
3.46	Différentes situations de la phase d'intersection de l'algorithme EstimEI. Données simulées.	101
3.47	Principe de l'algorithme EstimEC sur les composants de quaternion unitaire	102
3.48	Intervalles pour chacun des composants du quaternion obtenu avec l'algorithme EstimEC. Données simulées	102
3.49	Largeur des intervalles solution obtenus avec EstimEC pour chacune des composantes du quaternion des composantes de quaternion. Données simulées	103
3.50	$[\beta_e]$ (2.4.2) pour les résultats obtenus avec l'algorithme EstimEC. Données simulées	103
3.51	Comparaison entre $[\hat{q}_{k/k}]$ obtenus respectivement avec EstimEI et EstimEC. Données simulées	104
3.52	Comparaison des temps de calcul sur PC Intel i7 1.6GHz entre les algorithmes EstimEI et EstimEC. Données simulées	104
4.1	Les 3 étapes du diagnostic	109
4.2	Principe de location par une génération du vecteur de résidus. (a) schéma dédié, (b) schéma généralisé	112
4.3	Illustration des défauts considérés (en simulation)	114

Table des figures

4.4	Utilisation de l'algorithme LOOP (quaternion) avec un défaut sur m_4 , i.e. accéléromètre suivant l'axe x	115
4.5	Résultats pour l'algorithme LOOP (quaternion) avec un défaut sur m_4 (simulation)	115
4.6	Principe de l'algorithme LOOP (angle de Cardan) avec un défaut sur m_2 , i.e. magnétomètre suivant l'axe y	116
4.7	Résultats pour l'algorithme LOOP (angle de Cardan) avec un défaut sur m_2 (simulation)	116
4.8	Principe de l'algorithme EstimEC avec un défaut sur m_9 , i.e. gyromètre suivant l'axe z	117
4.9	Résultats pour l'algorithme EstimEC avec un défaut sur m_9 (simulation)	118
4.10	Estimation de l'attitude avec 5/6 mesures AM. Données simulées, [58] . .	119
4.11	Deuxième structure d'algorithme de diagnostic AM avec 5/6 mesures AM et un intervalle $[-1; 1]$	120
4.12	Composantes de $[q_{est}]$ par LOOP sans prendre m_1 pour le calcul, défaut sur m_4	121
4.13	Composantes de $[q_{est}]$ par LOOP sans prendre m_2 pour le calcul, défaut sur m_4	121
4.14	Composantes de $[q_{est}]$ par LOOP sans prendre m_3 pour le calcul, défaut sur m_4	122
4.15	Composantes de $[q_{est}]$ par LOOP sans prendre m_4 pour le calcul, défaut sur m_4	122
4.16	Composantes de $[q_{est}]$ par LOOP sans prendre m_5 pour le calcul, défaut sur m_4	123
4.17	Composantes de $[q_{est}]$ par LOOP sans prendre m_6 pour le calcul, défaut sur m_4	123
4.18	Angles de Cardan par LOOP sans prendre m_1 pour le calcul, défaut sur m_2	124
4.19	Angles de Cardan par LOOP sans prendre m_2 pour le calcul, défaut sur m_2	125
4.20	Angles de Cardan par LOOP sans prendre m_3 pour le calcul, défaut sur m_2	125
4.21	Angles de Cardan par LOOP sans prendre m_4 pour le calcul, défaut sur m_2	126
4.22	Angles de Cardan par LOOP sans prendre m_5 pour le calcul, défaut sur m_2	126

Table des figures

4.23	Angles de Cardan par LOOP sans prendre m_6 pour le calcul, défaut sur m_2	127
4.24	Troisième approche : génération des résidus avec <i>DiagES</i>	128
4.25	Composants de R_4 (cf. table de signature) pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	129
4.26	Composantes du vecteur de résidus r_1 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	129
4.27	Composantes du vecteur de résidus r_2 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	130
4.28	Composantes du vecteur de résidus r_3 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	130
4.29	Composantes du vecteur de résidus r_4 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	131
4.30	Composantes du vecteur de résidus r_5 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	131
4.31	Composantes du vecteur de résidus r_6 pour l'algorithme <i>DiagES</i> avec le défaut sur m_4	132
4.32	Principe de l'algorithme de diagnostic <i>DiagED</i> pour le cas AGM	134
4.33	Résultats l'algorithme <i>DiagED</i> pour le scénario 3 (défaut sur m_4)	136
4.34	Résultats l'algorithme <i>DiagED</i> pour le scénario 4 (défaut sur m_9)	137

Chapitre 1

Introduction

L'estimation de l'attitude (ou orientation) dans un espace à 3 dimensions est un sujet qui anime la communauté scientifique depuis plusieurs décennies (depuis au moins le milieu des années 60), avec des applications aussi variées que le contrôle en attitude des satellites et des aéronefs, la robotique aérienne et sous-marine, les applications liées à la santé (suivi des personnes dans leur environnement, ré-éducation et réhabilitation) ou bien encore l'explosion des applications liées aux films et aux jeux d'animation. Ces deux derniers domaines (santé et "entertainment") nécessitent de concevoir des systèmes fiables, fournissant une estimation de l'attitude (et parfois de la position) "précise", peu encombrant, portables et autonomes, donc peu gourmands en énergie. Notons que la précision est "problème-dépendant", une précision de quelques degrés étant suffisantes pour les jeux d'animation tandis qu'il faut atteindre une précision inférieure au degré, voire au dixième de degré, pour la robotique.

Ces contraintes ont conduit les chercheurs et les concepteurs de ces systèmes de capture d'attitude à se tourner vers des dispositifs à base de capteurs MEMS (Micro Electro-Mechanical System), le dispositif embarquant aujourd'hui traditionnellement un triaxe accéléromètre (i.e. 3 accéléromètres montés en triaxe idéalement orthogonal, avec une sensibilité similaire du capteur suivant les 3 axes de mesure), un triaxe magnétomètre et un triaxe gyromètre. Le triaxe accéléromètre mesure, dans le repère lié au capteur, la gravité et les accélérations propres du corps rigide sur lequel sont fixés les capteurs. Le triaxe magnétomètre mesure le champ magnétique terrestre ainsi que les perturbations magnétiques, et le triaxe gyromètre mesure la vitesse de rotation du corps rigide. La fusion de toutes ces données permet de remonter à l'attitude du corps rigide dans l'espace à 3 dimensions.

Les capteurs embarqués dans ces systèmes sont sujets à des défauts. Il est donc souhaitable d'être capable d'estimer "l'état de santé" de l'ensemble des capteurs afin de garantir qu'on a une estimation fiable de l'attitude. Or, les techniques standard d'estimation de

l'attitude ne permettent pas de savoir si le système de capteurs est sain. Il faut donc adjoindre des techniques de diagnostic à l'algorithme de fusion afin de garantir qu'on n'a pas de défaut dans l'ensemble des capteurs mis en jeu pour déterminer l'attitude. Le problème du diagnostic des centrales inertielles (i.e. dispositif contenant un triaxe accéléromètre, un triaxe magnétomètre et un triaxe gyromètre) a été abordé par exemple par C. Berbra [10]. Ce dernier utilise des techniques classiques de diagnostic : un banc d'estimateur/observateurs est mis en œuvre permettant de détecter, localiser et identifier des défauts. L'algorithme de diagnostic développé exploite des outils d'optimisation non-linéaire avec contraintes égalité, et des filtres de Kalman étendus.

Dans notre travail, on revisite le problème de l'estimation de l'attitude avec des outils ensemblistes. Le principal intérêt d'utiliser ces outils dans le cadre de la conception d'un système de capture de mouvement est que connaissant les caractéristiques des capteurs utilisés, on peut garantir un **ensemble de solutions**, en faisant une hypothèse sur le "bruit" inconnu mais borné. Ainsi, le concepteur peut vérifier que pour une configuration donnée de son système de mesures, les performances en sortie (i.e. la précision sur l'estimée) seront atteintes de manière **garantie**. Le second axe de notre travail concerne le diagnostic de la centrale inertielle. Ici encore, nous mettons en œuvre des outils ensemblistes, car ils offrent des résultats "garantis". L'objectif de cette seconde partie de l'étude était d'analyser la possibilité d'utiliser le calcul ensembliste pour le diagnostic de la centrale inertielle.

Aussi, ce manuscrit est composé de trois chapitres principaux et d'un chapitre de conclusions et perspectives :

- le chapitre 2 introduit les représentations de l'attitude d'un corps rigide, en particulier le quaternion unitaire et les angles de Cardan. On présente ensuite les capteurs qui seront considérés dans cette étude et on donne le modèle des différentes mesures. Rappelons que ce système de mesure est constitué d'un triaxe d'accéléromètres (A), d'un triaxe de magnétomètres (M) et d'un triaxe de gyromètres (G). Puis, on présente le calcul par intervalles ainsi que le concept d'un Problème de Satisfaction de Contraintes (CSP) ;
- le chapitre 3 étudie l'estimation ensembliste de l'attitude à partir de mesures fournies par une centrale d'attitude constituée d'un triaxe d'accéléromètres (A), d'un triaxe de magnétomètres (M) et d'un triaxe de gyromètres (G). On décompose ce chapitre en deux parties principales. La première partie traite des capteurs AM pour le cas statique et la deuxième traite des mesures AGM pour le cas dynamique. Plusieurs algorithmes sont proposés et validés avec des données simulées. Pour tous les algorithmes, le but est de vérifier que la valeur théorique de l'attitude (i.e. utilisée pour simuler les mesures) est bien comprise dans le domaine estimé pour l'attitude. Les algorithmes du cas AM ont également été testés avec des données réelles ;
- le chapitre 4 traite du diagnostic ensembliste de la centrale inertielle. Dans un pre-

mier temps, la capacité de détection d'un défaut apparaissant sur une des mesures avec les algorithmes développés au chapitre 3 est montrée sur les données simulées. A partir de cette capacité à détecter une incohérence dans l'ensemble des mesures utilisé pour estimer l'attitude, et en s'inspirant de techniques classiques de diagnostic basées sur des bancs d'observateurs, on a proposé deux algorithmes permettant respectivement de traiter du diagnostic des cas AM et AGM. Grâce à la table de signature, l'algorithme pour le cas statique permet de détecter, de localiser et aussi d'identifier un défaut unique apparu sur l'une des 6 mesures (AM) de la centrale inertielle. L'algorithme pour le cas dynamique permet quant à lui de détecter et de localiser le défaut apparu sur l'une des 9 mesures (AGM) de la centrale inertielle ;

- dans le chapitre 5 (conclusions et perspectives), nous résumons nos apports et nous dressons quelques perspectives de recherche.

Chapitre 2

Etat de l'art et choix préliminaires

L'estimation de l'attitude est depuis longtemps un domaine de recherche actif. Il couvre des domaines aussi larges que l'aéronautiques, les drones, la biologie et l'étude du comportement animal, les applications médicales, les jeux multimédias ...

Le laboratoire GIPSA mène depuis quelques années des travaux dans ce domaine, par exemple [26] pour le problème de la commande et de l'observation de l'attitude d'un drone quadrotor mais aussi [10] pour la problématique du diagnostic des systèmes en réseau, avec une application au drone quadrotor. Les études menées au GIPSA-Lab pour l'estimation d'attitude reposent sur une centrale inertielle constituée de triaxes accéléromètre, magnétomètre et gyromètre. On utilisera donc dans notre étude le même type de modalités de mesure.

Dans ce manuscrit, on a pris le parti de ne pas présenter exhaustivement l'état de l'art sur les techniques d'estimation d'attitude utilisant de l'arithmétique "standard" (c'est-à-dire de l'arithmétique non intervalle). En effet, ces dernières sont nombreuses et dépendent des modalités de mesures disponibles, de l'application visée, et de la représentation d'attitude choisie.

Par exemple, dans le cas où on dispose de mesures statiques vectorielles (e.g. un triaxe d'accéléromètre et un triaxe de magnétomètre, ou tout autre triaxe de mesure pointant sur une étoile), Whaba [80] a proposé une formulation basée sur une approche de moindres carrés. S'en sont déduits toute une famille d'algorithmes et on se reportera à l'article de F.J. Markley [49] qui donne un état de l'art de ces approches.

Dans le cas où on a également des équations de mesure dynamique (typiquement lorsqu'on utilise un gyroscope pour mesurer la vitesse de rotation), on se tourne alors vers des approches de filtrage ou d'observateurs. Ici encore, les solutions développées sont variées et nous avons choisi de ne pas en faire un développement complet. On pourra consulter par exemple [26] [50] [51].

A notre connaissance, il existe aujourd'hui peu de travaux sur l'estimation de l'atti-

tude avec des techniques ensemblistes, et c'est précisément la technique que nous nous proposons d'exploiter ici. Par essence, le calcul ensembliste fournit un **résultat garanti**, permettant par exemple de valider le choix d'un type de capteur donné compte tenu d'une précision attendue sur l'attitude estimée. La solution retournée est fournie sous la forme d'un **ensemble de solutions**.

Ce chapitre présente les outils utilisés durant ce travail de doctorat. Tout d'abord, nous rappelons la définition de l'*attitude* d'un corps rigide dans un espace à trois dimensions ($3D$) et nous donnons différentes *paramétrisations* de cette attitude. Nous verrons en particulier les deux représentations qui seront utilisées tout au long de cette étude : le quaternion unitaire et les angles de Cardan.

L'attitude tout au long de notre travail est estimée à partir de mesures fournies par une *centrale d'attitude* (IMU - Inertial Measurement Unit) constituée d'un triaxe d'accéléromètres, d'un triaxe de magnétomètres et d'un triaxe de gyromètres. Le modèle de chacune des mesures de la centrale d'attitude est présenté dans ce chapitre. On donne également des caractéristiques typiques (compte tenu d'une centrale d'attitude disponible dans le commerce) qui seront utilisées afin de simuler des données utilisées lors de la validation des algorithmes proposés.

Enfin, nous résumons quelques notions d'*arithmétique par intervalles*. Nous verrons ensuite l'algorithme SIVIA (Set Inverter Via Interval Analysis) [38] dédié au problème de l'inversion ensembliste. Enfin, le *problème de satisfaction de contraintes* (CSP - Constraints Satisfaction Problem) est défini et nous résumerons différents contracteurs pouvant être utilisés pour résoudre un CSP. Ces outils seront utilisés par la suite pour estimer l'attitude et aussi diagnostiquer de défaut de capteur ce qui seront étudiés dans les chapitres suivants.

2.1 Représentation de l'attitude

Tout d'abord, nous définissons ce que l'on entend par *attitude* (ou *orientation*) d'un corps rigide dans un espace à trois dimensions ($3D$). Nous donnons en particulier les deux repères qui permettent de mesurer cette orientation du corps rigide. Nous donnons ensuite 2 paramétrisations possibles de cette attitude, paramétrisations qui seront utilisées au cours de ce travail.

2.1.1 Motivation et définition

Quand un objet est dans un espace à trois dimensions, on s'intéresse en général à sa position, ou à sa trajectoire à chaque instant. En pratique, on donne la position de

son centre d'inertie G par rapport à un repère fixe de référence noté N . Pourtant, cette position du centre d'inertie G n'est pas suffisante pour décrire pleinement le corps rigide dans l'espace à trois dimensions. L'orientation de cet objet par rapport au repère fixe N constitue un paramètre important. Par exemple, la connaissance de la position du centre de gravité d'un véhicule est insuffisante pour contrôler sa trajectoire ; il faut également connaître son orientation.

C'est pourquoi il est nécessaire de définir l'orientation du repère $B(x_b, y_b, z_b)$ attaché au centre de gravité du corps rigide par rapport au repère de référence fixe $N(x_n, y_n, z_n)$ (voir figure 2.1). Un repère de référence classiquement utilisé est le repère inertiel Nord-Est-Bas, noté NED (North-East-Down). Il est défini à partir d'un plan tangent à la surface de la terre. Dans ce cas, les vecteurs unitaires (x_n, y_n, z_n) coïncident respectivement avec les directions Nord, Est et la pesanteur. Il sera utilisé dans ce document.

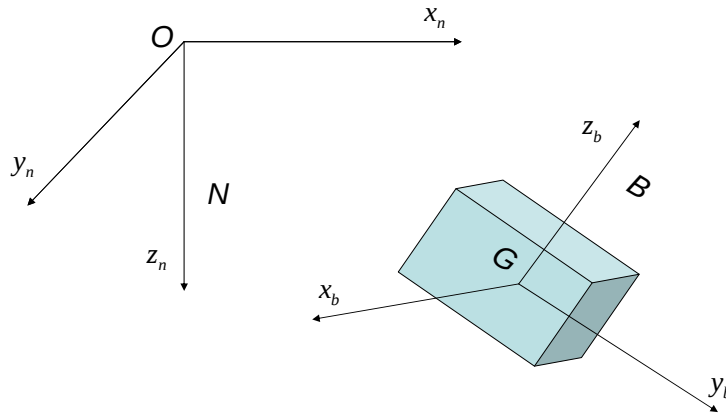


Figure 2.1 – Repères inertiel N et repère mobile B

Définition 1. L'attitude (ou orientation) d'un corps rigide dans un espace à trois dimensions est caractérisée par les directions des axes du repère B attaché au corps rigide en son centre de gravité, par rapport à un repère fixe N de référence. Les axes du repère B coïncident généralement avec les axes principaux d'inertie du corps rigide.

Dans la représentation de Cardan, il s'agit de déterminer les rotations d'avant en arrière (tangage), de gauche à droite (roulis) et autour d'un axe vertical (lacet) du corps rigide considéré.

2.1.2 Représentation de l'attitude

L'attitude d'un corps rigide dans l'espace peut être représentée avec différentes paramétrisations, chacune ayant ses avantages et inconvénients. En conséquence, il est

nécessaire, compte tenu de l'application traitée, de choisir une représentation adéquate de l'attitude. Le lecteur est invité à se reporter à l'article de M. Shuster [71] qui donne une vue d'ensemble de ces représentations. On résume ci-dessous les principales représentations d'attitude, qui seront utilisées dans le cadre de notre travail : les angles de Cardan et le quaternion unitaire.

Soient $\vec{b}$ et $\vec{r}$ les coordonnées d'un vecteur $\vec{x}$ exprimé dans B et N respectivement. Le vecteur $\vec{b}$ peut être écrit en termes du vecteur $\vec{r}$ en utilisant la matrice de rotation $\mathcal{C}$. Soit $\vec{e} = [e_1 \ e_2 \ e_3]^T$ un vecteur unitaire colinéaire à l'axe de rotation L autour duquel B est tourné d'un angle β afin de coïncider avec N . En conséquence, $\vec{b}$ est obtenu par :

$$\vec{b} = \cos \beta \vec{r} + (1 - \cos \beta) \vec{e} \vec{e}^T \vec{r} - \sin \beta \vec{e} \times \vec{r} \quad (2.1.1)$$

L'équation (2.1.1) permet de rappeler la définition suivante.

Définition 2 (Rotation simple [20]). Le mouvement d'un système de coordonnées B par rapport à un système de coordonnées N est appelé rotation simple de B à N s'il existe une droite L , appelée axe de rotation, dont l'orientation par rapport à B et N reste inchangée entre le début et la fin du mouvement.

En représentation matricielle, les coordonnées des vecteurs $\vec{b}$ et $\vec{r}$ sont liées par :

$$\vec{b} = \mathcal{C} \cdot \vec{r} \quad (2.1.2)$$

où la matrice $\mathcal{C}$ est un opérateur qui prend un vecteur fixe $\vec{r}$ exprimé dans N et l'exprime dans B . De l'équation (2.1.1), on déduit que :

$$\mathcal{C} = \cos \beta I_3 + (1 - \cos \beta) \vec{e} \vec{e}^T - \sin \beta [\vec{e} \times] \quad (2.1.3)$$

où I_3 représente la matrice identité de dimension 3×3 et $[\vec{\xi} \times]$ représente un tenseur anti-symétrique associé au vecteur $\vec{\xi} = (\xi_1 \ \xi_2 \ \xi_3)^T$, donné par :

$$[\vec{\xi} \times] = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}^\times = \begin{pmatrix} 0 & \xi_3 & -\xi_2 \\ -\xi_3 & 0 & \xi_1 \\ \xi_2 & -\xi_1 & 0 \end{pmatrix} \quad (2.1.4)$$

La matrice $\mathcal{C} \in \mathbb{R}^{3 \times 3}$ transforme complètement l'orientation du repère mobile B par rapport au repère inertiel N et elle permet de faire la transformation de coordonnées d'un vecteur d'un système de coordonnées à l'autre. Cette matrice est appelée matrice de cosinus directeurs, matrice de rotation, matrice de passage ou encore matrice d'attitude.

2.1.2.1 Propriétés de la matrice de rotation

La matrice de rotation $\mathcal{C}$ n'est pas quelconque, elle appartient au sous-espace des matrices orthogonales de dimension trois, dénoté $S0(3)$. Elle est définie par :

$$S0(3) = \{\mathcal{C} \mid \mathcal{C} \in \mathbb{R}^{3 \times 3}, \mathcal{C}^T \mathcal{C} = I_3, \det(\mathcal{C}) = 1\} \quad (2.1.5)$$

Chaque élément c_{ij} est un cosinus directeur, tel que :

$$\mathcal{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \quad (2.1.6)$$

Notons :

$$\mathbf{c}_1 = \begin{pmatrix} c_{11} \\ c_{21} \\ c_{31} \end{pmatrix}, \quad \mathbf{c}_2 = \begin{pmatrix} c_{12} \\ c_{22} \\ c_{32} \end{pmatrix}, \quad \mathbf{c}_3 = \begin{pmatrix} c_{13} \\ c_{23} \\ c_{33} \end{pmatrix} \quad (2.1.7)$$

On a donc

$$\mathcal{C} = (\mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3) \quad (2.1.8)$$

où

$$\mathbf{c}_i^T \mathbf{c}_i = 1 \quad \text{et} \quad \mathbf{c}_i^T \mathbf{c}_j = 0 \quad \forall i \neq j \quad (2.1.9)$$

2.1.2.2 Vitesse de rotation

Si le repère B tourne par rapport au repère N avec une vitesse angulaire $\vec{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T$ mesurée dans B , la matrice $\mathcal{C}$ sera fonction du temps $\mathcal{C}(t)$. La relation entre l'attitude du corps et la vitesse angulaire $\vec{\omega}$ est exprimée par :

$$\dot{\mathcal{C}}(t) = [\vec{\omega} \times] \cdot \mathcal{C}(t) \quad (2.1.10)$$

L'équation précédente est connue sous le nom d'"équation cinématique" ; elle est par exemple utilisée dans les systèmes de navigation.

2.1.2.3 Angles de Cardan

L'orientation d'un corps rigide dans un espace à trois dimensions peut être caractérisée par trois angles qui permettent de passer du repère de référence N au repère du corps rigide B . Le passage de N à B est réalisé via trois rotations successives R_1, R_2, R_3 d'angles φ, θ, ψ respectivement, autour de trois axes $\vec{n}_1, \vec{n}_2$ et $\vec{n}_3$ respectivement. M. Shuster [71] montre qu'il existe 12 combinaisons possibles de ces axes de rotation. Par

exemple, la plus connue est la suite de rotations 3 – 1 – 3 qui correspond aux angles classiques d'Euler : on part du repère $(Ox_0y_0z_0)$, on tourne autour de l'axe Oz_0 (le troisième axe du repère $(Ox_0y_0z_0)$) d'un angle φ et on obtient $(Ouvz_0)$; on tourne ensuite autour de Ou (le premier axe du repère $Ouvz_0$) d'un angle θ et on trouve le repère $(Ouv'Z)$; enfin on applique une rotation d'angle ψ d'axe OZ (le troisième axe du repère $(Ouv'Z)$) qui amène le repère $(Ouv'Z)$ en $(OXYZ)$. [71] définit 6 jeux de rotation "symétriques" (par exemple 1 – 2 – 1, 3 – 1 – 3, etc.) et 6 jeux anti-symétriques (par exemple 3 – 1 – 2, 3 – 2 – 1, etc.). Ces derniers constituent l'ensemble des jeux de type "Cardan", le plus classique étant celui correspondant à 3 – 2 – 1.

Dans notre travail, nous utiliserons les angles de Cardan 3 – 2 – 1, le passage d'un repère à l'autre s'effectuant en trois étapes (cf. figures 2.2 et 2.3) :

1. la rotation φ , autour de l'axe Oz_0 , transforme $(Ox_0y_0z_0)$ en $(Ouvz_0)$;
2. la rotation θ , autour de l'axe Ov , fait passer $(Ouvz_0)$ en $(OXvw)$;
3. la rotation ψ , autour de l'axe OX , transforme $(OXvw)$ en $(OXYZ)$. Ce dernier repère est lié au corps rigide.

Les angles satisfont les conditions suivantes :

$$\begin{cases} -\pi/2 \leq \varphi \leq \pi/2 \\ -\pi \leq \theta \leq \pi \\ -\pi \leq \psi \leq \pi \end{cases} \quad (2.1.11)$$

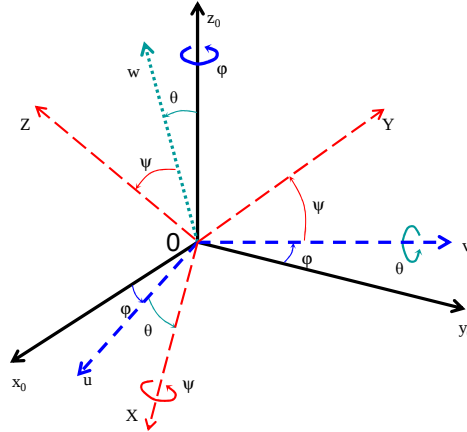


Figure 2.2 – Convention des angles de rotation (angles de Cardan)

Le passage des coordonnées d'un vecteur $\vec{r}$ exprimé dans le repère $(Ox_0y_0z_0)$ à $\vec{b}$ exprimé dans $(OXYZ)$ s'effectue via le produit de trois matrices de rotation :

$$\mathcal{C} = \mathcal{C}_{X,\theta} * \mathcal{C}_{v,\varphi} * \mathcal{C}_{z_0,\psi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\psi & s\psi \\ 0 & -s\psi & c\psi \end{pmatrix} \cdot \begin{pmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{pmatrix} \cdot \begin{pmatrix} c\varphi & s\varphi & 0 \\ -s\varphi & c\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.1.12)$$

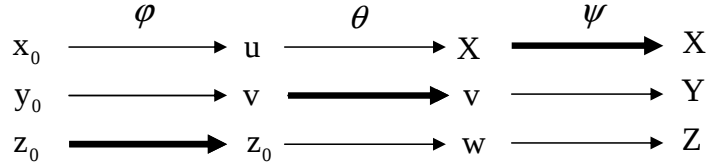


Figure 2.3 – Composition des trois rotations (angles de Cardan)

ce qui donne

$$\mathcal{C} = \begin{pmatrix} c\varphi c\theta & s\varphi c\theta & -s\theta \\ -s\varphi c\psi + c\varphi s\theta s\psi & c\varphi c\psi + s\varphi s\theta s\psi & c\theta s\psi \\ s\varphi s\psi + c\varphi s\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\theta c\psi \end{pmatrix} \quad (2.1.13)$$

où $c = \cos(\cdot)$ et $s = \sin(\cdot)$.

Le problème de cette représentation est l'existence de singularités : une même attitude dans l'espace peut être obtenue pour certains cas par une infinité de composition de rotations. Si l'angle $\theta = \pm\pi/2$, le repère de la troisième rotation est identique à celui de la première rotation. On est alors en présence de ce qui est appelé dans la littérature le *Gimbal lock*. De plus, cette singularité géométrique apparaît dans l'équation cinématique associée [20] :

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (2.1.14)$$

où $\vec{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ est la vitesse angulaire du corps rigide mesurée dans le système de coordonnées B , par rapport au système de coordonnées N .

Cette singularité peut être évitée si la plage d'orientation du corps rigide dans l'espace est limitée. Par conséquent, cette représentation peut être utilisée pour des applications pour lesquelles compte tenu des contraintes physiques, le phénomène de "Gimbal lock" n'est pas rencontré (par exemple, commande de certains robots manipulateurs ou aéronautique).

2.1.2.4 Quaternion

Pour s'affranchir du Gimbal lock, on peut utiliser la représentation "quaternion unitaire" $q \in \mathbb{R}^4$. Le quaternion de norme unité définit une rotation dans l'espace qui peut être réalisée par une simple rotation d'angle α autour d'un axe de rotation $\vec{u}$ (voir la figure 2.4), ce qui correspond à la représentation d'Euler-Rodrigues.

On définit $\mathbb{H}$ l'ensemble des quaternions unitaires par :

$$\mathbb{H} = \{q \mid q_0^2 + \vec{q}^T \vec{q} = 1, q = [q_0 \quad \vec{q}^T]^T, q_0 \in \mathbb{R}, \vec{q} \in \mathbb{R}^3\} \quad (2.1.15)$$

$\vec{q} = [q_1 \quad q_2 \quad q_3]^T$ et q_0 sont la partie vectorielle (partie imaginaire) et la partie scalaire (partie réelle) du quaternion respectivement. Le quaternion identité est défini par :

$$q_{id} = [1 \quad 0^T]^T \quad (2.1.16)$$

Le quaternion conjugué est donné par :

$$\bar{q} = [q_0 \quad -\vec{q}^T]^T \quad (2.1.17)$$

Comme le quaternion est unitaire, on a $q^{-1} = \bar{q}$.

Le quaternion unitaire peut être écrit en considérant un vecteur unitaire $\vec{u}$, nommé axe d'Euler, et un angle de rotation α autour de cet axe :

$$q = \begin{pmatrix} \cos \frac{\alpha}{2} \\ \sin \frac{\alpha}{2} \cdot \vec{u} \end{pmatrix} = \begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix} \in \mathbb{H} \quad (2.1.18)$$

avec $\|\vec{u}\| = 1$.

Remarque 1. Si l'angle de rotation α est égal à zéro (i.e. aucune rotation n'a été effectuée), le vecteur $\vec{u}$ est indéterminé, c'est-à-dire que tous les vecteurs $\vec{u}$ sont acceptables, et $q = [1 \ 0 \ 0 \ 0]^T$.

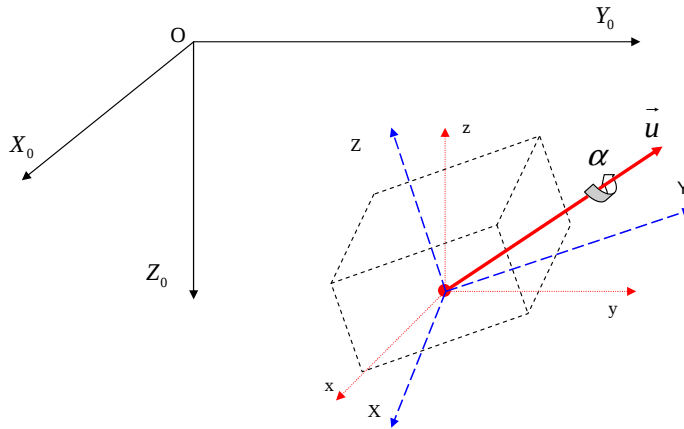


Figure 2.4 – Quaternion

La multiplication de deux quaternions quelconques $q_1 = [q_{10} \quad \vec{q}_1^T]^T$ et $q_2 = [q_{20} \quad \vec{q}_2^T]^T$ est définie par :

$$q_1 \otimes q_2 = \begin{pmatrix} q_{10} & -\vec{q}_1^T \\ \vec{q}_1 & I_3 q_{10} + [\vec{q}_1^\times] \end{pmatrix} \cdot \begin{pmatrix} q_{20} \\ \vec{q}_2 \end{pmatrix} \quad (2.1.19)$$

où

$$[\vec{q}_1^\times] = \begin{pmatrix} 0 & q_{13} & -q_{12} \\ -q_{13} & 0 & q_{11} \\ q_{12} & -q_{11} & 0 \end{pmatrix} \quad (2.1.20)$$

Grâce à l'algèbre des quaternions [16], des rotations finies dans l'espace peuvent être traitées d'une façon simple et élégante. Soient b_q et r_q les quaternions imaginaire associés aux vecteurs $\vec{b}$ et $\vec{r}$, et définis par :

$$b_q = [0 \ \vec{b}^T]^T \text{ et } r_q = [0 \ \vec{r}^T]^T \quad (2.1.21)$$

L'équation (2.1.2) exprimant le changement de coordonnées de N à B par la rotation $\mathcal{C}$ peut être donnée en termes de quaternion :

$$b_q = q^{-1} \otimes r_q \otimes q = \bar{q} \otimes r_q \otimes q \quad (2.1.22)$$

La matrice de rotation $\mathcal{C}$ devient alors :

$$\mathcal{C} = \mathcal{C}(q) = (q_0^2 - \vec{q}^T \vec{q})I_3 + 2(\vec{q} \vec{q}^T - q_0[\vec{q}^\times]) \quad (2.1.23)$$

Quelques manipulation conduisent à l'expression de $\mathcal{C}$ en fonction de q :

$$\mathcal{C}(q) = \begin{pmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_0q_1 + q_2q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{pmatrix} \quad (2.1.24)$$

ce qui donne :

$$\vec{b} = \mathcal{C}(q) \cdot \vec{r} \quad (2.1.25)$$

Remarque 2. Notons qu'une attitude dans l'espace peut être représentée par deux quaternions q et $-q$ correspondant respectivement à des rotations d'angle α et $2\pi - \alpha$ autour des axes $\vec{u}$ et $-\vec{u}$ respectivement.

Soit $\vec{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ la vitesse angulaire du corps rigide dans le système de coordonnées B par rapport à N , exprimée dans B . Alors, l'équation cinématique (2.1.10) peut être exprimée avec le quaternion unitaire q par :

$$\begin{pmatrix} \dot{q}_0 \\ \dot{\vec{q}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -\vec{q}^T \\ I_3 q_0 + [\vec{q}^\times] \end{pmatrix} \vec{\omega} \quad (2.1.26)$$

$$\dot{q} = \frac{1}{2} \Xi(q) \vec{\omega} = \frac{1}{2} \Omega(\vec{\omega}) q$$

où

$$\left\{ \begin{array}{l} \Xi(q) = \begin{pmatrix} -\vec{q}^T \\ I_3 q_0 + [\vec{q}^\times] \end{pmatrix} \\ \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \end{array} \right. \quad (2.1.27)$$

Remarque 3. En dehors du quaternion unitaire, d'autres représentations peuvent être utilisées telles que par exemple les paramètres de Rodrigues. Ces représentations n'étant pas utilisées dans le cadre de ce travail, elles ne sont pas présentées ici. Le lecteur pourra consulter [26] pour un exposé de ces différentes représentations.

2.1.3 Erreur d'attitude

Soient deux attitudes différentes d'un corps rigide données respectivement par les matrices de rotation $\mathcal{C}_1$ et $\mathcal{C}_2$. Les coordonnées associées sont :

$$\left\{ \begin{array}{l} \vec{b}_1 = \mathcal{C}_1 \cdot \vec{r} \\ \vec{b}_2 = \mathcal{C}_2 \cdot \vec{r} \end{array} \right. \quad (2.1.28)$$

L'attitude relative entre ces deux attitudes peut être calculée à partir de la matrice d'erreur :

$$\mathcal{C}_e = \mathcal{C}_2^{-1} \cdot \mathcal{C}_1 = \mathcal{C}_2^T \cdot \mathcal{C}_1 \quad (2.1.29)$$

où $\mathcal{C}_e$ représente un opérateur d'attitude qui fait tourner $\vec{b}_2$ en $\vec{b}_1$. $\mathcal{C}_e$ est utilisé dans le cadre de l'estimation et de la commande d'attitude comme une mesure de **l'erreur d'attitude**. Dans ce contexte, soit $\mathcal{C}_d = \mathcal{C}_1$ l'attitude désirée du corps rigide et $\mathcal{C}_r = \mathcal{C}_2$ l'attitude réelle de ce même corps. Alors, la matrice d'erreur d'attitude est obtenue par le calcul de :

$$\mathcal{C}_e = \mathcal{C}_r^{-1} \cdot \mathcal{C}_d \quad (2.1.30)$$

Lorsque $\mathcal{C}_1 = \mathcal{C}_2$ (attitudes identiques), l'erreur d'attitude vaut zéro et $\mathcal{C}_e = I_3$. L'attitude relative (ou erreur d'attitude) peut être également estimée en utilisant les quaternions unitaires q_1 et q_2 associés respectivement à $\mathcal{C}_1$ et $\mathcal{C}_2$ par :

$$\begin{aligned} q_e &= q_1^{-1} \otimes q_2 = \begin{pmatrix} q_{10} & \vec{q}_1^T \\ -\vec{q}_1 & I_3 q_{10} - [\vec{q}_1^\times] \end{pmatrix} \cdot \begin{pmatrix} q_{20} \\ \vec{q}_2 \end{pmatrix} \\ &= \bar{q}_1 \otimes q_2 \end{aligned} \quad (2.1.31)$$

Le quaternion q_e , lorsque l'erreur d'attitude est petite, peut être remplacé par une mesure d'erreur donnée par :

$$\beta_e = 2|\arccos(q_{e0})| \quad (2.1.32)$$

Notons que lorsque l'erreur d'attitude vaut zéro, on a $q_e = [\pm 1 \ 0]^T$, $\beta_e = 0$.

2.1.4 Commentaire sur la représentation d'attitude

On vient de présenter différentes représentations de l'attitude. Dans toutes ces représentations, on obtient des équations non-linéaires.

Le quaternion unitaire permet de s'affranchir des singularités géométriques. Cette représentation est très utilisée dans les problèmes d'estimation d'attitude [26], [10]. Les expressions de la matrice $\mathcal{C}(q)$ et de l'équation dynamique sont simples, ce qui limite le coût calcul (voir les chapitres suivants). On doit néanmoins satisfaire la contrainte $\|\vec{q}\| = 1$. Comme nous le verrons ultérieurement, cette contrainte est difficile à respecter dans le cadre du calcul ensembliste.

L'avantage des angles de Cardan est que les trois paramètres sont indépendants. Cependant, le problème du *Gimbal lock* limite leur utilisation à des applications pour lesquelles ce problème ne sera pas rencontré. Dans notre travail, on utilisera les angles de Cardan en limitant les angles "atteignables" par le corps rigide.

Ces deux représentations seront considérées dans ce travail et nous proposerons des algorithmes d'estimation de l'attitude ou de diagnostic de certains capteurs de l'IMU avec des outils ensemblistes.

2.2 IMU et leur mesures

Depuis les années 1950, il existe de nombreuses applications des centrales d'attitude (IMU - Inertial Measurement Unit) dans les domaines aussi variés que l'aéronautique, l'aérospatiale, la commande et la robotique. Ces applications nécessitent généralement l'estimation de l'attitude d'un corps rigide. Il existe plusieurs systèmes d'estimation d'attitude (optique, IMU, Ultra Large Bande, GPS, etc. [9]), chaque système ayant ses avantages et inconvénients, et étant dédié à un domaine d'applications.

Nous nous intéressons ici à une IMU qui contient différentes modalités de mesure, typiquement, des accéléromètres (A), des magnétomètres (M), des gyromètres (G). Les types de capteurs que nous utiliserons dépendront de l'application visée, en particulier, du caractère "quasi-statique" du mouvement, notion que nous définirons ci-après. Un estimateur d'attitude ou un observateur d'attitude est utilisé pour agréger l'information fournie par ces différentes modalités de mesure. Dans notre étude, une IMU est un capteur qui contient un triaxe de gyromètres, un triaxe d'accéléromètres et un triaxe de magnétomètres. Grâce à l'intégration des composants sur silicium, ces capteurs peuvent être intégrés dans un même composant électronique; on a alors un capteur

MEMS (Micro-Electro-Mechanical System). Une IMU réalisée avec de tels capteurs est de faible dimension, et d'un coût "raisonnable". Le capteur *MAG3* de chez MEMsense [30] est un exemple d'un tel capteur. Son coût est d'environ 900€ et ses dimensions sont de $18.288\text{ mm} \times 18.288\text{ mm} \times 12.236\text{ mm}$ (voir la figure 2.5 sur laquelle on donne sa photographie). Les caractéristiques de ce capteur seront utilisées pour simuler les données que nous utiliserons dans notre étude. On présente maintenant succinctement le principe

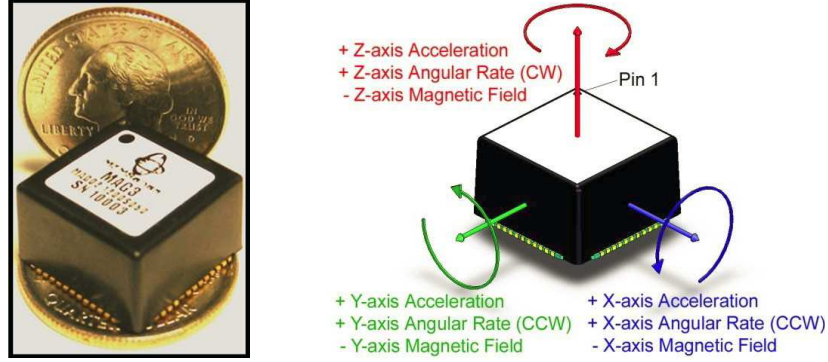


Figure 2.5 – Principe du capteur *MAG3* chez MEMsense [30]

de fonctionnement des capteurs utilisés dans notre étude, et on donne leur équation de fonctionnement.

2.2.1 Accéléromètre

Un accéléromètre [81] est un capteur qui, fixé sur un mobile, permet de mesurer l'accélération de ce dernier. Dans son principe physique, un accéléromètre peut être vu comme une masse M attachée à un ressort qui est lui-même attaché à un repère fixe (voir la figure 2.6).

Le principe de tout accéléromètre est le suivant : on a $\vec{F} = m\vec{a}$ où $\vec{F}$ est la force, M la

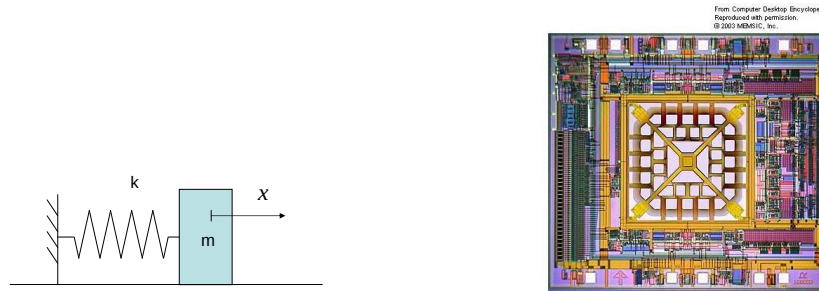


Figure 2.6 – (gauche) Principe de l'accéléromètre, (droite) Exemple d'accéléromètre MEMS (MEMSIC Inc)

masse d'épreuve et $\vec{a}$ l'accélération. D'après la loi fondamentale de la dynamique, nous

avons $F = ma = -kx$. Donc, en observant le déplacement x , il est possible d'en déduire l'accélération $a = -\frac{kx}{m}$. Donc une mesure de l'accélération consiste en réalité à mesurer le déplacement de la masse sismique attachée au ressort.

Remarque 4. $\vec{a}$ est un vecteur colinéaire à l'axe de coordonnées Ox . Il faut donc utiliser un accéléromètre tri-axes pour un mouvement dans un espace à trois dimensions. Dans ce cas, le capteur est également sensible au champ de gravité $\vec{g}$.

Les principaux paramètres d'un accéléromètre sont l'étendue de mesure généralement donnée en g (où g est l'accélération de la pesanteur au niveau de la mer pour une latitude de 45° , $g = 9.80665m/s^2$), la bande passante (Hz), la sensibilité (mV/g), la gamme de température d'utilisation ($^\circ C$), la précision, la masse du capteur, le nombre d'axes (1 à 3 axes), la construction mécanique, l'électronique intégrée. La principale raison de bruit existant dans la mesure fournie par un accéléromètre est due au mouvement Brownien. Le bruit résultant TNEA (Total Noise Equivalent Acceleration) est donné en $m/s^2/\sqrt{Hz}$.

Différents accéléromètres intégrés sont disponibles sur le marché. Leurs caractéristiques dépendent de leur technologie et de leur procédé de fabrication. Différents types de transducteurs peuvent être utilisés pour la construction des accéléromètres. Les plus utilisés sont à détection piézoélectrique, à détection capacitive, à jauge de contrainte/extensométrie, à poutre vibrante, à ondes de surface et à détection optique [12]. Les caractéristiques de l'accéléromètre intégré dans la centrale inertielle *MAG3* de chez MEMsense et qui sera utilisé dans notre étude sont résumés dans le tableau 2.1.

Paramètres	Caractéristiques
Étendue de mesure	$\pm 2 g$
Bande passante	$0-50Hz$
Précision	$< 4\mu g$
Bruit (X et Z)	$92\mu g/\sqrt{Hz}$
Bruit (Y)	$110\mu g/\sqrt{Hz}$
Gamme de température d'utilisation	$0^\circ C$ à $70^\circ C$

Tableau 2.1 – Caractéristiques de l'accéléromètre triaxe implanté dans la centrale d'attitude *MAG3* de MEMsense [30]

Modèle de mesure de l'accéléromètre. Le signal de sortie d'un accéléromètre est modélisé par :

$$\vec{b}_A = (\vec{a} - \vec{g}) + \vec{\eta}_A \quad (2.2.1)$$

où $\vec{g}$ est le champ de gravité, $\vec{a}$ est l'accélération propre et $\vec{\eta}_A$ est le bruit de mesure. Si les trois accéléromètres sont montés en triaxe orthogonal avec leurs axes sensibles qui

coïncident avec les principaux axes d'inertie du solide, le vecteur de mesure de l'accéléromètre triaxe est donné par :

$$\vec{b}_A = \mathcal{C} \cdot \vec{r}_A + \vec{\eta}_A \quad (2.2.2)$$

où $\vec{r}_A = (\vec{a} - \vec{g})$, $\vec{g} = [0 \ 0 \ g]^T$ et $\vec{a} \in \mathbb{R}^3$ représentent respectivement les vecteurs de gravité et d'accélération propre du corps, donnés dans le système de coordonnées inertiel N . $\vec{\eta}_A \in \mathbb{R}^3$ est le vecteur de bruits de mesure, supposés indépendants, blancs et gaussiens, à moyenne nulle. La matrice $\mathcal{C}$ est la matrice de rotation associée au quaternion q [16] ou aux angles de Cardan.

Remarque 5. Si l'accélération propre $\vec{a}$ est faible par rapport à la gravité, $\|\vec{a}\| \ll \|\vec{g}\|$, les mesures des accéléromètres donnent une projection du vecteur de gravité dans le repère attaché au corps rigide B . En conséquence, $\vec{r}_A \approx -\vec{g}$ et

$$\vec{b}_A = -\mathcal{C} \cdot \vec{g} + \vec{\eta}_A \quad (2.2.3)$$

Notons qu'en pratique $\vec{\eta}_A$ contient la somme du vecteur des bruits de mesure et des perturbations induites par des vibrations ou des accélérations parasites. Le vecteur $\vec{\eta}_A$ ne peut plus alors être considéré comme un vecteur de bruits blancs et gaussiens.

Remarque 6. On normalise habituellement le vecteur de gravité par g , et on obtient $\vec{g} = [0 \ 0 \ g]^T / \|\vec{g}\| = [0 \ 0 \ 1]^T$. Les mesures fournies par les accéléromètres seront données en g .

Remarque 7. Pour simplifier l'équation 2.2.3, le signe négatif de cette équation est transformé dans les composantes de vecteur $\vec{b}_A$ (les accéléromètres).

2.2.2 Gyromètre

Un gyromètre est un capteur qui permet d'effectuer une mesure de la vitesse angulaire. Les phénomènes physiques utilisés pour réaliser ce type de capteurs sont essentiellement les propriétés inertielles de la matière [82]. Notons qu'en français, on distingue le gyroscope qui donne une information sur la position angulaire et le gyromètre. Les gyromètres utilisent généralement plusieurs éléments vibrants qui produisent des signaux sinusoïdaux identiques et d'amplitude constante lorsque le capteur est au repos. Lorsque celui-ci est soumis à un mouvement de rotation, les forces de Coriolis induisent une variation de l'amplitude de ces signaux. Les principales caractéristiques d'un gyromètre sont la résolution, le biais, la sortie à vitesse nulle ZRO (Zero-Rate Output) et le facteur d'échelle.

Le tableau 2.2 donne les caractéristiques des gyromètres utilisés dans la centrale inertielle *MAG3* de chez MEMsense qui sera utilisée lors de nos expérimentations.

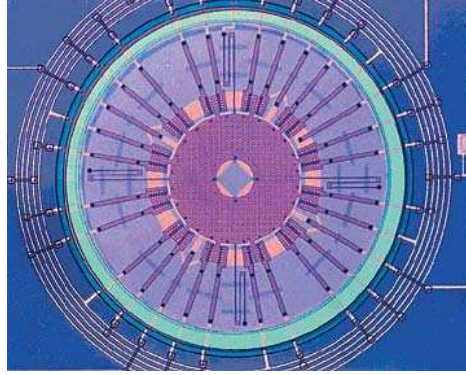


Figure 2.7 – Exemple de gyromètre MEMS de chez Berkeley Sensors and Actuators Center [70]

Paramètres	Caractéristiques
Etendue de mesure	± 150 ($^{\circ}/sec$)
Dérive	< 0.01 ($^{\circ}/h$)
Bande passante	0-50 (Hz)
Densité de Bruit	0.1 ($^{\circ}/sec/\sqrt{Hz}$)
Gamme de température	$0^{\circ}C$ à $70^{\circ}C$

Tableau 2.2 – Caractéristiques du gyromètre implanté dans la centrale d'attitude *MAG3* de MEMsense [30]

Modèle de mesure du gyromètre. Le signal de sortie du gyromètre est modélisé par la somme de trois grandeurs :

$$\omega_G = \omega + \nu + \eta_G \quad (2.2.4)$$

où ω est la vitesse réelle de rotation, ν est une fonction lentement variable dans le temps (biais lentement variable) et η_G est un bruit blanc.

La dynamique du biais ν est traditionnellement modélisée par un processus de Gauss-Markov [26] décrit par :

$$\dot{\nu} = -\frac{1}{\tau}\nu + \eta_{\nu} \quad (2.2.5)$$

Si trois gyromètres sont montés en triaxe orthogonal dans un solide, tels que leurs axes sensibles coïncident avec les axes principaux d'inertie du solide, la sortie des gyromètres et l'évolution du biais deviennent :

$$\vec{\omega}_G = \vec{\omega} + \vec{\nu} + \vec{\eta}_G \quad (2.2.6)$$

$$\dot{\vec{\nu}} = -T^{-1}\vec{\nu} + \vec{\eta}_{\nu} \quad (2.2.7)$$

où les composantes des vecteur $\vec{\eta}_G$ et $\vec{\eta}_{\nu} \in \mathbb{R}^3$ sont supposées être des bruits blancs gaussiens et $T = \tau I_3$ est une matrice diagonale de constantes de temps. En intégrant

l'équation cinématique (2.2.6) à partir d'une condition initiale, il est possible d'obtenir la variation de l'attitude au cours du temps. Si les sorties des gyromètres sont entachées de biais, celui-ci doit être estimé afin de limiter la dérive de l'angle obtenu par intégration.

Remarque 8. Notons que pour les capteurs récents et pour une utilisation de quelques heures en continu, la dérive $\vec{v}$ peut être négligée.

2.2.3 Magnétomètre

Parmi les instruments de navigation qui permettent de s'orienter par rapport à un système de référence absolu, nous trouvons le compas magnétique. Cet instrument utilise le champ magnétique terrestre comme référence. En conséquence, il permet de suivre le cap [83]. Le champ magnétique terrestre trouve son origine au centre de la terre où les atomes du magma portés à haute température ($5000^{\circ}C$) sont "dépouillés" d'une partie de leurs électrons et sont donc ionisés. A la surface de la terre, le champ magnétique varie d'un lieu à l'autre. Cette variation est à l'origine des mouvements de l'aiguille de la boussole lorsque l'on se déplace. Le Champ Géomagnétique International de Référence

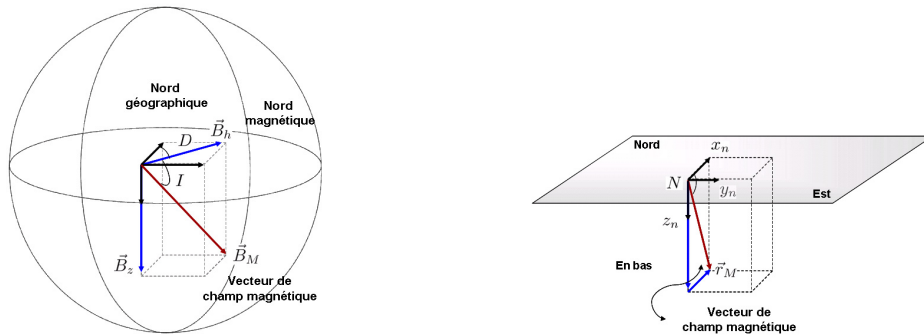


Figure 2.8 – (a) Champ magnétique terrestre, (b) Champ magnétique dans le repère NED

(IGRF) et le Modèle Magnétique Mondial (WMM) sont les deux modèles les plus utilisés pour modéliser le champ magnétique, notamment pour la navigation [29]. Ils donnent le champ magnétique en un lieu donné, et ce champ peut être considéré constant dans cette zone. Les composantes du champ magnétique terrestre (voir la figure 2.8 (a)) sont :

- $\vec{B}_h$, la composante horizontale du champ magnétique ;
- $\vec{B}_z$, la composante verticale du champ magnétique. Par convention, $\vec{B}$ est positif vers le bas.

Le champ magnétique $\vec{B}_M$ est la résultante. Sa norme représente l'intensité totale du champ magnétique. La composante horizontale $\vec{B}_h$ s'annule aux pôles magnétiques. L'angle I formé par $\vec{B}_M$ et le plan horizontal est appelé l'inclinaison magnétique. Cet angle augmente lorsqu'on se rapproche des pôles. L'angle D entre la direction du pôle nord géographique et du pôle nord magnétique est appelé déclinaison. Dans ce document,

le champ magnétique est modélisé par le vecteur $\vec{r}_M = [r_{M_x} \ 0 \ r_{M_z}]^T$ dans le repère NED (voir la figure 2.8 (b)). La variation d'orientation du champ magnétique terrestre dans un système de coordonnées mobile peut être déterminée à partir de l'évolution des trois composantes cartésiennes du champ dans ce système de coordonnées.

Il existe plusieurs méthodes de mesure des champs magnétiques et différents types de capteurs magnétiques ont été développés. En pratique, les capteurs à effets magnéto-résistifs répondent au cahier des charges des applications embarquées en raison de leur performance, de leur faible coût, de leur petite taille et leur facilité de mise en œuvre. Le tableau 2.3 résume les caractéristiques du magnétomètre implanté dans la centrale d'attitude *MAG3* de chez MEMsense [30] et qui sera utilisée dans ce rapport.

Paramètres	Caractéristiques
Étendue de mesure	$\pm 1.9 \text{ Gauss}$
Bande passante	$0\text{-}50\text{Hz}$
Densité de bruit	$0.14 \text{ mGauss}/\sqrt{\text{Hz}}$
Gamme de température	0°C à 70°C

Tableau 2.3 – Caractéristiques du magnétomètre implanté dans la centrale d'attitude *MAG3* [30]

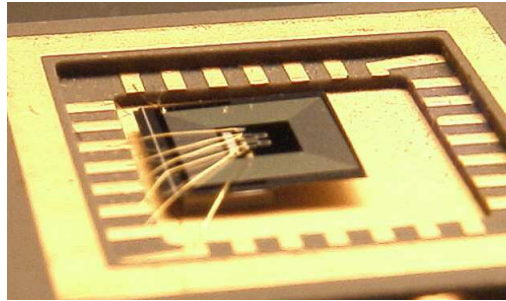


Figure 2.9 – Exemple d'un magnétomètre en technologie MEMS

Modèle de mesure pour le magnétomètre. Le signal de sortie d'un magnétomètre est modélisé par la somme d'une projection du champ magnétique terrestre $\vec{r}_M$ dans le repère B et un terme de bruit η_M considéré blanc et gaussien à moyenne nulle. Si trois magnétomètres sont montés en triaxe orthogonal, la sortie du magnétomètre triaxe dans le système de coordonnées mobile B peut être écrit sous la forme d'un vecteur de mesures :

$$\vec{b}_M = \mathcal{C} \cdot \vec{r}_M + \vec{\eta}_M \quad (2.2.8)$$

La matrice $\mathcal{C}$ est la matrice de rotation associée au quaternion q ou aux angles de Cardan. De plus, dans notre cas, le vecteur de champ magnétique correspond à un angle d'inclinaison $I = 60^\circ$. Sa valeur normalisée est $\vec{r}_M = [r_{M_x} \ 0 \ r_{M_z}]^T = [\frac{1}{2} \ 0 \ \frac{\sqrt{3}}{2}]^T$.

Remarque 9. À titre d'information l'intensité approximative du champ magnétique à Grenoble est $\|\vec{r}_M\| = 47 \text{ mT} = 400 \text{ mG}$. Les vecteurs de mesure seront normalisés par cette valeur.

2.2.4 Modèle des mesures de la centrale d'attitude

L'objet de notre travail est d'estimer l'attitude d'un corps rigide mobile à partir :

- d'un magnétomètre triaxe ;
- d'un accéléromètre triaxe ;
- d'un gyromètre triaxe.

En utilisant les modèles précédents, nous obtenons :

$$\begin{cases} \vec{b}_M = \mathcal{C} \cdot \vec{r}_M + \vec{\eta}_M \\ \vec{b}_A = \mathcal{C} \cdot \vec{g} + \vec{\eta}_A \\ \vec{\omega}_G = \vec{\omega} + \vec{\eta}_G \end{cases} \quad (2.2.9)$$

où $\vec{g} = [0 \ 0 \ 1]^T$, $\vec{r}_M = [\frac{1}{2} \ 0 \ \frac{\sqrt{3}}{2}]^T$, $\vec{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$. En utilisant le quaternion unitaire pour représenter l'attitude, on a :

$$\begin{cases} \mathcal{C} = \mathcal{C}(q) = \begin{pmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_0q_1 + q_2q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{pmatrix} \\ \dot{q} = \frac{1}{2}\Xi(q)\vec{\omega} = \frac{1}{2}\Omega(\vec{\omega})q \\ \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \end{cases} \quad (2.2.10)$$

L'utilisation des angles de Cardan conduit à :

$$\begin{cases} \mathcal{C} = \mathcal{C}(\varphi, \theta, \psi) = \begin{pmatrix} c\varphi c\theta & s\varphi c\theta & -s\theta \\ -s\varphi c\theta + c\varphi s\theta s\psi & c\varphi c\psi + s\varphi s\theta s\psi & c\theta s\psi \\ s\varphi s\psi + c\varphi s\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\theta c\psi \end{pmatrix} \\ \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan \theta \sin \varphi & \tan \theta \cos \varphi \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \end{cases} \quad (2.2.11)$$

Notre problème d'estimation de l'attitude se résume donc à l'estimation des 4 composantes du quaternion (avec la contrainte de norme unité) ou des 3 angles de Cardan à

partir du système d'équations (2.2.9)-(2.2.10) ou (2.2.9)-(2.2.11) en utilisant des outils ensemblistes. Les bruits ne sont plus supposés blancs, centrés et gaussiens, mais *bornés*. Les grandeurs (mesures et paramètres à estimer) sont représentées par des intervalles. Ce choix permet en outre de prendre en compte facilement des perturbations magnétiques ou de petites accélérations propres en les incluant dans le modèle de bruit "inconnu mais borné".

2.3 Calcul ensembliste

Dans cette partie, on présente les outils numériques qui serviront tout au long de notre étude. On s'intéresse tout d'abord au calcul par intervalles [38]. On présente ensuite l'algorithme SIVIA qui sera utilisé comme référence pour comparer nos résultats en termes de taille de l'ensemble solution et de temps calcul. Le problème de satisfaction de contraintes (CSP) est ensuite exposé. On résume enfin différentes approches dérivées de SIVIA.

2.3.1 Arithmétique par intervalles

Avant d'aborder les outils utilisés dans ce travail, on rappelle quelques concepts de base [38] de l'arithmétique par intervalles.

2.3.1.1 Calcul par intervalles

Un intervalle $[x]$ de bornes $\underline{x}$ et $\bar{x}$ réels est un sous-ensemble connexe, borné, fermé tel que [38] :

$$[x] = [\underline{x}, \bar{x}] \triangleq \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\} \quad (2.3.1)$$

où

$$\begin{cases} \bar{x} = ub([x]) \triangleq \sup\{b \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall x \in [x], x \leq b\} \\ \underline{x} = lb([x]) \triangleq \inf\{a \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall x \in [x], a \leq x\} \end{cases} \quad (2.3.2)$$

On note $\mathbb{IR}$ l'ensemble des intervalles $[x] \in \mathbb{IR}$. La largeur de l'intervalle $[x]$ est donnée par :

$$\delta([x]) \triangleq |\bar{x} - \underline{x}|. \quad (2.3.3)$$

Le centre de l'intervalle $[x]$ est calculé avec :

$$mid([x]) \triangleq \frac{\underline{x} + \bar{x}}{2}. \quad (2.3.4)$$

L'intersection de deux intervalles $[x]$ et $[y]$ est obtenu par :

$$[x] \cap [y] \triangleq \{z \in \mathbb{R} \mid z \in [x] \text{ et } z \in [y]\} \quad (2.3.5)$$

L'union de deux intervalles $[x]$ et $[y]$ est donnée par :

$$[x] \cup [y] \triangleq \{z \in \mathbb{R} | z \in [x] \text{ ou } z \in [y]\} \quad (2.3.6)$$

tandis que $[x] \setminus [y]$ est défini par :

$$[x] \setminus [y] = [[x] \setminus [y]] = \{x \in [x] | x \notin [y]\} \quad (2.3.7)$$

Les opérations de base sur les intervalles sont récapitulées dans le tableau 2.4.

$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$
$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$
$a * [x] = [a\underline{x}, a\bar{x}] \text{ si } a \geq 0$
$a * [x] = [a\bar{x}, a\underline{x}] \text{ si } a \leq 0$
$[x] * [y] = [\min\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}]$
$\frac{1}{[y]} = [\frac{1}{\bar{y}}, \frac{1}{\underline{y}}] \text{ si } 0 \notin [y]$
$\frac{1}{[y]} = [\frac{1}{\bar{y}}, +\infty] \text{ si } \underline{y} = 0 \text{ et } \bar{y} > 0$
$\frac{1}{[y]} = [-\infty, \frac{1}{\bar{y}}] \text{ si } \underline{y} < 0 \text{ et } \bar{y} = 0$
$\frac{1}{[y]} = [-\infty, +\infty] \text{ si } \underline{y} = 0 \text{ et } \bar{y} = 0$
$\frac{1}{[y]} = \emptyset \text{ si } [y] = [0, 0]$
$[\frac{x}{y}] = [x] * (\frac{1}{[y]})$

Tableau 2.4 – Opérations élémentaires en calcul par intervalles

2.3.1.2 Fonction d'inclusion

Soit $f : \mathbb{R} \mapsto \mathbb{R}$ une fonction dite élémentaire (tan, cos, sin, exp, log, sqr, etc.). Une fonction d'inclusion est définie par une extension intervalle de $[f]$ telle que [38]

$$\begin{aligned} [f] : \mathbb{IR} &\mapsto \mathbb{IR} \\ [x] &\mapsto [f]([x]) = \{f(x), x \in [x]\} \end{aligned} \quad (2.3.8)$$

où $\mathbb{IR}$ correspond à l'ensemble des intervalles de $\mathbb{R}$. La fonction d'inclusion $[f]([x])$ contient l'ensemble des valeurs de $f(x)$ où $x \in [x]$. La fonction d'inclusion d'une fonction continue et monotone f est donnée par :

$$[f]([x]) = [\min(f(\underline{x}), f(\bar{x})), \max(f(\underline{x}), f(\bar{x}))] \quad (2.3.9)$$

On a reporté sur la figure 2.10 un exemple où $[f^*]([x])$ est la fonction qui donne le domaine de taille minimum. La taille de l'ensemble résultat dépend de la fonction d'inclusion et cette fonction n'est **pas unique**. En particulier, il faut tenir compte des multi-occurrences de chaque variable et essayer de les minimiser [38] pour garantir que le pavé

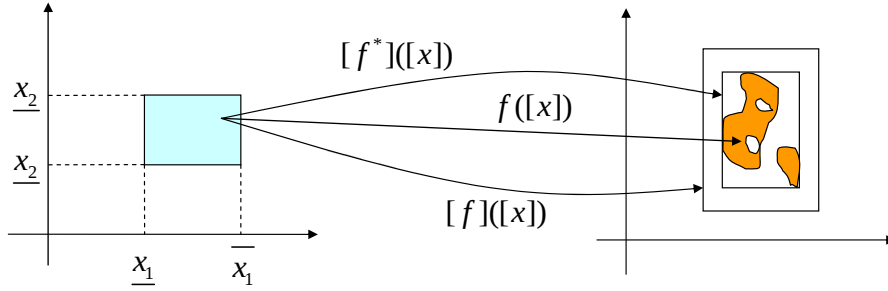


Figure 2.10 – Différentes fonctions d'inclusion

solution soit de taille minimale. Pour comprendre ce phénomène, considérons l'exemple ci-dessous. Soient différentes formulation d'une même fonction mathématique :

$$\begin{aligned}
 f_1(x) &= x^2 + 2x \\
 f_2(x) &= (x + 1)^2 - 1 \\
 f_3(x) &= x(x + 2) \\
 f_4(x) &= x * x + 2x
 \end{aligned}
 \tag{2.3.10}$$

On détermine l'ensemble englobant l'ensemble des solutions (calcul ensembliste) avec $x = [-2, 2]$ pour les différentes formulations f_1 à f_4 , ce qui donne :

$$\begin{aligned}
 f_1(x) &= [0, \max\{(-2)^2; (2)^2\}] + [2 * (-2), 2 * 2] \\
 &= [-4, 8] \\
 f_2(x) &= ([-2, 2] + 1)^2 - 1 \\
 &= ([-1, 3])^2 - 1 = [0, \max\{(-1)^2; (3)^2\}] - 1 \\
 &= [-1, 8] \\
 f_3(x) &= [-2, 2] * ([-2, 2] + 2) \\
 &= [-2, 2] * [0, 4] \\
 &= [\min\{(-2) * 0; (-2) * 4; 2 * 0; 2 * 4\}, \max\{(-2) * 0; (-2) * 4; 2 * 0; 2 * 4\}] \\
 &= [-8, 8] \\
 f_4(x) &= [\min\{(-2) * (-2); (-2) * 2; 2 * (-2); 2 * 2\}, \\
 &\quad \max\{(-2) * (-2); (-2) * 2; 2 * (-2); 2 * 2\}] + [2 * (-2), 2 * 2] \\
 &= [-4, 4] + [-4, 4] \\
 &= [-8, 8]
 \end{aligned}
 \tag{2.3.11}$$

Pour améliorer (c'est-à-dire minimiser) la taille de l'ensemble englobant l'ensemble des solutions (idéalement, on cherche à déterminer l'ensemble des solutions, sans pessimisme), on peut exploiter différentes techniques qui tentent de minimiser le nombre d'occurrences des variables. Ainsi, on peut [38] utiliser :

1. la fonction d'inclusion centrale ;
2. la fonction d'inclusion de Taylor.

Ces deux techniques sont résumées dans le tableau 2.5, ainsi que le résultat obtenu pour l'exemple que l'on vient de traiter. On voit d'ailleurs que les deux démarches proposées

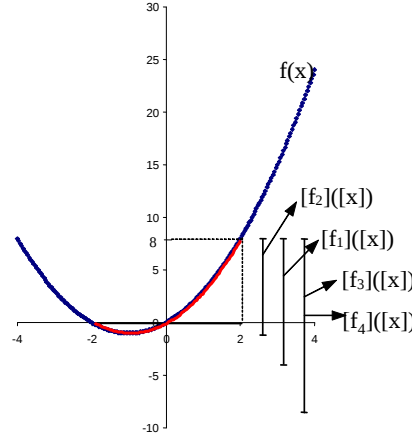


Figure 2.11 – Effet des multi-occurrences

ne conduisent pas à la même fonction d'inclusion et que les ensembles englobants sont différents. On voit donc que le choix de la fonction d'inclusion pour un problème donné est crucial afin de ne pas introduire "trop" de pessimisme (effet de "wrapping" à minimiser). Dans certains cas, on arrive à diminuer l'ensemble solution en travaillant sur

Méthode	Fonction mathématique	Ensemble résultat pour l'exemple ci-dessus
Fonction d'inclusion centrale	$[f_C](x) = f(m) + \sum_{i=1}^n [g_i]([x_1], [x_2], \dots, [x_i], m_{i+1}, \dots, m_n)([x_i] - m_i)$ $m = mid([x]), g_i = \delta f / \delta x_i$	$[-12, 12]$
Fonction d'inclusion de Taylor	$[f_T](x) = f(m) + g^T(m)([x] - m) + \frac{1}{2}([x] - m)^T [H]([x])([x] - m)$ $m = mid([x]), g_i = \delta f / \delta x_i, H \text{ Hessienne} \begin{cases} \delta^2 f / \delta x_i^2 & j = i \\ 2\delta^2 f / \delta x_i x_j & j \neq i \\ 0 & \end{cases}$	$[-4, 8]$

Tableau 2.5 – Détermination de la fonction d'inclusion par des techniques "systématiques" et ensemble solution associé

deux intervalles de données obtenus par la bisection de l'intervalle initial $[x]$.

2.3.1.3 Bisection

Soit un pavé $[x] \subset \mathbb{IR}^n : [x] = [x_1] \times [x_2] \times \dots \times [x_n]$. Définissons les pavés :

$$\begin{cases} L[x] = [x_1] \times [x_2] \times \dots \times [\underline{x}_j, x_{0j}] \times \dots \times [x_n] \\ R[x] = [x_1] \times [x_2] \times \dots \times [x_{0j}, \bar{x}_j] \times \dots \times [x_n] \\ j \in [1 : n] \end{cases} \quad (2.3.12)$$

Alors, l'opération qui consiste à obtenir les pavés $L[x]$ et $R[x]$ est appelée **bisection**

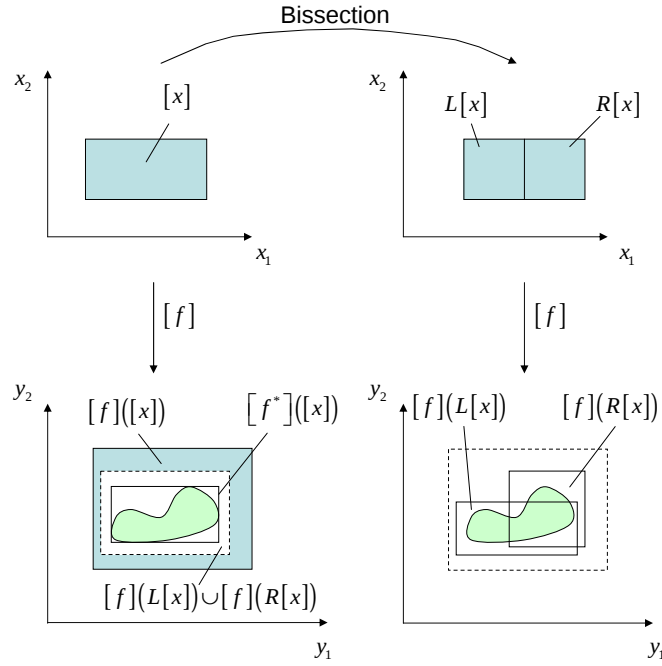


Figure 2.12 – Principe d'une fonction d'inclusion par bisection

[38]. Différents choix de l'intervalle j à bissecter peuvent être utilisés. Par exemple, on peut choisir :

- l'indice de l'opération de bisection j est donné par [38]

$$: j = \arg(\max_{i=1,\dots,n} (\delta([x_i]))); \quad (2.3.13)$$

- l'indice de l'opération de bisection j est limité à quelque valeur en analysant un graphe (ou structure de l'arbre) du problème [34] ;
- l'indice est obtenu à partir d'une fonction quelconque. Hansen a proposé de choisir j par [19] :

$$j = \arg(\max_{i=1,\dots,n} (\delta([f'_i]([x])) \cdot \delta([x_i]))); \quad (2.3.14)$$

- on peut chercher à minimiser la taille de $[f]([x])$ en utilisant la fonction d'inclusion centrée $[f_C]([x])$ [65] :

$$j = \arg(\max_{i=1,\dots,n} (\delta([f'_i]([x]))([x_i - m_i])))) \text{ avec } m_i = \text{mid}([x_i]); \quad (2.3.15)$$

- dans certains cas, on peut en choisissant j réduire le pessimisme de $([f]([x]) - f([x]))$ dû aux erreurs d'arrondi [74] :

$$j = \arg(\max_{i=1,\dots,n} (\delta(i)))$$

$$\text{où : } \delta(i) = \begin{cases} \delta([x_i]) & \text{si } 0 \in [x_i] \\ \delta([x_i]) / \min\{|x_i| \mid x_i \in [x_i]\} & \text{si } 0 \notin [x_i] \end{cases} \quad (2.3.16)$$

$SIVIA(in : [f], [Y], [x], \varepsilon; inout : \underline{X}, \overline{X})$
1.if $[f]([x]) \cap [Y] = \emptyset$ then return; 2.if $[f]([x]) \subset [Y]$ then $\{\underline{X} = \underline{X} \cup [x]; \overline{X} = \overline{X} \cup [x];$ return} 3.if $(\delta[x] < \varepsilon)$ then $\{\overline{X} = \overline{X} \cup [x];$ return} 4. $SIVIA([f], [Y], L[x], \varepsilon, \underline{X}, \overline{X}); SIVIA([f], [Y], R[x], \varepsilon, \underline{X}, \overline{X});$ où $\begin{cases} L[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [\underline{x}_j, (\underline{x}_j + \overline{x}_j)/2] \times \dots \times [\underline{x}_n, \overline{x}_n] \\ R[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [(\underline{x}_j + \overline{x}_j)/2, \overline{x}_j] \times \dots \times [\underline{x}_n, \overline{x}_n] \end{cases}$

Tableau 2.6 – Algorithme SIVIA

Remarque 10. Dans notre travail, nous choisirons l'intervalle sur lequel est réalisé la bisection avec l'équation (2.3.13).

Remarque 11. Soit une fonction d'inclusion $[f]$, grâce à la bisection de $[x]$, on a :

$$f([x]) \in [f^*]([x]) \subset [f](L([x])) \cup [f](R([x])) \subset [f]([x]) \quad (2.3.17)$$

L'opération de bisection a permis de développer un algorithme général de résolution du problème d'inversion ensembliste, nommé SIVIA. Les résultats obtenus avec cet algorithme serviront de référence pour analyser les performances des algorithmes que nous développerons au chapitre suivant.

2.3.2 SIVIA

L'algorithme SIVIA (Set Inverter Via Interval Analysis) [38] permet d'inverser un problème du type :

$$X = \{[x] \in \mathbb{IR}^n | [f]([x]) \subset [Y]\} = [f^{-1}]([Y]) \quad (2.3.18)$$

Pour tout ensemble $[Y] \subset \mathbb{IR}^m$ et pour toute fonction f admettant une fonction d'inclusion convergente $[f]([X])$, deux sous-pavages $\overline{X}$ et $\underline{X}$ tels que :

$$\underline{X} \subset X \subset \overline{X} \quad (2.3.19)$$

peuvent être obtenus avec l'algorithme SIVIA. Deux versions de SIVIA peuvent être développées, l'une basée sur un test d'inclusion, l'autre basée sur une fonction d'inclusion. On présente au tableau 2.6 la version de SIVIA basée sur une fonction d'inclusion. ε est la dimension minimale du pavé en deçà de laquelle le pavé n'est plus bisecté. Notons que, par essence, l'algorithme SIVIA est récursif. Il a été montré que SIVIA se termine avec moins de $(\frac{\delta([x](0)}{\varepsilon} + 1)^n$ bisections et que le temps calcul augmente exponentiellement avec la dimension de $[x]$ [47].

L'algorithme SIVIA sera utilisé dans notre travail d'estimation ensembliste de l'attitude. Il permet de déterminer l'ensemble des attitudes cohérentes avec le jeu de mesures données sous forme d'intervalles.

2.3.3 Problème de Satisfaction de Contraintes et contracteurs

Résoudre un problème de satisfaction de contraintes (Constraints Satisfaction Problem - CSP) consiste à trouver l'ensemble des solutions $[x]$ qui satisfait un ensemble de contraintes, généralement des contraintes égalité. On note :

$$H = \{f_i(x) = 0, i = 1 : n_f, x \in [x], x = (x_1, x_2, \dots, x_{n_x})^T\} \quad (2.3.20)$$

l'ensemble des contraintes. L'objectif est de calculer une approximation externe de l'ensemble $\mathcal{S}$ des solutions de l'équation (2.3.20). Les techniques de calcul utilisées exploitent des propriétés de "consistance". Ainsi on garantit que l'ensemble trouvé est bien une approximation externe de $\mathcal{S}$ et que cette approximation est calculée en temps polynomial [38]. Contracter H consiste en pratique à remplacer $[x]$ par $[x']$ tout en conservant, $\mathcal{S} \subset [x'] \subset [x]$. Un contracteur est un opérateur qui permet de réaliser cette contraction. Il existe un contracteur optimal qui permet d'obtenir l'ensemble $[x']$ de taille minimale. Dans la suite, un contracteur est noté C_{princ} , l'indice *princ* indiquant le principe sur lequel il s'appuie.

Dans le tableau 2.7 [38], on voit deux groupes de contracteurs, l'un dédié aux problèmes linéaires :

- C_{GE} , contracteur "Elimination de Gauss" ;
- C_{GS} , contracteur "Gauss-Seidel" ;
- C_{LP} , contracteur "Linéarisation Parallèle" ;
- C_{GSP} , contracteur "Gauss-Seidel avec Préconditionnement" ;
- C_{GEP} , contracteur "Elimination de Gauss avec Préconditionnement" ;

et l'autre aux problèmes non linéaires :

- C_K , contracteur de "Krawczyk" ;
- $C_{\uparrow\downarrow}$, contracteur "forward-backward" ;
- C_N , contracteur de "Newton" ;
- $C_{||}$, contracteur avec "Linéarisation Parallèle".

Notons que dans la plupart des méthodes non linéaires, on se ramène en fait à un problème linéaire. Il n'y a que $C_{\uparrow\downarrow}$ qui résout directement le problème non linéaire.

Linéaire	Non linéaire
$C_{GE}(Ap - b = 0, [A], [p], [b])$	$C_K(f(x) = 0, [x])$
$C_{GS}(Ap - b = 0, [A], [p], [b])$	$C_{\uparrow\downarrow}(f(x) = 0, [x])$
$C_{LP}(Ap - b = 0, [A], [p], [b])$	$C_N(f(x) = 0, [x])$
$C_{GSP}(Ap - b = 0, [A], [p], [b])$	$C_{ }(f(x) = 0, [x])$
$C_{GEP}(Ap - b = 0, [A], [p], [b])$	

Tableau 2.7 – Liste de différents contracteurs

Le problème d'estimation d'attitude étant intrinsèquement non linéaire, on donne

maintenant un rapide aperçu des différents contracteurs utilisables pour les problèmes non linéaires. Notons dès à présent que $C_{\uparrow\downarrow}$ sera utilisé dans la suite de ce travail.

2.3.3.1 Contracteur C_K (méthode de Krawczyk)

On s'intéresse au problème non linéaire :

$$H = \{f(x) = 0, x \in [x]\} \quad (2.3.21)$$

pour lequel $n_f = n_x$ (i.e. le nombre d'équations n_f est égal au nombre d'inconnues n_x) et f est différentiable. Puisque pour toute matrice inversible M on a :

$$f(x) = 0 \Leftrightarrow x - Mf(x) = x \quad (2.3.22)$$

La fonction $\Psi(x) = x - Mf(x)$ est un solveur de type point fixe pour H . La fonction d'inclusion centrée pour le contracteur Ψ est donnée par :

$$[\Psi]([x]) = \Psi(x_0) + [J_\Psi]([x]) * ([x] - x_0) \quad (2.3.23)$$

où $[J_\Psi]$ est une fonction d'inclusion pour le Jacobien de la fonction Ψ et $x_0 = \text{mid}([x])$ est le centre de l'intervalle sur la variable $[x]$. Alors, le contracteur de type point fixe de Krawczyk est obtenu, après remplacement de Ψ par $x - Mf(x)$:

$$C_K : [x] \mapsto [x] \cap (x_0 - Mf(x_0) + (I - M[J_f]([x])) * ([x] - x_0)) \quad (2.3.24)$$

où $[J_f]$ est la fonction d'inclusion du Jacobien de f . Notons que l'on choisit souvent $M = J_f^{-1}(x_0)$, ce qui peut être vu comme un préconditionnement.

Remarque 12. Le contracteur C_K n'a pas été choisi dans ce travail car $n_f > n_x$.

2.3.3.2 Contracteur C_N (méthode de Newton)

Le problème non linéaire traité est ici encore :

$$H = \{f(x) = 0, x \in [x]\} \quad (2.3.25)$$

pour lequel $n_f = n_x$ avec un solveur de type point fixe donné par $\Psi(x) = x - Mf(x)$. Si f est affine, $f(x) = Ax + b$, alors $\Psi(x) = x - M\{Ax + b\}$. En choisissant $M = A^{-1}$, la séquence $x_{k+1} = \Psi(x_k)$ converge vers la solution $x^* = A^{-1}b$ en un pas. Si f est non linéaire mais différentiable, elle peut être approximée par son développement de Taylor au premier ordre, ce qui conduit à un solveur approximé $\Psi(x) = x - J_f^{-1}(x)f(x)$. La fonction d'inclusion peut s'écrire :

$$[\Psi]([x]) = [x] - [J_\Psi]^{-1}([x]) * [f]([x]) \quad (2.3.26)$$

tandis que le contracteur de Newton est défini par :

$$C_N : [x] \longmapsto [x] \cap ([x] - [J_f]^{-1}([x])) * [f](x) \quad (2.3.27)$$

Notons que d'autres formes de la fonction d'inclusion peuvent être exploitées, comme par exemple une fonction d'inclusion centrée.

Remarque 13. C_N n'a pas été exploité car pour le problème d'estimation d'attitude, $n_f > n_x$.

2.3.3.3 Contracteur $C_{\parallel}$ (linéarisation parallèle)

On suppose que $n_f \neq n_x$. L'objectif est de prendre en compte toutes les contraintes en utilisant une approche "linéarisation parallèle". Le principe est de borner $f_i(x)$ sur $[x]$ par deux hyperplans parallèles. En conséquence, on a :

$$Ax + \underline{b} \leq f(x) \leq Ax + \bar{b} \quad (2.3.28)$$

Cet encadrement peut être obtenu via le théorème de la valeur moyenne sous l'hypothèse que f soit dérivable. Soit $x_0 \in [x]$ (on peut choisir le centre de l'intervalle par exemple). Alors $\exists \xi \in [x]$ tel que :

$$\begin{aligned} f(x) &= f(x_0) + J_f(\xi)(x - x_0) \\ &= f(x_0) + J_f(x_0)(x - x_0) + J_f(\xi)(x - x_0) - J_f(x_0)(x - x_0) \\ &= \underline{J_f(x_0)x} + \frac{\{f(x_0) - J_f(x_0)x_0 + (J_f(\xi) - J_f(x_0))(x - x_0)\}}{B} \\ &= A * x + \end{aligned} \quad (2.3.29)$$

avec $A = J_f(x_0)$ et $b = f(x_0) - Ax_0 + (J_f(\xi) - A)(x - x_0)$. A partir de l'équation (2.3.28), le contracteur $C_{\parallel}$ s'écrit :

$$C_{\parallel}(in \ f; \ inout[p]) \left| \begin{array}{l} x_0 = mid([x]) \\ A = J_f(x_0) \\ [b] = f(x_0) - Ax_0 + ([J_f]([x]) - A)(x - x_0) \\ C_{LP}(Ax - b = 0, x \in [x], b \in [b]) \end{array} \right. \quad (2.3.30)$$

où : C_{LP} est donné dans [38].

Remarque 14. Comme C_K et C_N , $C_{\parallel}$ ne peut pas être appliqué dans notre cas car $n_f > n_x$.

2.3.3.4 Contracteur $C_{\uparrow\downarrow}$ "Forward-backward"

Le contracteur Forward-backward ($\uparrow\downarrow$) [13] [46] est basé sur les techniques de propagation de contraintes [79] [17] [18]. Le contracteur permet de contracter le domaine du CSP $H = \{f(x) = 0, x \in [x]\}$ en prenant en compte chacune des n_f contraintes indépendamment $f_i(x_1, x_2, \dots, x_{n_x}), i = 1 : n_f$, n_f n'étant pas forcément égal à n_x [38]. En pratique, chaque contrainte $f_i(x_1, x_2, \dots, x_{n_x}), i = 1 : n_f$ est décomposée en une sé-

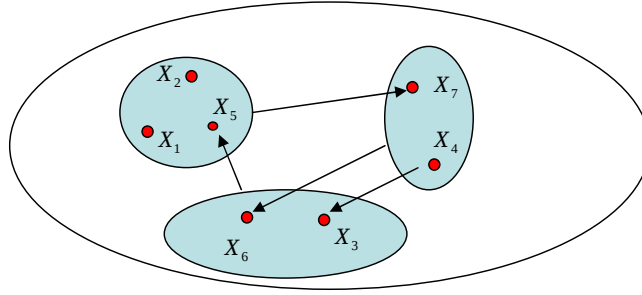


Figure 2.13 – Contracteur $C_{\uparrow\downarrow}$

quence d'opérations élémentaires. Il est ensuite possible de décomposer la contrainte en contraintes primitives impliquant un opérateur mathématique simple (+, -, *, /) ou une fonction simple (sin, exp, ...) [45]. On est par conséquent amené à définir des variables intermédiaires dont le domaine de définition est $] -\infty, +\infty[$. La méthode consiste donc à contracter H en contractant chacune des contraintes primitives jusqu'à ce que les contracteurs soient inefficaces. C'est exactement le principe de la propagation de contraintes introduite par [79]. La propagation $\uparrow\downarrow$ choisit l'ordre d'utilisation des contraintes primitives d'une manière optimale en fonction de la taille du domaine finalement obtenu [13]. Un exemple de l'algorithme $C_{\uparrow\downarrow}$ est décrit par la figure 2.13. Donc X_7 est déterminé par les composantes : X_1, X_2, X_5 . Mais X_5 est calculé par (X_6, X_3) lorsque X_6 dépend de X_7 . Alors, il faut calculer plusieurs fois pour diminuer la largeur des composantes estimés comme X_7 .

Remarque 15. Le contracteur "forward-backward" n'impose pas d'avoir $n_f = n_x$. Il sera utilisé dans la suite de notre travail.

2.3.4 Phase de contraction intégrée à SIVIA

Différentes améliorations de l'algorithme SIVIA ont été proposées dans la littérature [38] [33]. En particulier, il est possible d'associer une phase de contraction avant la phase de bisection. Cela permet d'améliorer la vitesse de convergence de l'algorithme. En utilisant cette méthodologie, différents problèmes peuvent être considérés. Le tableau 2.8 résume les versions les plus classiquement rencontrées dans la littérature :

1. SiviaX a pour objectif la résolution d'un système d'équations non linéaires $f(x) = 0$ pour lequel le nombre d'équations est égal au nombre d'inconnues $n_f = n_x$;
2. SiviaPY s'intéresse à un ensemble d'inégalités non linéaires $g_i(p_1, \dots, p_n) \in [y_i], i = 1 : n_g$, écrit encore sous la forme $[g]([p]) \subset [y]$. En fait, dans ce cas, on est amené à résoudre $\mathcal{S}_p \triangleq \{p \in [p] | g(p) \in [y]\} = [g^{-1}([y]) \cap [p]$, et on note $[x] = [p] \times [y]$. Contrairement à SiviaX dans lequel seulement une boîte est considérée, SiviaPY différencie $[p]$ et $[y]$;
3. lorsqu'on s'intéresse uniquement à la caractérisation de $\mathcal{S}_p$, il est préférable d'utiliser SiviaP [38].

Dans le tableau 2.8, C_{S_x} correspond au contracteur employé.

Remarque 16. En utilisant l'opération de contraction, on améliore la vitesse de convergence des algorithmes puisque, avant de bissecter, on contracte, ce qui permet de rejeter des parties du domaine trouvées à l'étape précédente. Rappelons que la bisection permet en pratique d'obtenir une représentation plus "fine" de la forme de l'ensemble des solutions. Néanmoins, si l'objectif est de déterminer des bornes minimales et maximales sur chacune des inconnues (donc un unique pavé qui contienne l'ensemble des solutions), il n'est pas forcément nécessaire d'appliquer l'opération de bisection. Ce problème est typiquement le cas si on veut garantir qu'étant donné un ensemble de capteurs et un problème à inverser, les estimées satisferont à coup sûr une erreur limite préfixée par un cahier des charges.

2.4 Calcul du quaternion d'erreur dans le cas d'une représentation par intervalles

La définition de l'erreur donnée à l'équation (2.1.32) ne peut pas être directement appliquée lorsque l'on va s'intéresser à une représentation du quaternion sous forme d'un pavé. En effet, on verra dans la suite de ce document que la contrainte de norme unité pour le quaternion a été levée, ce qui rend cette définition inexploitable en l'état. En s'inspirant de cette définition, on va définir un quaternion erreur q_e qui va permettre de mesurer la taille du pavé solution. Notons $[q_{est}]$ le pavé solution retourné par les algorithmes d'estimation ensembliste de l'attitude et définissons $\overline{q_{est}}$ et $\underline{q_{est}}$ suivant la formule (2.3.2). Ces quaternions sont a priori non normalisés mais nous les normalisons pour pouvoir calculer le quaternion d'erreur q_e . Ainsi, on pose

$$q_e = \frac{\overline{q_{est}}}{\|\overline{q_{est}}\|} \otimes \frac{\underline{q_{est}}}{\|\underline{q_{est}}\|} = (q_{e0}, q_{e1}, q_{e2}, q_{e3})^T \in \mathbb{R}^4 \quad (2.4.1)$$

Alors si le quaternion théorique $q_{th} \in [q_{est}]$, la mesure de l'erreur que nous utiliserons est donnée par

$$[\beta_e] = [0, \quad 2 \mid \arccos(q_{e0}) \mid] \quad (rad) \quad (2.4.2)$$

Algorithme SiviaX	SiviaX ($in : [x], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, inout : \mathcal{L}$) 1. $[x] = \mathcal{C}_{\mathcal{S}_x}([x])$ 2. <i>if</i> ($[x] = \emptyset$) <i>then return</i> ; 3. <i>if</i> ($\delta[x] < \varepsilon$) <i>then</i> $\mathcal{L} = \mathcal{L} \cup \{[x]\}$; <i>return</i> ; 4. <i>bisect</i> $[x]$ <i>into</i> $[x_1]$ <i>and</i> $[x_2]$ 5. <i>SiviaX</i> ($[x_1], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, \mathcal{L}$); <i>SiviaX</i> ($[x_2], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, \mathcal{L}$)
Algorithme SiviaPY	SiviaPY ($in : [x], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, inout : \mathcal{L}$) 1. $[x] = \mathcal{C}_{\mathcal{S}_x}([x])$ 2. <i>if</i> ($[x] = \emptyset$) <i>then return</i> ; 3. $([p], [y]) = [x]$; 4. <i>if</i> ($\delta([x]) < \varepsilon$) <i>then</i> $\mathcal{L} = \mathcal{L} \cup \{[x]\}$; <i>return</i> ; 5. <i>bisect</i> $[p]$ <i>into</i> $[p_1]$ <i>and</i> $[p_2]$ 6. $[x_1] = ([p_1], [y])$; $[x_2] = ([p_2], [y])$; 7. <i>SiviaPY</i> ($[x_1], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, \mathcal{L}$); <i>SiviaPY</i> ($[x_2], \mathcal{C}_{\mathcal{S}_x}, \varepsilon, \mathcal{L}$)
Algorithme SiviaP	SiviaP ($in : [p], \mathcal{C}_{\mathcal{S}_p}, \varepsilon, g, [y], inout : \underline{\mathcal{S}}_p, \overline{\mathcal{S}}_p$) 1. $[p] = \mathcal{C}_{\mathcal{S}_p}([p])$ 2. <i>if</i> ($[p] = \emptyset$) <i>then return</i> ; 3. <i>if</i> ($[g]([p]) \subset [y]$) $\underline{\mathcal{S}}_p = \underline{\mathcal{S}}_p \cup [p]$; $\overline{\mathcal{S}}_p = \overline{\mathcal{S}}_p \cup [p]$; <i>return</i> ; 4. <i>if</i> ($\delta[p] < \varepsilon$) <i>then</i> $\overline{\mathcal{S}}_p = \overline{\mathcal{S}}_p \cup [p]$; <i>return</i> ; 5. <i>bisect</i> $[p]$ <i>into</i> $[p_1]$ <i>and</i> $[p_2]$ 6. <i>SiviaP</i> ($[p_1], \mathcal{C}_{\mathcal{S}_p}, \varepsilon, g, [y], \underline{\mathcal{S}}_p, \overline{\mathcal{S}}_p$); <i>SiviaP</i> ($[p_2], \mathcal{C}_{\mathcal{S}_p}, \varepsilon, g, [y], \underline{\mathcal{S}}_p, \overline{\mathcal{S}}_p$)

Tableau 2.8 – Déclinaison de différents solveurs obtenus à partir de SIVIA. ε est la dimension minimale du pavé en deçà de laquelle le pavé n'est plus bissecté

2.5 Conclusion

En conclusion, résoudre le problème d'estimation de l'attitude revient à résoudre un problème non-linéaire, que ce soit dans le cas avec 6 mesures (accéléromètre triaxe et magnétomètre triaxe) ou 9 mesures (accéléromètre triaxe, magnétomètre triaxe et gyromètre triaxe). Dans la suite de ce document, nous nous intéresserons uniquement à la représentation de l'attitude avec les formalismes quaternion unitaire et angles de Cardan.

Pour le quaternion unitaire $q \in \mathbb{R}^4$, la contrainte $\|\vec{q}\| = 1$ crée une difficulté pour la résolution par les approches ensemblistes. En effet, on verra dans le chapitre sur l'estimation de l'attitude que cette paramétrisation conduit, si on n'y prend garde, à des solutions qui ne sont pas strictement des quaternions de rotation. Les équations pour les mesures issues des accéléromètres et magnétomètres sont non linéaires, mais quadratiques en q_i , $i = 1 : 4$, donc facilement calculables. Les équations dynamiques (gyromètres) sont quant à elles linéaires en q .

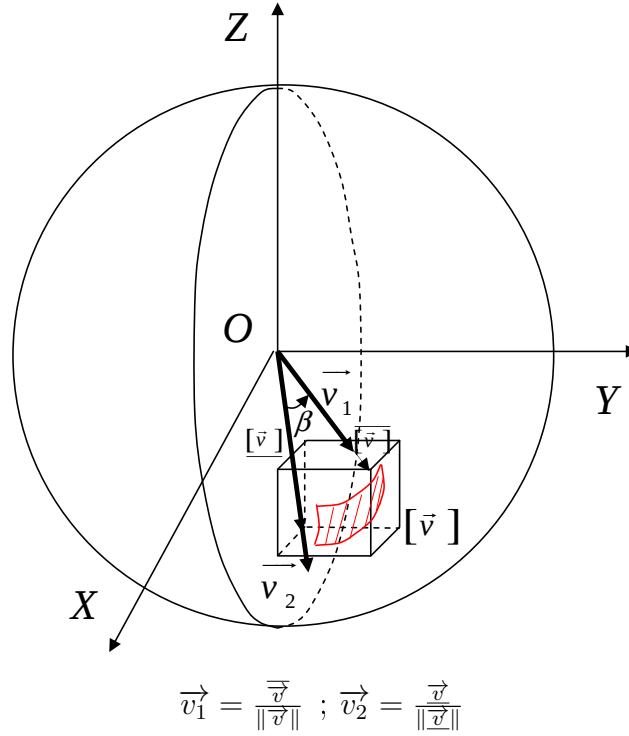


Figure 2.14 – Illustration de l'erreur de quaternion dans un espace à 3 dimensions

La représentation sous forme d'angles de Cardan est sujette à la perte d'un degré de liberté, communément appelée *Gimbal lock*. Les équations de mesure accéléromètre et magnétomètre sont non linéaires, et font apparaître des fonctions sinus et cosinus qui rendront le problème "délicat" à inverser. Cette représentation de l'attitude dans le cadre du calcul ensembliste est toutefois intéressante car pour toute solution de l'ensemble des solutions, la matrice $\mathcal{C}$ obtenue est bien une matrice de rotation. L'équation dynamique (qui fait intervenir les mesures des gyromètres) est quant à elle non linéaire.

Pour les simulations que nous présenterons par la suite, nous nous sommes inspirés des caractéristiques du capteur *MAG3* de chez MEMsense.

On comparera nos résultats à ceux obtenus avec l'algorithme SIVIA, en particulier en termes de temps calcul et de pessimisme introduit par les algorithmes que nous proposerons dans les chapitres suivants.

Chapitre 3

Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

Dans ce chapitre, nous introduisons tout d'abord le problème de l'estimation ensembliste de l'attitude dans les cas :

- *statique*, lorsque seules les mesures issues du triaxe accéléromètre et du triaxe magnétomètre sont utilisées (par abus de langage, on notera ces mesures ou ce problème d'estimation AM). On utilise alors un ensemble d'équations statiques non linéaires reliant les mesures à l'attitude du corps rigide en trois dimensions ;
- *dynamique* pour lequel les mesures fournies par les triaxes de capteurs accéléromètre, magnétomètre et gyromètre sont utilisés (par abus de langage, on notera ces mesures ou ce problème d'estimation AGM ci-après). On est alors en présence du même ensemble d'équations statiques non linéaires auquel vient s'ajouter un ensemble d'équations non linéaires reliant l'attitude, sa dérivée et les mesures gyrométriques.

Pour chacune des deux situations, on présente dans la suite de ce chapitre des techniques d'estimation de l'attitude via des approches par intervalles et on expose les algorithmes que nous avons développés. En particulier :

- on introduit les algorithmes LOOP et LoopBorne développés dans le cas AM et qui seront également exploités dans le chapitre suivant pour le diagnostic de la centrale d'attitude ;
- on présente ensuite l'observateur d'attitude pour le problème AGM. Dans ce cas, on exploite les neuf mesures fournies par la centrale d'attitude.

3.1 Définition des cas statique et dynamique et utilisation de mesures intervalles

Ici, on s'intéresse au cas où les données sont fournies par une centrale inertielle (IMU - Inertial Measurement Unit) constituée d'un ensemble de capteurs, à savoir un triaxe magnétomètre, un triaxe accéléromètre et un triaxe gyromètre. Nos travaux traitent respectivement :

1. du cas statique pour lequel nous utilisons les trois mesures du triaxe magnétomètre et les trois mesures du triaxe accéléromètres. On suppose le mouvement quasi-statique, ce qui signifie que $\|\vec{a}\| \ll \|\vec{g}\|$ où a est l'accélération propre du corps rigide dont on cherche à déterminer l'attitude, et g est le champ de gravité. Le problème de l'estimation d'attitude revient alors à construire une estimation ensembliste des paramètres qui caractérisent l'attitude, et ce à chaque instant d'échantillonnage. Dans ce cas, les algorithmes LOOP ou LoopBorne sont proposés en considérant le système d'équations non linéaires (2.2.9) ;
2. du cas dynamique pour lequel, outre les mesures précédentes, nous utilisons les mesures délivrées par un triaxe gyromètre. Ici encore, on suppose $\|\vec{a}\| \ll \|\vec{g}\|$. Le but est l'estimation de l'attitude avec un observateur ensembliste. Pour cela, les algorithmes, appelés l'algorithme *EstimEI* et l'algorithme *EstimEC* sont proposés avec le système d'équations défini par (2.2.10) ou (2.2.11).

Le principal intérêt du calcul ensembliste est la prise en compte des incertitudes des mesures fournies par un capteur sur la forme d'un intervalle, sans hypothèse sur la distribution statistique sous-jacente. En fait, dans le cas d'un fonctionnement sans défaut, la différence entre la mesure fournie par le capteur et sa valeur théorique attendue correspond au bruit de mesure plus éventuellement du bruit lié à l'environnement d'utilisation ainsi que des erreurs de modélisation. La donnée expérimentale utile y^* ce qu'on veut mesurer doit satisfaire [33] :

$$y - \delta_y \leq y^* \leq y + \delta_y \quad (3.1.1)$$

où l'incertitude de mesure δ_y est la valeur absolue de l'erreur maximale sur la mesure y . Ici, δ_y regroupe les erreurs telles que l'erreur due à la manipulation, l'erreur de l'expérimentateur et/ou l'erreur de traitement de l'information comme l'erreur de conversion. Soit $y(t)$ une mesure et δ_y la borne de l'erreur sur cette mesure, alors l'intervalle de mesure est défini par :

$$[y](t) = y(t) + [-\delta_y, +\delta_y] \quad (3.1.2)$$

La figure 3.1 présente la relation entre l'intervalle de mesure $[y](t)$ (en noir) obtenu à partir de la mesure $y(t)$ (en bleu) fournie par le capteur et la valeur expérimentale utile y^* (en rouge). Il est clair que l'erreur entre y^* et $[y](t)$ peut être caractérisée par un

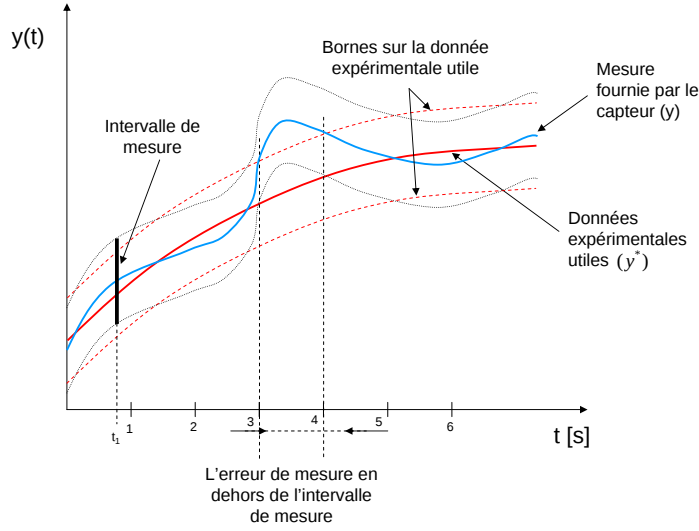


Figure 3.1 – Donnée expérimentale utile y^* , donnée fournie par le capteur y et incertitude de mesure $\pm\delta_y$

intervalle. Notons qu'entre les instants $t \leq 3$ s et $t \geq 4$ s, la représentation par intervalle permet de savoir que le capteur fonctionne correctement car la donnée expérimentale y^* est incluse dans l'intervalle de mesure $[y](t)$. Cependant, entre les instants $t = 3$ s et $t = 4$ s, la donnée expérimentale y^* n'est plus incluse dans l'intervalle $[y](t)$ ce qui signifie que la donnée fournie par le capteur n'est pas cohérente avec le modèle de mesure. Cette situation peut apparaître par exemple en cas de panne sur le capteur.

Avant d'aborder le problème de l'estimation d'attitude par des outils ensemblistes, nous avons besoin de donner quelques notations et définitions.

Un vecteur intervalle $\overrightarrow{[x]} = ([x_1], [x_2], \dots, [x_n])^T \in \mathbb{IR}^n$ est défini par :

$$\overrightarrow{[x]} = \{ \vec{x} \in \mathbb{R}^n \mid \underline{x}_i \leq x \leq \overline{x}_i, i = 1 : n \} \quad (3.1.3)$$

De plus, on note :

$$\begin{cases} \overrightarrow{[x]} = (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)^T \\ \underline{\overrightarrow{[x]}} = (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n)^T \\ x_i, i=1:n \in \mathbb{R} \end{cases} \quad (3.1.4)$$

Cette définition sera utilisée pour représenter le vecteur de mesures et le vecteur des solutions estimées.

La figure 3.2 présente une vue graphique des différentes situations que l'on peut rencontrer lorsque l'on résout un problème d'estimation par les outils ensemblistes (problème de satisfaction de contraintes). On y voit des pavés solutions (rouges) complètement inclus dans le domaine de solution (bleu), des pavés (jaunes) dits indécidables car ils ne

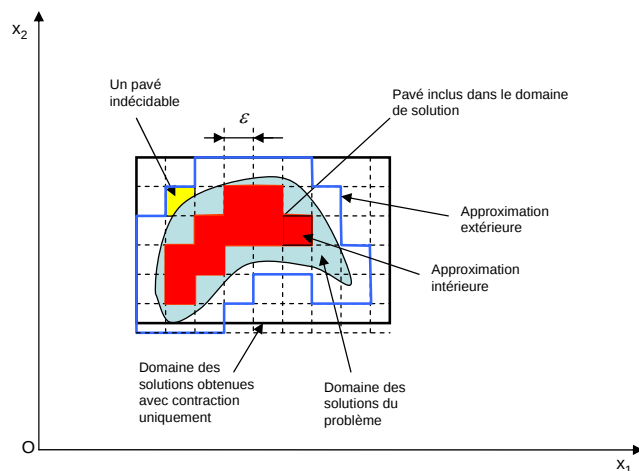


Figure 3.2 – Définition des différents pavés de l'ensemble des solutions obtenu par calcul ensembliste

sont pas complètement inclus dans le domaine de solutions et ils ne sont pas exclus du domaine de solution, et des pavés qui sont complètement exclus du domaine de solution (blancs). Le code de couleur est donné ici à titre indicatif, d'autres auteurs préférant les couleurs rouge/jaune/bleu. Sur ce schéma, ε représente la dimension minimale du pavé au delà de laquelle l'opération de bisection ne sera pas réalisée.

Remarque 17. Notons que l'ensemble des pavés solution (rouges) constitue l'approximation intérieure tandis que la surface délimitée par la ligne bleue constitue l'approximation extérieure (voir figure 3.2).

Remarque 18. Pour résoudre un problème de satisfaction de contraintes, il est clair que l'algorithme SIVIA décrit le domaine des solutions plus précisément que si l'on utilisait uniquement la solution obtenue par contraction. Cependant, SIVIA nécessite un temps de calcul non négligeable à cause du grand nombre de bisections que l'on est habituellement obligé de réaliser.

Le but de notre travail est de rechercher une approximation extérieure de l'ensemble des solutions, avec un temps de calcul "raisonnable". L'objectif est de qualifier la précision de la solution obtenue pour une configuration donnée des capteurs utilisés. En conséquence, une approximation de l'ensemble des solutions par une boîte extérieure sera suffisante, la connaissance d'un ensemble de boîtes donnant une description plus fine de l'ensemble des solutions, mais sans apporter d'information complémentaire sur la précision globale de la solution. Pour préciser notre pensée, prenons un exemple. On résout le problème de l'estimation ensembliste de l'attitude et on obtient que l'erreur maximale sur l'attitude, compte tenu des intervalles sur les mesures, est de 1° . Si ce résultat est compatible avec la précision attendue, il n'est pas nécessaire d'obtenir une description plus précise de l'ensemble des solutions. Cette remarque nous fait pressentir

que l'utilisation de la phase de bisection ne sera pas indispensable dans notre travail, ce qui nous permettra en outre, de diminuer le temps de calcul. Cela conduira au premier algorithme, nommé LOOP. Notons que nous avons constaté que la différence entre les solutions obtenues avec bisection (SIVIA par exemple) ou sans bisection est négligeable pour le problème que nous traitons. Une phase de bisection peut néanmoins être appliquée, et nous verrons dans l'algorithme LoopBorne que contrairement à ce qui est fait dans la littérature, la phase de bisection ne sera pas appliquée au centre de l'intervalle.

Il existe plusieurs environnements pour manipuler les opérations de calcul ensembliste. Dans ce travail, nous avons choisi d'utiliser la boîte à outils Intlab version 5.5 [32], gratuite et interfacée à l'environnement Matlab. Intlab exploite entre autres les facilités de surcharge d'opérateurs offertes par Matlab. Elle permet donc de créer un environnement de calcul par intervalles ouvert, et les algorithmes peuvent être codés avec le "look & feel" Matlab.

3.2 Estimation de l'attitude à partir des mesures AM : cas statique

Ici, nous considérons le cas où les mesures utilisées pour résoudre le problème de l'estimation d'attitude sont les 6 mesures fournies par un triaxe magnétomètre et un triaxe accéléromètre (voir la figure 3.3). Comme on se place dans le cas statique (on suppose que l'accélération propre est faible devant la gravité, $\|\vec{a}\| \ll \|\vec{g}\|$), le système d'équations de mesures est donné par :

$$\begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix}_{3 \times 1} + \begin{pmatrix} \vec{\eta}_M \\ \vec{\eta}_A \end{pmatrix}_{3 \times 1} = \mathcal{C}([\cdot]) \begin{pmatrix} \vec{h} \\ \vec{g} \end{pmatrix}_{3 \times 1} \quad (3.2.1)$$

$\vec{b}_M = (m_1 \ m_2 \ m_3)^T$ et $\vec{b}_A = (m_4 \ m_5 \ m_6)^T$ représentent respectivement les mesures des triaxes magnétomètre et accéléromètre fournies par la centrale inertielle MAG3. $\vec{h} = (\frac{1}{2} \ 0 \ \frac{\sqrt{3}}{2})^T$ est le vecteur du champ magnétique terrestre normalisé tandis que $\vec{g} = (0 \ 0 \ 1)^T$ est le vecteur du champ de gravité normalisé. La matrice de rotation $\mathcal{C}$ peut être exprimée en utilisant par exemple le formalisme quaternion ou le formalisme angles de Cardan. Les vecteurs $\vec{\eta}_M$ et $\vec{\eta}_A \in \mathbb{R}^3$ représentent les vecteurs de bruit de mesures qui seront considérés dans ce travail "inconnus mais bornés", c'est-à-dire $\vec{\eta}_M \in \mathbb{R}^3$ et $\vec{\eta}_A \in \mathbb{R}^3$.

Remarque 19. Rappelons que l'estimation d'attitude dans le cas statique est en fait un problème d'identification de paramètres. A chaque instant d'échantillonnage, nous traitons les mesures fournies par le triaxe magnétomètre et le triaxe accéléromètre pour

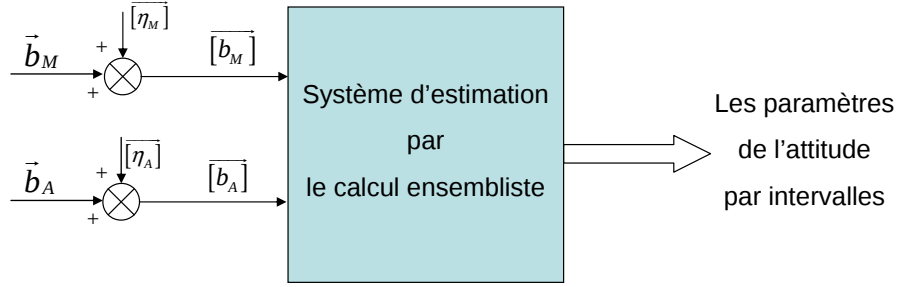


Figure 3.3 – Estimation de l'attitude à partir des mesures AM

trouver l'ensemble des attitudes satisfaisant les équations de mesures. Différents algorithmes d'identification ensembliste ont été proposés dans la littérature, et le lecteur intéressé pourra se reporter à [53], [33], ou encore [59].

Formulation du problème avec le quaternion unitaire. En utilisant le formalisme quaternion et les notations ensemblistes, le problème d'estimation de l'attitude dans le cas statique défini par l'équation (3.2.1) devient :

$$\left\{ \begin{array}{l} [q_0]^2 + [q_1]^2 + \sqrt{3}[q_1][q_3] - \sqrt{3}[q_0][q_2] - \frac{1}{2} = [m_1] \\ [q_1][q_2] - [q_0][q_3] + \sqrt{3}[q_2][q_3] + \sqrt{3}[q_0][q_1] = [m_2] \\ [q_1][q_3] + [q_0][q_2] + \sqrt{3}[q_0]^2 + \sqrt{3}[q_3]^2 - \frac{\sqrt{3}}{2} = [m_3] \\ [q_1][q_3] - [q_0][q_2] = \frac{[m_4]}{2} \\ [q_2][q_3] + [q_0][q_1] = \frac{[m_5]}{2} \\ [q_0]^2 + [q_3]^2 - \frac{1}{2} = \frac{[m_6]}{2} \\ [q_0]^2 + [q_1]^2 + [q_2]^2 + [q_3]^2 = 1 \end{array} \right. \quad (3.2.2)$$

avec $\vec{q} = ([q_0] \ [q_1] \ [q_2] \ [q_3])^T \subset ([0, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1])^T$.

Remarque 20. Notons que pour représenter une rotation, le quaternion solution doit satisfaire la contrainte de norme unité, $\|q\| = 1$. Cela signifie que, dans l'ensemble des solutions obtenu, seuls les quaternions situés sur l'hypersphère unité sont des solutions valides. Comme on l'a signalé ci-avant, notre objectif n'est pas d'obtenir une description fine de l'ensemble des solutions, mais un domaine extérieur de forme simple afin de faciliter l'exploitation du domaine obtenu. En conséquence, même si en toute rigueur la totalité de l'ensemble obtenu ne satisfait pas la contrainte égalité $\|q\| = 1$, le domaine obtenu contiendra les quaternions recherchés.

En analysant les relations entre les composantes du quaternion, on obtient le graphe des contraintes de la figure 3.4. Ce graphe ne peut pas être ramené à un arbre donc il

semble qu'on ne puisse pas utiliser la technique de vectorisation proposée par A. Gning [24].

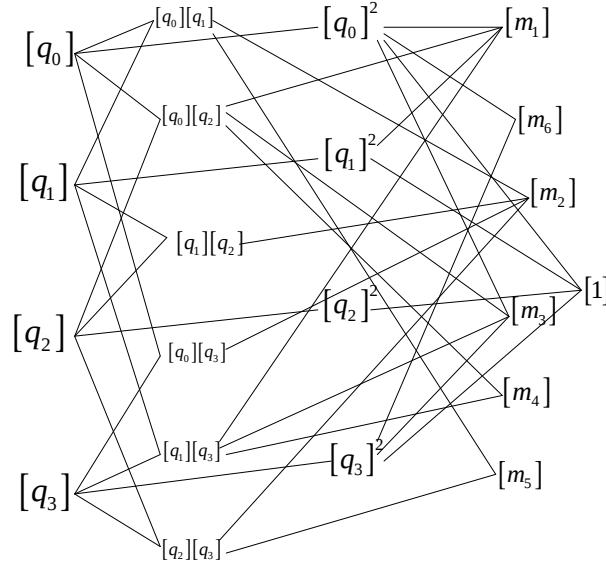


Figure 3.4 – Graphe des contraintes utilisés pour l'estimation d'attitude lorsque cette dernière est modélisée par le quaternion unitaire

Formulation du problème avec les angles de Cardan. En utilisant le formalisme des angles de Cardan, l'équation (3.2.1) devient :

$$\left\{ \begin{array}{l} [c]([\varphi])[c](\theta) - \sqrt{3}[s](\theta) = 2[m_1] \\ -[s](\varphi)[c](\psi) + [c](\varphi)[s](\theta)[s](\psi) + \sqrt{3}[c](\theta)[s](\psi) = 2[m_2] \\ [s](\varphi)[s](\psi) + [c](\varphi)[s](\theta)[c](\psi) + \sqrt{3}[c](\theta)[c](\psi) = 2[m_3] \\ -[s](\theta) = [m_4] \\ [c](\theta)[s](\psi) = [m_5] \\ [c](\theta)[c](\psi) = [m_6] \end{array} \right. \quad (3.2.3)$$

où $([\varphi] \theta [\psi])^T \subset ([-\pi/2, \pi/2] \times [-\pi, \pi] \times [-\pi, \pi])^T$, $[c] = [\cos](\cdot)$ et $[s] = [\sin](\cdot)$. Le graphe de contraintes dans le cas des angles de Cardan, est représenté sur la figure 3.5. Ici encore, il semble que le graphe ne puisse pas se ramener à un arbre, et les techniques proposées par A. Gning ne pourront être exploitées.

Dans la suite de cette partie, nous présentons différentes méthodes qui ont été implantées pour estimer l'attitude avec des outils ensemblistes dans le cas (quasi-)statique. On est donc amené à résoudre le système d'équations non-linéaires (3.2.2) lorsque l'on utilise le formalisme quaternion et (3.2.3) dans le cas d'une paramétrisation de l'attitude avec les angles de Cardan.

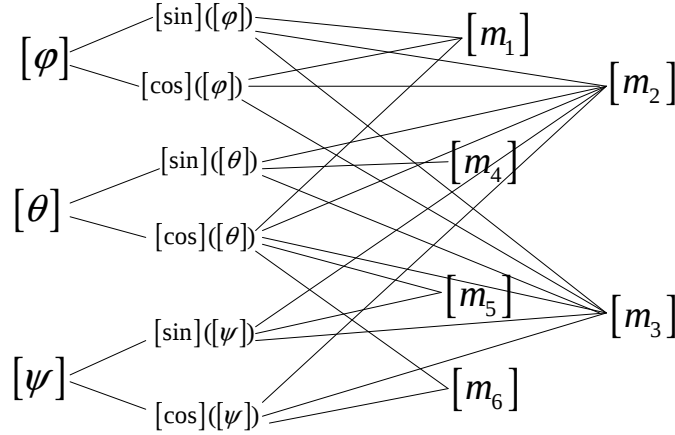


Figure 3.5 – Graphe des contraintes lorsque l'attitude est modélisée par les angles de Cardan

3.2.1 Estimation de l'attitude AM : un problème d'identification

Comme on vient de le voir, l'estimation de l'attitude dans le cas AM (quasi-)statique s'apparente à un problème d'identification de paramètres. Dans le contexte ensembliste, on peut résoudre ce problème d'identification avec un problème de satisfaction de contraintes (CSP). On résume ci-dessous quelques algorithmes (SIVIA, Project) pouvant être appliqués à notre problème d'estimation d'attitude AM puis on expose l'algorithme *LOOP* et *LoopBorne* que nous avons implanté.

Remarque 21. Dans la suite du manuscrit, les résultats obtenus avec l'algorithme *LOOP* et avec l'algorithme *LoopBorne* seront comparés à ceux obtenus par *SIVIA*, *SIVIA* étant utilisé ici comme étalon afin d'évaluer le pessimisme induit par nos choix méthodologiques, mais également le gain en temps calcul.

3.2.1.1 Estimation de l'attitude AM : utilisation de SIVIA

Afin d'analyser les performances de notre approche, nous considérons dans un premier temps l'estimation de l'attitude dans le cas statique en utilisant l'algorithme *SIVIA*.

Résolution avec SIVIA : cas de la formulation avec le quaternion. Les fonctions

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

d'inclusion que nous allons utiliser sont données par :

$$\Pi_q : \begin{cases} [f_1] = [q_0]^2 + [q_1]^2 + \sqrt{3}[q_1][q_3] - \sqrt{3}[q_0][q_2] - \frac{1}{2} \\ [f_2] = [q_1][q_2] - [q_0][q_3] + \sqrt{3}[q_2][q_3] + \sqrt{3}[q_0][q_1] \\ [f_3] = [q_1][q_3] + [q_0][q_2] + \sqrt{3}[q_0]^2 + \sqrt{3}[q_3]^2 - \frac{\sqrt{3}}{2} \\ [f_4] = [q_1][q_3] - [q_0][q_2] \\ [f_5] = [q_2][q_3] + [q_0][q_1] \\ [f_6] = [q_0]^2 + [q_3]^2 - \frac{1}{2} \\ [f_7] = [q_0]^2 + [q_1]^2 + [q_2]^2 + [q_3]^2 \end{cases} \quad (3.2.4)$$

Cependant, la contrainte $\|q\| = 1$ (f_7 de l'équation (3.2.4)) sera difficilement prise en compte puisqu'elle revient à ne retenir que les solutions situées sur l'hypersphère unité, (épaisseur nulle). Il en résulte que, pour ε donné, on n'obtiendra que des pavés admissibles. C'est pourquoi nous avons choisi de relâcher cette contrainte en considérant :

$$[1] = [1 - \varrho, 1 + \varrho] \quad (3.2.5)$$

avec ϱ l'erreur maximale que l'on s'autorise sur la norme du quaternion. Ainsi on peut définir $[Y_q] = ([m_i]_{i=1:6}, [1 - \varrho, 1 + \varrho])^T$. Pour estimer les composantes du quaternion unitaire $[x] = ([q_0] [q_1] [q_2] [q_3])^T$, l'algorithme SIVIA est utilisé, voir tableau 3.1.

$SIVIA(in : \Pi_q, [Y_q], [x], \varepsilon; inout : \mathcal{L}_1, \mathcal{L}_2)$
1.if $\Pi_q \cap [Y_q] = \emptyset$ then return; 2.if $\Pi_q \subset [Y_q]$ then $\{\mathcal{L}_1 = \mathcal{L}_1 \cup [x];$ return} 3.if $(\delta[x] < \varepsilon)$ then $\{\mathcal{L}_2 = \mathcal{L}_2 \cup [x];$ return} 4. $SIVIA(\Pi_q, [Y_q], L[x], \varepsilon, \mathcal{L}_1, \mathcal{L}_2);$ $SIVIA(\Pi_q, [Y_q], R[x], \varepsilon, \mathcal{L}_1, \mathcal{L}_2);$ where $\begin{cases} L[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [\underline{x}_j, (\underline{x}_j + \overline{x}_j)/2] \times \dots \times [\underline{x}_4, \overline{x}_4] \\ R[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [(\underline{x}_j + \overline{x}_j)/2, \overline{x}_j] \times \dots \times [\underline{x}_4, \overline{x}_4] \\ j \in 1 : 4 \end{cases}$

Tableau 3.1 – Algorithme SIVIA pour l'estimation de l'attitude avec la formulation quaternion. ε est la dimension minimale du pavé en deçà de laquelle le pavé n'est plus bissecté

Après traitement par SIVIA, on obtient deux types de solutions notées $\mathcal{L}_1$ et $\mathcal{L}_2$. $\mathcal{L}_1$ contient l'ensemble des pavés solution, i.e. lorsque toutes les contraintes du problème sont satisfaites. $\mathcal{L}_2$ correspond aux pavés indécidables.

Remarque 22. Le problème de l'estimation de l'attitude avec un quaternion est en fait surparamétré : on a 4 composantes du quaternion à estimer, ces 4 composantes devant satisfaire une contrainte (nonlinéaire) égalité. Il aurait été possible de se ramener à un

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

problème d'estimation de paramètres à 3 dimensions en remplaçant une des composantes du quaternion par son expression en fonction des 3 autres. Par exemple, on aurait pu choisir de remplacer q_0 par

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2} \quad (3.2.6)$$

et de réécrire l'ensemble des contraintes (3.2.4) en fonction uniquement de q_1 , q_2 et q_3 . Cette voie reste à explorer. En outre, le choix du signe "+" au niveau de la racine carrée dans l'équation (3.2.6) n'a pas d'incidence puisque q et $-q$ représentent physiquement la même attitude.

Notons qu'on a proposé au paragraphe 3.2.2.3 une paramétrisation du quaternion utilisant 3 paramètres (i.e. l'angle β et une représentation du vecteur $\vec{u}$ en coordonnées sphériques) qui permet d'obtenir un quaternion forcément unitaire. Cette paramétrisation "sphérique" du quaternion, on le verra tout à l'heure, n'a pas été pleinement explorée car elle induit des équations similaires à celles rencontrées avec les angles de Cardan.

Résolution avec SIVIA : cas de la formulation avec les angles de Cardan.

Lorsque l'on utilise les angles de Cardan pour paramétrer l'attitude, on obtient les fonctions d'inclusion :

$$\Pi_C : \begin{cases} [f_1] = [c]([\varphi])[c](\theta) - \sqrt{3}[s](\theta) \\ [f_2] = -[s](\varphi)[c](\psi) + [c](\varphi)[s](\theta)[s](\psi) + \sqrt{3}[c](\theta)[s](\psi) \\ [f_3] = [s](\varphi)[s](\psi) + [c](\varphi)[s](\theta)[c](\psi) + \sqrt{3}[c](\theta)[c](\psi) \\ [f_4] = -[s](\theta) \\ [f_5] = [c](\theta)[s](\psi) \\ [f_6] = [c](\theta)[c](\psi) \end{cases} \quad (3.2.7)$$

avec $[c] = [\cos](\cdot)$ et $[s] = [\sin](\cdot)$. Le vecteur intervalle de mesures est noté $[Y_C] = ([m_i]_{i=1:6})^T$ et la solution intervalle recherchée est notée $[x] = ([\varphi] \ \theta \ [\psi])^T$. L'algorithme SIVIA utilisé pour estimer les angles de Cardan est décrit dans le tableau 3.2.

$SIVIA(in : \Pi_C, [Y_C], [x], \varepsilon; inout : \mathcal{L}_1, \mathcal{L}_2)$	
1.if $\Pi_C \cap [Y_C] = \emptyset$ then return; 2.if $\Pi_C \subset [Y_C]$ then $\{\mathcal{L}_1 = \mathcal{L}_1 \cup [x];$ return} 3.if $(\delta[x] < \varepsilon)$ then $\{\mathcal{L}_2 = \mathcal{L}_2 \cup [x];$ return} 4. $SIVIA(\Pi_C, [Y_C], L[x], \varepsilon, \mathcal{L}_1, \mathcal{L}_2);$ $SIVIA(\Pi_C, [Y_C], R[x], \varepsilon, \mathcal{L}_1, \mathcal{L}_2);$	
where	$\begin{cases} L[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [\underline{x}_j, (\underline{x}_j + \overline{x}_j)/2] \times \dots \times [\underline{x}_3, \overline{x}_3] \\ R[x] \triangleq [\underline{x}_1, \overline{x}_1] \times \dots \times [(\underline{x}_j + \overline{x}_j)/2, \overline{x}_j] \times \dots \times [\underline{x}_3, \overline{x}_3] \\ j \in 1 : 3 \end{cases}$

Tableau 3.2 – Algorithme SIVIA pour l'estimation de l'attitude avec la formulation angles de Cardan. ε est la dimension minimale du pavé en deçà de laquelle le pavé n'est plus bissecté

3.2.1.2 Autre approche dérivée de SIVIA

En utilisant le principe de SiviaP, I.Braems a proposé une structure générale pour la famille d'algorithmes Project [33] qui a été appliquée au problème d'identification paramétrique en électrochimie. Le principe de l'algorithme Project est donné dans le tableau 3.3.

<i>Algorithme Project</i> (<i>in</i> : $[p], [q]$; <i>out</i> : $\mathcal{L}^-, \Delta\mathcal{L}$)	
1	$\mathcal{L} = ([p], [q]); \mathcal{L}^- = \emptyset; \Delta\mathcal{L} = \emptyset;$
2	<i>while</i> $\mathcal{L} \neq \emptyset,$
3	<i>pop first box out of</i> $\mathcal{L}$ <i>into</i> $([p], [q]);$
4	<i>Outside</i> $([p], [q]);$
5	<i>if</i> $[p] = \emptyset$ <i>go to</i> 2;
6	<i>if</i> (<i>Inside</i> $([p], [q])$) <i>then</i> $\mathcal{L}^- = \mathcal{L} \cup \{[p]\};$ <i>go to</i> 2;
7	<i>if</i> $\delta([p]) > \varepsilon,$
8	<i>bisect</i> $[p]$ <i>in</i> $[p_1]$ <i>and</i> $[p_2];$
9	$\mathcal{L} = \mathcal{L} \cup \{[p_1], [q]\}; \mathcal{L} = \mathcal{L} \cup \{[p_2], [q]\};$
10	<i>else</i> $\Delta\mathcal{L} = \Delta\mathcal{L} \cup \{[p]\};$
11	<i>endwhile.</i>

Tableau 3.3 – Structure générale de la famille d'algorithmes Project. ε est la dimension minimale du pavé en deçà de laquelle le pavé n'est plus bisecté

La technique développée par I. Braems décompose l'espace de la variable $\mathbb{X}$ en deux parties :

- une partie des composantes de $\mathbb{X}$ sera bissectée. Les composantes bissectées sont rassemblées dans le vecteur $\mathbb{X}_1$;
- l'autre partie des composantes sera mise à jour en utilisant uniquement l'étape de contraction, sans bissection. Ces composantes sont rassemblées dans le vecteur $\mathbb{X}_2$.

On a également $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2$. Cette méthode doit permettre de réduire le nombre de bisections, ce qui tend à diminuer le temps calcul. En outre, un choix judicieux de $\mathbb{X}_1$ permet de ramener le graphe de représentation du CSP à un arbre. Chaque domaine des composantes de $\mathbb{X}_1$ conduit à des pavés de petite dimension (de largeur ε) tandis que les pavées des composantes de $\mathbb{X}_2$ sont de "grande" dimension (pas de phase de bissection pour les composantes de $\mathbb{X}_2$) [33].

La solution retournée par l'algorithme *Project* contient un ensemble de pavés, ce qui n'est pas tout à fait cohérent avec les objectifs que nous nous sommes fixés. En outre, il nous semble difficile de choisir a priori quelles composantes du quaternion q ou quels angles du triplet de Cardan seront bissectés. Enfin, il n'y a aucune garantie sur le fait que la taille du domaine des composantes qui ne sont pas bissectées va être diminuée par le simple fait que d'autres composantes du vecteur de solutions sont elles

bissectées. C'est pourquoi il nous semble préférable pour notre problème, et dans le cas où nous souhaiterions utiliser une étape de bisection en plus de l'étape de contraction, de réaliser des bisections sur toutes les composantes de $\mathbb{X}$, comme le fait par exemple, SiviaP. Compte tenu des remarques ci-avant, nous avons choisi de ne pas nous comparer aux résultats obtenus avec l'algorithme Project.

3.2.1.3 Algorithmes LOOP et LoopBorne

L'algorithme LOOP est une méthode de résolution par propagation de contraintes telle que introduite par Waltz [79] et dont l'étude de la convergence peut être trouvée dans [17], [18]. De plus, LOOP utilise la variable $[\mathcal{L}]$ pour regarder tous les domaines de solution. Pour résoudre un problème inverse (ici notre problème d'identification de paramètres) avec un ensemble de n contraintes $[f_i]([x]) = 0, i = 1 : n$, nous cherchons un ensemble de m contracteurs $[x] = [f_j]([x]), j = 1 : m$. Cet ensemble est appelé fonction d'inclusion projection. Les $[f_j]([x])$ permettent de contracter l'ensemble initial de $[x]$ à chaque réalisation [45]. Lorsqu'il n'y a plus contraction, l'algorithme s'arrête et le résultat obtenu par contraction est renvoyé à l'utilisateur.

LOOP(in : $[\Gamma]$; inout : $[x]$; out : $[\mathcal{L}]$)	
1	<i>if</i> ($[x] = \emptyset$) <i>then</i> out = $[\mathcal{L}]$; <i>Exit</i> ;
2	$[a] = [\Gamma]([x])$;
3	<i>if</i> ($[a] \cap [x] = \emptyset$) <i>then</i> $[x] = \emptyset$; <i>return</i> ;
4	<i>else</i> ,
5	<i>if</i> ($[a] \cap [x] = [x]$),
6	$[\mathcal{L}] = [\mathcal{L}] \cap [x]$;
7	<i>return</i> ;
8	<i>else</i> ,
9	$[x] = [x] \cap [a]$;
10	LOOP(in : $[\Gamma]$, inout : $[x]$, out : $[\mathcal{L}]$);
11	<i>End</i> ;

Tableau 3.4 – Principe de l'algorithme LOOP

$[\Gamma]$ est l'ensemble des fonctions projection de la forme $[\Gamma] : \{[x] = [f_j]([x]), j = 1 : m\}$. L'ensemble $[x]$ retourné par LOOP est en fait un ensemble de solutions indécidables. Le temps de calcul nécessaire à la résolution du problème par l'algorithme LOOP ne dépend plus que de la largeur des intervalles en entrée et de l'ensemble des fonctions à traiter $[\Gamma]$. Un point critique de l'algorithme est donc de construire $[\Gamma]$ tel que l'algorithme converge le plus rapidement possible. Si $[\Gamma]$ ne permet pas de contracter l'ensemble initial, alors LOOP ne converge pas et la solution recherchée ne peut pas être obtenue.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

Pour améliorer la convergence de l'algorithme LOOP, on va choisir des fonctions d'inclusions du type $[x]_k = [f_j](x)$ où x_k est une des composantes de la variable x . Pour notre problème, ces fonctions d'inclusions ont été écrites pour la représentation de l'attitude avec les angles de Cardan et le quaternion unitaire. Ces fonctions sont données respectivement dans les équations (3.2.8) et (3.2.9).

Fonction d'inclusion : cas de la formulation avec les angles de Cardan. En utilisant le formalisme des angles de Cardan, $[\Gamma]$ de LOOP est donné par :

$$\left\{ \begin{array}{l} [s](\varphi) = (\sqrt{3}[m_5] + [c](\varphi)[s](\theta)[s](\psi) - 2[m_2])/[c](\psi) \\ [s](\varphi) = (2[m_3] - \sqrt{3}[m_6] - [c](\varphi)[s](\theta)[c](\psi))/[s](\psi) \\ [s](\varphi) = (\sqrt{3}[m_5] - 2[m_2])[c](\psi) + (2[m_3] - \sqrt{3}[m_6])[s](\psi) \\ [c](\varphi) = (2[m_1] + \sqrt{3}[s](\theta))/[c](\theta) \\ [c](\varphi) = (2[m_2] - \sqrt{3}[m_5] + [s](\varphi)[c](\psi))/([s](\theta)[s](\psi)) \\ [c](\varphi) = (2[m_3] - \sqrt{3}[m_6] - [s](\varphi)[s](\psi))/([s](\theta)[c](\psi)) \\ [c](\varphi) = (2[m_1] - \sqrt{3}[m_4])/[c](\theta) \\ [s](\theta) = -[m_4] \\ [s](\theta) = ([c](\varphi)[s](\theta) - 2[m_1])/ \sqrt{3} \\ [s](\theta) = (2[m_2] - \sqrt{3}[m_5] + [s](\varphi)[c](\psi))/([c](\varphi)[s](\psi)) \\ [s](\theta) = (2[m_3] - \sqrt{3}[m_6] - [s](\varphi)[s](\psi))/([c](\varphi)[c](\psi)) \\ [c](\theta) = (2[m_1] + \sqrt{3}[s](\theta))/[c](\varphi) \\ [c](\theta) = (2[m_2] + [s](\varphi)[c](\psi) - [c](\varphi)[s](\theta)[s](\psi))/(\sqrt{3}[s](\psi)) \quad (3.2.8) \\ [c](\theta) = (2[m_3] - [s](\varphi)[s](\psi) - [c](\varphi)[s](\theta)[c](\psi))/(\sqrt{3}[c](\psi)) \\ [c](\theta) = [m_5]/[s](\psi) \\ [c](\theta) = [m_6]/[c](\psi) \\ [c](\theta) = [m_5][s](\psi) + [m_6][c](\psi) \\ [s](\psi) = (2[m_2] + [s](\varphi)[c](\psi) - \sqrt{3}[m_5])/([c](\varphi)[s](\theta)) \\ [s](\psi) = (2[m_3] - \sqrt{3}[m_6] - [c](\varphi)[s](\theta)[c](\psi))/[s](\varphi) \\ [s](\psi) = [m_5]/[c](\theta) \\ [c](\psi) = (\sqrt{3}[m_5] + [c](\varphi)[s](\theta)[s](\psi) - 2[m_2])/[s](\varphi) \\ [c](\psi) = (2[m_3] - [s](\varphi)[s](\psi) - \sqrt{3}[m_6])/([c](\varphi)[s](\theta)) \\ [c](\psi) = [m_6]/[c](\theta) \\ [tan](\psi) = [m_5]/[m_6] \\ [cotan](\psi) = [m_6]/[m_5] \end{array} \right.$$

avec $[c] = [\cos](\cdot)$ et $[s] = [\sin](\cdot)$.

Fonction d'inclusion : cas de la formulation avec le quaternion unitaire. Pour la représentation de l'attitude avec un quaternion unitaire, les fonctions à considérer deviennent :

$$\left\{ \begin{array}{l} A = 2 * [m_1] - \sqrt{3} * [m_4] - [m_6] \\ B = 2 * [m_3] + [m_4] - \sqrt{3} * [m_6] \\ C = 2 * [m_3] - \sqrt{3} * [m_6] - [m_4] \\ [q_0]^2 = \frac{2+2*[m_6]+A-\sqrt{A+B}}{4} \\ [q_1]^2 = \frac{A+\sqrt{A+B}}{4} \\ [q_2]^2 = \frac{2-2*[m_6]-A-\sqrt{A+B}}{4} \\ [q_3]^2 = \frac{-A+\sqrt{A+B}}{4} \\ [q_0] = C/(4 * [q_2]) \\ [q_1] = B/(4 * [q_3]) \\ [q_2] = C/(4 * [q_0]) \\ [q_3] = B/(4 * [q_1]) \\ [q_0] * [q_1] + [q_2] * [q_3] = [m_5]/2 \\ [q_1] * [q_2] - [q_0] * [q_3] = [m_2] - \sqrt{3} * [m_5]/2 \\ [q_0] * [q_2] + [q_1] * [q_3] = [m_3] - \sqrt{3} * [m_6]/2 \\ [q_1] * [q_2] - [q_0] * [q_3] = [m_2] - \sqrt{3} * [m_5]/2 \\ [q_0]^2 + [q_1]^2 + \sqrt{3}[q_1][q_3] - \sqrt{3}[q_0][q_2] - \frac{1}{2} = [m_1] \\ [q_1] * [q_2] - [q_0] * [q_3] + \sqrt{3} * [q_2] * [q_3] + \sqrt{3} * [q_0] * [q_1] = [m_2] \\ [q_1] * [q_3] + [q_0] * [q_2] + \sqrt{3} * [q_0]^2 + \sqrt{3} * [q_3]^2 - \frac{\sqrt{3}}{2} = [m_3] \\ [q_1] * [q_3] - [q_0] * [q_2] = [m_4]/2 \\ [q_2] * [q_3] + [q_0] * [q_1] = [m_5]/2 \\ [q_0]^2 + [q_3]^2 - \frac{1}{2} = [m_6]/2 \\ [q_0]^2 + [q_1]^2 + [q_2]^2 + [q_3]^2 = 1 \end{array} \right. \quad (3.2.9)$$

Sur les systèmes d'équations (3.2.8) et (3.2.9), nous pouvons noter que pour la plupart des équations, les composantes du vecteur solution $\vec{x}$ du problème considéré sont indépendantes. Cela limite les effets dits *FALL-CLIMB*, qui sont décrits dans la littérature, par exemple [46], [33] ou [24].

Remarque 23. $[\Gamma]$ n'est pas unique. Par conséquent, il se peut que la solution obtenue puisse être réduite. Pour limiter l'effet du pessimisme introduit, nous proposons un algorithme nommé *LoopBorne* qui introduit une phase simpliste de bisection. En fait, on va découper dans le domaine solution obtenu une petite "tranche" de taille choisie ε et on va vérifier s'il existe effectivement des solutions dans cette petite tranche du domaine

précédemment obtenu. Si l'on n'a pas de solution, la tranche est éliminée et le domaine est diminué, sinon, le domaine précédemment obtenu est conservé. On constate que la pénalité introduite par cette bisection sur le temps calcul est minime puisque la coupe n'a pas pour effet de casser le domaine pour obtenir une description fine du domaine de solutions, mais de "rognier les bords". Le tableau 3.5 résume l'algorithme LoopBorne.

LoopBorne(<i>in</i> : $[\Gamma]$, ε ; <i>inout</i> : $[x]$; <i>out</i> : $[\mathcal{L}]$)	
1	<i>LOOP</i> ($[\Gamma]$, $[x]$, $[\mathcal{L}]$);
2	<i>While</i> $\mathcal{L} \neq \emptyset$
3	$[x_R] = \text{upper}(\mathcal{L}, \varepsilon)$;
4	<i>LOOP</i> ($[\Gamma]$, $[x_R]$, $[\mathcal{L}_R]$);
5	<i>if</i> $[\mathcal{L}_R] = \emptyset$
6	$[\mathcal{L}] = [\mathcal{L}] \setminus [x_R]$;
7	<i>go to</i> 2;
8	<i>elseif</i>
9	$[x_L] = \text{lower}(\mathcal{L}, \varepsilon)$;
10	<i>LOOP</i> ($[\Gamma]$, $[x_L]$, $[\mathcal{L}_L]$);
11	<i>if</i> $[\mathcal{L}_L] = \emptyset$
12	$[\mathcal{L}] = [\mathcal{L}] \setminus [x_L]$;
13	<i>go to</i> 2;
14	<i>elseif</i>
15	<i>return</i> ;
16	<i>endif</i>
17	<i>endif</i>
18	<i>endwhile</i>

Tableau 3.5 – Principe de l'algorithme LoopBorne. ε est l'épaisseur de la tranche que l'on traite

Dans cet algorithme, on a :

$$\begin{cases} \text{upper}([x], \varepsilon) \triangleq ([\underline{x}_1, \overline{x}_1] \times \dots \times [\overline{x}_j - \varepsilon, \overline{x}_j] \times \dots \times [\underline{x}_n, \overline{x}_n])^T \\ \text{lower}([x], \varepsilon) \triangleq ([\underline{x}_1, \overline{x}_1] \times \dots \times [\underline{x}_j, \underline{x}_j + \varepsilon] \times \dots \times [\underline{x}_n, \overline{x}_n])^T \\ j = \arg(\max_{i=1, \dots, n} (\delta([x_i]))) \end{cases} \quad (3.2.10)$$

Le fonctionnement de cet algorithme sera illustré dans la suite de ce chapitre.

3.2.2 Application des algorithmes sur des données simulées et réelles

Dans cette partie, on présente les résultats obtenus par les algorithmes LOOP et LoopBorne et on compare ces résultats à ceux obtenus par l'algorithme SIVIA.

3.2.2.1 Génération des données simulées

Les résultats numériques présentés ici ont été obtenus à partir d'une "trajectoire d'attitude" identique pour tous les tests numériques. On suppose que l'objet rigide sur lequel est attachée la centrale inertielle est dans l'attitude représentée par le quaternion $q = (0.9989, 0.0171, -0.0438, -0.0080)^T$ ce qui, en angles de Cardan, correspond à $(\varphi \ \theta \ \psi) = (-1^\circ, -5^\circ, 2^\circ)^T$. Au bout d'environ 2 s de transition, l'objet rigide se stabilise dans la position représentée par $q = (1, 0, 0, 0)^T$. A partir de $t = 3$ s, la nouvelle consigne en attitude est $q = (0.7036, 0.5325, -0.2361, 0.4070)^T$ soit $(\varphi \ \theta \ \psi) = (30^\circ, -50^\circ, 60^\circ)^T$, attitude que l'objet rigide mettra environ 2 s à atteindre. A partir de $t = 8$ s, la nouvelle consigne ramène l'objet rigide en $q = (1, 0, 0, 0)^T$. Les niveaux de bruits aléatoires inconnus mais bornés ajoutés tout au long de la simulation sont donnés dans le tableau 3.6. Les figures 3.7 à 3.9 montrent les attitudes et mesures obtenues. Les données étant simulées, les attitudes intermédiaires sont parfaitement connues. Pour comparer les différents résultats obtenus, nous considérons dans tous les cas la même structure de génération des données et d'estimation de l'attitude telle que décrite par la figure 3.6.

Paramètre	Borne
Bruit sur la mesure magnétomètre (η_M)	$\pm 0.0008 \text{ Gauss}$
Bruit sur la mesure accéléromètre (η_A)	$\pm 0.002 \text{ g}$
Bruit sur la mesure gyromètre η_G	$\pm 0.004 \text{ rad/s}$

Tableau 3.6 – Bornes des bruits ajoutés aux mesures simulées

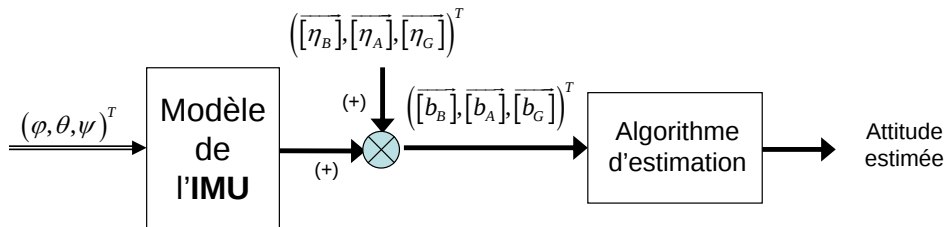


Figure 3.6 – Génération des données test et estimation de l'attitude ensembliste

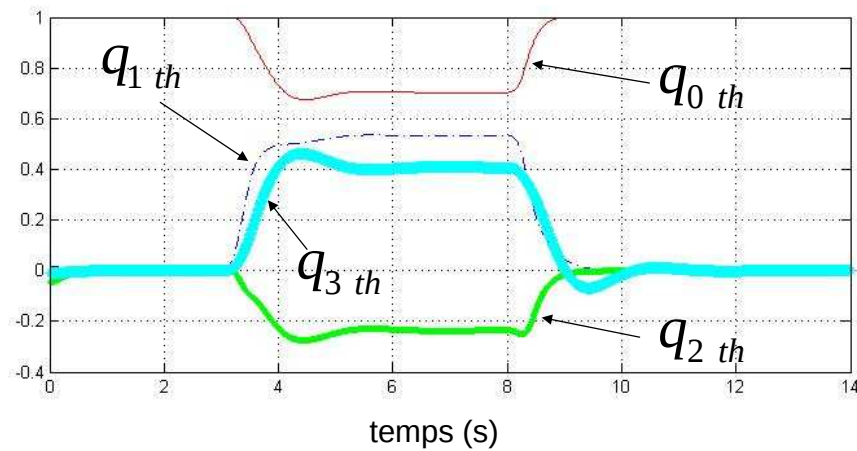


Figure 3.7 – Composantes du quaternion théorique en simulation

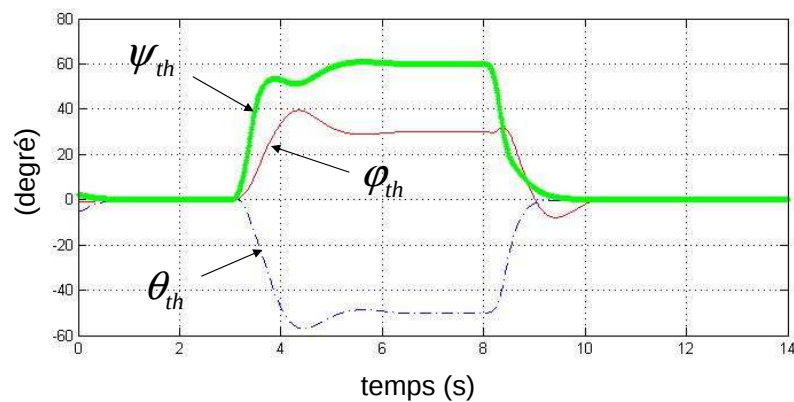


Figure 3.8 – Angles de Cardan théoriques en simulation

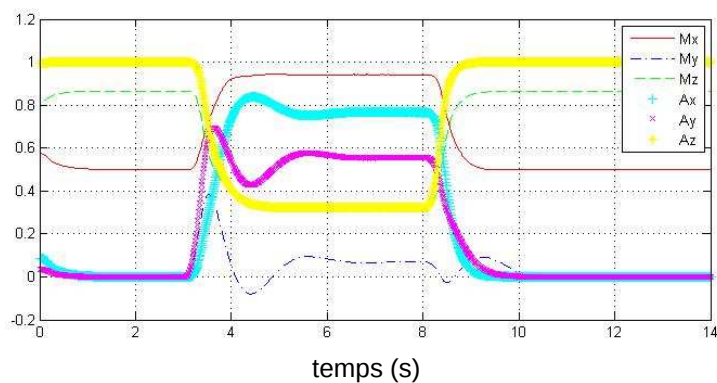


Figure 3.9 – Mesures du triaxe magnétomètres et du triaxe accéléromètres (données simulées)

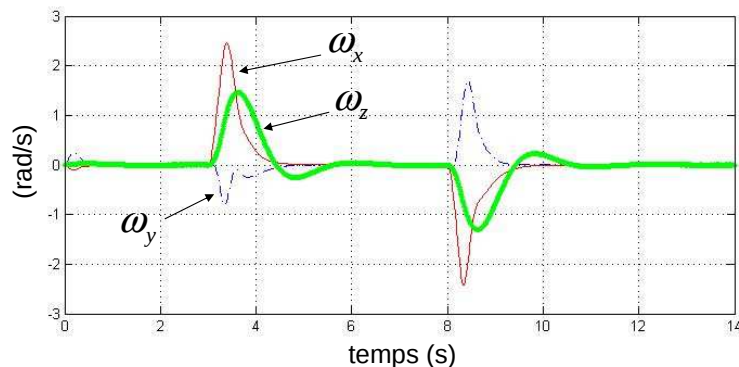


Figure 3.10 – Composantes de la vitesse angulaire (données simulées)

3.2.2.2 Description des données réelles

Pour valider les résultats obtenus en simulation pour les différents algorithmes utilisés et montrer la viabilité des démarches proposées sur un vrai système physique, nous utiliserons des données réelles acquises sur la centrale inertielle 3DM-GX1 de chez MicroStrain [31]. Les principales caractéristiques des erreurs de ce capteur sont résumées dans le tableau 3.7.

Paramètre	Valeur
Étendue de mesure	Accéléromètre : $\pm 5g$
	Gyromètre : $\pm 300^\circ/sec$
	Magnétomètre : $\pm 1.2 Gauss$
Bruit accéléromètre	$0.004 g$
Bruit gyromètre	$0.7^\circ/sec$ maximum
Bruit magnétomètre	$0.010 Gauss$ maximum
Gamme de température	-40 à $+85^\circ C$ without enclosure
Bande passante	$0 - 350 Hz$ pour neuf mesures

Tableau 3.7 – Caractéristiques de la centrale d'attitude 3DM-GX1 chez MicroStrain, le "bruit" correspond à l'écart type.

3.2.2.3 Résultats obtenus en simulation, représentation quaternion

Dans cette partie, nous considérons le cas où l'ensemble des six mesures à $t = 0 s$ est considéré pour l'estimation de l'attitude (en utilisant le formalisme du quaternion). Les résultats obtenus par SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$, sont présentés sur la figure 3.11. La valeur ε ici est choisie pour obtenir un ensemble

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

de solution par SIVIA qui ne soit pas trop grossier et pour que la tranche traitée par LoopBorne soit de dimension similaire à celle de la boîte de taille minimale dans SIVIA. Notons que les domaines obtenus par LOOP et LoopBorne sont indécidables. En outre, un choix de ε plus petit permettrait d'obtenir une description du domaine de solution via SIVIA plus fine, au détriment d'une augmentation du temps calcul.

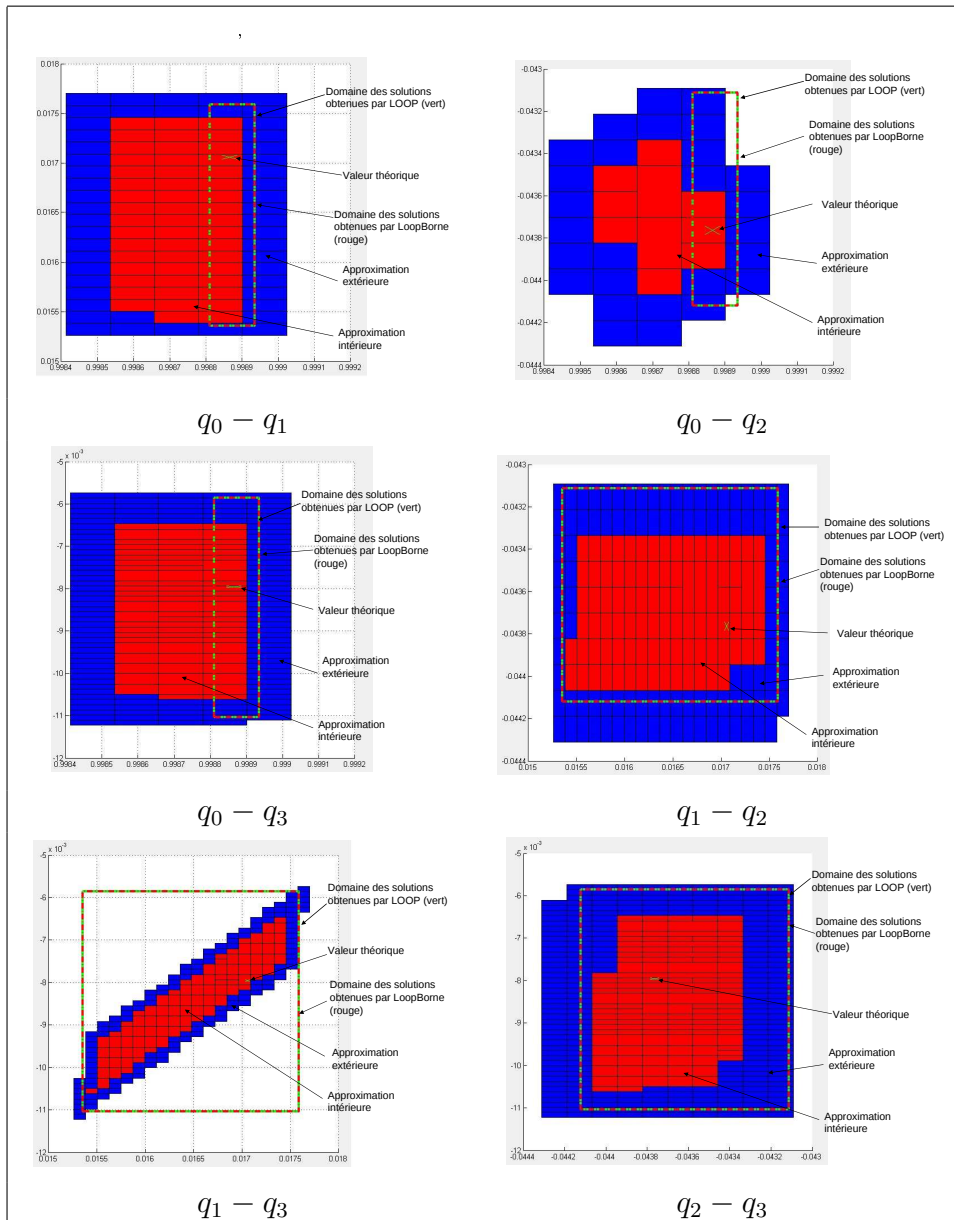


Figure 3.11 – Projection de l'ensemble des solutions obtenues par SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$ pour les mesures à $t = 0$ s

Algorithme	Temps de calcul (s)	Largeur maximale du domaine ($\delta([q])$)
SIVIA	480.7261	$(0.000732, 0.002441, 0.001343, 0.005493)^T$ (domaine extérieur)
LOOP	5.0764	$(0.000129, 0.002236, 0.001109, 0.005187)^T$
LoopBorne	14.2566	$(0.000129, 0.002260, 0.001109, 0.005187)^T$

Tableau 3.8 – Comparaison des temps de calcul obtenus avec SIVIA pour $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne. Largeur du domaine solution pour chacune des composantes pour les mesures à $t = 0$ s

On peut constater sur le tableau 3.8 que LOOP est presque 100 fois plus rapide que SIVIA. Il est également 3 fois plus rapide que LoopBorne, ce qui montre que le surcoût de LoopBorne est négligeable. Pour les données traitées, LoopBorne ne permet pas de réduire le domaine solution.

Remarque 24. La description du domaine extérieur avec SIVIA pour $\varepsilon = 0.0002$ nécessite une liste de 8930 boites lorsque nous choisissons de bissecter suivant la plus grande dimension.

Remarque 25. LOOP ne permet pas de diminuer le domaine solution lorsque l'on n'utilise que l'équation (3.2.9). Par contre, les domaines obtenus avec SIVIA et LoopBorne peuvent être encore diminués si l'on choisit ε plus petit. En effet, ces algorithmes ont la capacité de réduire la taille du domaine solution. Le temps calcul est par contre augmenté.

On remarque sur les projections de $(q_0 - q_1)$ à $(q_0 - q_3)$ de la figure 3.11 un problème sur l'approximation intérieure obtenue avec SIVIA. En effet, on constate que l'approximation intérieure de SIVIA n'est pas similaire à celle du domaine obtenu par LoopBorne. La raison de ce problème est la fonction d'inclusion $[f_7]$ de l'équation (3.2.5) qui autorise de trouver des solutions de norme non unité $\|q\| \neq 1$. En choisissant comme fonction $1 \in [[q] \times [q]^T]$, nous obtenons les résultats de la figure 3.12. On a également repéré en jaune les pavés de l'approximation intérieure obtenue par SIVIA qui ne contiennent pas de quaternion de norme unité, $\|q\| = 1$. Cela montre la performance de l'algorithme LOOP qui pour un temps de calcul raisonnable, et sans bisection, permet d'obtenir une approximation extérieure de l'ensemble des solutions. La figure 3.13, partie de gauche, donne une représentation 3D de l'ensemble des solutions de $\vec{[q]}$. En fait, la solution obtenue par LOOP ou LoopBorne englobe une fraction de la sphère unité alors que les solutions effectivement valides doivent rester sur la surface de la sphère unité.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

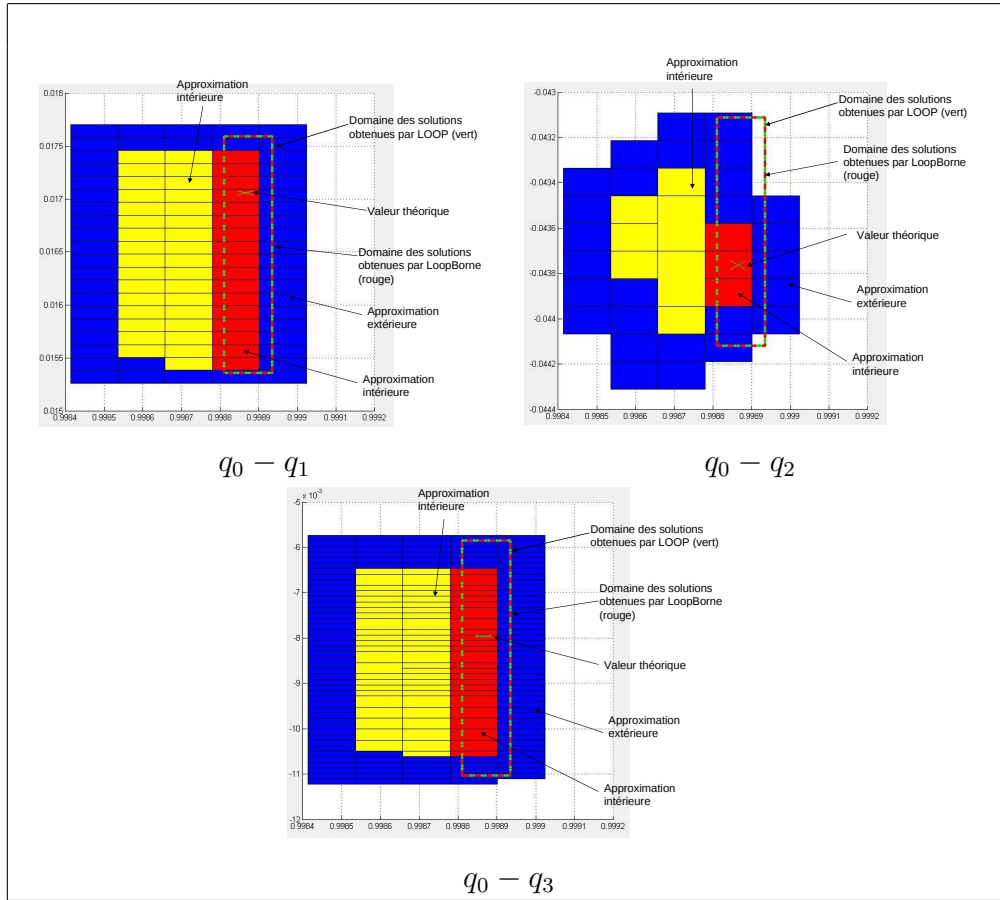


Figure 3.12 – Résultats obtenus avec SIVIA, LOOP et LoopBorne après modification de la fonction $[f_7]$ de l'équation (3.2.4). Les pavés jaunes (SIVIA) ne contiennent pas de quaternion de norme unité.

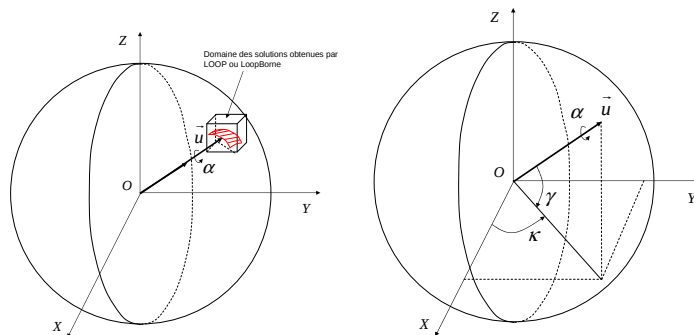


Figure 3.13 – (gauche) Composantes du domaine solution $[q]$ représenté sur la sphère unité (le quaternion est exprimé en termes de paramètres d'Euler α et $\vec{u}$), (droite) coordonnées polaires $(\alpha, \gamma, \kappa)^T$ permettant de définir q .

Pour prendre naturellement en compte la contrainte de norme unité du quaternion,

on propose de représenter le quaternion unitaire q avec les paramètres d'Euler α et $\vec{u}$ en paramétrant le vecteur unitaire $\vec{u}$ en utilisant les coordonnées sphériques. On obtient alors une paramétrisation du quaternion d'attitude avec les angles $(\alpha \ \gamma \ \kappa)$ (voir figure 3.13 (droite)), ce qui conduit à :

$$\begin{cases} q_0 = \cos(\frac{\alpha}{2}) \\ \vec{q} = \sin(\frac{\alpha}{2}) \begin{pmatrix} \cos(\gamma)\cos(\kappa) \\ \cos(\gamma)\sin(\kappa) \\ \sin(\gamma) \end{pmatrix} \end{cases} \quad (3.2.11)$$

Notons que si $\alpha = 0$, les deux angles γ et κ sont indéfinis. De l'équation (3.2.11), les angles $([\alpha] \ [\gamma] \ [\kappa])$ peuvent être calculés à partir de $[q]$:

$$\begin{cases} [\alpha] = 2 * [acos]([q_0]) \\ [\kappa] = [atan](\frac{[q_2]}{[q_1]}) \\ [\kappa] = [acos](\frac{[q_1]}{[\sin]([\alpha]/2)*[\cos]([\gamma])}) \\ [\kappa] = [asin](\frac{[q_2]}{[\sin]([\alpha]/2)*[\cos]([\gamma])}) \\ [\gamma] = [[asin](\frac{[q_3]}{[\sin]([\alpha]/2})] \end{cases} \quad (3.2.12)$$

Notons qu'il est impossible de transformer $[q]$ obtenu par SIVIA en coordonnées polaires $([\alpha] \ [\gamma] \ [\kappa])$ car l'ensemble des quaternions obtenu contient des quaternions qui ne sont pas strictement de norme unité. Si l'on réalise la transformation donnée par l'équation (3.2.12), on peut constater des situations telles que illustrées sur la figure 3.14 où l'approximation extérieure n'inclut pas l'approximation intérieure compte tenu de la remarque faite précédemment.

Remarque 26. L'utilisation du quaternion unité dans la détermination de l'attitude par les approches ensemblistes pose une vraie difficulté de résolution. Pour la contourner, nous avons proposé de lever cette contrainte en acceptant des quaternions de norme appartenant à l'intervalle $[1 - \varrho, 1 + \varrho]$. Cette approximation n'est pas complètement satisfaisante puisque nous savons que nous obtiendrons dans l'ensemble solution retourné des quaternions qui ne représentent pas des rotations (puisque de norme non unité).

Nous avons alors proposé d'utiliser une représentation "sphérique" $([\alpha] \ [\gamma] \ [\kappa])$ du quaternion. Cela permet de garantir que pour l'ensemble des solutions, les quaternions sont bien de norme unité. Même si cette approche est **parfaitement valide pour garantir les propriétés théoriques de l'ensemble des quaternions solutions**, elle n'a pas été poursuivie car elle nous a semblé induire des difficultés similaires à la représentation avec les angles de Cardan (présence de fonctions trigonométriques qui augmentent le temps de calcul).

Notons enfin que si la paramétrisation sphérique du quaternion est retenue, le quaternion doit être déduit de $([\alpha] \ [\gamma] \ [\kappa])$.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

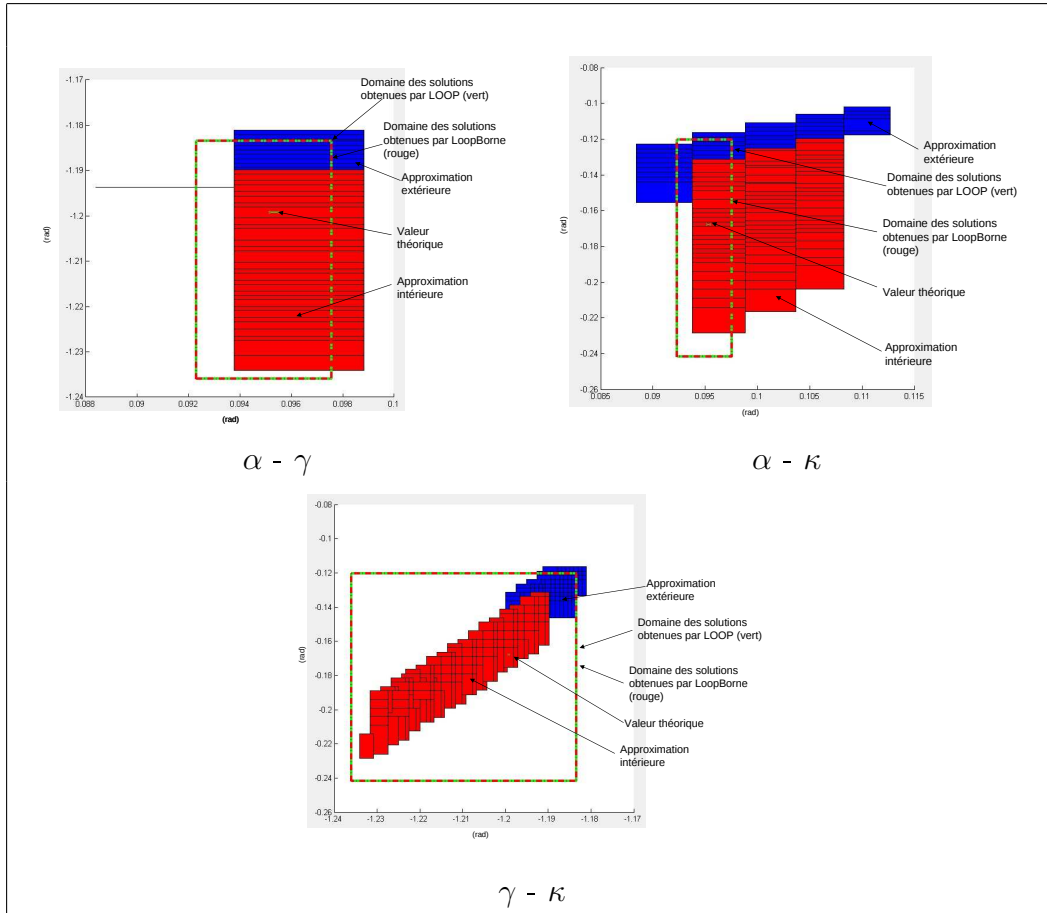


Figure 3.14 – Angles $([\alpha], [\gamma], [\kappa])$ à partir du quaternion $[q]$ obtenu en utilisant SIVIA, LOOP et LoopBorne, mesures à $t = 0$ s

Considérons maintenant le cas où l'estimation de l'attitude est réalisée à l'instant $t = 0.09$ s. La figure 3.15 montre les résultats obtenus pour les différents algorithmes. On constate que pour SIVIA, on trouve plusieurs pavés intérieurs (pour $\varepsilon = 2 \cdot 10^{-4}$) et que pour aucun d'eux la contrainte $\|q\| = 1$ n'est satisfaite (pavés repérés en jaune sur figure 3.15). Pour palier à ce problème, il suffit de diminuer la valeur de ε afin d'obtenir une description plus fine du domaine de solution. Cependant, cela augmente le temps de calcul. Il faut noter que ce problème n'existe pas sous la représentation des angles de Cardan puisqu'on n'a pas de contrainte égalité à satisfaire.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

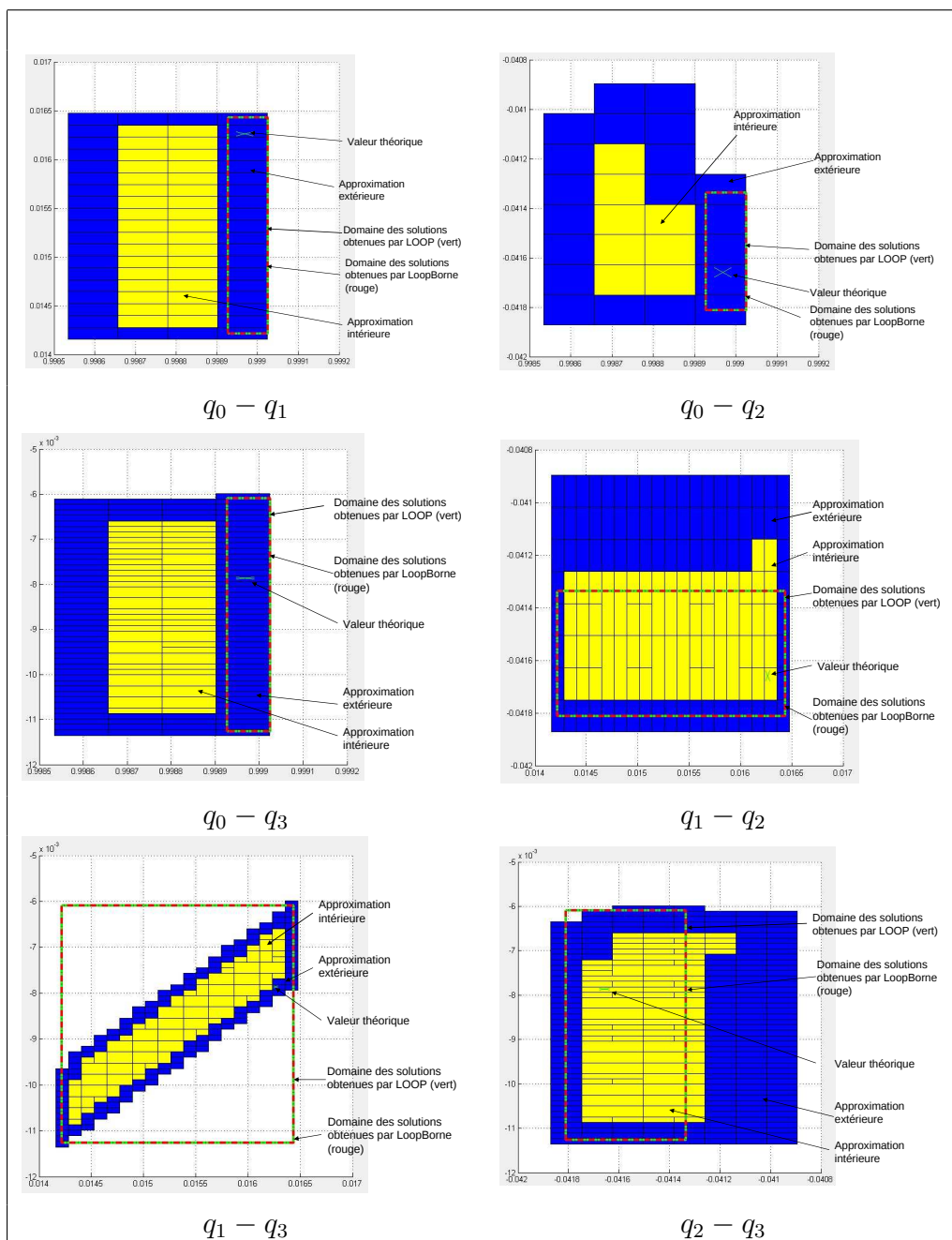


Figure 3.15 – Composantes du quaternion obtenues par SIVIA, LOOP et LoopBorne, mesures à $t = 0.09$ s

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

Toutes les données décrites au paragraphe 3.2.2.1 ont été traitées avec LOOP. Pour la représentation quaternion de l'attitude, les résultats obtenus sont présentés sur les figures 3.16 à 3.19. La figure 3.16 donne les 4 composantes du quaternion au cours du temps tandis que la figure 3.17 présente la largeur de l'intervalle pour chaque composante du quaternion. On constate que cette largeur dépend de l'instant d'échantillonnage, donc des mesures et donc de l'attitude dans laquelle se trouve le corps rigide sur lequel sont fixés les capteurs. La figure 3.18 présente la grandeur $[\beta_e]$ définie à l'équation (2.4.2). Rappelons que cette erreur est en pratique une mesure de la taille du domaine solution obtenu. On voit qu'on a au maximum $\beta_e = 1.5^\circ$, ce qui constitue de fait, un domaine solution de faible dimension. La figure 3.19 présente le temps de calcul nécessaire pour obtenir l'ensemble solution à chaque instant d'échantillonnage. Ici aussi, on constate que le temps de calcul dépend de l'attitude du corps rigide sur lequel sont fixés les capteurs.

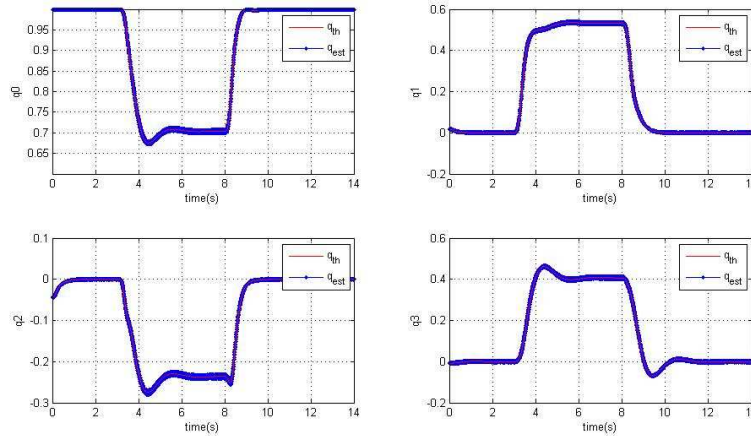


Figure 3.16 – Ensemble des solutions obtenues par LOOP, représentation quaternion de l'attitude

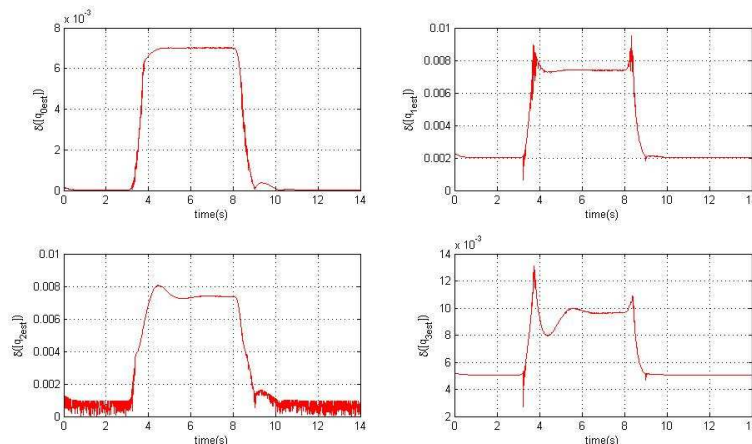


Figure 3.17 – Largeur des intervalles pour chaque composante des quaternions calculés par LOOP

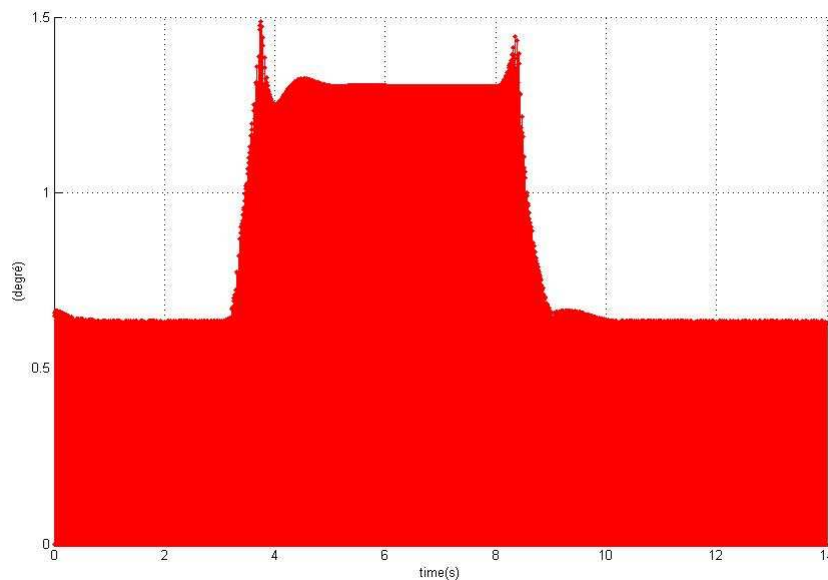


Figure 3.18 – $[\beta_e]$ (2.4.2) par LOOP pour la représentation quaternion. Données simulées

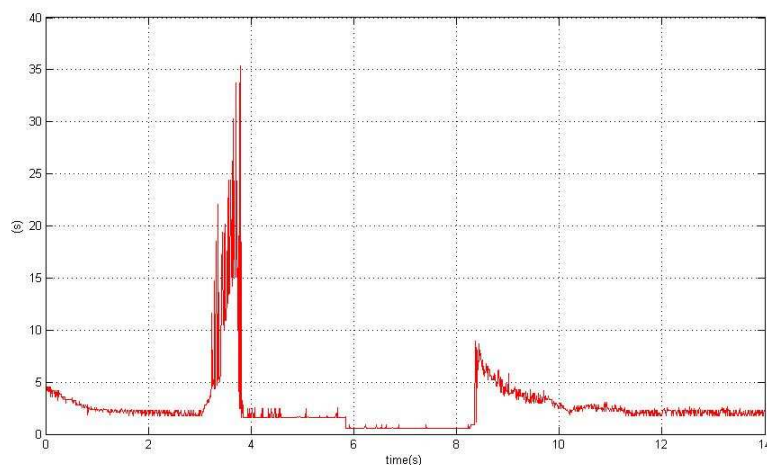


Figure 3.19 – Temps calcul de LOOP en représentation quaternion sur PC Intel i7 1.6GHz

Les résultats obtenus avec l'algorithme LoopBorne sont donnés sur les figures 3.20 à 3.23. En comparant aux résultats des figures 3.16 à 3.19, on voit que la largeur des pavés solution obtenus par LoopBorne est plus petite que celle obtenue par LOOP, ce qui justifie pleinement son utilisation. Par contre, le temps de calcul peut devenir prohibitif, mais le ratio des temps de calcul avec LOOP et LoopBorne dépend des données utilisés pour déterminer le quaternion, donc de l'attitude du corps rigide sur lequel sont fixés les capteurs.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

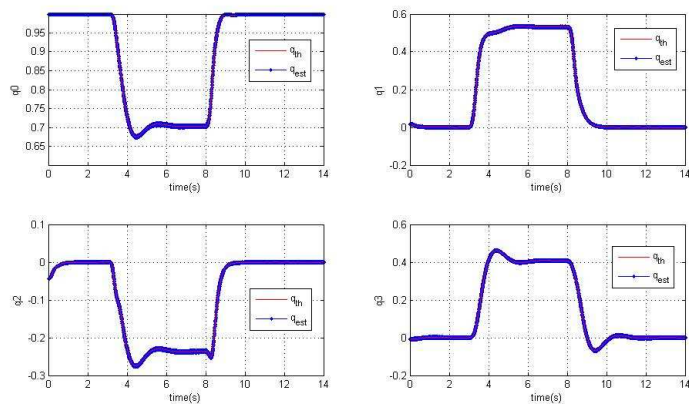


Figure 3.20 – Ensemble solution obtenu par LoopBorne, représentation quaternion de l'attitude

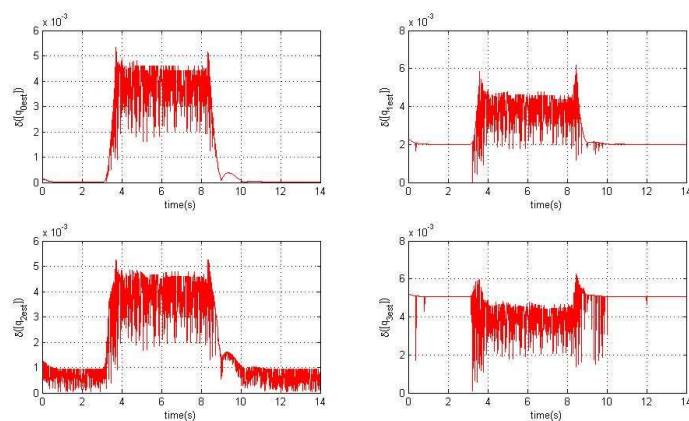


Figure 3.21 – Largeur des pavés solution pour les différentes composantes du quaternion, utilisation de LoopBorne

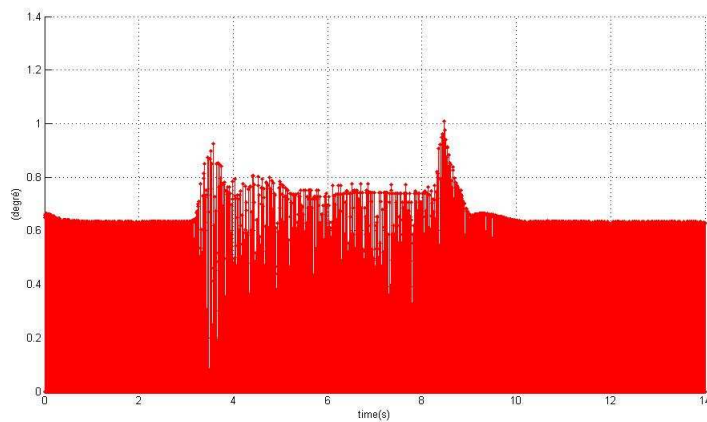


Figure 3.22 – $[\beta_e]$ (2.4.2) par LoopBorne en représentation quaternion. Données simulées

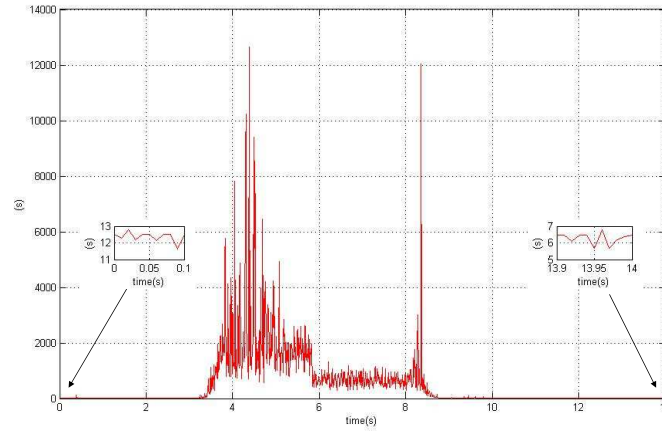


Figure 3.23 – Temps calcul pour déterminer l'ensemble solution par LoopBorne en représentation quaternion sur PC Intel i7 1.6GHz. Données simulées

3.2.2.4 Résultats obtenus en simulation, représentation angles de Cardan

La figure 3.24 montre les résultats obtenus à $t = 0$ s pour SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$ dans le formalisme des angles de Cardan. On voit une très bonne concordance entre tous les résultats, en particulier, les résultats obtenus par LOOP et LoopBorne contiennent l'approximation interne retournée par SIVIA. Notons que pour tout triplet $(\varphi, \theta, \psi) \in [\varphi] \times [\theta] \times [\psi]$, la matrice $\mathcal{C}$ est bien une matrice de rotation, ce qui ne pouvait pas être garanti dans la représentation quaternion lorsque la contrainte de norme unité est levée.

Algorithme	Temps de calcul (s)	Largeur des intervalles $(\delta([\varphi]) \ \delta([\theta]) \ \delta([\psi]))^T$
SIVIA	327.4983	$(0.593262, 0.878906, 0.241699)^T$ (domaine extérieur)
LOOP	5.3396	$(0.604385, 0.102645, 0.230486)^T$
LoopBorne	22.7675	$(0.581466, 0.102645, 0.230486)^T$

Tableau 3.9 – Comparaison des performances des algorithmes SIVIA avec $\varepsilon = 2e - 4$, LOOP et LoopBorne, représentation angles de Cardan, mesures à $t = 0.09$ s

Toutes les données décrites au paragraphe 3.2.2.1 ont été traitées avec LOOP. Pour la représentation avec les angles de Cardan de l'attitude, les résultats obtenus sont présentés sur les figures 3.25 à 3.27. La figure 3.25 donne la solution pour les trois angles de Cardan au cours du temps. La largeur des différents intervalles est reportée sur la figure 3.26 et le temps de calcul est donné sur la figure 3.27. On constate ici encore que ces grandeurs dépendent des mesures disponibles, c'est-à-dire de l'instant d'échantillonnage. En effet, intuitivement, on comprend bien que suivant l'attitude, certains capteurs n'apporteront

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

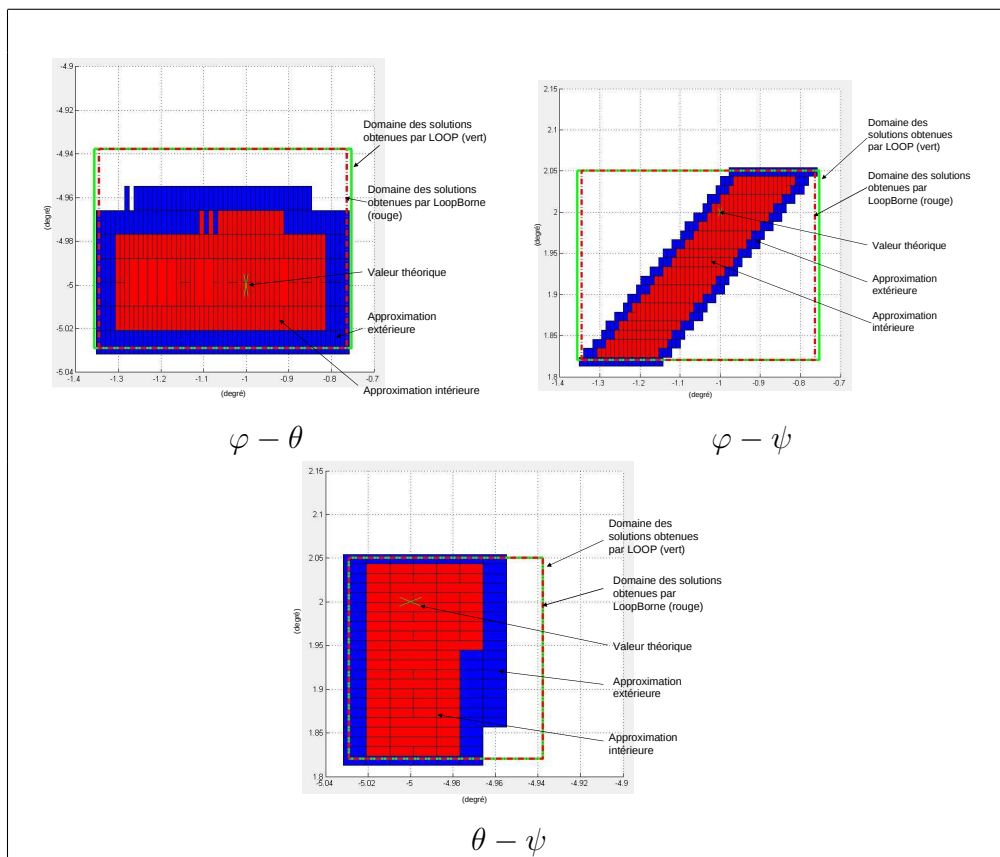


Figure 3.24 – Angles de Cadran obtenus par SIVIA avec $\varepsilon = 2 \cdot 10^{-4}$, LOOP et LoopBorne avec $\varepsilon = 2 \cdot 10^{-4}$, mesures à $t = 0$ s

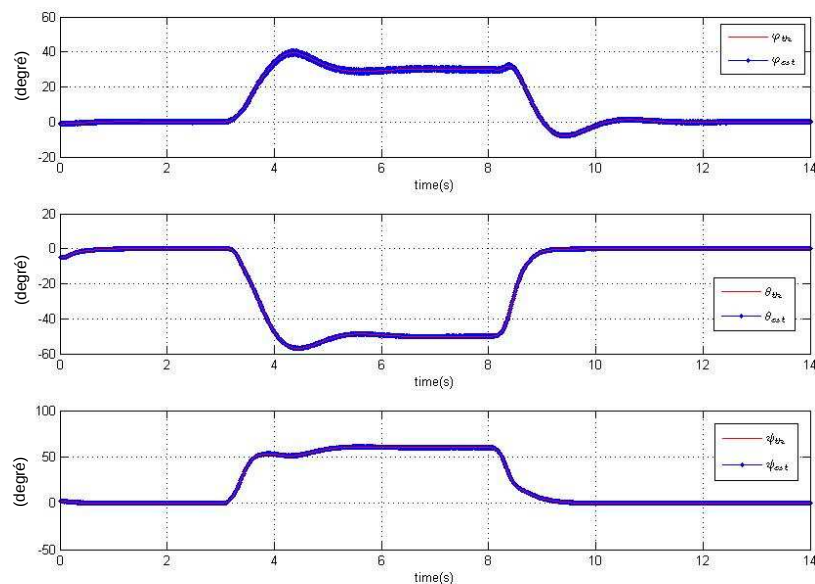


Figure 3.25 – Angles de Cardan obtenus par LOOP

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

pas d'information. De ce fait certaines fonctions dans le jeu de contraintes ne permettent pas de contracter "significativement" le domaine solution à chaque itération. Par exemple dans le cas où la centrale inertielle est à l'horizontale, les mesures des accéléromètres n'apportent pas d'information sur le cap, ce dernier ne pouvant être estimé qu'à partir des magnétomètres.

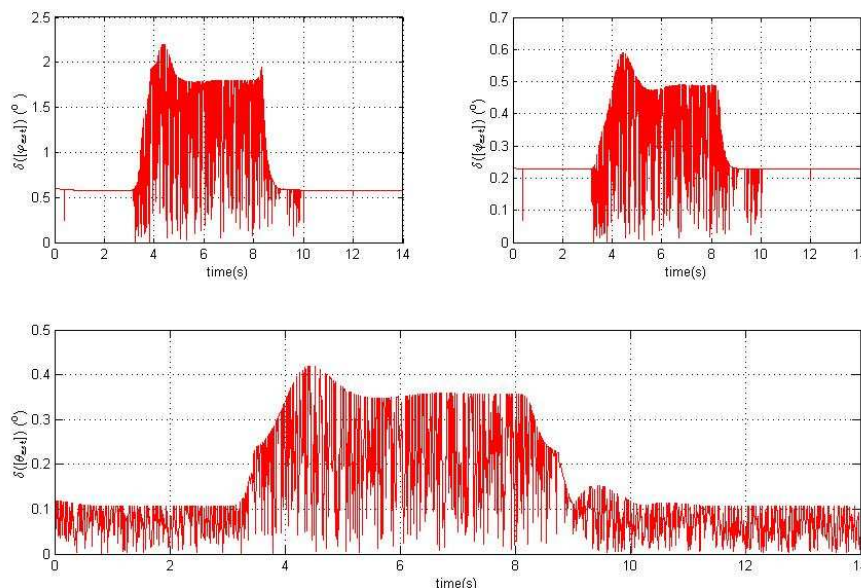


Figure 3.26 – Largeur des intervalles pour chacun des angles de Cardan obtenus par LOOP

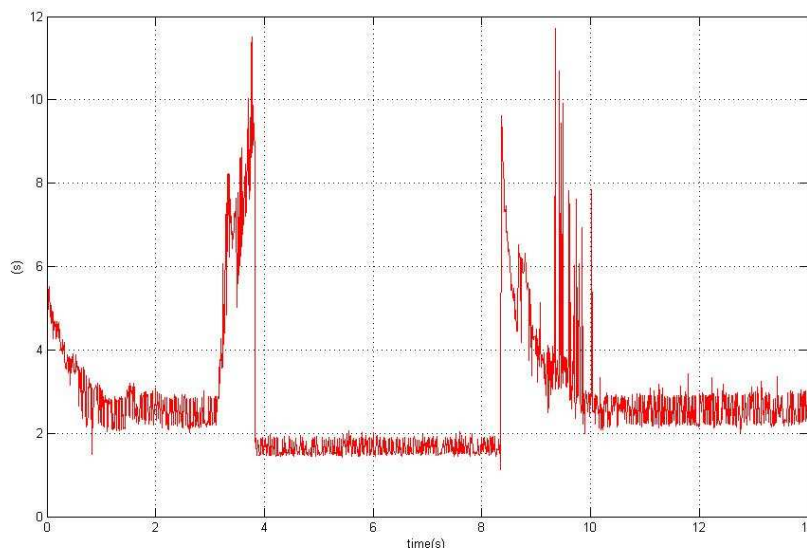


Figure 3.27 – Temps de calcul par LOOP en représentation des angles de Cardan sur PC Intel i7 1.6GHz

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

Les figures 3.28 à 3.30 montrent les résultats obtenus pour la représentation des angles de Cardan avec LoopBorne. Comme précédemment, le temps d'exécution de LoopBorne est plus important que celui de LOOP mais les pavés solution sont de taille réduite avec LoopBorne.

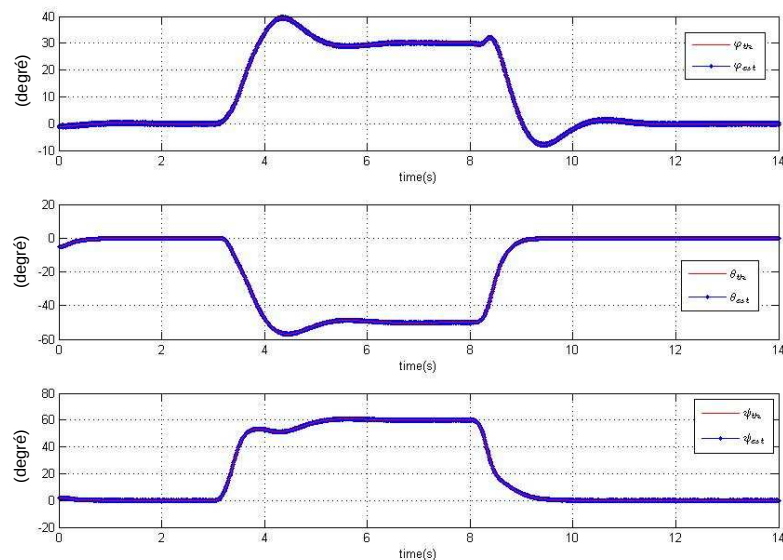


Figure 3.28 – Angles de Cardan par LoopBorne

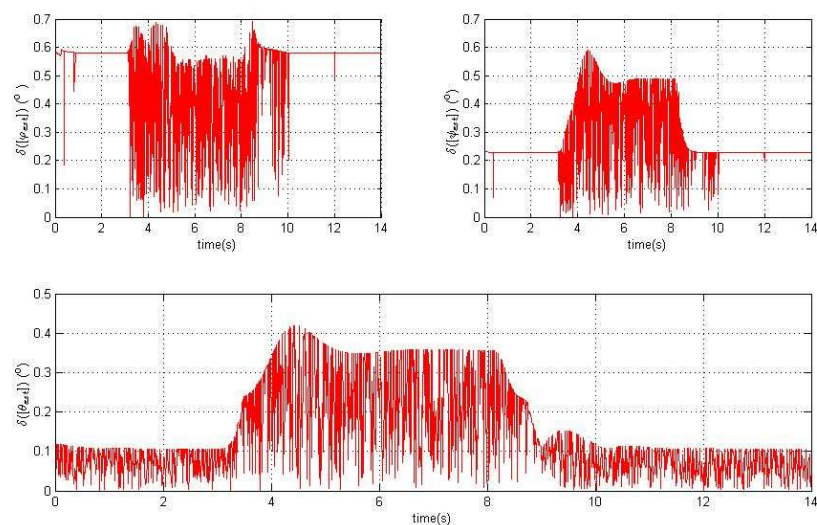


Figure 3.29 – Largeur des pavés des angles de Cardan par LoopBorne

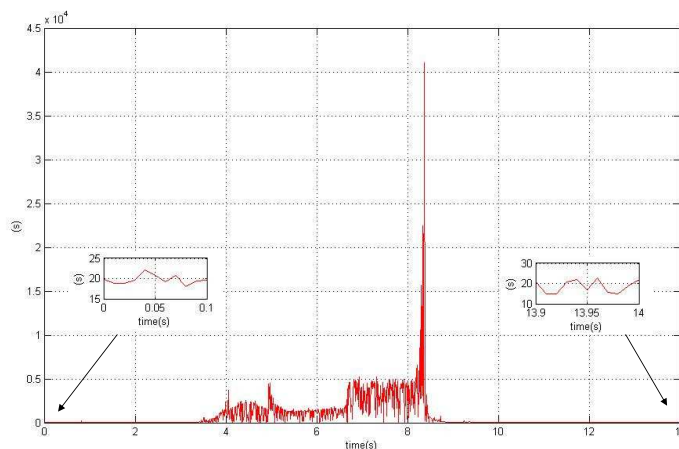


Figure 3.30 – Temps de calcul par LoopBorne en représentation angles de Cardan sur PC Intel i7 1.6GHz

3.2.2.5 Résultats obtenus avec des données réelles

Les algorithmes présentés ci-avant ont été utilisés sur des données réelles obtenues avec le capteur 3DM-GX1 de chez MEMsense. Les mesures sont données sur la figure 3.31. La centrale inertielle est posée sur un plan horizontal (ou supposé comme tel!). Des mouvements de rotation lents sont ensuite appliqués puis la centrale est maintenue dans une attitude proche de celle de départ, mais sans être posée. La fin de l'expérimentation est donc parfaitement statique, aux tremblements près induits par l'expérimentateur. Les mesures ont été effectuées en extérieur pour limiter l'effet des perturbations magnétiques. L'attitude a été représentée par un quaternion unitaire.

Les paramètres du tableau 3.7 page 70 sont utilisés pour choisir les bornes sur les différentes mesures. Notons que ces bornes sont en pratique supérieures à celles du tableau afin de tenir compte des petites perturbations magnétiques et des faibles accélérations propres supportées par la centrale inertielle. En outre, les grandeurs données dans le tableau 3.7 sont les écarts type. Ainsi, nous prendrons pour borne

- 0.02 (donnée normalisée) pour l'accéléromètre ;
- 0.02 (donnée normalisée) pour le magnétomètre.

Les données ont été traitées avec les algorithmes LOOP et LoopBorne, ce dernier utilisant $\varepsilon = 0.002$.

Pour cette expérimentation, nous ne possédons pas à proprement parler de mesure de référence de l'attitude. En pratique, cette mesure de référence de l'attitude pourrait être fournie par un système optique de type "CodaMotion", <http://www.codamotion.com/>. Malheureusement, au GIPSA-Lab, nous ne disposons pas d'un tel dispositif. Notons que ce dispositif nécessite l'installation dans une salle. Or les mesures ont été réalisées en

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

extérieur afin de limiter les perturbations magnétiques induites par les éléments métalliques présents dans un bureau. La centrale inertielle 3DM-GX1 possède, en interne, un algorithme d'estimation de l'attitude. On peut donc récupérer à la fois les données brutes (utilisées par nos propres algorithmes) et le quaternion estimé par la centrale 3DM-GX1. Ce dernier servira de référence q_{th} dans les résultats présentés ci-après. Notons dès à présent qu'on ne connaît pas l'algorithme embarqué dans la centrale inertielle, en particulier, on ne sait pas comment il gère les accélérations parasites et les perturbations magnétiques.

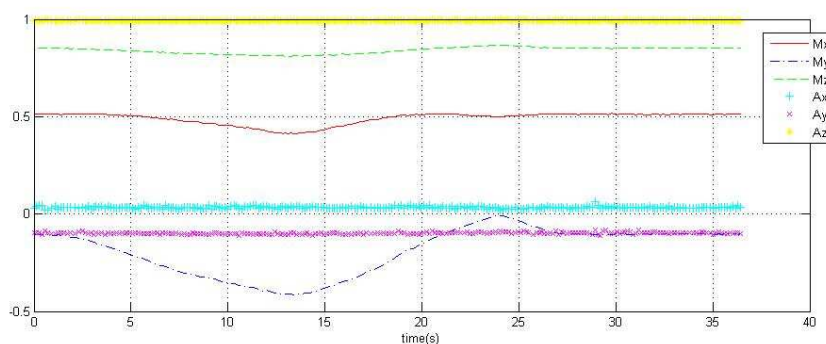


Figure 3.31 – Mesures réelles des triaxes magnétomètre et accéléromètre

Les résultats sont reportés sur les figures 3.32 à 3.37. Sur les figures 3.32 et 3.35, on donne les intervalles obtenus sur chacune des composantes du quaternion estimé à chaque instant d'échantillonnage avec LOOP et LoopBorne respectivement. On constate que les intervalles sont du même ordre de grandeur, la phase de bisection introduite dans LoopBorne n'apportant pas de diminution de la taille du domaine solution retourné. Les figures 3.33 et 3.36 représentent l'angle β_e pour les algorithmes LOOP et LoopBorne respectivement. On constate ici encore que les erreurs sont similaires pour les deux algorithmes. Notons que pour certains instants d'échantillonnage, le quaternion retourné par la centrale inertielle et qui nous sert de référence, q_{th} , n'est pas inclus dans le domaine solution estimé. Comme on ne maîtrise pas l'algorithme d'estimation de l'attitude embarqué dans la centrale inertielle, on donne ce quaternion q_{th} à titre indicatif, sans avoir réellement confiance dans sa valeur. La comparaison des figures 3.34 et 3.37 montre que la phase de bisection introduite dans LoopBorne augmente significativement le temps de calcul. On voit donc sur cet exemple que pour la trajectoire suivie, LoopBorne ne produit pas des résultats significativement meilleurs que LOOP, i.e. la taille de l'ensemble solution retourné n'est pas diminué tandis que le temps calcul est clairement augmenté.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

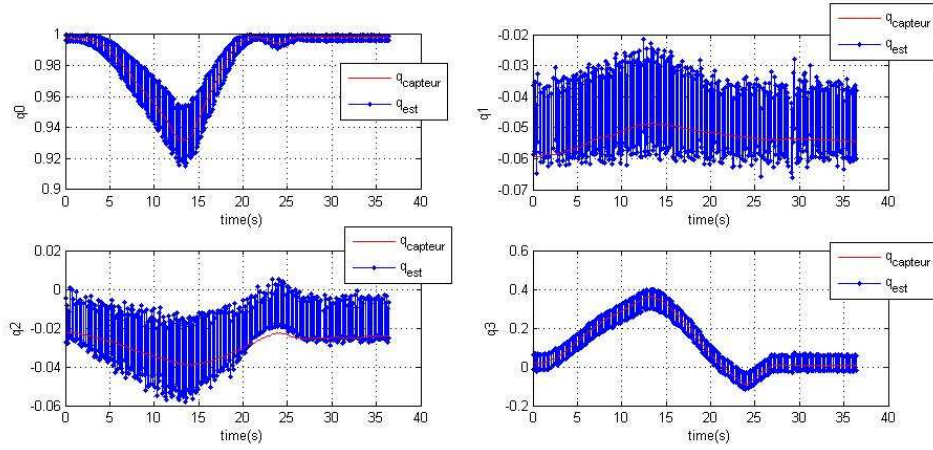


Figure 3.32 – Estimation des composantes du quaternion par l'algorithme LOOP au cours du temps. Données réelles. q_{est} en bleu et $q_{capteur}$ fourni par l'algorithme interne de l'IMU en rouge

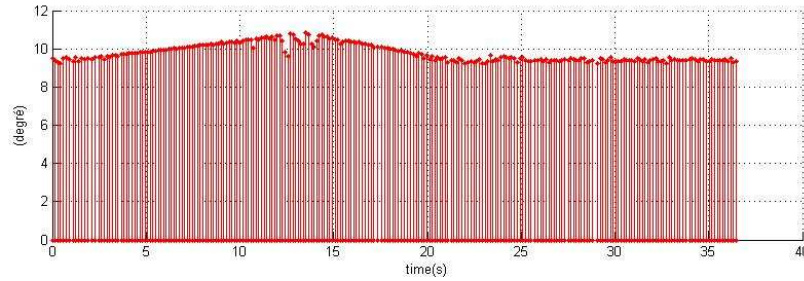


Figure 3.33 – Valeur de $[\beta_e]$ (2.4.2) au cours du temps. Utilisation de l'algorithme LOOP. Données réelles

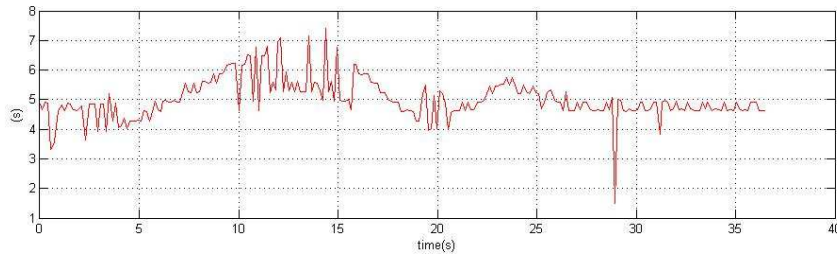


Figure 3.34 – Temps de calcul par LOOP à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation quaternion de l'attitude

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

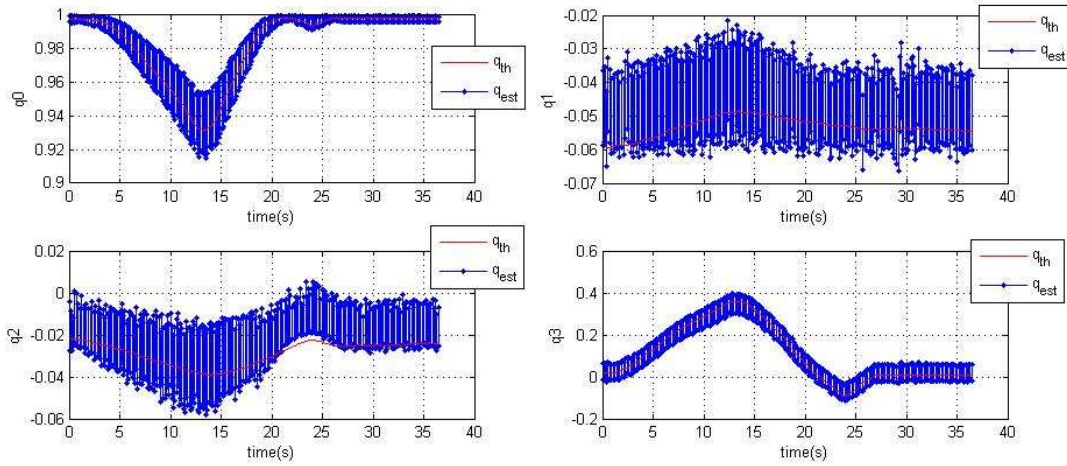


Figure 3.35 – Estimation des composantes du quaternion par l'algorithme LoopBorne au cours du temps. Données réelles

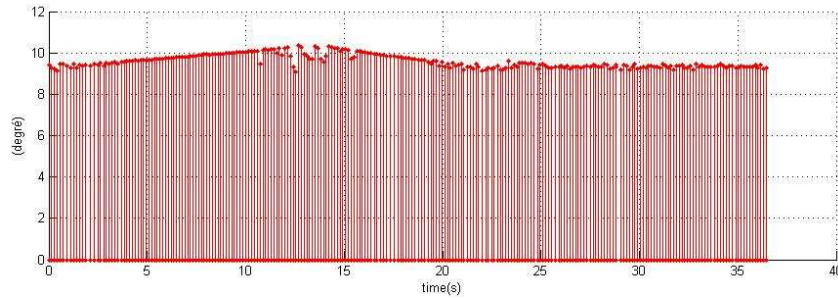


Figure 3.36 – Valeur de $[\beta_e]$ (2.4.2) au cours du temps. Utilisation de l'algorithme LoopBorne. Données réelles

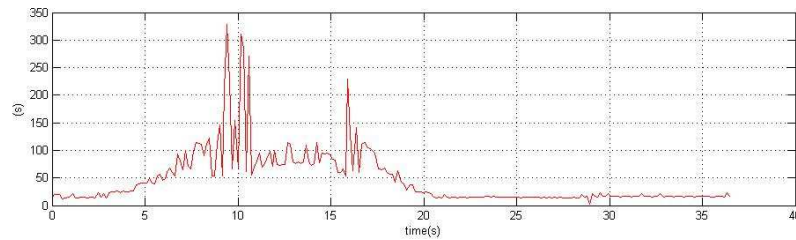


Figure 3.37 – Temps de calcul par LoopBorne à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation quaternion de l'attitude

Sur les figures 3.38 à 3.39 on a reporté les résultats obtenus avec la paramétrisation de l'attitude avec les angles de Cardan et pour l'algorithme LoopBorne. Le domaine

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

de solutions retourné est quand à lui tout à fait acceptable. En effet, on constate des intervalles sur les différents angles de 1° à 8° , en fonction de l'instant d'échantillonnage. On voit également que les angles "théoriques" obtenus à partir du quaternion q_{th} calculé par l'algorithme embarqué dans la centrale inertielle ne sont pas toujours inclus dans l'intervalle solution retourné. On est ici encore en droit de mettre en doute la valeur du quaternion retourné par la centrale inertielle car on ne connaît pas l'algorithme implanté (en particulier, comment gère-t-il les accélérations parasites ?), alors que ces accélérations parasites sont naturellement prises en compte dans le modèle de bruit "inconnu mais borné".

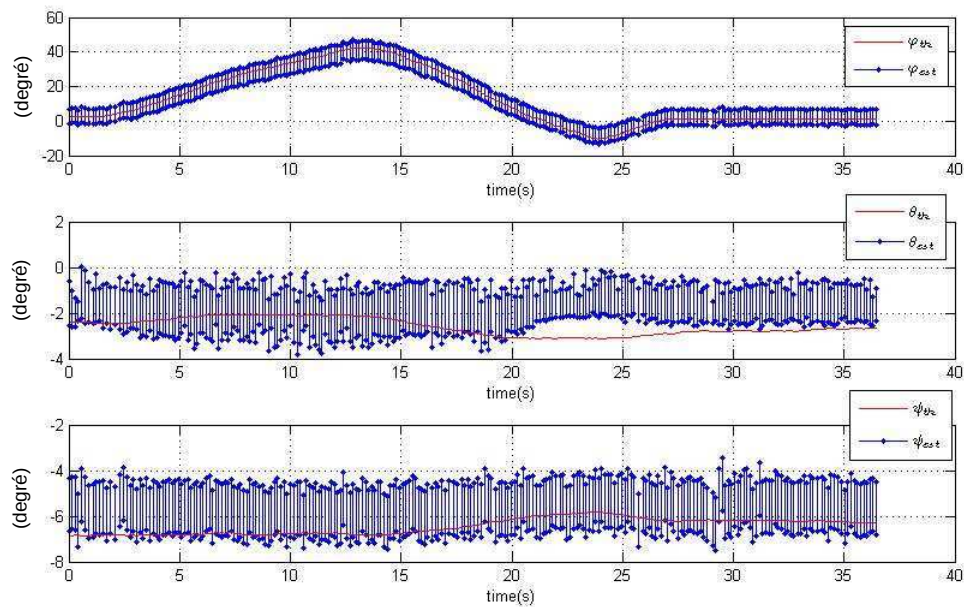


Figure 3.38 – Estimation des composantes des angles de Cardan par l'algorithme LoopBorne au cours du temps. Données réelles

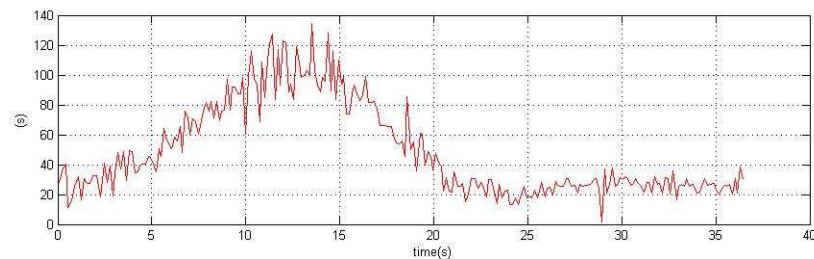


Figure 3.39 – Temps de calcul par LoopBorne à chaque instant d'échantillonnage sur PC Intel i7 1.6GHz. Traitement des données réelles et utilisation de la représentation angles de Cardan de l'attitude

3.2.3 Conclusion partielle sur l'estimation de l'attitude dans le cas statique

Dans cette partie, nous nous sommes intéressés à l'estimation de l'attitude par des outils ensemblistes, dans le cas de mesures fournies par un triaxe accéléromètre et un triaxe magnétomètre. Rappelons que notre objectif est d'obtenir un ensemble de forme simple englobant l'ensemble des solutions, ceci dans le but de caractériser la précision des estimées obtenues compte tenu de caractéristiques données pour les capteurs utilisés. Le coût calcul étant aussi un facteur limitant à l'utilisation de ces techniques ensemblistes, nous avons choisi de simplifier, voire d'éliminer la phase de bisection.

Nous avons proposés deux algorithmes dédiés au problème de l'estimation d'attitude AM :

- l'algorithme LOOP qui n'implante pas de phase de bisection, mais uniquement une phase de contraction ;
- l'algorithme LoopBorne qui effectue une bisection "décentrée" mais qui ne gère pas de liste pour le domaine solution.

Au regard des résultats obtenus, plusieurs conclusions se dégagent :

- si on a des contraintes d'exécution temporelle de l'algorithme d'identification, par exemple parce que l'estimation de l'attitude est réalisée en temps réel, il faut considérer l'algorithme LOOP. En outre, le domaine englobant l'ensemble des solutions sera simple (un parallélépipède rectangle) ;
- en revanche, si l'utilisateur n'a pas de contraintes temporelles, l'algorithme LoopBorne permet d'obtenir un domaine englobant l'ensemble des solutions de forme simple, en un temps calcul "raisonnable" avec en général, une précision accrue par rapport à LOOP ;
- si l'utilisateur souhaite une description fine du domaine des solutions, il faudra implanter un algorithme classique d'identification ensembliste intégrant les phases de bisection et de contraction.

La capacité des algorithmes a été démontrée sur des données simulées et réelles. Dans le premier cas, plusieurs paramétrisations de l'attitude ont été explorées, à savoir avec les angles de Cardan, ou bien encore avec un quaternion. Notons que pour ce dernier, on a levé l'hypothèse de norme strictement égale à l'unité. En conséquence, l'ensemble solution retourné contient tous les quaternions de norme unité recherchés, ces derniers étant situés sur l'hypersphère de rayon 1 dans l'espace à 4 dimensions. Par contre, certains quaternions retournés ne sont pas de norme unité : il ne sont pas à strictement parler des solutions au problème d'estimation de l'attitude. Cet artifice nous a permis de résoudre le problème d'estimation de l'attitude par des techniques de calcul par intervalles avec un temps de calcul fini.

Dans la suite de ce chapitre, nous nous intéressons à l'estimation de l'attitude dans le cas où le problème à résoudre est dynamique c'est-à-dire le cas où les gyromètres sont

pris en compte. Ce problème sera appelé AGM.

3.3 Estimation de l'attitude à partir des mesures AGM : cas dynamique

Le deuxième objectif de ce chapitre est de traiter le problème de l'estimation de l'attitude par des techniques ensemblistes dans le cas dynamique. La résolution de ce problème se fera par une approche par observateur d'état. Dans le cadre du calcul ensembliste, ce type d'observateur est appelé *observateur à intervalles*. [26] donne la structure d'un observateur d'attitude pour le problème dynamique dans le cas de calcul standard. Ici, nous avons mis en œuvre un observateur ensembliste qui permet d'obtenir une attitude garantie d'un corps rigide dans le domaine ensembliste. On utilisera avantageusement les algorithmes LOOP et LoopBorne. A partir des 9 mesures fournies par un triaxe magnétomètre, un triaxe accéléromètre et un triaxe gyromètre, nous utiliserons le système d'estimation de l'attitude décrit par la figure 3.40 qui permet de choisir une des deux équations suivantes selon le formalisme retenu pour paramétrer l'attitude, à savoir le quaternion unitaire ou les angles d'Euler (voir chapitre 2, équations (2.2.10) et (2.2.11)) :

$$\begin{cases} [\dot{q}] = \frac{1}{2} * [\Omega](\vec{\omega} + \overrightarrow{[\eta_G]}) * [q] \\ \begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix} + \begin{pmatrix} \overrightarrow{[\eta_M]} \\ \overrightarrow{[\eta_A]} \end{pmatrix} = [\mathcal{C}]([q]) * \begin{pmatrix} \vec{g} \\ \vec{h} \end{pmatrix} \end{cases} \quad (3.3.1)$$

ou bien

$$\begin{cases} \begin{pmatrix} [\dot{\varphi}] \\ [\dot{\theta}] \\ [\dot{\psi}] \end{pmatrix} = \begin{pmatrix} 1 & [\tan]([\theta])[\sin]([\varphi]) & [\tan]([\theta])[\cos]([\varphi]) \\ 0 & [\cos]([\varphi]) & -[\sin]([\varphi]) \\ 0 & \frac{[\sin]([\varphi])}{[\cos]([\theta])} & \frac{[\cos]([\varphi])}{[\cos]([\theta])} \end{pmatrix} * \begin{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \\ \begin{pmatrix} [\eta_G \ x] \\ [\eta_G \ y] \\ [\eta_G \ z] \end{pmatrix} \end{pmatrix} \\ \begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix} + \begin{pmatrix} \overrightarrow{[\eta_M]} \\ \overrightarrow{[\eta_A]} \end{pmatrix} = [\mathcal{C}]([\varphi], [\theta], [\psi]) * \begin{pmatrix} \vec{g} \\ \vec{h} \end{pmatrix} \end{cases} \quad (3.3.2)$$

$\mathcal{C}(\cdot)$ est la matrice de rotation dépendant du quaternion ou des angles de Cardan. $\vec{b}_M = (m_1 \ m_2 \ m_3)^T$ sont les mesures de triaxe magnétomètre. $\vec{b}_A = (m_4 \ m_5 \ m_6)^T$ sont les mesures de triaxe accéléromètre. $\vec{\omega} = (\omega_x \ \omega_y \ \omega_z)^T = (m_7 \ m_8 \ m_9)^T$ sont les mesures de triaxe gyromètre. Ces mesures pourront par exemple être fournies par le capteur MAG3. Le vecteur de champ magnétique normalisé est donné par $\vec{h} = (\frac{1}{2} \ 0 \ \frac{\sqrt{3}}{2})^T$. Le vecteur gravité normalisé est $\vec{g} = (0 \ 0 \ 1)^T$. Les vecteurs $\vec{\eta}_M$, $\vec{\eta}_A$ et $\vec{\eta}_G \in \mathbb{R}^3$ représentent les vecteurs des bruits de mesures. On rappelle que dans le cadre de ce travail, ces derniers sont considérés inconnus mais bornés, et on les note $\overrightarrow{[\eta_M]}_{3 \times 1} \subset \mathbb{IR}^3$, $\overrightarrow{[\eta_A]}_{3 \times 1} \subset \mathbb{IR}^3$ et $\overrightarrow{[\eta_G]}_{3 \times 1} \subset \mathbb{IR}^3$.

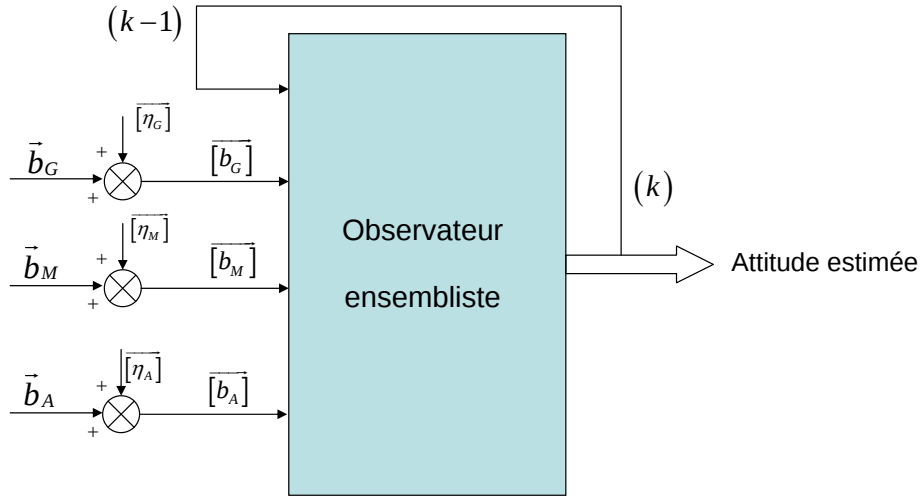


Figure 3.40 – Principe d'un observateur ensembliste d'attitude. Les données d'entrée sont fournies par une centrale inertielle

Remarque 27. On voit que l'équation (3.3.2) est non linéaire et que sa discrétisation est délicate à réaliser. Pour ces raisons, et compte tenu de la simplicité de l'équation (3.3.3), nous avons choisi de ne retenir que la formulation quaternion pour la fusion des données AGM.

L'équation dynamique à temps discret fait intervenir une récurrence temporelle entre les variables du système échantillonné. Pour la période d'échantillonnage T_e (ici $T_e = 0.01$ s) et à l'instant d'échantillonnage $t_k = T_e * k, k = 1, 2, \dots, N$, le modèle à temps discret est donné par :

$$\begin{cases} [q_k] = [\Theta](\vec{\omega}, [\eta_G]) * [q_{k-1}] \\ \begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix} + \begin{pmatrix} [\eta_M] \\ [\eta_A] \end{pmatrix} = [C]([q_k]) * \begin{pmatrix} \vec{g} \\ \vec{h} \end{pmatrix} \end{cases} \quad (3.3.3)$$

où $[\Theta](\vec{\omega}, [\eta_G]) = e^{\frac{1}{2}[\Omega](\vec{\omega} + [\eta_G]) * T_e}$ est la matrice exponentielle [60] qui peut être calculée par l'équation suivante :

$$\begin{cases} [\Theta](\vec{\omega}, [\eta_G]) = I_{4 \times 4} + \sum_{i=1}^n \frac{\left(\frac{1}{2} * [\Omega](\vec{\omega} + [\eta_G]) * T_e\right)^i}{i!} \\ \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \\ \vec{\omega} = (\omega_x \ \omega_y \ \omega_z)^T \end{cases} \quad (3.3.4)$$

Remarque 28. Dans notre travail, nous proposons de choisir n tel que $\max(\delta([\Omega]^i) < 1 \cdot 10^{-20}$, ce qui correspond à $n \leq 10$ avec les données décrites au paragraphe 3.2.2.1.

Notons également que pour réaliser une intégration garantie, il faut également majorer le reste de Taylor. Cette étape n'a pas été faite dans les résultats numériques rapportés ici.

L'équation (3.3.3) montre que l'observateur d'attitude d'un corps rigide sous le formalisme quaternion q est un problème dynamique non linéaire. On peut réécrire ce problème sous la forme d'un système dynamique général :

$$\begin{cases} [x_k] = [f_k]([x_{k-1}], [u_k]) + [H]([\xi_e]) \\ [y_k] = [h]([x_k], [u_k]) + [\xi_s] \end{cases} \quad (3.3.5)$$

où $[u]$ représente le vecteur intervalle des données d'entrées, $[y]$ est le vecteur intervalle des variables de sorties, $[\xi_e]$ et $[\xi_s]$ sont les vecteurs intervalle des bruits du système transformés par H , $[f]$ et $[h]$ sont les fonctions d'inclusion du problème.

Pour résoudre le système (3.3.5) avec un intervalle, V.G. Héctor-Moisés propose un observateur ensembliste [76] pour lequel *"Il s'agit d'un système auxiliaire capable de déterminer sur un domaine admissible $\mathbb{X}$ l'intervalle d'états estimés $[\tilde{x}]$, tel que les trajectoires $[\tilde{y}]$ issues de $[\tilde{x}]$ restent à l'intérieur de la mesure bornée de la sortie $[y]$ (tolérance du capteur)".* Trois cas doivent être considéré (voir figure 3.41) :

- $[\tilde{x}_1] \subset \mathbb{X}$ et $[y_1] = [h]([\tilde{x}_1])$ tels que $[y_1] \subset [y]$, alors $[x_1]$ est solution du problème d'estimation, $[\tilde{x}_1] \subset [\tilde{x}]$;
- $[\tilde{x}_2] \subset \mathbb{X}$ et $[y_2] = [h]([\tilde{x}_2])$ tels que $[y_2] \not\subset [y]$, mais $[y_2] \cap [y] \neq \emptyset$ donc nous ne pouvons rien dire par rapport à $[x_2]$ et donc $[x_2] \subset [\tilde{x}]$. Cependant comme il existe certains états de $[\tilde{x}_2]$ qui ne sont pas de solutions du problème, $[\tilde{x}_2]$ pourrait être encore réduit ;
- $[\tilde{x}_3] \subset \mathbb{X}$ et $[y_3] = [h]([\tilde{x}_3])$ tels que $[y_3] \not\subset [y]$, mais $[y_3] \cap [y] = \emptyset$, pourtant $[\tilde{x}_3]$ ne peut pas faire partie des solution et nous pouvons l'éliminer sans autres considérations.

Remarque 29. Dans le cadre de ce travail, l'observateur ensembliste/intervalle doit garantir la convergence des composantes de l'attitude (les variables d'état). En d'autres termes, notre intervalle d'état estimé $[\hat{x}]$ doit satisfaire :

$$[\hat{x}] \triangleq \{\forall x \in [x] \mid x \in [\tilde{x}_1]\} \cup \{\exists x \in [x] \mid x \in [\tilde{x}_2]\} \quad (3.3.6)$$

Différentes méthodes seront maintenant présentées pour résoudre le problème dynamique (3.3.5) avec les équations non-linéaires définies par les équations (3.3.3) et (3.3.2).

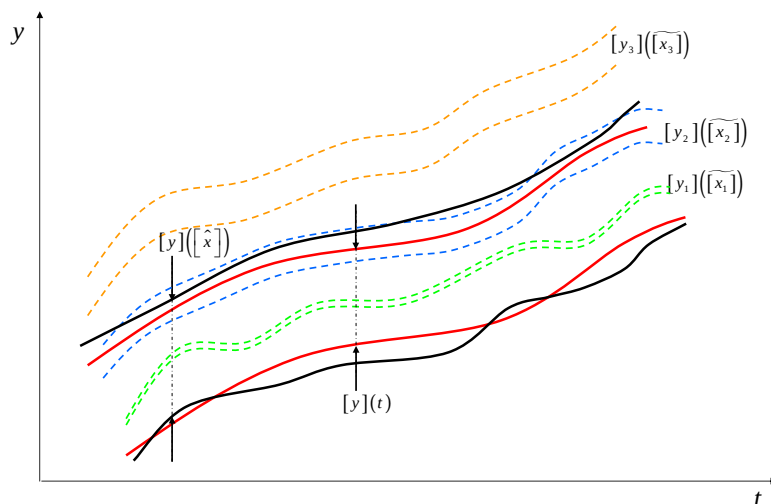


Figure 3.41 – Exemple d’une trajectoire $|\mathbf{y}|$ obtenue par un observateur ensembliste

3.3.1 Résumé de différents observateurs par intervalles

3.3.1.1 Filtre de Kalman à intervalles

[14] et [72] ont introduit la notion de filtre de Kalman avec utilisation du calcul ensembliste. Ainsi, les filtres *IKF* (Interval Kalman Filter) et *IEKF* (Interval Extended Kalman Filter) sont construits sur un principe similaire au filtre de Kalman standard. Le but de l'*IEKF* est de déterminer l'état sous forme intervalle $[\tilde{x}]$.

Extended Interval Kalman Filter

Main process :

$$\begin{aligned} [\tilde{x}_0] &= E\{[x_0]\} \quad (give) \\ [\tilde{x}_{k/k-1}] &= [f_k]([\tilde{x}_{k-1}]) \\ [\tilde{x}_k] &= [\tilde{x}_{k/k-1}] + G_k * ([y_k] - [h]([\tilde{x}_{k/k-1}])) \\ k &= 1, 2, \dots \end{aligned}$$

Coprocess :

$$\begin{aligned} [P_0] &= cov\{[x_0]\} \quad (give) \\ [M_{k-1}] &= \left[\frac{\partial f_k}{\partial x_k} \right]([\tilde{x}_{k-1}]) * [P_{k-1}] * \\ &\quad \left(\left[\frac{\partial f_k}{\partial x_k} \right]([\tilde{x}_{k-1}]) \right)^T + [H_{k-1}]([\tilde{x}_{k-1}]) * Q_{k-1} * ([H_{k-1}]([\tilde{x}_{k-1}]))^T \\ [G_k] &= [M_{k-1}] * \left(\left[\frac{\partial h}{\partial x} \right]([\tilde{x}_{k/k-1}]) \right)^T * \\ &\quad \left(\left[\frac{\partial h}{\partial x} \right]([\tilde{x}_{k/k-1}]) * [M_{k-1}] * \left(\left[\frac{\partial h}{\partial x} \right]([\tilde{x}_{k/k-1}]) \right)^T + R_k \right)^{-1} \\ [P_{k-1}] &= (I - [G_k] * \left[\frac{\partial h}{\partial x} \right]([\tilde{x}_{k/k-1}])) * [M_{k-1}] \\ k &= 1, 2, \dots \end{aligned}$$

Tableau 3.10 – Filtre de Kalman étendu sous forme intervalle (IEKF)

Pour le système (3.3.5), le filtre *IEKF* est décrit dans le tableau 3.10. Dans ce tableau, $[x_k] \subset \mathbb{R}^4$ et $[y_k] \subset \mathbb{R}^6$ sont les états donnés sous forme intervalle et les mesures intervalles des capteurs accéléromètre et magnétomètre. $E[x_0]$ est l'état initial (sous forme intervalle) et $[P_0]$ est la matrice de covariance initiale (sous forme intervalle). Q_k et R_k représentent les matrices de covariance sur le bruit de l'état et sur le bruit de mesure respectivement.

L'atout principal de l'*IEFK* est d'utiliser $[G_k] * ([y_k] - [h](\tilde{x}_{k/k-1}))$ dans la phase de correction. Cela permet de corriger l'état obtenu dans la phase de prédiction par intégration ensembliste à l'aide du vecteur d'innovation et d'une matrice de gain. Cette approche a été testée sur notre problème, et les résultats obtenus n'étaient pas satisfaisants (augmentation au cours du temps de la taille de l'ensemble solution). Aussi, ces résultats ne sont pas présentés dans ce manuscrit.

3.3.1.2 Algorithme Waltz

L'algorithme *FALL/CLIMB* qui est appliqué dans le problème de localisation de véhicule à partir de système multi-capteurs est décrit par Gning dans [5], [4], [24]. L'objectif principal de ces travaux est de résoudre le problème de localisation avec une nouvelle technique de calcul par intervalles, appelé algorithme de *FALL/CLIMB*. Il permet d'améliorer l'algorithme de Waltz, en particulier en découpant l'algorithme de Waltz en deux parties "FALL" et "CLIMB" afin d'améliorer le temps de calcul.

Lorsque l'on considère le cas dynamique de notre problème d'estimation, notre système et le système de Gning [24] ont plusieurs points communs mais possèdent également un point différent important. Ces derniers sont résumés ci-dessous :

1. points communs :
 - Les deux systèmes permettent d'estimer les états du problème car ils incluent l'ensemble des modèles statiques et dynamiques ;
 - Les données intervalles sont obtenues avec l'erreur maximale de mesure ;
2. différences :
 - Dans [24], il y a une modification de la période d'échantillonnage en fonction de la situation "cas statique" ou "cas dynamique" permettant de recorriger les états dynamiques grâce à l'identification des états par l'algorithme *FALL/CLIMB* après un certain temps ;
 - Dans notre approche, il n'y a aucune modification de la période d'échantillonnage. La période reste inchangée que l'on soit dans le cas statique ou le cas dynamique.

Cependant, l'analyse de notre problème montre qu'il existe des cycles, ce qui limite l'application des résultats de [24]. En outre, il n'est pas prouvé que l'intégration de la version discrétisée de l'équation dynamique converge vers la solution de l'équation à

temps continue. Ce point sera abordé au paragraphe 3.3.2.

L'algorithme *BPF* décrit dans [1] et [2] montre qu'il y a des cas où il y a non-intersection entre les solutions des équations dynamiques et statiques.

Remarque 30. Compte tenu des remarques ci-dessus, l'approche de Gning ne sera pas utilisée dans le cadre de ce manuscrit.

3.3.1.3 Algorithme max/min

Le principe de l'observateur décrit dans [39], [64], [63], [78], [40] où il est appliqué à différentes applications, est basé sur l'analyse du maximum ($\bar{f}(\cdot)$) et du minimum ($\underline{f}(\cdot)$) d'une fonction dynamique définie par (3.3.7). Pour un système d'équations $\dot{x}(t)$, ils ont proposé d'observer l'intervalle d'état à partir des équations :

$$\begin{cases} \tilde{x}^-(t) \leq \tilde{x}(t) \leq \tilde{x}^+(t) \\ \dot{x}^+(t) = \bar{f}(x^+(t)) \\ \dot{x}^-(t) = \underline{f}(x^-(t)) \end{cases} \quad (3.3.7)$$

La difficulté de cet approche est de déterminer les fonctions mathématiques de l'équation (3.3.7) afin de déterminer un encadrement de l'état recherché. Cette étape semble très difficile pour notre problème. Par exemple, pour $\dot{x}_3(t) = f(x_1, x_2) = x_1 * x_2$ avec x_1, x_2, x_3 les états du système, il n'existe pas de fonctions mathématiques pour $\dot{x}^+(t)$ et $\dot{x}^-(t)$. Dans ce cas, il est préférable d'utiliser le calcul par intervalles.

3.3.2 Algorithmes d'estimation de l'attitude dans le cas AGM utilisant une intersection ou de la rétropropagation dans la phase de correction

Dans cette partie, nous proposons deux algorithmes permettant de résoudre le problème de l'estimation ensembliste de l'attitude dans le cas AGM, à savoir les algorithmes EstimEI et EstimEC. EstimEI utilise une intersection des domaines respectivement prédits par l'équation dynamique et calculé avec les équation statiques dans la phase de correction. EstimEC quant à lui utilise la rétropropagation du domaine estimé par l'équation dynamique dans la résolution des équations statiques. Auparavant, on rappelle les problèmes liés à l'intégration de l'équation dynamique.

3.3.2.1 Problèmes liés à l'intégration numérique

L'intégration numérique d'une fonction à temps continu est un problème en soi. Cela peut être illustré par l'exemple suivant. Soit l'équation différentielle :

$$\dot{x}(t) = t^2 \quad (3.3.8)$$

sur l'intervalle $t \in [0.5, 1]$ s avec pour condition initiale $x_{t=0.5\text{ s}} = x_0 = 2$. Dans le domaine à temps continu, l'intégration de cette équation différentielle donne :

$$S_0 = \int_{0.5}^1 t^2 * dt = \left. \frac{t^3}{3} \right|_{0.5}^1 = \frac{1}{3} * \left(1 - \frac{1}{8}\right) = \frac{7}{24} \quad (3.3.9)$$

Dans l'espace discret avec un pas d'intégration $T_s = 0.5$ s, cette intégration, si elle est calculée par la méthode d'Euler, conduit à :

$$S_1 = discret \left\{ \int_{t=0.5}^{t=1} t^2 dt \right\} = t^2|_{t=1} * (1 - 0.5) = \frac{1}{2} > S_0 \quad (3.3.10)$$

Si les pas d'intégration Δt tend vers 0, on a $\lim_{\Delta t \rightarrow 0} (S_1 - S_0) = 0$, mais on ne peut pas choisir pour pas d'intégration discrete $T_s \approx 0$.

Lors d'une intégration par intervalles, notre objectif est également de calculer :

$$\begin{cases} S_2 = discret \left\{ \int_{t=0.5}^{t=1} t^2 dt \right\} = t^2|_{t=0.5} * (1 - 0.5) = \frac{1}{8} < S_0 \\ \lim_{\Delta t \rightarrow 0} (S_2 - S_0) = 0 \end{cases} \quad (3.3.11)$$

Dans l'exemple présent, t^2 est une fonction croissante sur l'intervalle considéré. On a donc toujours $S_2 < S_0 < S_1$, ce qui revient à $S_0 \in [S_2, S_1]$. Notons que suivant la fonction considérée, S_2 peut être plus grande que S_1 .

Exemple 3.3.1

Dans l'espace discret, l'intégration intervalle de $\left(\left[\int_{t_1}^{t_2} \dot{x} * dt \right] \right)$ (communément appelées forward et backward) peut être calculée par deux intégrations classiques $\int_{t_1}^{t_2} \dot{x}|_{t_1} * dt$ et $\int_{t_1}^{t_2} \dot{x}|_{t_2} * dt$. En outre, cela satisfait les propriétés d'une fonction d'inclusion :

$$\int_{t_1}^{t_2} \dot{x} * dt \in \left(\left[\int_{t_1}^{t_2} \dot{x} * dt \right] \right) \quad \forall x(t) : t \in [t_1, t_2] \quad (3.3.12)$$

Remarque 31. Si $\dot{x}(t)$ est monotone sur l'intervalle $[t_1, t_2]$ ou si $\dot{x}(t)$ n'est pas monotone mais satisfaisait $\dot{x}(t) \in [\dot{x}] \forall t \in [t_1, t_2]$, on obtient :

$$\int_{t_1}^{t_2} \dot{x} * dt \in \left[\int_{t_1}^{t_2} \dot{x} * dt \right] \in \left[\int_{t_1}^{t_2} [\dot{x}] * dt \right] \quad (3.3.13)$$

où $[\dot{x}] = [\dot{x}]_{t_1} \cup [\dot{x}]_{t_2}$.

Pour diminuer l'erreur entre l'intégration à temps continu et l'intégration numérique [25], on peut utiliser d'autres approches que les méthodes d'Euler, par exemple une méthode de Runge-Kutta¹ d'ordre 4. On peut également utiliser une approche de type Taylor. L'application de la méthode de Taylor dans l'arithmétique par intervalle est présentée par exemple dans [56] [57] [62]. On a alors :

$$\begin{aligned} \dot{x} &= f(x, t), x(t_0) \in X_0 \\ \Rightarrow [x_{j+1}] &= [x_j] + \sum_{i=1}^{k-1} h_j^i * f^{[i]}(t_j, [x_j]) + h_j^k * f^{[k]}([t_j, t_{j+1}], [\tilde{x}_j]) \end{aligned} \quad (3.3.14)$$

$$\text{où : } \begin{cases} h_j = t_{j+1} - t_j \\ [x_j] + [0, h_j] * f^{[k]}([t_j, t_{j+1}], [\tilde{x}_j]) \subseteq [\tilde{x}_j] \end{cases}$$

[57] a montré que :

$$\delta([x_{j+1}]) = \delta([x_j]) + \sum_{i=1}^{k-1} h_j^i * \delta(f^{[i]}(t_j, [x_j])) + h_j^k * \delta(f^{[k]}([t_j, t_{j+1}], [\tilde{x}_j])) \geq \delta([x_j]) \quad (3.3.15)$$

Ainsi, $\delta([x_{j+1}])$ peut augmenter plus rapidement qu'une correction de l'erreur lorsque k augmente. Dans le cas du problème que nous traitons (voir l'équation dynamique (3.3.3) avec les quaternions, on peut observer une divergence due à l'intégration (voir les résultats donnés dans le tableau 3.11). Pour appliquer l'approche de Taylor à notre travail, on simplifie (3.3.15) et on prend :

$$\delta([q_{j+1}]) = \delta([q_j]) + \sum_{i=1}^k h_j^i * \delta([f]^{[i]}(t_j, [q_j])) \quad (3.3.16)$$

Le tableau 3.11 représente l'augmentation de $\delta([q_{j+1}])$ lorsque l'on utilise l'équation (3.3.16), pour les données simulées à l'instant $t = 3.3$ s et pour k variant de 0 à 2.

Dans le cadre de ce travail, nous utiliserons l'équation (3.3.13) pour l'intégration par intervalles de l'équation dynamique. On supposera que pour T_s suffisamment petit, la monotonie de $\dot{x}$ est garantie. Notons que la contrainte de norme unité du quaternion est

1. voir par exemple <http://media4.obspm.fr/public/dea/cours/chapitre3/souschapitre2/section4/INTRODUCTION/page1.html>

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

i=330	$\delta([q_{i+1}])$	Augmentation largeur (0.1%)
$k = 0$	$(3.62471\text{e-}4, 1.55779\text{e-}3, 2.02544\text{e-}3, 3.94875\text{e-}3)^T$	x x x
$k = 1$	$(3.75862\text{e-}4, 1.56329\text{e-}3, 2.03497\text{e-}3, 3.95919\text{e-}3)^T$	$(3.4, 1.4, 2.4, 2.6)^T$
$k = 2$	$(3.75993\text{e-}4, 1.56355\text{e-}3, 2.03517\text{e-}3, 3.95938\text{e-}3)^T$	$(3.4, 1.5, 2.5, 2.7)^T$

Mesures à $t = 3.3$ s ; $[q_i] = ([9.92974 \cdot 10^{-1}, 9.93245 \cdot 10^{-1}], [1.11611 \cdot 10^{-1}, 1.12687 \cdot 10^{-1}], [-2.16661 \cdot 10^{-2}, -2.00002 \cdot 10^{-2}], [2.52473 \cdot 10^{-2}, 2.88808 \cdot 10^{-2}])^T$ avec $T_s = 0.01$ s.

Tableau 3.11 – Comparaison de la largeur des composantes de $[q]$ pour différents ordres du développement de Taylor

levée afin de pouvoir traiter le problème simplement avec la représentation quaternion. On s'assure néanmoins que l'intervalle retourné par l'étape d'intégration est de très petite largeur devant 1.

Le système de l'équation (3.3.3) devient :

$$\begin{cases} [\hat{q}_k] = [\Theta]([\tilde{\omega}], [\eta_G]) * [\hat{q}_{k-1}] \\ [\tilde{\omega}] = [\omega(k)] \cup [\omega(k-1)] \\ \begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix} + \begin{pmatrix} \overrightarrow{[\eta_M]} \\ \overrightarrow{[\eta_A]} \end{pmatrix} = [\mathcal{C}]([\hat{q}_k]) * \begin{pmatrix} \vec{g} \\ \vec{h} \end{pmatrix} \end{cases} \quad (3.3.17)$$

On présente maintenant les algorithmes *EstimEI* et *EstimEC* qui seront utilisés pour résoudre le problème de l'observation de l'attitude d'un corps rigide à partir de mesures fournies par une centrale inertielle constituée d'un triaxe accéléromètre, d'un triaxe magnétomètre et d'un triaxe gyromètre lorsque les bruits sont supposés inconnus mais bornés.

3.3.2.2 Algorithme EstimEI : mise en œuvre d'une intersection dans la phase de correction

L'objectif de cet algorithme est de remplacer dans le filtre de Kalman par intervalles *IEKF* la phase de correction qui utilise G_k par une opération d'intersection entre l'intervalle solution calculé par intégration et l'intervalle solution obtenu par la fusion des mesures accéléromètre et magnétomètre, cette fusion des mesures statiques étant réalisée par l'algorithme LOOP. Ce mode de calcul semble naturel puisque l'état estimé doit satisfaire les deux équations du système (3.3.5).

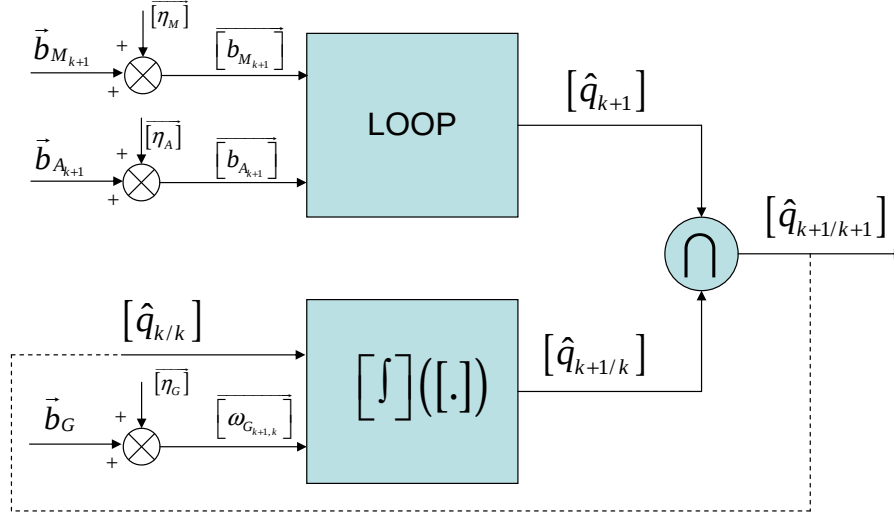


Figure 3.42 – Principe de l'algorithme EstimEI pour une paramétrisation de l'attitude avec un quaternion

Les résultats obtenus pour les données simulées avec l'algorithme EstimEI sont présentés par les figures 3.43 à 3.45. Sur la figure 3.43, on voit que la valeur du quaternion théorique q_{th} est bien incluse dans l'intervalle de solutions $[\hat{q}_{est}]$ retourné par l'algorithme, avec une largeur maximale pour les différentes composantes 0.01191 et une largeur minimale pour les composantes de l'intervalle solution retourné de $3.9861 \cdot 10^{-6}$, voir figure 3.44. L'erreur d'angle $[\beta_e]$ sur le vecteur unitaire associé au quaternion est donnée sur la figure 3.45 : on voit que cet angle est du même ordre de grandeur que celui estimé dans le cas statique. La figure 3.46 montre les différentes situations de correction pouvant être rencontrées. Il existe des situations où $[\hat{q}_{k+1/k}]$ (calculé via l'équation dynamique) n'améliore pas $[\hat{q}_{k+1}]$ (calculé par équation statique) (valeur 2 sur le graphique). Dans d'autres situations, il est nécessaire de corriger $[\hat{q}_{k+1/k}]$ par l'opération d'intersection avec $[\hat{q}_{k+1}]$ (valeur de 0 sur la figure). Enfin, aux instants $t = 3.24 \text{ s}$ et $t = 8.54 \text{ s}$, $[\hat{q}_{k+1/k}]$ est inclus dans $[\hat{q}_{k+1}]$. Remarquons que la valeur 2 (i.e. $[\hat{q}_{k+1}] \in [\hat{q}_{k+1/k}]$) correspond aux situations quasi-statiques tandis que 0 (i.e. $[\hat{q}_{k+1}] \cap [\hat{q}_{k+1/k}] \neq \emptyset$ et un intervalle n'est pas inclus entièrement dans l'autre) apparaît dans les phases dynamiques.

Nous présentons maintenant l'algorithme EstimEC qui est une modification de EstimEI dans lequel l'exploitation des mesures accéléromètre et magnétomètre est modifiée.

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

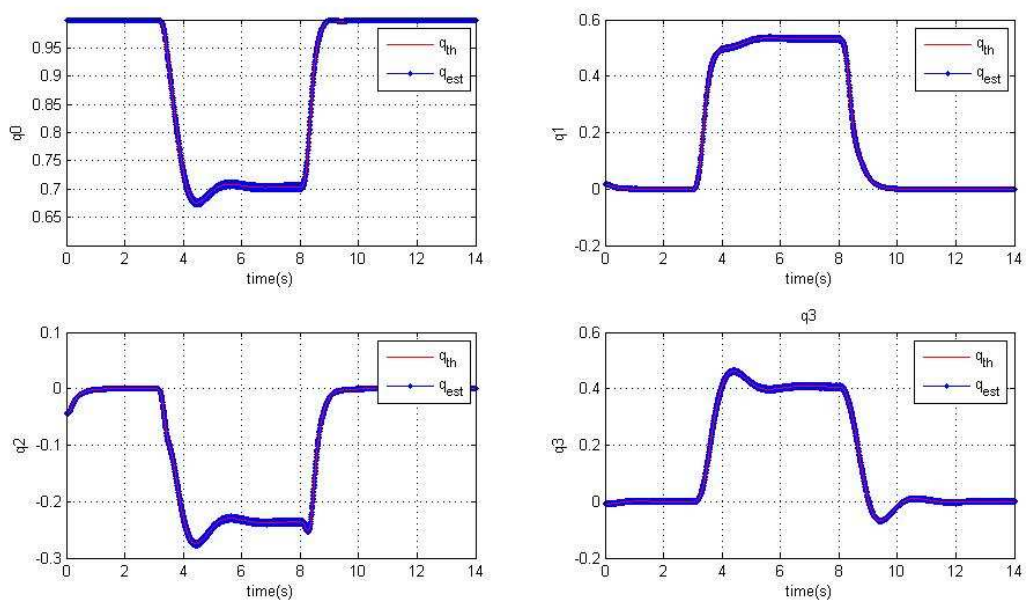


Figure 3.43 – Domaine solution obtenu avec l'algorithme EstimEI pour la paramétrisation avec le quaternion de l'attitude. Données simulées

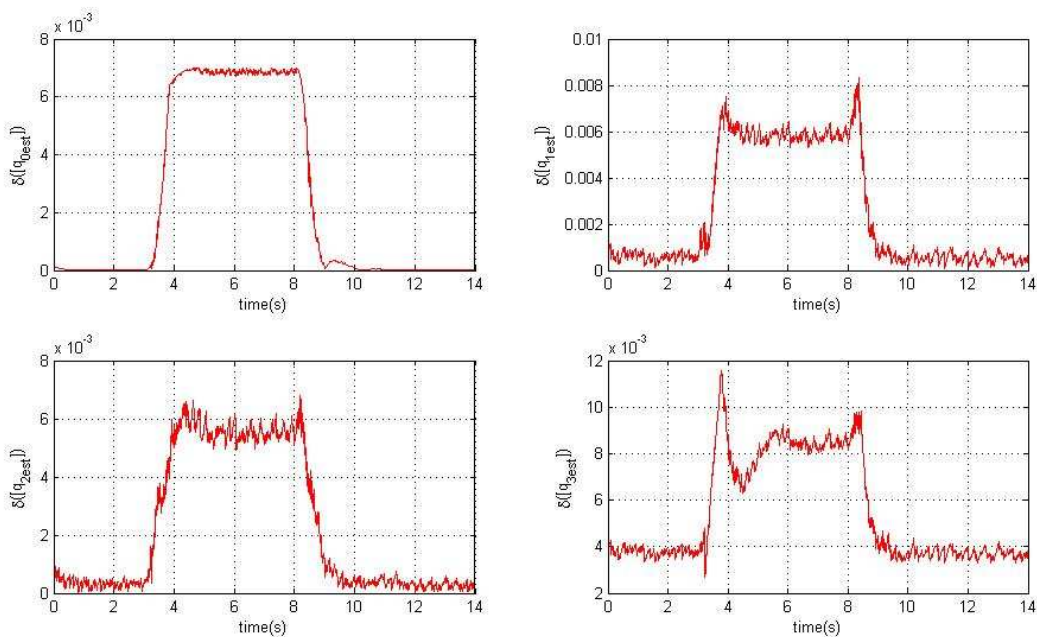


Figure 3.44 – Largeur du domaine solution suivant les 4 composantes du quaternion lorsque l'on utilise l'algorithme EstimEI. Données simulées

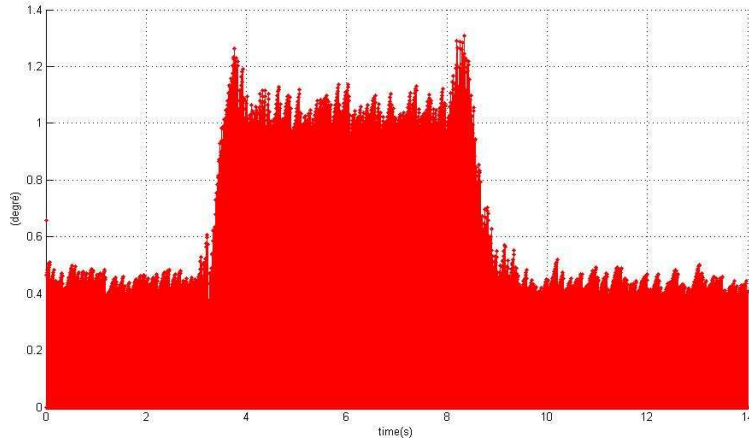
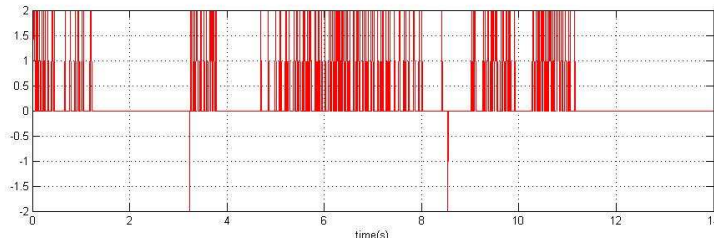


Figure 3.45 – $[\beta_e]$ (2.4.2) pour l'algorithme EstimEI. Données simulées



2 correspond aux cas où $[\hat{q}_{k+1}]$ est inclus dans $[\hat{q}_{k+1/k}]$

0 correspond aux cas où l'intervalle $[\hat{q}_{k+1/k}]$ et $[\hat{q}_{k+1}]$ s'intersectent

-2 : correspond aux cas où $[\hat{q}_{k+1/k}]$ est inclus dans $[\hat{q}_{k+1}]$

Figure 3.46 – Différentes situations de la phase d'intersection de l'algorithme EstimEI. Données simulées.

3.3.2.3 Algorithme EstimEC : prise en compte de l'intervalle obtenu via l'équation dynamique lors du traitement des données AM

L'algorithme EstimEC, donné dans la figure 3.47, est proposé pour observer les états du système (3.3.17) à partir des mesures AGM. Ici, l'algorithme LOOP est utilisé de manière à fusionner les mesures AM avec le $[\hat{q}_{k+1/k}]$ estimé obtenue à partir des mesures gyrométriques et de l'intégration de l'équation dynamique. Ce choix est guidé par le fait que l'attitude obtenue doit satisfaire à la fois les équations statiques AM et les équations dynamiques.

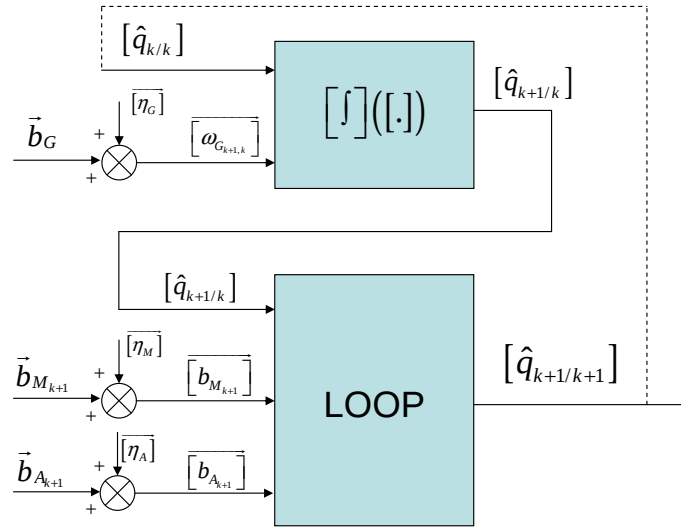


Figure 3.47 – Principe de l'algorithme EstimEC sur les composants de quaternion unitaire

Les résultats obtenus avec l'algorithme EstimEC pour les données simulées sont exposés par les figures 3.48 à 3.50. Sur la figure 3.48, on a tracé les intervalles obtenus sur chacun des composants du quaternion ainsi que le quaternion théorique. On peut constater que ce dernier est bien inclus dans l'ensemble solution retourné par l'algorithme EstimEC. La figure 3.49 donne la largeur des intervalles pour chacune des composantes du quaternion calculé avec EstimEC. On constate une légère diminution de la largeur des intervalles par rapport à ceux obtenus avec EstimEI. L'erreur d'angle $[\beta_e]$ sur le vecteur

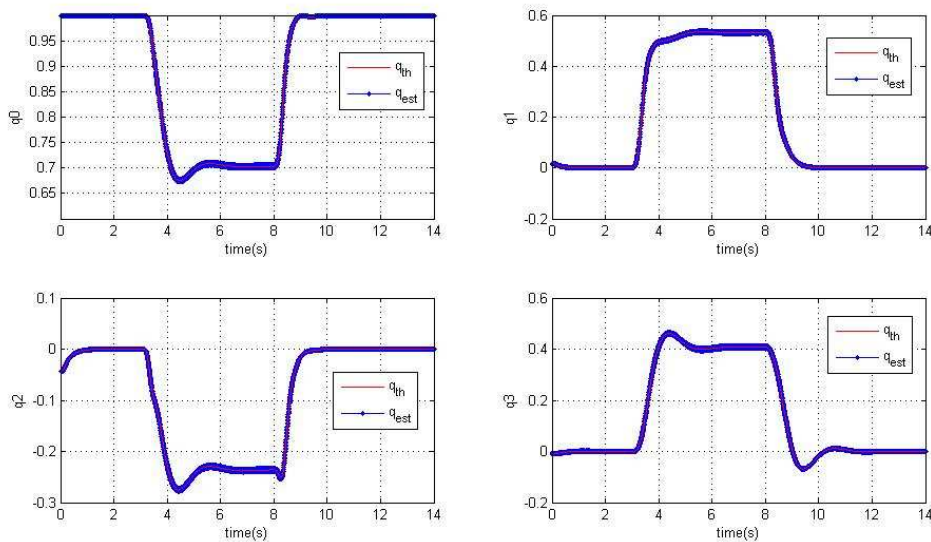


Figure 3.48 – Intervalles pour chacun des composants du quaternion obtenu avec l'algorithme EstimEC. Données simulées

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

unitaire associé au quaternion est donnée sur la figure 3.50 : on voit que cet angle est pratiquement tout le temps plus faible que celui obtenu avec EstimEI, ce qui confirme que la taille des intervalles obtenus avec EstimEC a diminué par rapport à celle obtenue par EstimEI. Ainsi, la figure 3.46 montre clairement que les domaines solutions obtenus avec EstimEC sont plus petits (puisqu'inclus) que ceux obtenus avec EstimEI, sauf aux instants $t = 3.24\text{ s}$, $t = 3.25\text{ s}$ et $t = 8.37\text{ s}$ où les résultats obtenus par l'algorithme EstimEC ne sont pas intégralement inclus dans ceux obtenus par EstimEI.

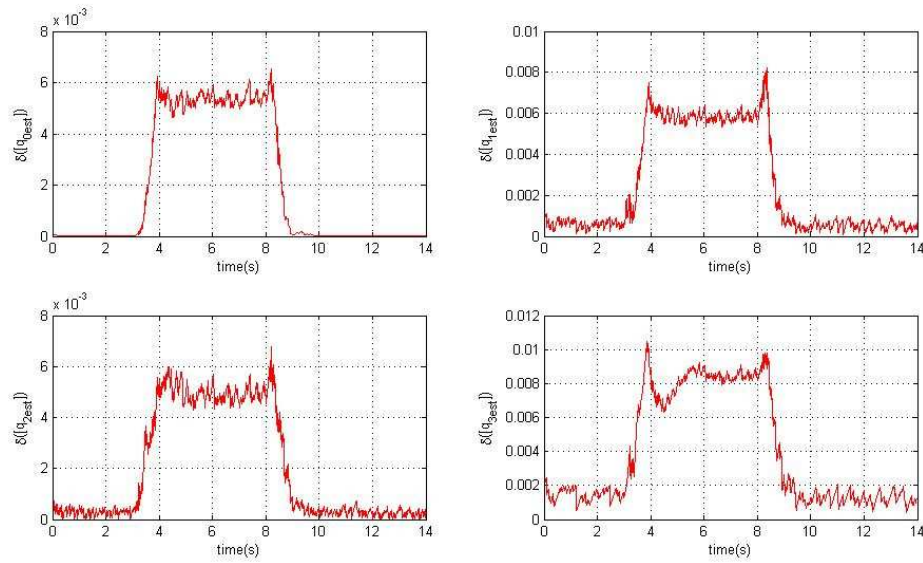


Figure 3.49 – Largeur des intervalles solution obtenus avec EstimEC pour chacune des composantes du quaternion des composantes de quaternion. Données simulées

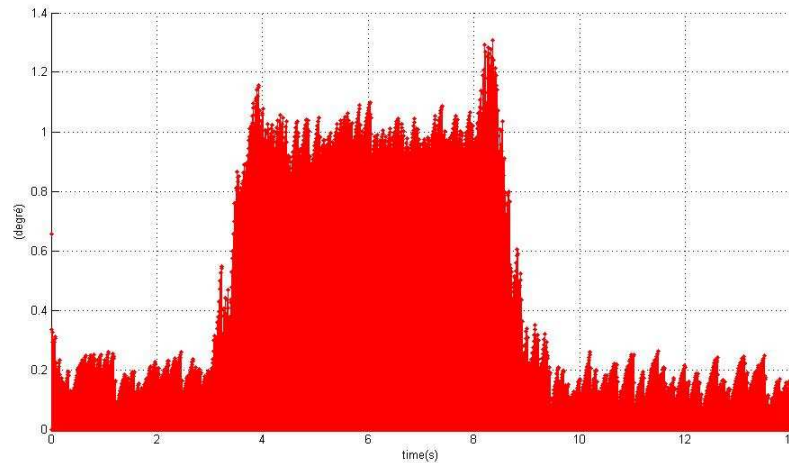
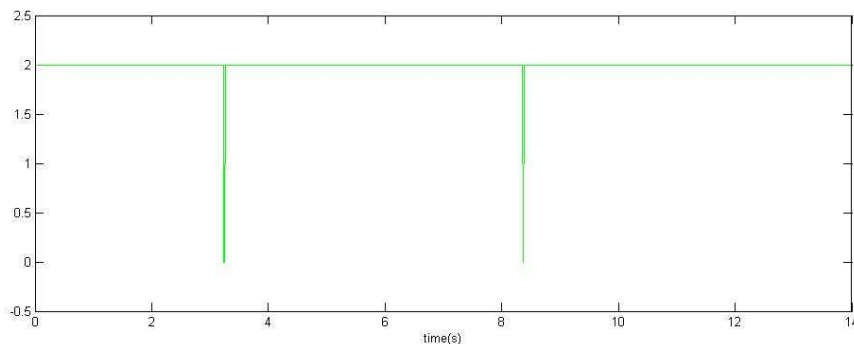


Figure 3.50 – $[\beta_e]$ (2.4.2) pour les résultats obtenus avec l'algorithme EstimEC. Données simulées



2 correspond aux cas où $[\hat{q}_{k/k}]$ obtenu par EstimEC est inclus dans $[\hat{q}_{k/k}]$ obtenu par EstimEI
 0 correspond aux cas où $[\hat{q}_{k/k}]$ de EstimEC n'est pas inclus dans $[\hat{q}_{k/k}]$ de EstimEI

Figure 3.51 – Comparaison entre $[\hat{q}_{k/k}]$ obtenus respectivement avec EstimEI et EstimEC. Données simulées

Le temps de calcul de l'algorithme EstimEC est inférieur à celui d'EstimEI pour quasiment tous les instants d'échantillonnage, cf. la figure 3.52. Cette diminution du temps de calcul s'explique par le fait que le domaine de recherche de l'algorithme LOOP est restreint dans l'algorithme EstimEC puisqu'on exploite le domaine obtenu dans la phase de prédiction.

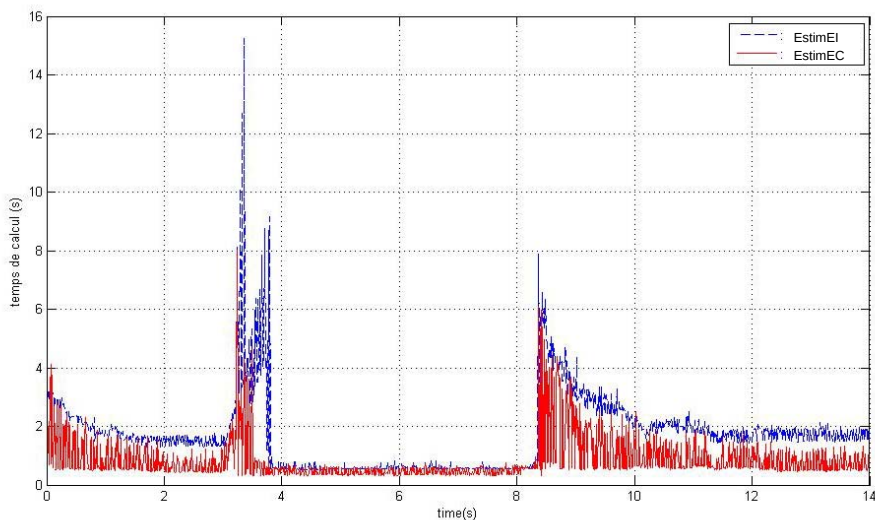


Figure 3.52 – Comparaison des temps de calcul sur PC Intel i7 1.6GHz entre les algorithmes EstimEI et EstimEC. Données simulées

3.3.3 Conclusion partielle sur l'estimation de l'attitude dans le cas dynamique

Dans cette partie, nous avons présenté les résultats obtenus avec l'algorithme EstimEI puis avec l'algorithme EstimEC pour le problème de l'estimation de l'attitude à partir de mesures fournies par une centrale inertielle constituée d'un triaxe gyromètre (mesures G), d'un triaxe accéléromètre (mesures A) et d'un triaxe magnétomètre (mesures M).

L'algorithme EstimEI implémente deux phases disjointes (et parallèle), à savoir une phase d'intégration de l'équation dynamique en prenant en compte les mesures G et une phase d'exploitation des mesures AM . Une troisième phase d'intersection des domaines obtenus lors des deux premières phases permet d'obtenir l'estimation ensembliste de l'attitude recherchée.

L'algorithme EstimEC est quant à lui constitué de deux phases réalisées séquentiellement. La première consiste en une phase d'intégration de l'équation dynamique, en utilisant les mesures G . Il donne de meilleurs résultats mais le calcul ne peut pas être réalisé en deux parties pouvant "tourner" en parallèle. L'algorithme EstimEC ne peut pas trouver de solution lorsque les mesures AGM sont aberrantes (à cause des fautes par exemple). Ce problème sera traité en détail dans le chapitre suivant.

3.4 Conclusion sur l'estimation d'attitude par des techniques ensemblistes

Dans ce chapitre, nous avons présenté des méthodes développées pour résoudre le problème de l'estimation de l'attitude par des approches ensemblistes dans les cas dits statique (mesures accéléromètre et magnétomètre) et dynamique (mesures accéléromètre, gyromètre et magnétomètre). L'intérêt d'utiliser ces approches ensemblistes est qu'elles offrent des résultats garantis : on est certain que l'ensemble solution retourné contient les solutions recherchées. De plus, les bruits sont représentés par des intervalles, sans hypothèses sur les distributions sous-jacentes. On dit alors qu'on est en présence de "bruit inconnu mais borné". Les algorithmes que nous avons développés ont été validés sur des données simulées, ce qui a permis d'évaluer la pertinence des démarches proposées et de comparer nos résultats à ceux obtenus par des approches de l'état de l'art, en particulier l'algorithme SIVIA dans le cas statique. Les algorithmes développés dans le cas AM ont également été utilisés sur des données réelles acquises à partir d'une centrale inertielle 3DM-GX1.

L'estimation de l'attitude par des approches ensemblistes retourne, à partir de mesures données sous forme d'intervalles, un ensemble de solutions contenant toutes les at-

Chapitre 3. Estimation ensembliste de l'attitude à partir de mesures fournies par une centrale inertielle

titudes cohérentes avec les mesures. Deux paramétrisations de l'attitude ont été considérées. La première est la représentation classique avec les angles de Cardan, l'ensemble solution retourné étant donné sous la forme d'intervalles sur chacun des angles (i.e un cube). La deuxième représentation de l'attitude utilise le quaternion unitaire. Dans notre travail, on a levé la contrainte de norme du quaternion égale à l'unité et on a vérifié que l'ensemble retourné (sous forme d'hypercube) contient bien toutes les solutions recherchées.

Dans le cas statique, nous avons développé deux algorithmes pour déterminer l'attitude à partir de mesures AM : l'algorithme LOOP et LoopBorne. L'algorithme LOOP est en pratique uniquement basée sur un contracteur ; il est particulièrement adapté si on souhaite faire une implantation en temps réel. L'algorithme LoopBorne contient une phase de bisection, mais décentrée. Cette phase de bisection augmente le temps calcul.

Dans le cas dynamique (mesures AGM), nous avons proposé deux algorithmes, à savoir l'algorithme EstimEI et l'algorithme EstimEC. L'algorithme EstimEI fait tourner en parallèle deux tâches disjointes, à savoir l'intégration de l'équation dynamique et exploitation des mesures AM. Une troisième phase d'intersection des domaines obtenus lors des deux premières phases permet d'obtenir l'estimation ensembliste de l'attitude recherchée. L'algorithme EstimEC exploite dans un premier temps l'équation dynamique (phase d'intégration) puis dans un second temps les mesures AM.

De plus, on a à la fois une estimation de l'attitude mais également une information sur la précision de cette valeur qui est donné par la taille de l'intervalle.

Le chapitre suivant se propose d'adapter les algorithmes développés ici au cas du diagnostic de la centrale d'attitude.

Chapitre 4

Diagnostic des capteurs d'une centrale inertielle AGM

Dans le chapitre 3, nous avons estimé l'attitude d'un corps rigide dans un espace à 3 dimensions dans les cas "statique" et "dynamique" à partir des mesures fournies par une centrale inertielle constituée d'un triaxe d'accéléromètres, d'un triaxe de magnétomètres et d'un triaxe de gyromètres. Nos résultats garantissent que la valeur "théorique" est toujours comprise dans le pavé solution obtenu. Cependant, si un défaut apparaît sur la mesure, nous devons pouvoir le détecter afin de ne pas prendre en considération le capteur en défaut dans la phase d'estimation de l'attitude.

Dans ce chapitre, le problème du diagnostic d'une centrale d'attitude est considéré. Pour cela, le diagnostic sera basé sur des techniques de calcul par intervalles à partir des mesures AM (pour le cas statique) et AGM (pour traiter le cas dynamique). Dans un premier temps, quelques principes généraux sur le diagnostic seront rappelés. Dans un second temps, en se basant sur le principe de l'algorithme LOOP présenté au chapitre précédent, nous proposerons une méthode de diagnostic utilisant une approche par intervalles, pour détecter, localiser et identifier (FDI) un défaut sur un des capteurs de la centrale inertielle. Dans la suite du chapitre, deux algorithmes vont être considérés pour :

1. le cas statique :

- dans ce cas, la représentation par le formalisme des angles de Cardan sera considérée ;
- on analysera la capacité de détecter des défauts (ou des perturbations) sur l'ensemble des 6 mesures AM, grâce à un calcul par intervalles de différents sous-ensembles de mesures ;
- on évaluera la capacité de localiser les défauts, grâce à la comparaison du symptôme avec une table de signature ;
- on évaluera également la capacité d'identifier les défauts par difference entre le

vecteur de mesures et sa reconstruction ;

2. le cas dynamique :

- on analysera la capacité de détecter des défauts sur l'ensemble de mesures AGM en utilisant le formalisme quaternion pour la paramétrisation de l'attitude ;
- on étudiera la possibilité de localiser les défauts capteurs grâce à une comparaison avec une table de signature ;
- on verra que l'étape d'identification n'est pas réalisable sur l'ensemble des 9 mesures. En effet, l'identification peut, pour le moment, se faire uniquement sur les mesures AM. Malheureusement, la reconstruction des mesures gyromètre n'a pu être réalisée avec le calcul par intervalles.

4.1 Diagnostic et centrales inertielles

Une facette de ce travail est le diagnostic de défaut pouvant survenir sur une centrale inertielle. En pratique, des mesures issues de la centrale inertielle peuvent être erronées à cause de perturbations mais aussi à cause d'une panne. L'objectif est dans ce dernier cas de détecter au plus vite les capteurs en défaut.

En effet, dans de nombreuses applications, il est imprudent de se focaliser uniquement sur le problème de l'estimation de l'attitude sans avoir la garantie d'utiliser des mesures saines. C'est en particulier le cas pour les applications où la sécurité des personnes et des biens sont en jeu.

Nous présenterons maintenant succinctement quelques notions et approches existantes rencontrées dans la littérature diagnostic. L'objectif de ce paragraphe n'est pas de faire une revue étendue de ces approches, le lecteur intéressé étant invité à se reporter aux nombreux ouvrages publiés sur ce thème.

4.1.1 Définitions et méthodes de diagnostic utilisant une arithmétique "standard"

Remarque 32. On appelle arithmétique standard une arithmétique qui manipule des grandeurs "ponctuelles" à l'opposé de grandeurs "intervalles".

Un défaut capteur est interprétable comme une incohérence entre la valeur réelle y^* et la valeur mesurée y . [22] considère plusieurs situations pour un défaut capteur :

- le blocage. Dans ce cas, la mesure reste bloquée à une certaine valeur à partir d'un instant t_f inconnu :

$$y = \text{const}, \quad \forall k > t_f ; \quad (4.1.1)$$

- la mise à zéro de la mesure. Le capteur cesse de fonctionner et sa valeur de sortie est zéro après l'apparition du défaut à un instant donné. Ce défaut est usuellement dû à des problèmes électriques ou de communication :

$$y = 0, \quad \forall k > t_f ; \quad (4.1.2)$$

- la dérive ou défaut de type additif. Ce type de défaut est le plus courant dans les capteurs analogiques, souvent produit par un changement de température interne ou à des problèmes de calibration. La valeur de sortie du capteur est la valeur théorique plus un terme additif :

$$y = y + \xi, \quad \forall k > t_f \quad (4.1.3)$$

où ξ peut être constant ou variant dans le temps (par exemple, dérive lente) ;

- un défaut de type multiplicatif. Un facteur multiplicatif modifie la valeur nominale :

$$y = \xi y^*, \quad \forall k > t_f. \quad (4.1.4)$$

Le problème de diagnostic est de réaliser une surveillance du comportement du système et d'apporter toutes les informations possibles quand un dysfonctionnement apparaît et ceci, le plus tôt possible. En règle générale, la procédure complète du diagnostic d'un système peut être décomposée en trois "sous-tâches" :

- *détection du défaut* : l'objectif est d'avoir une indication le plus tôt possible quand "quelque chose ne va pas bien sur le système" ;
- *localisation du défaut* : l'objectif ici est de déterminer précisément le composant en défaut ;
- *identification du défaut* : la forme et l'amplitude du défaut sont déterminées dans cette sous-tâche.

comme la figure 4.1 [15] le montre.

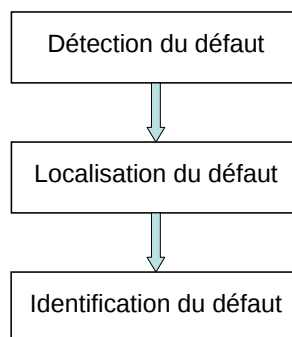


Figure 4.1 – Les 3 étapes du diagnostic

Dans la littérature du diagnostic par des outils numériques standard, la tâche du diagnostic est réalisée suivant les étapes suivantes [21] :

- *générer le vecteur de résidus*. Il est nécessaire de rechercher des indicateurs qui soient sensibles aux défauts. Ces indicateurs sont généralement intégrés dans un vecteur de résidus. La génération du vecteur de résidus est réalisée en général en temps réel ;
- *évaluer le résidu*. Cette étape est aussi appelée prise de décision. Dans cette étape, on analyse chaque composante du vecteur du résidu pour décider s'il y a ou non présence d'un défaut. Plusieurs techniques peuvent être utilisées comme un simple test de dépassement de seuil sur les valeurs instantanées [84], ou bien encore on peut faire appel à la théorie de la décision statistique [8], [7], ou bien encore des techniques de décision floue. On génère aussi un symptôme qui est l'image du vecteur de résidu dans le domaine booléen ("0" correspond à un système sain, "1" à un système en défaut).

Remarque 33. Sans la tâche de détection de défaut, on ne peut pas réaliser la tâche de localisation et par conséquent, il est impossible d'identifier le défaut. Par conséquent, il est nécessaire de se concentrer sur la génération du résidu [85]. En effet, si la génération du résidu est correctement conçue, la détection du défaut est une tâche aisée.

Plusieurs méthodes, qui permettent de générer des résidus en utilisant des modèles analytiques, ont été considérés par les automaticiens [15],[43]. Ces approches utilisent le modèle mathématique et sont généralement classées en trois catégories [22],[36] :

- les techniques à base d'observateur [41][75][37] ;
- les approches espace de parité [23],[48] ;
- l'utilisation de l'estimation paramétrique [35].

Dans le cas le plus simple, le résidu est obtenu par la différence entre les sorties du système et celles estimées avec un modèle :

$$r_k = y_k - \hat{y}_k \quad (4.1.5)$$

où k est le numéro de la composante considérée des vecteurs, $\hat{y}$ peut être obtenu par simulation du modèle à partir des entrées mesurées du système ou par l'observation à partir d'entrées/sorties. En théorie, ce résidu est égal à zéro en présence d'un système sain, et différent de zéro lorsqu'un système présente un défaut :

$$\begin{aligned} r_k = 0 &\Rightarrow \text{système sain} \\ r_k \neq 0 &\Rightarrow \text{système en défaut} \end{aligned} \quad (4.1.6)$$

Chaque fois qu'un vecteur de résidus est généré, celui-ci est transformé en *symptôme*. Lorsque les symptômes sont comparés avec une table appelée *tableau de signature* (voir un exemple de table de signature dans le tableau 4.1), l'étape de *localisation* est réalisée. Dans ce tableau, "1" correspond au cas où le résidu est sensible au défaut considéré et "0" correspond au fait que le résidu est insensible à ce défaut. Par exemple, pour la table

	f_1	f_2	f_3
r_1	1	0	0
r_2	0	1	0
r_3	0	0	1

Tableau 4.1 – Tableau de signature pour l'étape de *localisation*

4.1, le résidu r_1 est sensible à la faute f_1 mais ne l'est pas à aux fautes f_2 et f_3 . Par contre, le résidu r_2 est sensible à la faute f_2 mais ne l'est pas aux fautes f_1 et f_3 . Enfin, le résidu r_3 est sensible à la faute f_3 mais ne l'est pas aux fautes f_1 et f_2 . De plus, si les colonnes de la table sont différentes entre elles, alors le défaut peut être localisé. Inversement, si deux colonnes sont identiques, alors ces deux défauts ne sont pas différenciables.

L'exemple donné ci-dessus est simpliste et la situation sur des systèmes réels est beaucoup plus complexe.

L'étape de localisation est effectuée, comme on vient de le voir, via une table de signature, localisante généralement. Deux schémas ont été proposés [15],[36] (voir également la figure 4.2) :

- un schéma dédié, où chaque r_i , $i = 1, \dots, N$ est sensible à un défaut seulement et reste insensible aux autres. Si T_i est un seuil prédéfini, alors on a :

$$|r_{i,k}| > T_i \Rightarrow f_{i,k} \neq 0, \quad i = 1 \dots N, \quad (4.1.7)$$

- un schéma généralisé, où chaque résidu r_i , $i = 1, \dots, N$ est sensible à tous les défauts sauf un. Par conséquent, on a [85] :

$$\begin{aligned} |r_{i,k}| &< T_i \\ |r_{j,k}| &> T_j, j = 1, \dots, i-1, i+1, \dots, N \\ &\Rightarrow f_{i,k} \neq 0, i = 1 \dots N, \end{aligned} \quad (4.1.8)$$

L'avantage de l'approche par schéma généralisé est qu'elle est moins restrictive que celle du schéma dédié. En effet, compte tenu des mesures disponibles, si on prend moins de mesures, on obtient un problème sans solution. La figure 4.2 montre ces deux approches à travers un banc de N observateurs surveillant un système à N sorties. Pour localiser les défauts sur les capteurs, on ne prend qu'une seule sortie dans le cas du schéma dédié. Par contre, on prend toutes les entrées et $(N - 1)$ sorties pour chaque observateur dans le cas du schéma généralisé.

4.1.2 Utilisation d'approches ensemblistes pour le diagnostic

Un grand nombre de méthodes présentées dans la littérature du diagnostic sont basées sur une arithmétique dite "standard", les calculs étant effectués avec des grandeurs

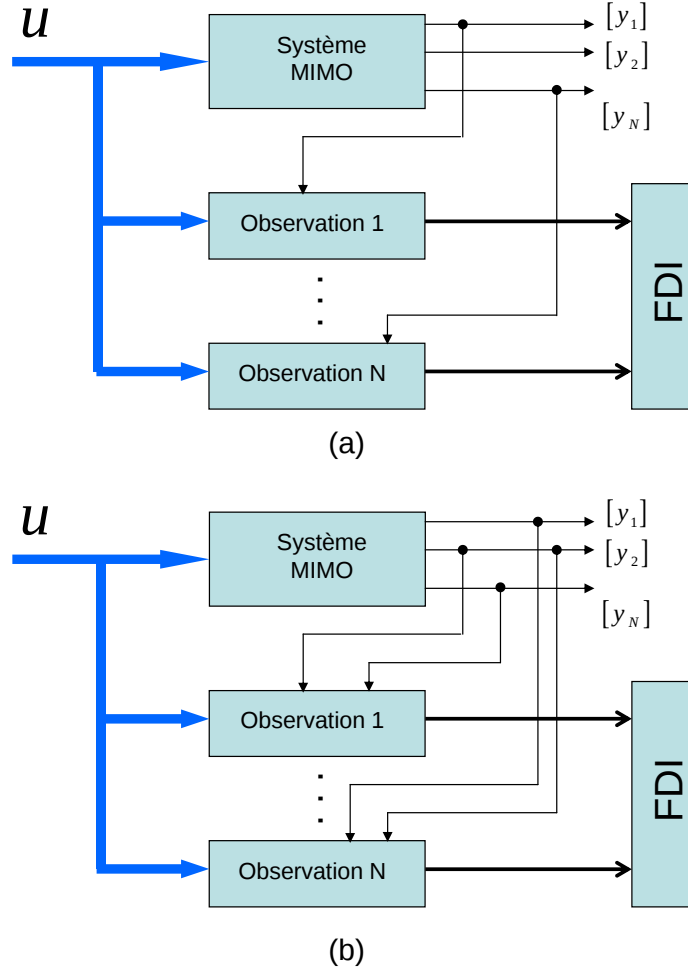


Figure 4.2 – Principe de location par une génération du vecteur de résidus. (a) schéma dédié, (b) schéma généralisé

ponctuelles, à l'opposé de grandeurs représentées par un intervalle. Néanmoins, différents travaux ont abordé le diagnostic par des approches ensemblistes. Par exemple, [52] [3] recherchent la valeur 0 dans les résidus intervalles $[r_k]$ pour détecter les défauts. [66] utilise l'opération d'intersection entre la mesure et son estimée pour la recherche de $[r_k] \ni 0$. [61] propose une détermination plus stricte basée sur la vérification d'un sous-ensemble entre l'estimé et sa limite. On détermine les max/min des mesures estimées à un instant donné [42] ou sur une fenêtre temporelle glissante [6] qui n'inclue pas la mesure car des défauts sont survenus dans le système. Cependant, ces approches n'exploitent pas explicitement un CSP.

4.1.3 Diagnostic d'une centrale inertielle

Comme déjà signalé à plusieurs reprises dans le chapitre précédent, les équations de mesures considérées sont non linéaires, quelque soit la paramétrisation de l'attitude retenue. En conséquence, dans le cadre de ce travail, nous devons mettre au point un algorithme de diagnostic adapté à un système non-linéaire.

Dans [27], nous trouvons une étude de différentes approches que l'on peut rencontrer pour le diagnostic et la reconfiguration en cas de défaut dans le domaine de l'aéronautique et du spatial. La première approche utilise l'espace de parité, visiblement très utilisée dans l'aérospatiale pour diagnostiquer les centrales inertielles. Par exemple, dans [67], 7 équations ont été générées et utilisées pour détecter et localiser un défaut simple sur un axe uniquement. Notons que le capteur utilisé est composé de gyromètres et d'accéléromètres et que le défaut ne peut pas être identifié. La deuxième approche utilise les réseaux neurones. L'erreur quadratique d'estimation entre les mesures des gyromètres et leur estimées est calculée [54] [55]. Si cette erreur dépasse un certain seuil, alors cela est interprété comme la présence d'un défaut sur le système. La troisième approche est basée sur la génération de résidu par observateur ou estimateur. Les résidus sont générés à partir de la différence entre la mesure et son estimée. Dans [28], la détection de défauts d'une centrale d'attitude du vaisseau spatial MARS Express est considérée. Les défauts sur 12 capteurs (3 accéléromètres, 3 magnétomètres, 3 gyromètres, plus un GPS) embarqués dans un hélicoptère ont été détectés et localisés. Chaque résidu est insensible à tous les capteurs excepté un. Enfin, les études dans [73] sont réalisées autour d'un quadrotor avec des défauts sur tous les capteurs embarqués dans le drone (accéléromètres, magnétomètres et gyromètres).

Notons cependant que toutes ces méthodes utilisent le modèle mécanique du système sur lequel sont montés les capteurs, ce qui comporte des limitations, en particulier si ce modèle est mal ou pas connu (cas des applications bio-médicales par exemple où les capteurs sont montés sur un être vivant dont on ne connaît pas a priori les mouvements).

4.2 Description des défauts ajoutés aux mesures simulées

Dans la suite de ce chapitre, nous traiterons uniquement le cas d'un défaut additif sur un capteur. Nous allons considérer le même scénario simulé qu'au chapitre 3. Cependant, nous allons introduire un défaut additif sur l'une des mesures suivant le protocole suivant :

- un défaut transitoire constant entre $t = 2\text{ s}$ et $t = 4\text{ s}$. Un biais d'amplitude 0.5 (unité à définir suivant la mesure considérée) est ajouté à la mesure. Ce défaut, d'une grande amplitude pour les mesures AGM, permet de réaliser une analyse

qualitative de l'intérêt de la démarche proposée ;

- un défaut en rampe, appliqué de $t = 6$ s à $t = 9$ s. La rampe de la pente égale à 0.2 (unité à définir suivant la mesure considérée) est ajouté. Ce défaut a pour valeur maximale 60% du maximum d'une mesure AM. Ce défaut en rampe permettra d'analyser le niveau de défaut minimum que l'on peut détecter avec l'algorithme proposé.

Notons d'ores déjà que la sensibilité de détection (i.e. l'amplitude de défaut minimum que l'on peut détecter) dépend de l'attitude dans laquelle on est. La figure 4.3 illustre le protocole de défaut considéré ici, c'est-à-dire le défaut transitoire constant ainsi que le défaut transitoire en rampe.

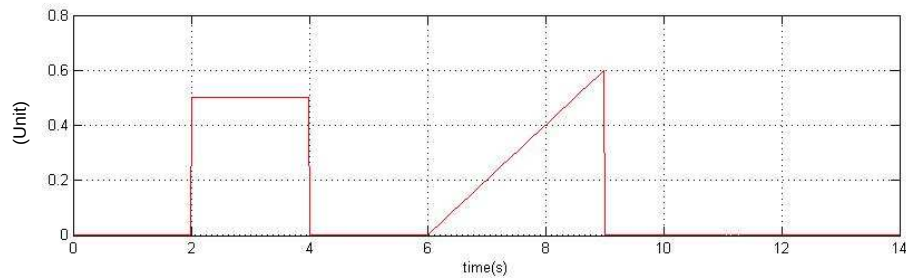


Figure 4.3 – Illustration des défauts considérés (en simulation)

4.3 Comportement des algorithmes d'estimation d'attitude en présence d'un défaut dans le jeu de mesures

Les algorithmes présentés dans le chapitre 3 présentent une particularité intéressante par rapport aux approches classiques : lorsque le jeu de mesures utilisé pour déterminer l'attitude est incohérent, l'ensemble solution retourné est vide [58]. Cette incohérence peut en particulier être due à la présence d'un (ou des) défaut(s).

Les figures 4.4 et 4.5 montrent respectivement l'utilisation de LOOP lorsqu'un défaut additif est simulé sur le capteur accéléromètre suivant l'axe x , et le résultat obtenu au cours du temps lorsque l'attitude est paramétrée avec le quaternion. On constate effectivement que lorsque le capteur est en défaut, l'ensemble solution retourné est vide.

Les figures 4.6 et 4.7 montrent une situation similaire lorsque le magnétomètre suivant l'axe y est en défaut. Cette fois, on utilise la paramétrisation de l'attitude avec les angles de Cardan. On constate ici encore que le quaternion simulé q_{th} est bien contenu dans

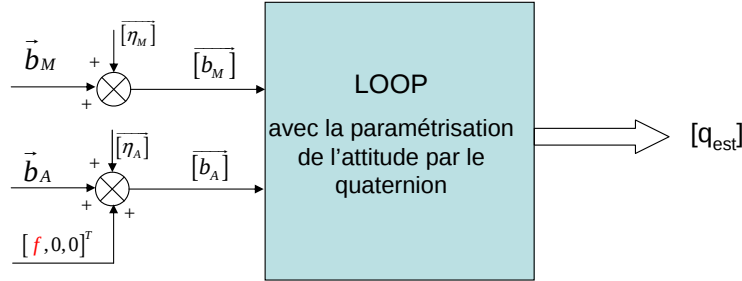


Figure 4.4 – Utilisation de l'algorithme LOOP (quaternion) avec un défaut sur m_4 , i.e. accéléromètre suivant l'axe x

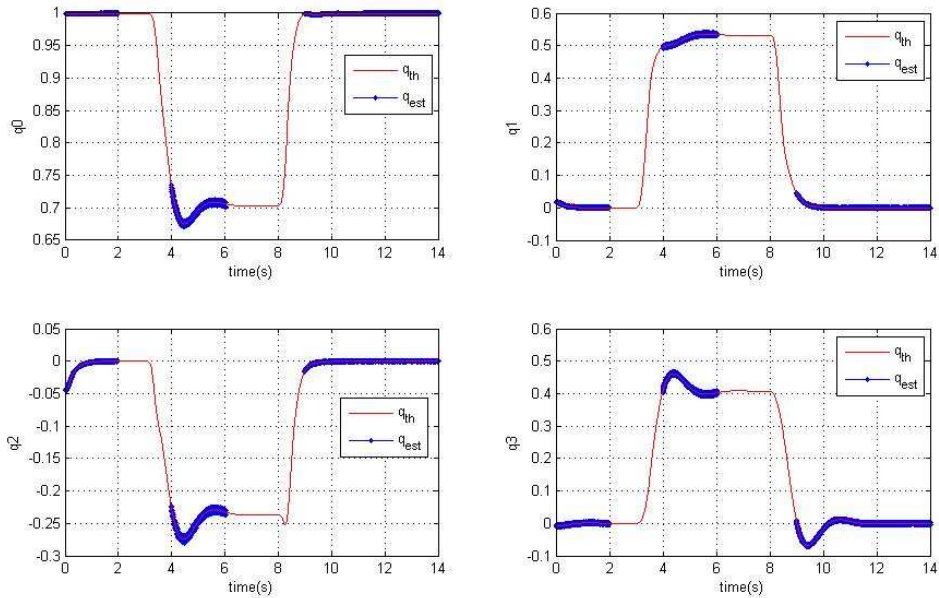


Figure 4.5 – Résultats pour l'algorithme LOOP (quaternion) avec un défaut sur m_4 (simulation)

l'ensemble solution estimé $[q_{est}]$ durant les phases saines et que l'algorithme détecte les incohérences entre les mesures puisque le domaine retourné lorsqu'un défaut est apparu est vide.

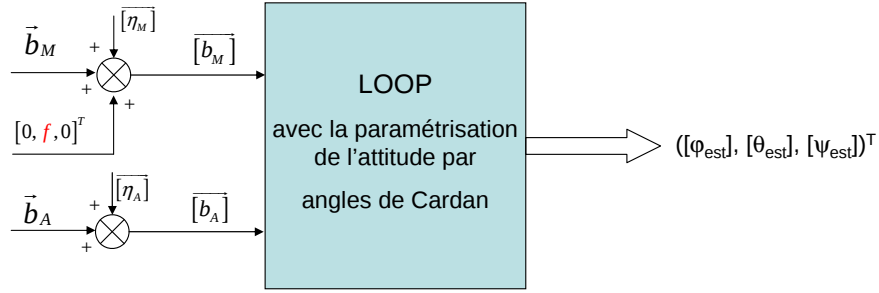


Figure 4.6 – Principe de l'algorithme LOOP (angle de Cardan) avec un défaut sur m_2 , i.e. magnétomètre suivant l'axe y

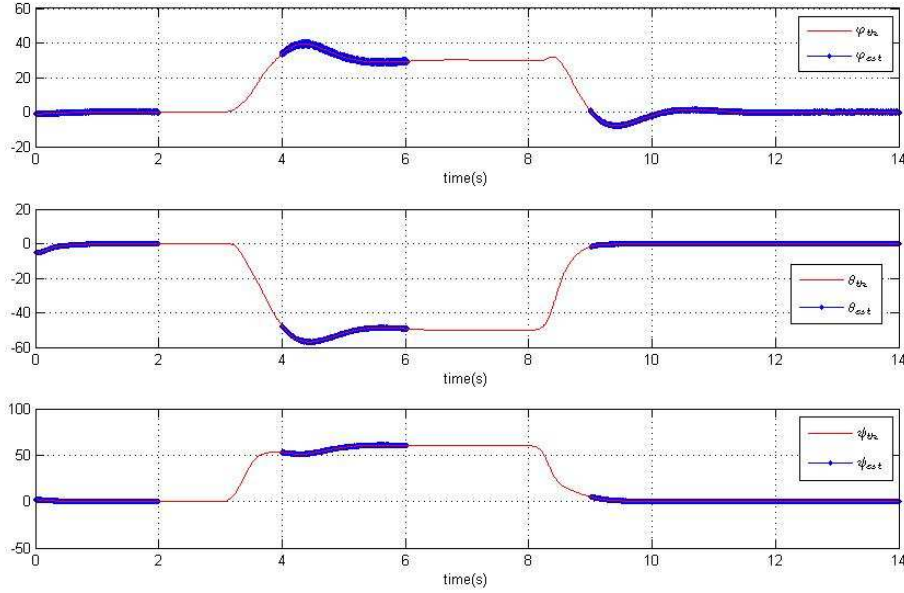


Figure 4.7 – Résultats pour l'algorithme LOOP (angle de Cardan) avec un défaut sur m_2 (simulation)

Remarque 34. Pour le cas statique, on a constaté que la détection est sans retard à la détection dans le cas où le niveau du défaut est suffisant pour être détecté. Grâce à la structure "forward-backup" utilisée par LOOP, on obtient en quelques itérations pour solution l'ensemble vide pour les systèmes d'équations (3.2.8) ou (3.2.9) comme par exemple, aux instants $t = 2 s$ ou $t = 6.02 s$. De plus, le temps de traitement avec LOOP (pour la représentation quaternion) dans le cas de mesures incohérentes est égal par exemple à l'instant $t = 2 s$ à $0.104 s$ ou encore à $0.296 s$ à l'instant $t = 6.03 s$. Ces résultats ont été obtenus dans l'environnement Matlab, avec la boîte à outils IntLab, sur un ordinateur intel i7 1.6GHz. Ce temps de calcul n'est pas compatible

avec un traitement en temps réel des mesures puisque l'échantillonnage est réalisé avec une période d'échantillonnage de 10 ms . Néanmoins, un recodage de nos fonctions en C pourrait permettre d'obtenir un temps calcul compatible avec l'application temps réel.

Les figures 4.8 et 4.9 présentent une situation similaire lorsqu'un défaut sur le troisième gyromètre est simulé. L'algorithme d'estimation utilisé est EstimEC présenté au chapitre précédent, avec une paramétrisation de l'attitude par un quaternion. Les conclusions sont similaires aux deux situations présentées ci-dessus dans le cas statique.

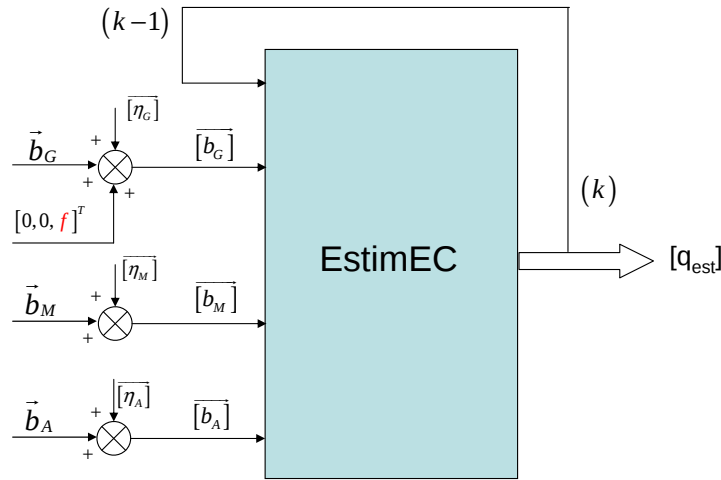


Figure 4.8 – Principe de l'algorithme EstimEC avec un défaut sur m_g , i.e. gyromètre suivant l'axe z

Remarque 35. Dans le cas dynamique, l'algorithme EstimEC détecte le défaut appliqué à $t = 2\text{ s}$ à l'instant $t = 2.01\text{ s}$. Par contre on constate que le défaut en rampe appliqué à partir de $t = 6\text{ s}$ est détecté avec retard puisque ce défaut est détecté seulement à partir de $t = 6.41\text{ s}$. Ce retard à la détection est explicable par le fait que l'amplitude du défaut est trop faible pour être détectée. Notons que lorsque l'algorithme détecte à l'instant $(k-1)$ une incohérence, le quaternion à l'instant k ne peut pas être calculé car il nécessite le quaternion à $(k-1)$. Aussi, afin de reprendre l'estimation d'attitude, nous avons fait une remise à jour de l'attitude aux instants $t = 5\text{ s}$ et $t = 9\text{ s}$.

Remarque 36. Notons que les algorithmes développés dans le chapitre précédent (ainsi qu'illustré avec LOOP et EstimEC) permettent de détecter des défauts simples ou simultanés. L'incohérence de l'ensemble de mesures utilisé par l'algorithme implique un ensemble solution vide.

Compte tenu des observations faites ci-avant, nous allons mettre en œuvre deux

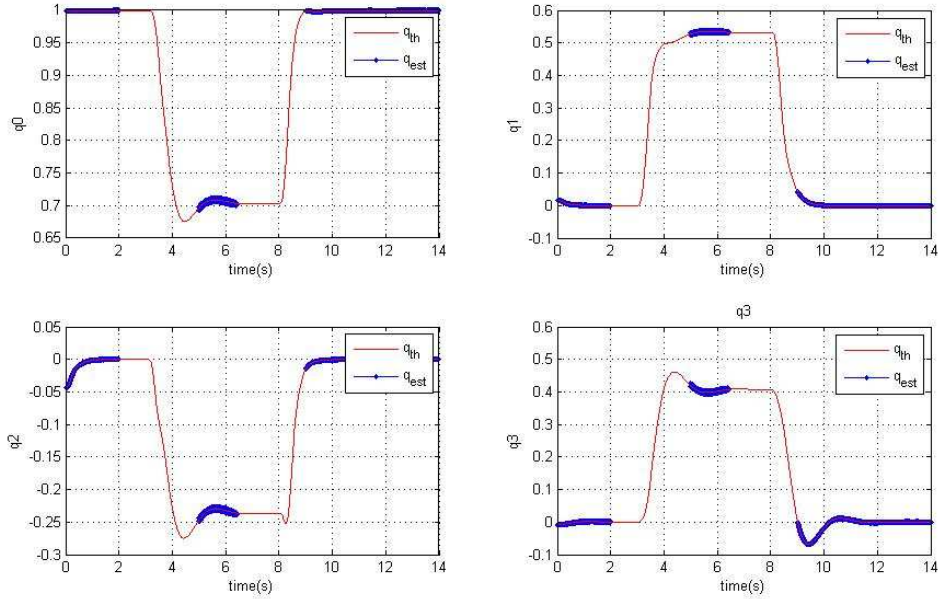


Figure 4.9 – Résultats pour l'algorithme EstimEC avec un défaut sur m_9 (simulation)

familles d'algorithmes de diagnostic pour une centrale inertielle basées sur des approches ensemblistes, respectivement pour le cas statique AM et le cas dynamique AGM.

4.4 Diagnostic d'une centrale inertielle, cas des mesures AM

On présente maintenant trois algorithmes qui ont été développés pour le diagnostic des capteurs AM d'une centrale inertielle. Le premier algorithme est adapté de l'algorithme LOOP pour laquelle on prend 5 des 6 mesures AM. En sortie, on obtient soit l'ensemble des attitudes recherchées, soit un ensemble vide dans le cas où une mesure parmi les 5 n'est pas "consistante", ce qui permet de conclure que dans le sous-ensemble de 5 mesures considéré, une des mesures est en défaut. Le second algorithme est une utilisation directe de LOOP dans laquelle on remplace la mesure non prise en considération dans l'estimation de l'attitude par l'intervalle $[-1; 1]$. Le principal intérêt de cette seconde approche est qu'on utilise le même contracteur dans les 6 estimations d'attitude. La troisième approche associe à la seconde une phase de reconstruction de la mesure, ce qui permet de construire un résidu classique en faisant une différence entre les 6 mesures réelles et les 6 mesures simulées.

Retenons que dans les trois approches, l'algorithme LOOP joue un rôle central puis-

qu'il ne retourne pas de résultat (ensemble solution vide) lorsque les données en entrée ne sont pas consistantes.

4.4.1 Détection/localisation de défauts AM avec 5/6 mesures AM

Nous avons vu précédemment comment une mesure aberrante pouvait être détectée en utilisant l'algorithme LOOP.

Cependant, l'étape de localisation doit être réalisée. C'est pourquoi, dans un deuxième temps, nous allons considérer le cas statique avec moins de 6 mesures AM afin de développer une structure d'estimateurs permettant la localisation.

Dans le paragraphe 3.2, nous avons expliqué que "le graphe de contraintes", dans le cas des angles de Cardan ou du quaternion, ne peut pas être ramené clairement à un arbre. De plus, il faut répéter plusieurs fois certains calculs (avec des équations complexes, voir (3.2.9) et (3.2.8) pour diminuer la largeur du pavé solution obtenu. Alors, il est possible que le pavé ne diminue pas significativement lorsque l'on ne prend pas en compte l'une des 6 mesures AM. En outre, on peut rencontrer des situations (comme dans le cas avec 6 mesures où la valeur "0" est incluse dans des intervalles utilisés dans des divisions. Les résultats obtenus avec un algorithme de diagnostic estimant l'attitude à partir de 5 sur 6 mesures en utilisant le formalisme des angles de Cardan ont été présentés dans [58]. L'algorithme implanté dans [58] est schématisé sur la figure 4.10. Comme on peut le voir, on est en présence d'un schéma généralisé. 6 algorithmes LOOP sont implantés en parallèle, chacun nécessitant l'écriture d'un système différent d'équations ne considérant que 5 mesures sur 6. La sortie de chaque estimateur est l'attitude obtenue à partir du sous-ensemble de 5 mesures sur 6 disponibles.

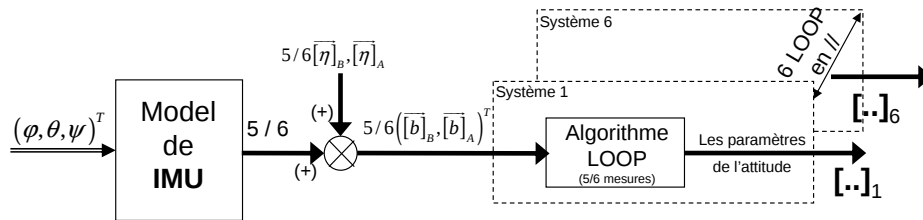


Figure 4.10 – Estimation de l'attitude avec 5/6 mesures AM. Données simulées, [58]

En l'absence de défaut, les 6 algorithmes en parallèle retournent chacun une estimation de l'attitude. Par contre lorsque l'un des capteurs est en défaut, seul un des estimateurs ensemblistes retourne un ensemble non vide de solutions, les 5 autres estimateurs utilisant en entrée la mesure corrompue.

Remarque 37. Le principe de l'algorithme que nous avons proposé dans [58] présente une structure compliquée, d'autant plus lorsque le problème possède de nombreuses sorties puisqu'il faut réécrire une structure d'observateur pour chacun des sous-ensembles. Pour cette raison, on propose une autre approche.

4.4.2 Détection/localisation de défauts AM avec 5/6 mesures AM et un intervalle $[-1; 1]$

Dans le cas d'un banc d'observateurs en arithmétique "standard", lorsqu'une entrée est manquante (i.e. non considérée), l'observateur doit être repensé et recodé. Par contre, lorsque l'on utilise du calcul par intervalles, cette entrée manquante peut être remplacée par un intervalle contenant toutes les valeurs possibles. Par conséquent, le système d'équations n'est pas modifié. Cette démarche proposée a été appliquée pour la seconde démarche ; elle est schématisée dans la figure 4.11.

L'algorithme de diagnostic ainsi développé possède les mêmes propriétés que celui du paragraphe 4.4.1 : il permet de détecter et localiser un défaut apparaissant dans l'une des mesures AM.

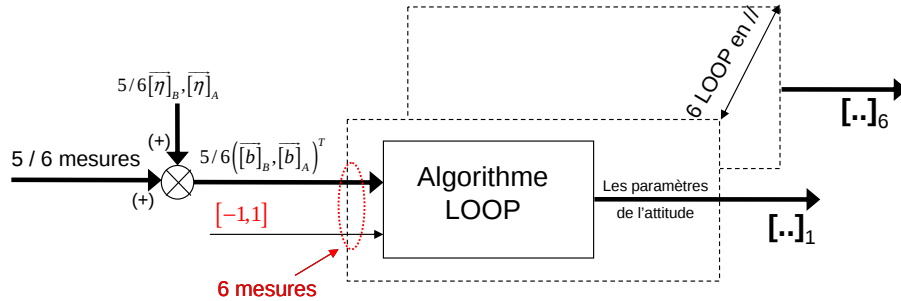


Figure 4.11 – Deuxième structure d'algorithme de diagnostic AM avec 5/6 mesures AM et un intervalle $[-1; 1]$

Les figures 4.12 à 4.17 montrent les résultats obtenus pour la représentation de l'attitude avec un quaternion lorsqu'un défaut est ajouté à la mesure m_4 (i.e. l'accéléromètre suivant l'axe x). On voit que pour tous les estimateurs prenant en considération cette mesure, on obtient un ensemble vide comme solution pour les intervalles de temps sur lesquels le capteur est en défaut. Par exemple, à l'instant $t = 3$ s, on a $0 \in [q_3]$, ce qui induit que toutes les équations qui font intervenir une division par q_3 ne sont pas utilisées dans le jeu d'équations. Notons que certaines composantes du quaternion peuvent être calculées, si leur calcul ne présente pas une division par q_3 .

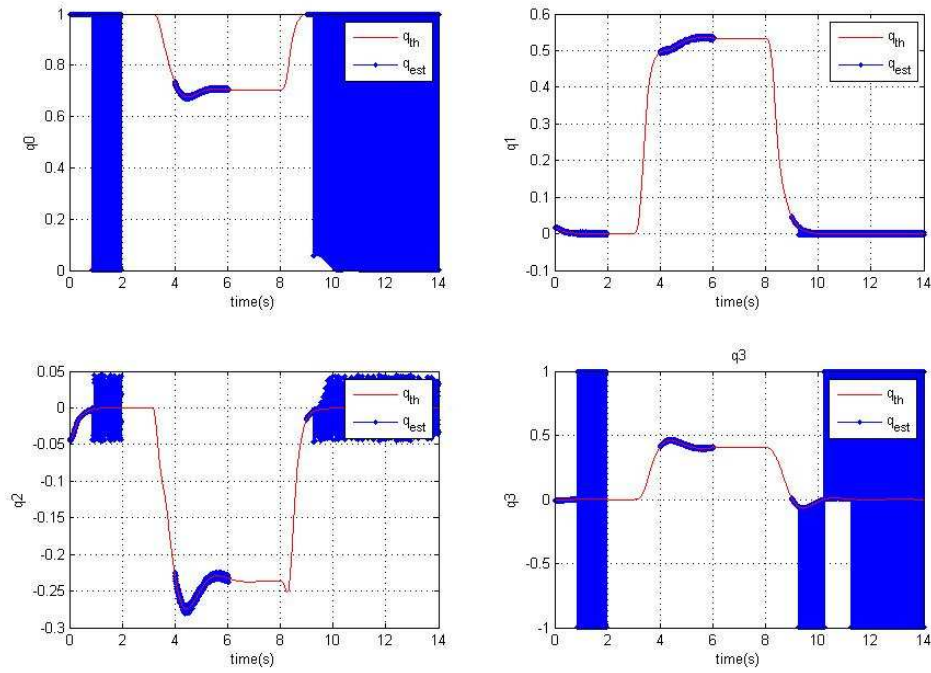


Figure 4.12 – Composantes de $[q_{est}]$ par LOOP sans prendre m_1 pour le calcul, défaut sur m_4

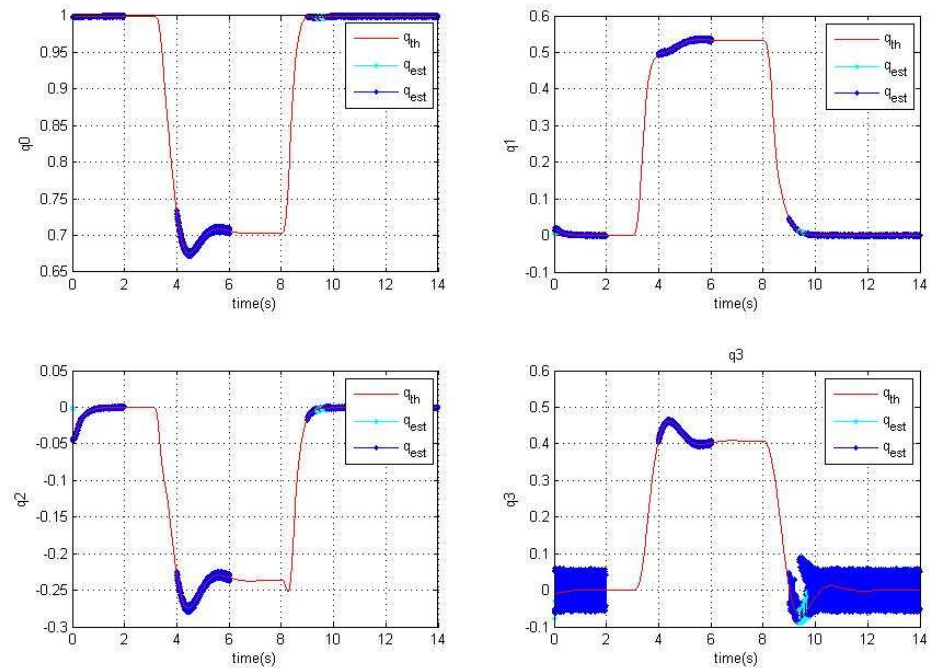


Figure 4.13 – Composantes de $[q_{est}]$ par LOOP sans prendre m_2 pour le calcul, défaut sur m_4

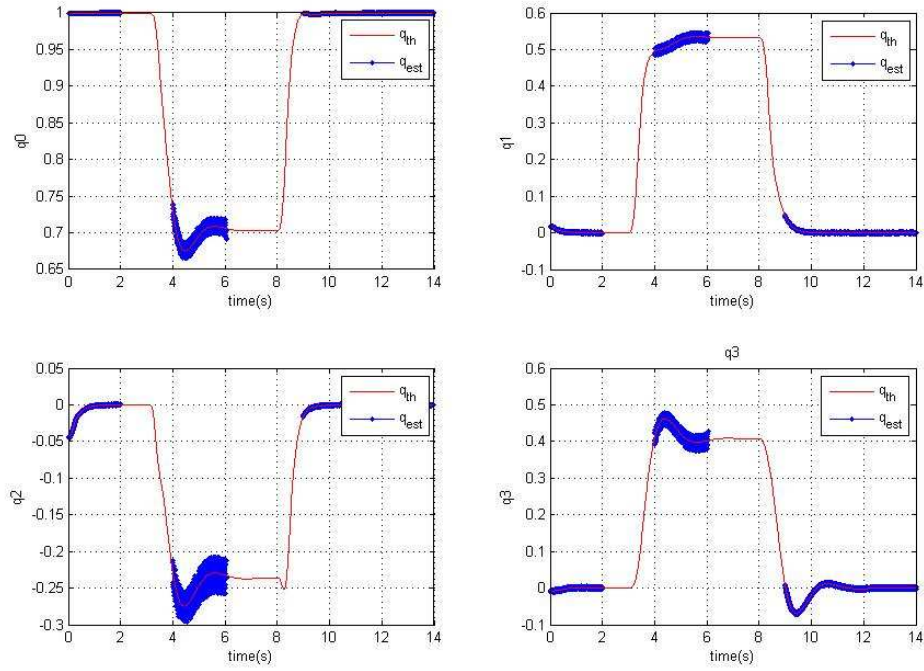


Figure 4.14 – Composantes de $[q_{est}]$ par LOOP sans prendre m_3 pour le calcul, défaut sur m_4

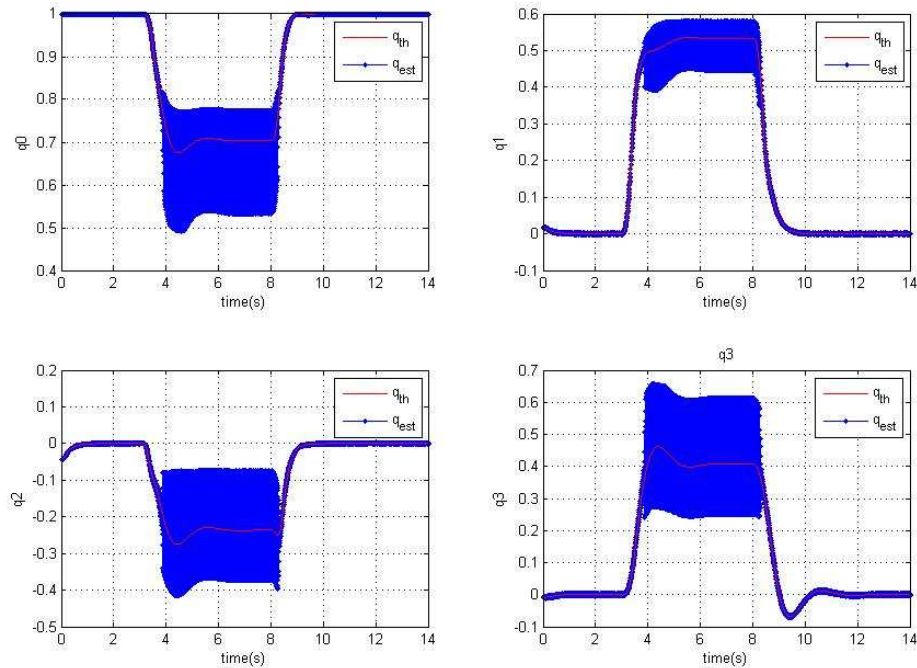


Figure 4.15 – Composantes de $[q_{est}]$ par LOOP sans prendre m_4 pour le calcul, défaut sur m_4

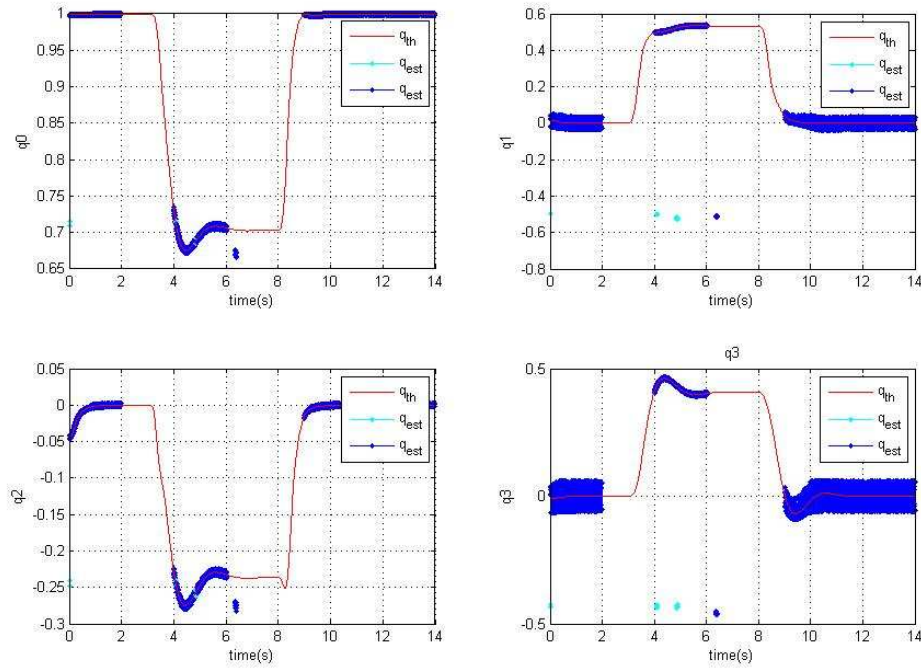


Figure 4.16 – Composantes de $[q_{est}]$ par LOOP sans prendre m_5 pour le calcul, défaut sur m_4

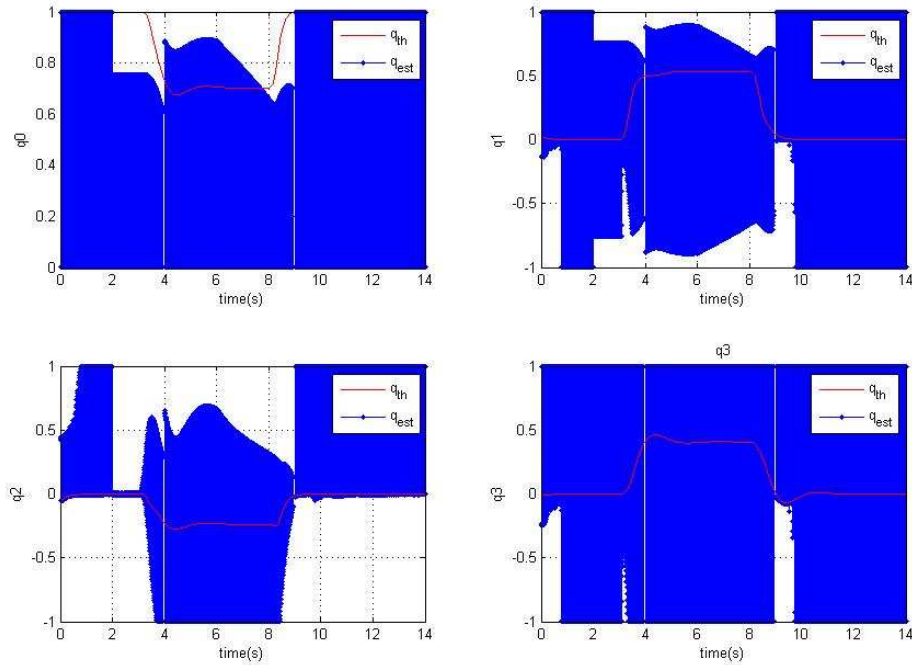


Figure 4.17 – Composantes de $[q_{est}]$ par LOOP sans prendre m_6 pour le calcul, défaut sur m_4

Les figures 4.18 à 4.23 montrent les résultats obtenus avec la paramétrisation de l'attitude avec les angles de Cardan, avec un défaut additif, considéré à des fins d'illustration, sur m_2 (i.e. le magnétomètre d'axe y). Ici encore, on constate bien un intervalle vide retourné pour l'ensemble solution lorsque le capteur en défaut est dans le sous-ensemble de capteurs considérés pour l'estimation de l'attitude. Dans cette configuration, il n'y a pas de situation où $\sin(\cdot)$ et $\cos(\cdot)$ sont égaux en même temps à 0. Ainsi, l'estimation des angles de Cardan retourne un ensemble solution, sauf situations où il existe plusieurs ensembles solutions. Cette situation est illustrée sur les figures 4.18 à 4.23. De plus, s'il n'y a pas d'information avec m_1 (voir la figure 4.18) ou d'information avec m_2 (voir la figure 4.19), l'estimation par LOOP permet de trouver deux solutions très différentes entre $t = 2$ s et $t = 3$ s ou deux solutions proches entre $t = 9.7$ s et $t = 10.1$ s. Par conséquent, l'outil du calcul ensembliste permet d'obtenir toutes les solutions possibles pour un ensemble de mesures considéré.

Cette situation ne peut pas être détectée avec une arithmétique standard, comme on peut le voir dans le travail de C. Berbra [10] qui utilise une routine d'optimisation pour estimer l'attitude à partir de 5/5 mesures AM. Dans ce dernier cas, l'algorithme d'optimisation ne retourne qu'une seule attitude, sans possibilité pour l'utilisateur de détecter que les mesures dont il dispose lui retourne une solution alors qu'il existe 2 minima locaux.

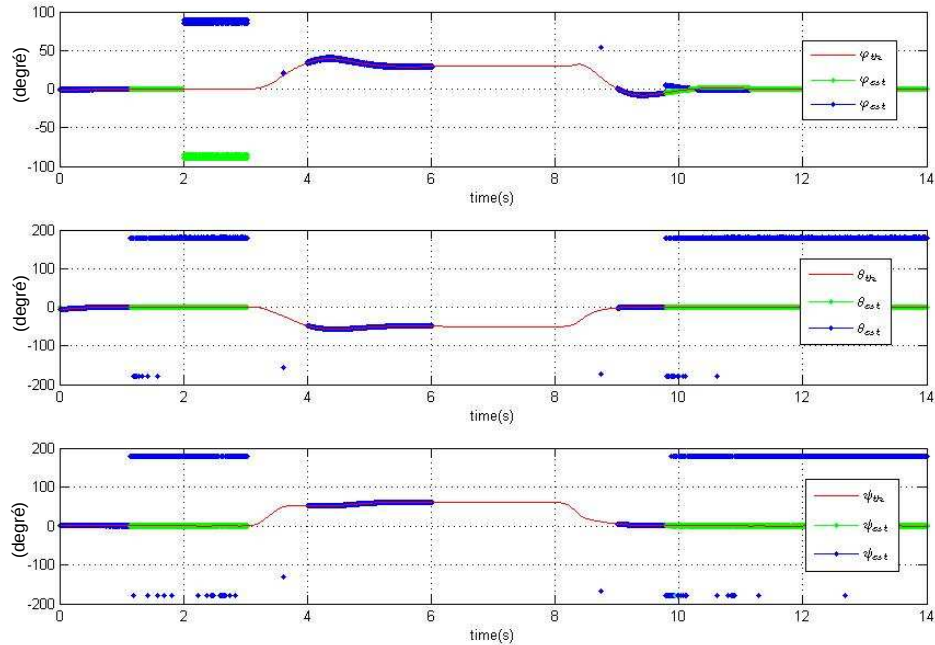


Figure 4.18 – Angles de Cardan par LOOP sans prendre m_1 pour le calcul, défaut sur m_2

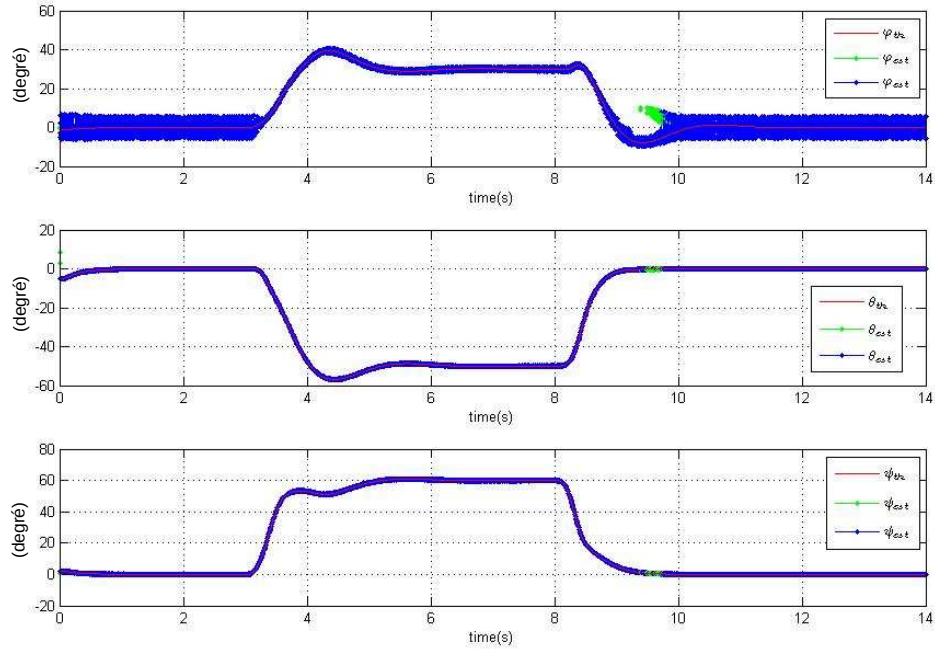


Figure 4.19 – Angles de Cardan par LOOP sans prendre m_2 pour le calcul, défaut sur m_2

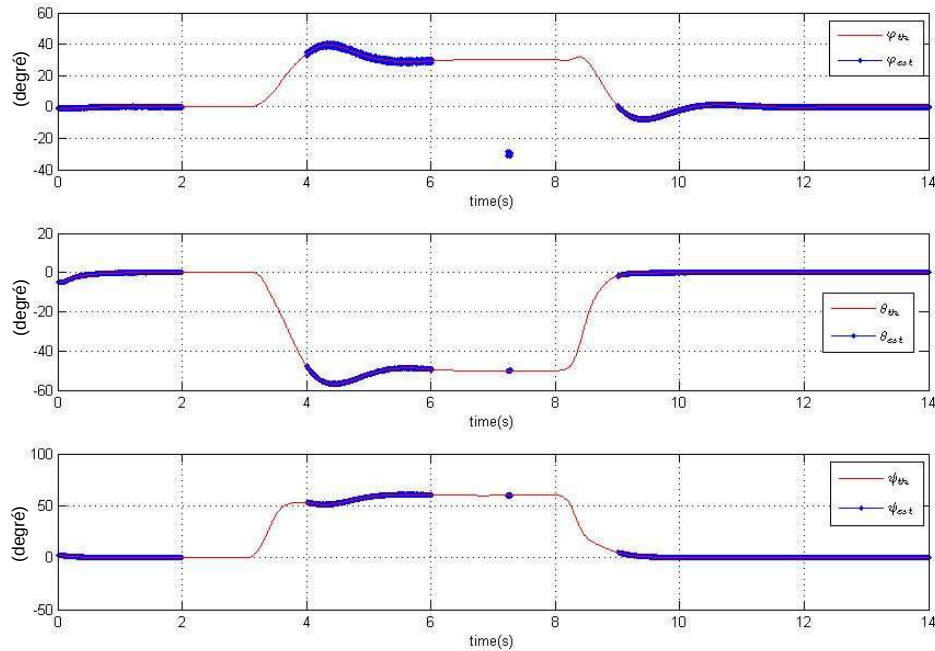


Figure 4.20 – Angles de Cardan par LOOP sans prendre m_3 pour le calcul, défaut sur m_2

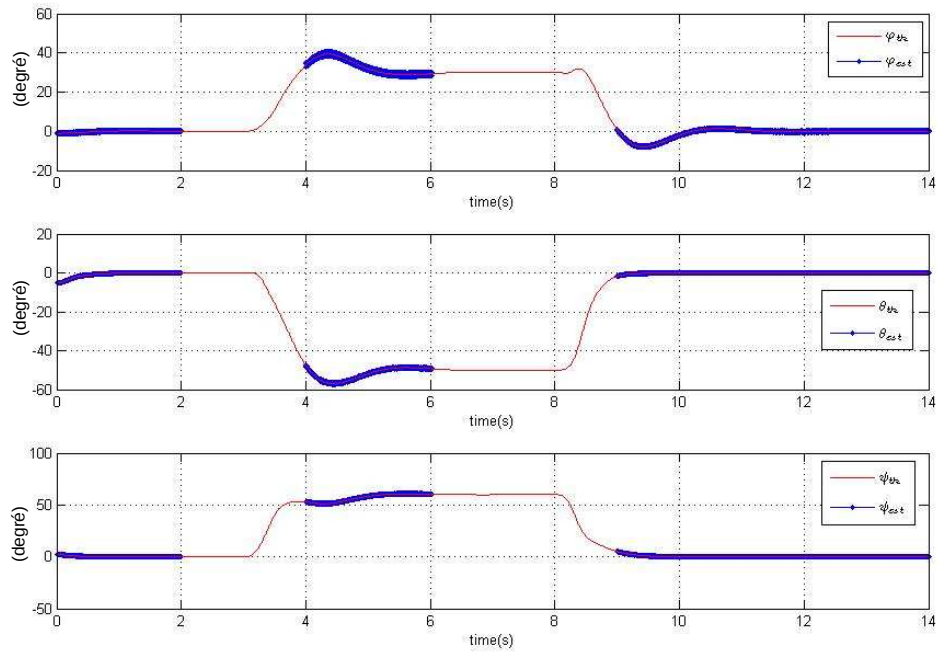


Figure 4.21 – Angles de Cardan par LOOP sans prendre m_4 pour le calcul, défaut sur m_2

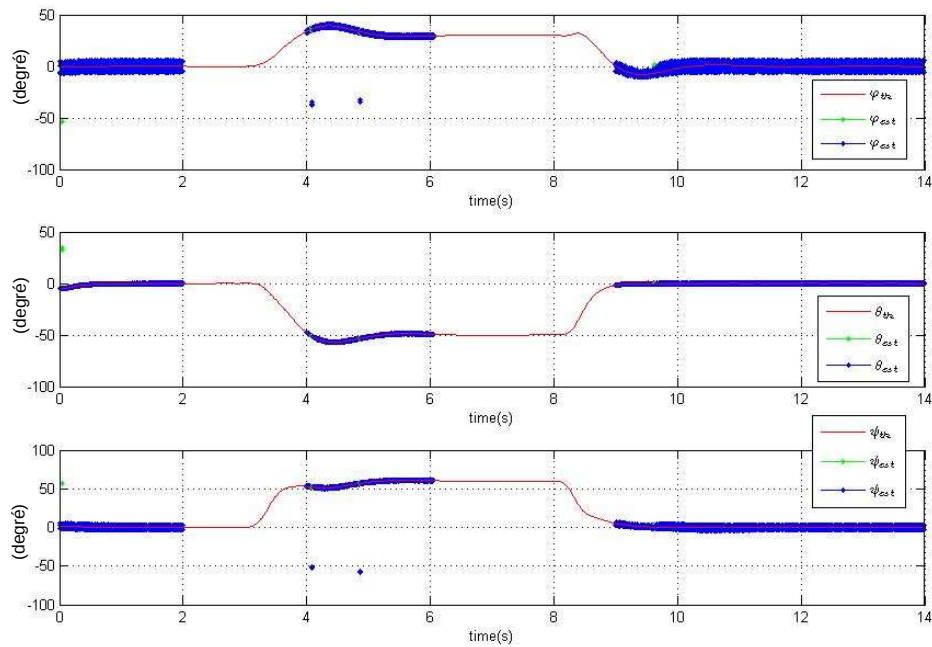


Figure 4.22 – Angles de Cardan par LOOP sans prendre m_5 pour le calcul, défaut sur m_2

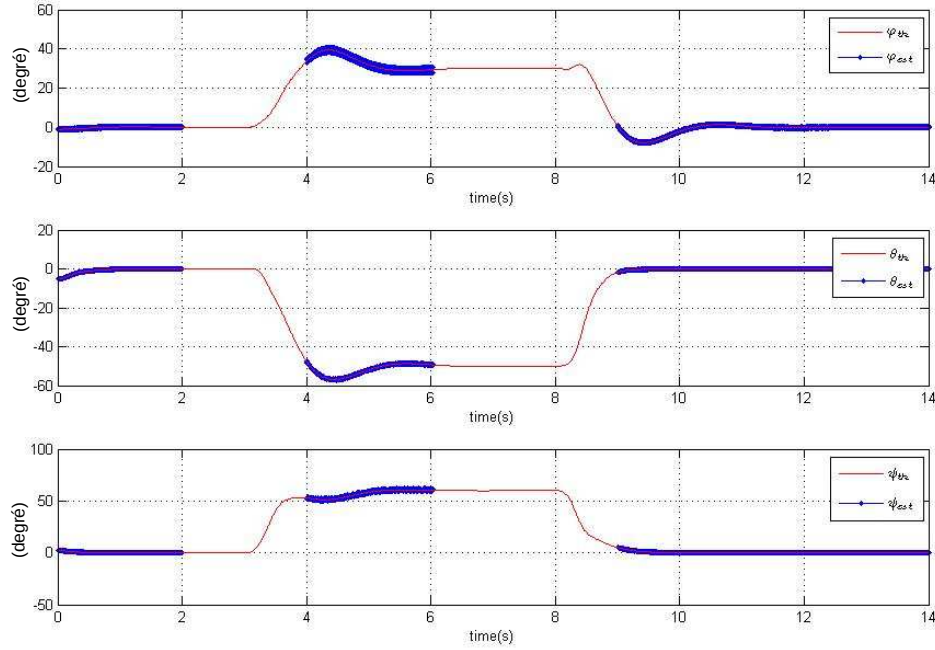
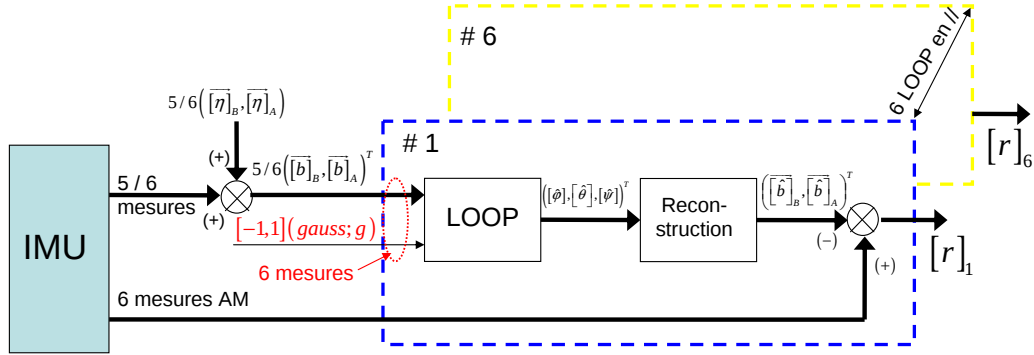


Figure 4.23 – Angles de Cardan par LOOP sans prendre m_6 pour le calcul, défaut sur m_2

4.4.3 Détection/localisation/identification de défauts AM avec 5/6 mesures et un intervalle $[-1; 1]$

Dans le paragraphe précédent, nous avons traité le cas de 6 estimateurs ensemblistes mis en parallèle, chacun prenant 5/6 mesures AM et remplaçant la mesure manquante par l'intervalle de toutes les mesures possibles $[-1; 1]$. Les deux approches précédentes (4.4.1 et 4.4.2) ont montré que l'ensemble vide était obtenu lorsque la mesure contenant le défaut est prise en compte dans l'estimation de l'attitude. Par contre, on obtient bien l'ensemble solution (i.e. l'attitude) voulu lorsque la mesure fautive n'est pas prise en compte.

En se basant sur les principes classiquement utilisés en diagnostic, nous proposons un algorithme permettant non seulement de détecter et localiser le défaut, mais aussi de l'identifier (défaut additif). Pour cela, l'estimateur d'attitude qui utilise un sous-ensemble de mesures saines, est utilisé pour reconstruire les mesures via l'équation (3.2.1). Le résidu est ensuite calculé en calculant la différence entre les mesures et leurs estimées. Le principe de l'algorithme *DiagES* est illustré sur la figure 4.24. Ici encore, l'utilisation de LOOP est centrale. On utilise la paramétrisation de l'attitude avec les angles de Cardan. Les vecteurs résidus $[r_i]$ obtenus sont comparés à la table de signature intervalles défini par le tableau 4.2. Notons que les valeurs des résidus $[r_i]$ générés par l'algorithme *DiagES* montrent les informations de défaut pour chaque mesure. Dans la table de signature 4.2,


 Figure 4.24 – Troisième approche : génération des résidus avec *DiagES*

le vecteur R_i , $i = 1 : 6$ est le vecteur de symptômes obtenus par l'algorithme *DiagES* en écartant respectivement : $m_1, m_2, m_3, m_4, m_5, m_6$. $[000000]$ correspond au cas où l'algorithme *DiagES* ne trouve pas de solutions à cause d'un sous ensemble de mesures incohérentes, i.e, ce vecteur est sensible au défaut. Le symbole $[\emptyset]$ signifie que LOOP retourne un ensemble vide, et donc que la mesure ne peut pas être reconstruite. **[100000]** correspond au cas où LOOP trouve un ensemble solution non vide pour l'estimation de l'attitude. **[0]** correspond au fait que l'intervalle calculé en faisant la différence entre les mesures reconstruite et réelle contient 0. **[1]** correspond au cas où la différence entre les mesures réelle et reconstruite ne contient pas 0, ce qui s'interprète donc par le fait qu'un défaut est apparu sur le capteur considéré.

	f_{m_1}	f_{m_2}	f_{m_3}	f_{m_4}	f_{m_5}	f_{m_6}
R_1	[100000]	[000000]	[000000]	[000000]	[000000]	[000000]
R_2	[000000]	[010000]	[000000]	[000000]	[000000]	[000000]
R_3	[000000]	[000000]	[001000]	[000000]	[000000]	[000000]
R_4	[000000]	[000000]	[000000]	[000100]	[000000]	[000000]
R_5	[000000]	[000000]	[000000]	[000000]	[000010]	[000000]
R_6	[000000]	[000000]	[000000]	[000000]	[000000]	[000001]

 Tableau 4.2 – Table de signatures de l'algorithme *DiagES* pour les mesures AM

La validation de l'algorithme *DiagES* est réalisée en simulation en considérant le scénario ci-dessous :

- les mesures AM considérées sont représentées sur la figure 3.8 avec $\eta_M = 0.0008$ et $\eta_A = 0.002$;
- le défaut est ajouté sur m_4 (i.e. l'accéléromètre d'axe x) avec l'allure telle que donnée sur la figure 4.3.

Les résultats obtenus avec l'algorithme *DiagES* sont présentés sur les figures 4.26 à 4.31 et les composantes de R_4 données sur la figure 4.25. Ces résultats confirment les

capacités de l'algorithme *DiagES*, à savoir :

- détection d'un défaut additif sur l'une des mesures AM ;
- localisation du défaut ;
- identification du défaut.

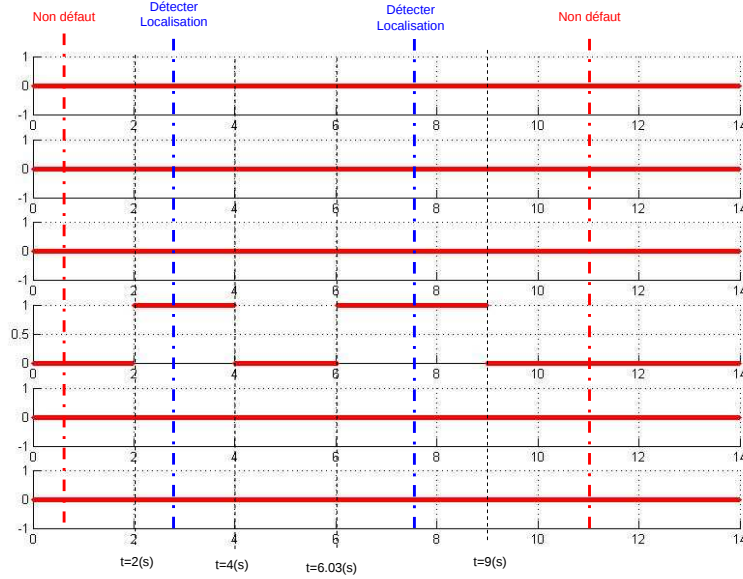


Figure 4.25 – Composants de R_4 (cf. table de signature) pour l'algorithme *DiagES* avec le défaut sur m_4

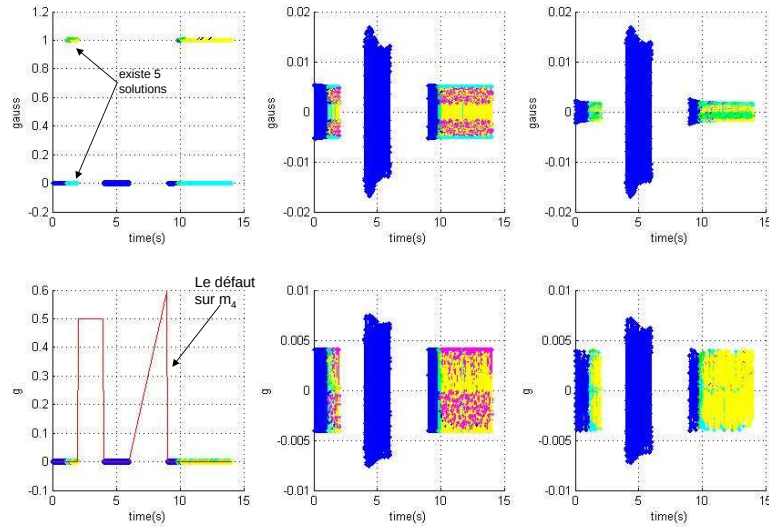


Figure 4.26 – Composantes du vecteur de résidus r_1 pour l'algorithme *DiagES* avec le défaut sur m_4

Remarque 38. Les figures 4.26 et 4.27 montrent que LOOP renvoie parfois plusieurs solutions à partir d'un ensemble de 5/6 mesures. Cela est évidemment dû à la résolution

des systèmes (3.2.9) et (3.2.8) qui conduisent au calcul de racines carrées ou de fonctions trigonométriques inverses ($asin$, ...).

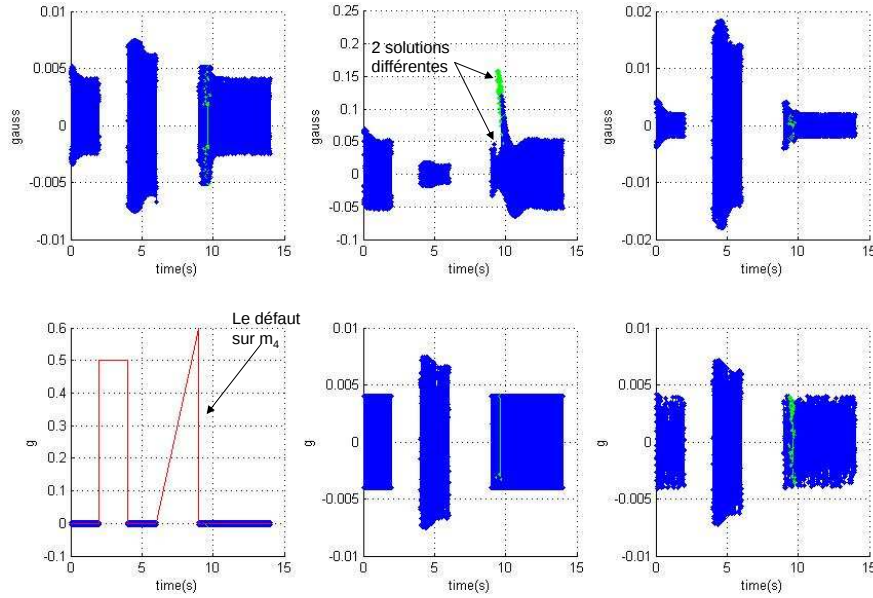


Figure 4.27 – Composantes du vecteur de résidus r_2 pour l'algorithme *DiagES* avec le défaut sur m_4

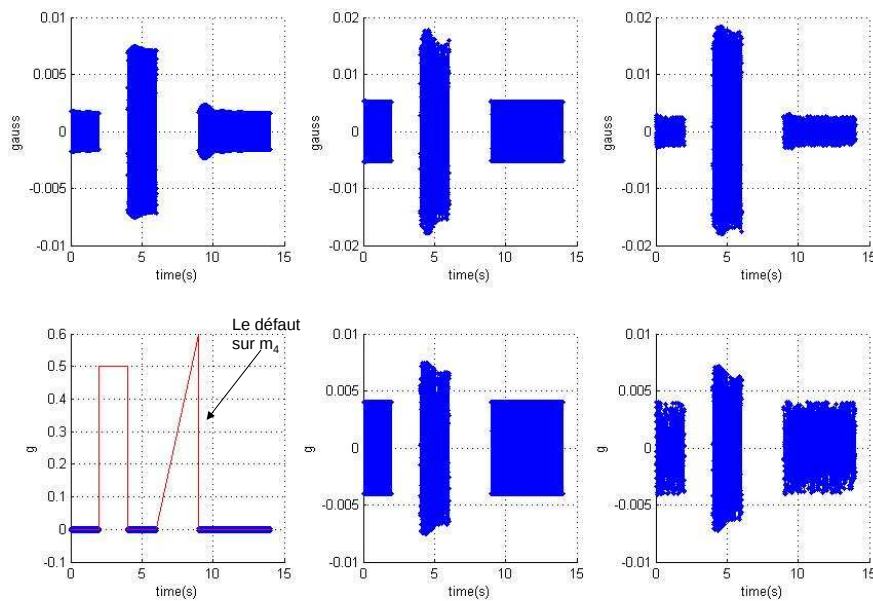


Figure 4.28 – Composantes du vecteur de résidus r_3 pour l'algorithme *DiagES* avec le défaut sur m_4

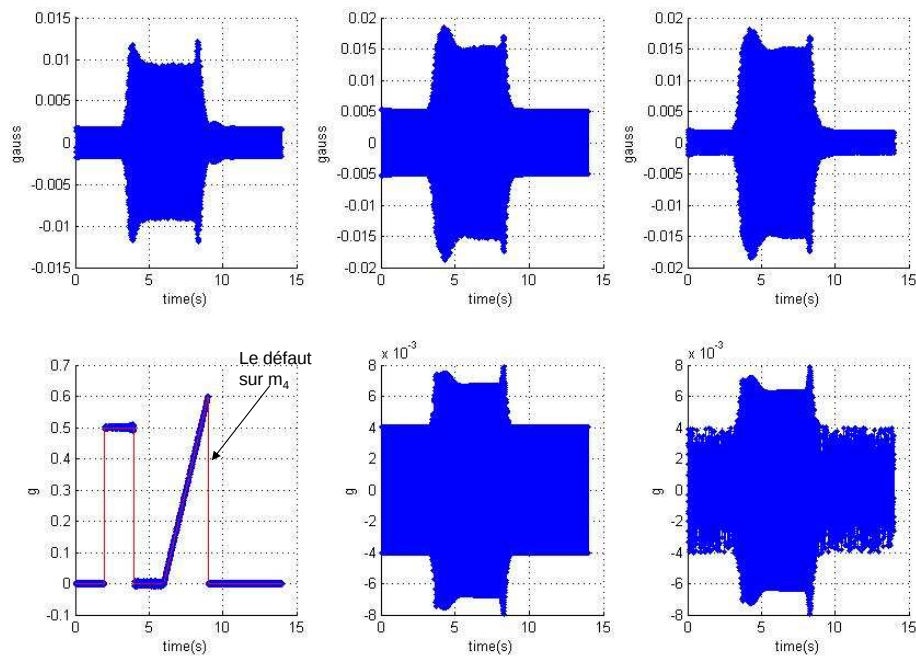


Figure 4.29 – Composantes du vecteur de résidus r_4 pour l'algorithme *DiagES* avec le défaut sur m_4

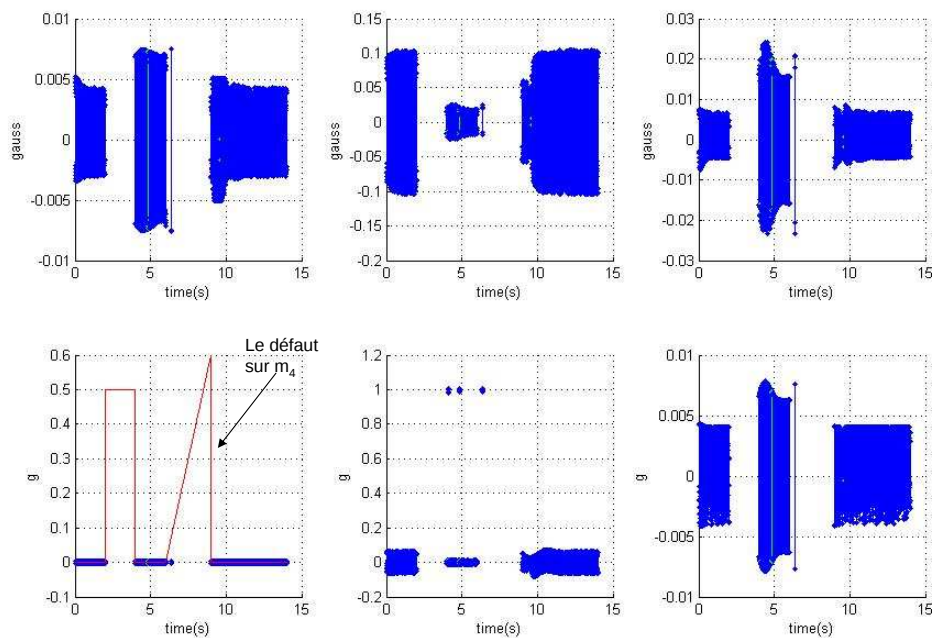


Figure 4.30 – Composantes du vecteur de résidus r_5 pour l'algorithme *DiagES* avec le défaut sur m_4

Remarque 39. Dans le cas où m_1 n'est pas considéré dans le sous-ensemble de mesures

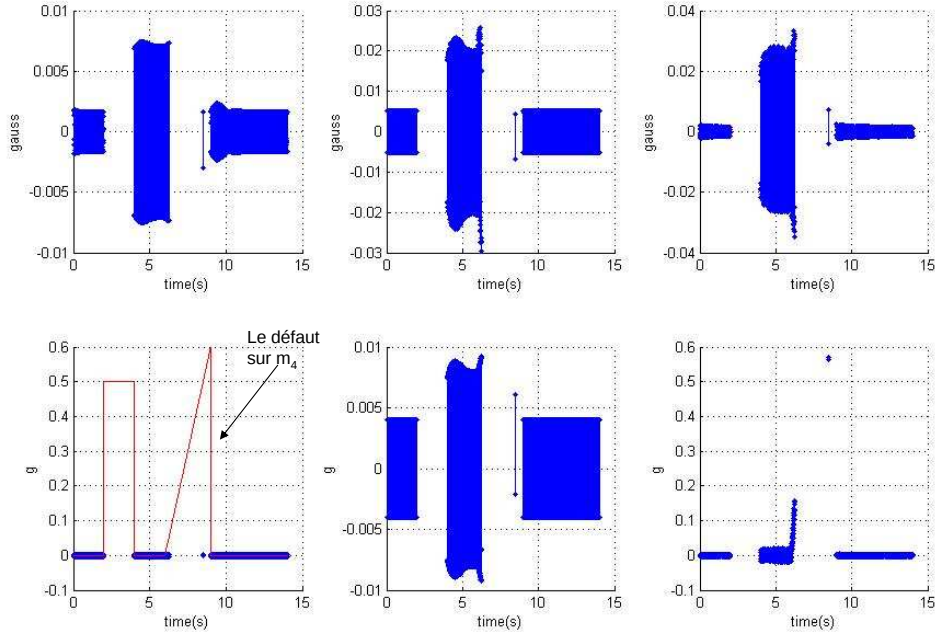


Figure 4.31 – Composantes du vecteur de résidus r_6 pour l'algorithme *DiagES* avec le défaut sur m_4

utilisé par LOOP, pour le mouvement simulé, on observe à plusieurs instants d'échantillonnage plusieurs ensembles de solution disjoints. Ceci est clairement visible sur la figure 4.26 dans les intervalles de temps $t = [1.15; 2] s$ et $t = [9.79; 14] s$. En fait, cette situation apparaît car l'on n'a pas considéré m_1 , la mesure du magnétomètre suivant x pour l'estimation de l'attitude, et les autres mesures ne permettent pas de déterminer le cap de l'objet rigide sur lequel sont fixés les triaxes de capteurs. Pour certains des instants, la mesure estimée $\hat{m}_1$ de m_1 est constituée de deux intervalles disjoints. Cette situation apparaît lorsque les informations contenues dans le sous-ensemble de mesures considéré pour estimer l'attitude ne sont pas suffisantes pour estimer de manière unique l'attitude. La figure 4.27 montre une situation similaire lorsque m_2 n'est pas dans le sous-ensemble de mesures considérées pour estimer l'attitude. Cette difficulté n'a pas été mentionnée dans [10].

4.5 Diagnostic d'une centrale inertielle, cas des mesures AGM

On s'intéresse maintenant au diagnostic de la centrale inertielle AGM, en particulier des capteurs G. Le système d'équations à considérer pour le problème de l'estimation de l'attitude dans le cas dynamique est décrit par (3.3.17) dans lequel on voit apparaître 3

mesures accéléromètre, 3 mesures magnétomètre et 3 mesures gyromètre :

$$\begin{cases} [\hat{q}_k] = [\Theta](\tilde{\omega}, \overrightarrow{[\eta_G]}) * [\hat{q}_{k-1}] \\ \tilde{\omega} = [\omega_k] \cup [\omega_{k-1}] \\ \begin{pmatrix} \vec{b}_M \\ \vec{b}_A \end{pmatrix}_k + \begin{pmatrix} \overrightarrow{[\eta_M]} \\ \overrightarrow{[\eta_A]} \end{pmatrix} = [\mathcal{C}]([\hat{q}_k]) * \begin{pmatrix} \vec{g} \\ \vec{h} \end{pmatrix} \end{cases} \quad (4.5.1)$$

où

$$\begin{cases} [\mathcal{C}](q) = \begin{pmatrix} 2([q_0]^2 + [q_1]^2) - 1 & 2([q_1][q_2] + [q_0][q_3]) & 2([q_1][q_3] - [q_0][q_2]) \\ 2([q_1][q_2] - [q_0][q_3]) & 2([q_0]^2 + [q_2]^2) - 1 & 2([q_0][q_1] + [q_2][q_3]) \\ 2([q_0][q_2] + [q_1][q_3]) & 2([q_2][q_3] - [q_0][q_1]) & 2([q_0]^2 + [q_3]^2) - 1 \end{pmatrix} \\ [\Theta](\omega, [\eta_G]) = I_{4 \times 4} + \sum_{i=1}^n \frac{(\frac{1}{2} * [\Omega](\vec{\omega} + \overrightarrow{[\eta_G]}) * T_e)^i}{i!} \\ \Omega(\vec{\omega}) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \end{cases} \quad (4.5.2)$$

S'il manque une mesure (i.e. si elle n'est pas prise en compte) parmi les trois mesures du gyromètre, $\hat{q}_k$ ne peut pas être calculé. En conséquence, l'estimation de l'attitude est impossible. Pour remplacer la mesure gyromètre non prise en compte, nous allons considérer un pavé qui contient toutes les valeurs possibles de la mesure dans le cadre de l'application considérée.

4.5.1 Algorithme de diagnostic dans le cas AGM

Dans ses travaux sur le diagnostic des gyromètres avec une arithmétique standard, C. Berbra [10] a proposé d'estimer les mesures gyrométriques en résolvant un système non linéaire déduit de l'équation dynamique. Dans un premier temps, nous avons essayé d'appliquer une démarche similaire lorsqu'on utilise du calcul par intervalles. Si l'on veut reconstruire la mesure gyromètre $\vec{\omega}$ non prise en considération à partir des relations (4.5.1) et (4.5.2), on doit résoudre un problème CSP supplémentaire. Cependant, ce CSP est très délicat à résoudre car on doit approximer la dérivée (par intervalles) [11] [77]. Ici, on considère la formule suivante :

$$\delta \left(\dot{\hat{q}} \right) \approx \delta \left(\frac{\hat{q}_k - \hat{q}_{k-1}}{T_e} \right) = (\delta(\hat{q}_k) + \delta(\hat{q}_{k-1})) \times \frac{1}{T_e} \quad (4.5.3)$$

A partir de cette équation, et en prenant $T_e = 10 \text{ ms}$, on a une augmentation très importante de la largeur de $\dot{\hat{q}}$ qui devient 200 fois plus grande que celle de $\hat{q}_{k-1}$, à

chaque instant d'échantillonnage. Cette augmentation de l'intervalle de $\dot{q}$ conduit à une estimation de la mesure gyromètre non considérée avec un intervalle très large. Quand on compare cette mesure estimée avec la mesure fournie par le gyromètre, on constate de nombreux manques à la détection, principalement à cause de cet effet d'augmentation de la taille du pavé. Pour ces raisons, nous n'avons pas poursuivi cette démarche et avons proposé un autre algorithme de diagnostic, plus approprié nous semble-t-il à l'approche ensembliste.

Pour résoudre le problème de diagnostic des mesures AGM, en particulier les mesures gyromètre, l'algorithme *DiagED*, représenté sur la figure 4.32, va être utilisé pour détecter et localiser les défauts additifs sur l'ensemble de 9 mesures AGM grâce à la table de signature 4.3. En pratique, l'algorithme *DiagED* est constitué d'un ensemble de 9 algorithmes *EstimEC* en parallèle. Si l'ensemble vide est trouvé, alors cela veut dire que la mesure du capteur en défaut est considéré dans le sous-ensemble de mesures

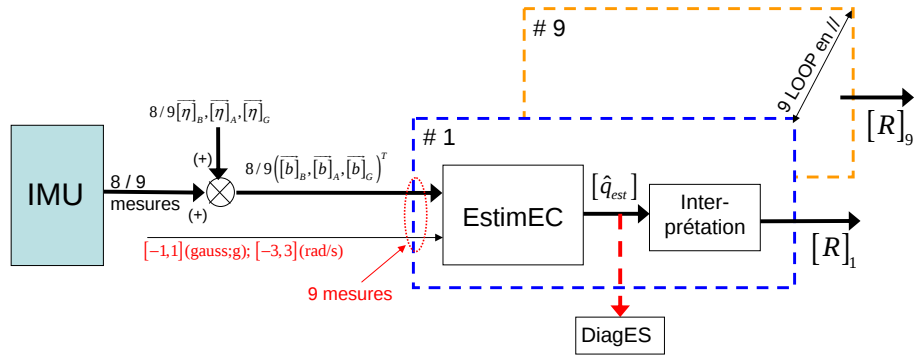


Figure 4.32 – Principe de l'algorithme de diagnostic *DiagED* pour le cas AGM

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9
f_{m_1}	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_2}	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_3}	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_4}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_5}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_6}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
f_{m_7}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$	$[\emptyset]$
f_{m_8}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$	$[\emptyset]$
f_{m_9}	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\mathbf{x}]$

$\mathbf{x}$: mesures cohérentes ; $\emptyset$: mesures incohérentes

Tableau 4.3 – Table de signatures pour l'algorithme *DiagED*, cas des défauts capteur AGM

utilisé déterminer l'attitude et que cette dernière n'a pas pu être calculée puisque l'ensemble retourné est vide (symbole $\emptyset$ dans la table de signature 4.3). Inversement, si on obtient un pavé solution pour l'attitude, cela correspond au cas où la mesure fautive est écartée lors de la génération du résidu. Cela correspond au cas **[x]** dans la table de signature 4.3.

Remarque 40. Notons que seul le cas d'un défaut simple sur une des mesures AM peut être identifié. En outre, les défauts gyromètres ne pourront pas être localisés.

4.5.2 Application de l'algorithme DiagED

Pour valider l'algorithme *DiagED*, différents scénarios vont être traités en simulation :

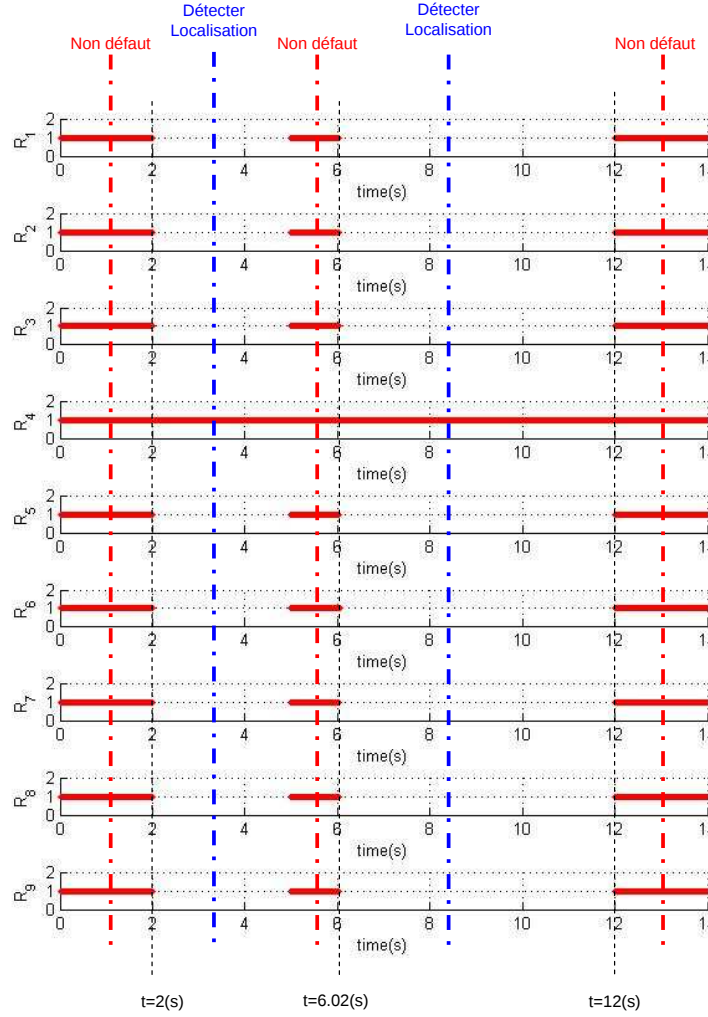
- scénario 3, un défaut additif de type biais (tel que sur la figure 4.3) est ajouté sur m_4 (l'accéléromètre d'axe x) ;
- scénario 4, un défaut additif de type rampe (tel que sur la figure 4.3) est ajouté sur m_9 (le gyromètre d'axe Oz).

A noter qu'une mise à jour de l'attitude sera effectuée aux instants $t = 5\text{ s}$ et $t = 12\text{ s}$. En effet, pour générer le vecteur de résidu, nous avons besoin d'avoir q_{k-1} pour estimer q_k . Or si un ensemble vide apparaît lors du calcul de q_k , on ne pourra pas calculer q_{k+1} .

4.5.2.1 Scénario 3 : défaut sur m_4

La figure 4.33 montre le résultat pour les données simulées obtenus avec l'algorithme de diagnostic *DiagED*. Nous pouvons constater que le défaut est détecté quasiment sans retard pour le défaut constant. Un retard à la détection de $3 \times T_e$ est présent pour le cas du défaut additif en rampe. Cela signifie qu'il faut 3 périodes d'échantillonnage pour que l'amplitude du défaut soit suffisante pour que le défaut soit détectable (détection du biais : $t = 2.0\text{ s}$ et de la rampe : $t = 6.03\text{ s}$).

Après comparaison avec la table de signature 4.3, le défaut peut être localisé à $t = 2.0\text{ s}$ pour le biais et $t = 6.06\text{ s}$ pour la rampe. Le défaut sur m_4 est donc bien détecté et localisé. De plus, il n'existe pas de cas où l'on obtient deux solutions pour l'attitude, ou bien encore une solution pour l'attitude qui n'inclue pas la valeur théorique (i.e. simulée) de l'attitude. C'est une amélioration certaine par rapport aux approches AM seules (voir le paragraphe 4.4) où seulement 6 mesures AM étaient utilisées. Cela montre que dans des situations pas tout à fait (quasi-)statiques, les gyromètres apportent une information complémentaire aux informations fournies par les 5 mesures considérées.



1 : correspond à un ensemble non vide pour le quaternion estimé

Figure 4.33 – Résultats l'algorithme *DiagED* pour le scénario 3 (défaut sur m_4)

4.5.2.2 Scénario 4 : défaut sur m_9

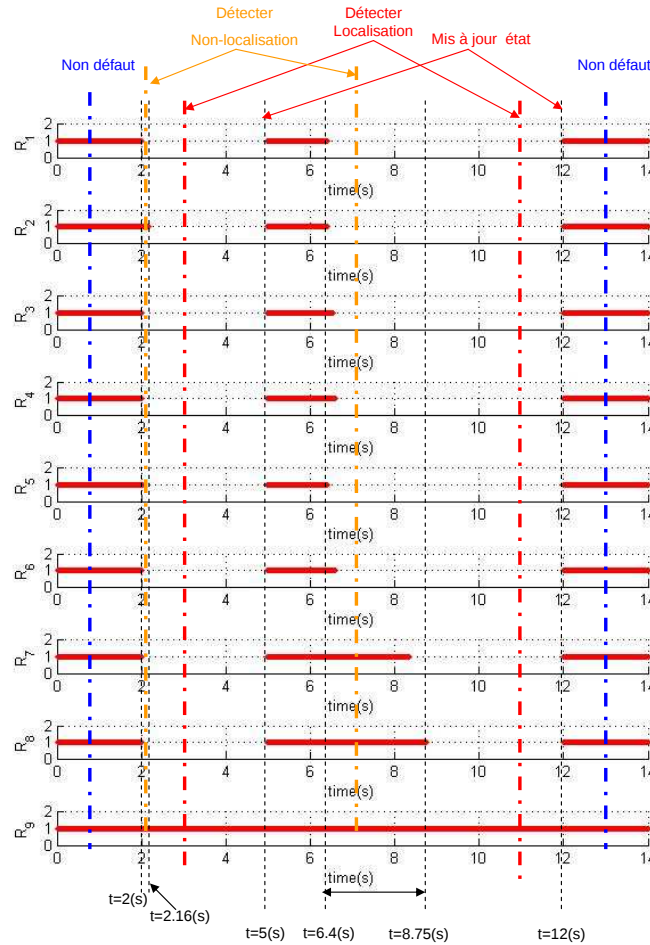
La figure 4.34 montre les résultats pour ce scénario. On peut remarquer que la détection du défaut sur une des trois mesures gyromètre est en retard. Ce manque à la détection est explicable par le fait que différentes valeurs de l'ensemble de (R_7, R_8, R_9) du vecteur résidu R ont été trouvées sur les périodes $t = [2.01; 2.16]$ s et $t = [6.4; 8.75]$ s.

Avec le défaut de type biais, l'algorithme *DiagED* détecte le défaut à partir de $t = 2.01$ s, mais on ne peut pas encore localiser la mesure fautive avec le vecteur R . La localisation est réalisée à $t = 2.16$ s. Le temps nécessaire entre la détection et la localisation (entre $t = 2.01$ s et $t = 2.16$ s) s'explique par l'effet de l'intégration des mesures gyromètres.

Remarque 41. Il est important de remarquer que pour le défaut de type biais, l'algorithme *DiagED* (voir la figure 4.34) et l'algorithme *DiagEC* (voir la figure 4.9) donnent les mêmes résultats sur la détection du défaut.

Avec le défaut de type rampe, l'algorithme *DiagED* détecte le défaut avec un retard de 0.4 s car il faut que l'amplitude du défaut soit suffisante pour être détecté. Ensuite, on constate un retard important pour que l'algorithme puisse localiser ce défaut (retard de 2.75 s). Cela est inéluctable car la mesure non prise en compte dans le calcul est remplacée par l'intervalle $[-3, 3] \text{ rad/s}$. Notons que dans [10], avec une technique de diagnostic mettant en œuvre une arithmétique standard, on peut localiser le défaut en estimant $\hat{q}_{est}$ et en utilisant l'équation dynamique du quaternion.

Il serait intéressant de trouver une technique similaire pour les défauts sur les gyro-mètres en utilisant les techniques par intervalle. Par manque de temps, cette étude n'a pas pu être menée.



1 : correspond à un ensemble non vide pour le quaternion estimé

Figure 4.34 – Résultats l'algorithme *DiagED* pour le scénario 4 (défaut sur m_9)

4.6 Conclusion

Dans ce chapitre, le problème de détection et de localisation de défaut avec des techniques basées sur le calcul par intervalles d'une centrale d'attitude a été étudié. Différents algorithmes ont été mis en œuvre : un pour le cas statique, l'algorithme *DiagES*, et un pour le cas dynamique, l'algorithme *DiagED*.

Dans le cas statique, l'algorithme *DiagES* permet de détecter, localiser des défauts sur une des 6 mesures AM. Nous avons montré également que l'étape d'identification pouvait être résolue. Dans la table de signature, le symbole signifie que la mesure fautive est prise en compte dans le sous-ensemble de mesures lors de l'estimation de l'attitude, conduisant à un ensemble vide lors du calcul de l'attitude.

Dans le cas dynamique, l'algorithme *DiagED* appliqué à un modèle dynamique non-linéaire, permet de détecter et localiser des défauts sur une des 9 mesures AGM. Il a été montré que l'algorithme *DiagED* permet de détecter (rapidement) un défaut mais présente un retard certain pour la localisation du défaut.

Chapitre 5

Conclusions et perspectives

Ce travail a été dédié à l'estimation de l'attitude et au diagnostic d'une centrale d'attitude constitué d'un triaxe d'accéléromètres, d'un triaxe de magnétomètres et d'un triaxe de gyromètres en utilisant le calcul ensembliste. L'objectif était d'obtenir un résultat garanti pour l'attitude compte tenu d'un bruit inconnu mais borné (estimation d'attitude) et de détecter le capteur en défaut. Notons que seuls des défauts additifs ont été considérés.

Les méthodes développées dans ce manuscrit pour l'estimation de l'attitude possèdent des avantages très intéressants permettant de "configurer" l'ensemble des capteurs (i.e. de choisir les capteurs appropriés) afin de garantir la précision voulue pour l'attitude estimée. Les algorithmes développés pour l'estimation d'attitude permettent de calculer un ensemble solution de forme simple, directement exploitable par le concepteur du système, et qui contient toutes les solutions recherchées compte tenu de bruits inconnus mais bornés sur les différentes mesures. Cette hypothèse de "bruit inconnu mais borné" permet en outre de prendre en compte simplement des vibrations (ou accélérations réputées parasites) ou des perturbations magnétiques.

Pour l'estimation de l'attitude d'un corps rigide dans l'espace à 3 dimensions, deux familles d'algorithmes ont été proposées :

- dans le cas statique on considère que les accélérations propres du corps rigide sur lequel sont fixés les capteurs sont négligeables devant la gravité. On a alors une configuration avec les triaxes accéléromètre et magnétomètre. Deux algorithmes ont été développés en considérant pour chacun d'eux, deux représentations de l'attitude avec respectivement les angles de Cardan et le quaternion unitaire. Le premier algorithme proposé est nommé LOOP, le second est nommé LoopBorne. Ces deux algorithmes sont basés sur le principe de l'algorithme Waltz. Nous avons vu qu'à partir de mesures fournies par un triaxe accéléromètre et un triaxe magnétomètre, LOOP et LoopBorne permettent d'estimer l'attitude et de fournir un domaine

solution de faible dimension.

L'algorithme LOOP qui ne contient qu'une phase de contraction, sans bisection, peut être implanté pour une application en temps réel, alors que l'algorithme LoopBorne qui intègre une phase de bisection permet d'obtenir un domaine solution pour l'attitude de taille réduite mais avec un temps de calcul augmenté. De plus, tous les deux peuvent détecter des mesures incohérentes. La présence de mesures incohérentes peut être expliquée par la présence d'un défaut.

Les résultats obtenus avec LOOP et LoopBorne ont été comparés à ceux obtenus par SIVIA, avec des données simulées. Ce dernier algorithme permet une description plus fine du domaine solution au détriment d'un temps calcul augmenté. Ces algorithmes ont également été validés sur des données réelles enregistrées avec une centrale inertielle *3DM – GX1* chez MicroStrain ;

- dans le cas de l'estimation d'attitude dynamique, on utilise les mesures fournies par les 3 triaxes (accéléromètre, gyromètre et magnétomètre). Deux algorithmes *EstimEI* et *EstimEC* ont été développés dans ce travail. L'équation dynamique est intégrée en utilisant une étape d'intégration garantie et en utilisant les mesures gyromètre. Les mesures accéléromètre et magnétomètre sont fusionnées avec l'algorithme LOOP. La phase de correction est réalisée soit par l'intersection des deux domaines obtenus par respectivement, LOOP et l'intégration de l'équation dynamique, soit en considérant dans la phase d'intégration le résultat retourné par LOOP.

Une deuxième problématique a été abordée dans ce travail de doctorat. Nous avons proposé une structure permettant de répondre au problème du diagnostic pour une centrale d'attitude par l'outil du calcul ensembliste. Pour cela, deux algorithmes ont été proposés :

- le premier, appelée *DiagES*, plante un banc d'estimateurs/observateurs prenant 5/6 mesures AM, la mesure non considérée dans le calcul de l'attitude étant remplacée par un intervalle contenant toutes les valeurs possibles de la mesure, i.e. $[-1; 1]$. Cet algorithme permet de réaliser les étapes de détection, de localisation et aussi d'identification grâce à la reconstruction des mesures ;
- le deuxième, appelé *DiagED*, utilise les 9 mesures de la centrale inertielle. Il permet de réaliser l'étape de détection de manière similaire à l'algorithme *DiagES*. Ici encore, on remplace la mesure non considérée dans le sous-ensemble de mesures utilisé pour estimer l'attitude par un intervalle contenant toutes les mesures possibles.

Les algorithmes ont été validés sur des données simulées.

Perspectives

A l'issue de ce travail, de nombreuses pistes restent encore à explorer.

La première d'entre elles est que le temps de calcul nécessaire pour nos algorithmes n'est pas négligeable. Il est donc nécessaire de travailler sur la diminution de ce temps

si on veut pouvoir utiliser les algorithmes développés pour une application temps réel. Notons que pour des questions de facilité d'implémentation, nos algorithmes ont été développés sous Matlab. Il suffirait donc de re-coder les routines en C ou en C++ par exemple, afin d'obtenir un gain de temps certain.

En analysant le principe de traitement de nos algorithmes, nous avons constaté que la parallélisation du calcul (par exemple en utilisant la boîte à outils "Parallel Computing Toolbox" et en ayant plusieurs cœurs de calcul) peut être appliquée. Actuellement, nous ne disposons pas de cette boîte à outils, et nous n'avons pas pu exploiter la possibilité de diminuer le temps calcul en parallélisant le calcul. On pourrait aussi imaginer d'utiliser des processeurs dédiés tels que décrits dans [44] [69] [68]. Par exemple, [68] montre que la vitesse de calcul pour des opérations de base comme l'addition, la multiplication et la division est plus rapide avec une accélération d'un facteur 200 à 1000. C'est pourquoi, il serait intéressant d'exploiter les capacités de ces processeurs dédiés dans le cadre d'une application temps réel.

La deuxième piste à poursuivre est liée à la contrainte sur la norme unité du quaternion, $\|\vec{q}\| = 1$, qu'il est difficile de respecter dans le cadre du calcul par intervalles. Notons que dans le cas dynamique, l'utilisation du quaternion est plus facile à manipuler que les angles de Cardan (ou d'Euler), ce qui justifie pleinement son utilisation dans une approche par intervalles. Dans nos travaux, nous avons exploité une représentation du quaternion avec les paramètres d'Euler $q = (\alpha, \vec{u})$, le vecteur $\vec{u}$ étant lui-même donné en représentation sphérique. Cette approche doit être poursuivie et évaluée de manière à vérifier, en particulier qu'il n'existe pas de problèmes de singularité.

La troisième piste de recherche touche au sujet du diagnostic des capteurs de la centrale inertielle, qui s'avère être, lorsqu'on utilise le calcul ensembliste, un problème difficile. En particulier, on a étudié uniquement le cas de défauts additifs, et on a été capable de localiser un défaut unique. Il serait souhaitable de poursuivre l'étude afin d'être capable de détecter et/ou localiser et/ou identifier des défauts multiples. Ce travail nécessitera, à notre avis de proposer une structure modifiée de l'algorithme de diagnostic ensembliste des 9 capteurs de la centrale inertielle pour prendre en compte le cas des défauts multiples car les algorithmes développés dans ce manuscrit ne peuvent pas en l'état, traiter ce problème.

La quatrième piste de recherche touche au sujet de la détection d'une perturbation magnétique ou d'une accélération propre ou d'un biais sur les gyromètres. Dans ces cas, la "perturbation" affecte les 3 mesures et l'hypothèse de défaut unique n'est plus valide. Il faut donc étendre les algorithmes proposés au cas d'un défaut affectant l'ensemble des 3 mesures d'un triaxe.

Bibliographie

- [1] F. Abdallah, A. Gning, and P. Bonnifait. Adapting particle filter on interval data for dynamic state estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 44, Honolulu, Hawaii, USA, April 2007.
- [2] F. Abdallah, A. Gning, and P. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44 :807 – 815, 2008.
- [3] O. Adrot, H. Janati-Idrissi, and D. Maquin. Fault detection based on interval analysis. In *15th IFAC World Congress on Automatic Control*, Barcelonne, Spain, July 2002.
- [4] A.Gning, F.Abdallah, and Ph.Bonnifait. Dynamic vehicle localisation using constraints propagation techniques on intervals a comparison with kalman filtering. In *Proceedings of the 2005 international conference on Robotics and Automation*, Barcelona, Spain, May 2005.
- [5] A.Gning and P.Bonnifait. Guaranteed dynamic localizaton using constraints propagation techniques on real intervals. In *Proceedings of the 2004 international conference on Robotics and Automation*, New Orleans, LA, USA, April 2004.
- [6] J. Armengol, J. Vehi, L. Travé-Massuyès, and M. A. Sainz. Interval model-based fault detection using multiple sliding time windows. In *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 2000*, Budapest, Hungary, 2000.
- [7] M. Basseville. Detecting changes in signals and systems - A survey. *Automatica*, 24 :309 – 326, 1988.
- [8] M. Basseville and I. Nikiforov. *Detection of abrupt changes- Theory and applications*. Prentice Hall Information and System Sciences Series, 1993.
- [9] B.B.Salmeron-Quiro. *Fusion de Données multi-capteurs pour la capture de mouvement*. PhD thesis, Université Joshep Fourier de Grenoble, 2007.
- [10] B.Cédric. *Diagnostic des systèmes embarqués en réseau. Application à mini drone hélicoptère*. PhD thesis, Institut polytechnique de Grenoble, 2009.

Bibliographie

- [11] Barnabas Bede and Luciano Stefanini. Numerical solution of interval differential equations with generalized hukuhara differentiability. In *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*, Lisbon, Portugal, July 2009.
- [12] S. Beeby, G. Ensell, M. Kraft, and N. White. *MEMS Mechanical Sensors*. Artech House, Inc, 2004.
- [13] Frédéric Benhamou, Frédéric Goualard, Laurent Granvilliers, and Jean-François Puges. Revising hull and box consistency. In *Proceedings of the 16th International Conference on Logic Programming*, Las Cruces, New Mexico, United States, November 1999.
- [14] Guanrong Chen, Jianrong Wang, and L. S. Shieh. Interval kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 33 :250 – 259, 1997.
- [15] J. Chen and R.J. Patton. *Robust Model Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [16] J. C.K. Chou. Quaternion kinematic and dynamic differential equations. *IEEE Transactions on robotics and automation*, 8 :53 – 64, 1992.
- [17] J.G. Cleary. Logical arithmetic. *Future Computing Systems*, 2 :125 – 149, 1987.
- [18] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32 :281–331, 1987.
- [19] E.Hansen. *Global Optimization Using Interval Analysis*. Dekker, New York, 1992.
- [20] T.I. Fossen. *Guidance and control of ocean vehicles*. New York, 1994.
- [21] P.M. Frank and S.X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 6 :403–424, 1997.
- [22] J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, 9 :3 – 11, 1988.
- [23] J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5 :653 – 661, 1997.
- [24] A. Gning. *Localisation garantie d’automobiles. Contribution aux techniques de satisfaction de contraintes sur les intervalles*. PhD thesis, Université de Technologie Compiègne, 2006.
- [25] David F. Griffiths and Desmond J. Higham. *Numerical Methods for Ordinary Differential Equations*. Springer, 2010.

Bibliographie

- [26] J.F. Guerrero-Castellanos. *Estimation de l'attitude et commande bornée en attitude d'un corps rigide : Application à un mini hélicoptère à quatre rotors*. PhD thesis, Université Joseph Fourier-Grenoble I, 2008.
- [27] D. Henry. From fault diagnosis to recovery actions for aeronautic and aerospace missions : a model-based point of view. In *23rd IAR Workshop on Advanced Control and Diagnosis*, Coventry, UK, November 2008. Royaume-Uni.
- [28] G. Heredia, A. Oiler, M. Bejar, and R. Mahtani. Sensor and actuator fault detection in small autonomous helicopter. *Mechatronics*, 18 :90 – 99, 2008.
- [29] Astrosurf ([http ://www.astrosurf.com/luxorion/terre-champ_magnetique2.htm](http://www.astrosurf.com/luxorion/terre-champ_magnetique2.htm)). Magnetometers.
- [30] MEMSense ([http ://www.memsense.com/](http://www.memsense.com/)). Mag3 : Triaxial magnetometer, accelerometer & gyroscope analog inertial sensor.
- [31] [http ://www.microstrain.com](http://www.microstrain.com). Microstrain.
- [32] [http ://www.ti3.tu-harburg.de/rump/intlab/](http://www.ti3.tu-harburg.de/rump/intlab/). Intlab.
- [33] I.Braems. *Méthodes ensemblistes garanties pour l'estimation de grandeurs physiques*. PhD thesis, Université de Paris-Sud 11, Orsay, 2002.
- [34] I.Braems, N.Ramdani, M.Kieffer, and E.Walter. Caractérisation garantie d'un dispositif de mesure de grandeurs thermiques. *Journal européen des systèmes automatisés*, spécial Outils de calculs ensemblistes :1129 – 1143, 2003.
- [35] R. Isermann. Process fault detection based on modeling and estimation methods : A survey. *Automatica*, 20 :387 – 404, 1984.
- [36] R. Isermann. *Fault Diagnosis Systems : An introduction from fault detection to fault tolerance*. Springer-Verlag, 2006.
- [37] R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5 :709 – 719, 1997.
- [38] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. *Applied Interval Analysis*. Springer, 2001.
- [39] J.L.Gouzé, A.Rapaport, and M. Z.Hadj-Sadok. Interval observers for uncertain biological systems. *Ecological Modelling*, 133 :45 – 56, 2000.
- [40] J.L.Gouzé, M.Moisan, and O.Bernard. Simple interval observers for biotechnological processes. In *Proceedings of the ISCCSP 2006 conference*, Marrakech, Morocco, March 2006.

Bibliographie

- [41] R. J. Patto and J. Chen. Observer-based fault detection and isolation : Robustness and applications. *Control Engineering Practice*, 5 :671 – 682, 1997.
- [42] A. Khosravi, J.A. Llobet, and E.R. Gelso. An interval intelligent-based approach for fault detection and modelling. In *Fuzzy Systems Conference, FUZZ-IEEE 2007*, London (UK), July 2007.
- [43] J. Korbicz, J. Kościelny, Z. Kowalczyk, and W. Cholewa. *Fault diagnosis. Models, Artificial Intelligence, Applications*. Springer-Verlag, 2004.
- [44] K. Reinhard and W.K. Ulrich. Hardware support for interval arithmetic. *Reliable Computing*, 12 :225–237, 2006.
- [45] O. Lhomme. Consistency techniques for numeric CSPs. In *Proceeding of IJCAI 1993*, volume 1, pages 232–238, Chambéry, France, August 1993.
- [46] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74 :1772 –1782, 2001.
- [47] L. Jaulin and É. Walter. Guaranteed nonlinear parameter estimation from bounded error data via interval analysis. *Mathematics and Computers in Simulation*, 35 :123 – 137, 1993.
- [48] D. Maquin and J. Ragot. *Diagnostic des systèmes linéaires*. Hermes, 2000.
- [49] F.L. Markley. Attitude determination using vector observations : Theory. *Journal of the Astronautical Sciences*, 37 :41 – 58, 1989.
- [50] F.L. Markley. Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics*, 63 :311 – 317, 2003.
- [51] F.L. Markley, J.L. Crassidis, and Y. Cheng. Nonlinear attitude filtering methods. *AIAA Guidance, Navigation, and Control Conference*, 2005.
- [52] B. Marx, D. Maquin, and J. Ragot. State estimation and fault detection of uncertain systems based on an interval approach. In *Conference on Control and Fault Tolerant Systems, SysTol'10*, Nice, France, October 2010.
- [53] M. Kieffer and É. Walter. Interval analysis for guaranteed nonlinear parameter and state estimation. *Mathematical and Computer Modelling of Dynamical Systems*, 11 :171 – 181, 2005.
- [54] M.R. Napolitano, Y. An, and A. Seanor. A fault tolerant flight control system for sensor and actuator failure using neural network. *Aircraft Design*, 3 :103 – 128, 2000.

Bibliographie

- [55] M.R. Napolitano, D.A. Windson, and J.L. Casanova. Kalman filters and neural-network schemes for sensor validation in flight control systems. *IEEE Transaction on Control Systems Technology*, 6 :596 – 611, 1998.
- [56] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105 :21 – 68, 1999.
- [57] Nedialko S. Nedialkov. *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Canada, 1999.
- [58] H.V. Nguyen, C. Berbra, S. Lesecq, S. Gentil, A.Barraud, and C. Godin. Diagnosis of an inertial measurement unit based on set membership estimation. In *17th Mediterranean Conference on Control and Automation, MED '09.*, Thessaloniki, Greece, June 2009.
- [59] N.Meslem. *Atteignabilité hybride des systèmes dynamiques continus par analyse par intervalles. Application à l'estimation ensembliste*. PhD thesis, Université Paris Est, 2008.
- [60] E. P. Oppenheimer. Application of interval analysis techniques to linear systems. Part II : The interval matrix exponential function. *IEEE Transactions on Circuits and Systems*, 35 :1230 – 1242, 1988.
- [61] S. A. Raka and C. Combastel. A stable interval observer for fault detection in the context of verified model-based design. In *Conference on Control and Fault Tolerant Systems, SysTol'10*, Nice, France, October 2010.
- [62] N. Ramdani and S. N. Nedialkov. Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques. *Nonlinear Analysis : Hybrid Systems*, 2010.
- [63] A. Rapaport and D. Docha. Interval observers for biochemical processes with uncertain kinetics and inputs. *Mathematical Biosciences*, 193 :235 – 253, 2005.
- [64] A. Rapaport and J. L. Gouzé. Parallelotopic and practical observers for non-linear uncertain systems. *International Journal of Control*, 76 :237 – 251, 2003.
- [65] D. Ratz. *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen*. PhD thesis, Universität Karlsruhe, Nancy, 1992.
- [66] V. Reppa and A. Tzes. Fault detection and diagnosis relying on set membership identification for time varying systems. In *Conference on Control and Fault Tolerant Systems, SysTol'10*, Nice, France, October 2010.

Bibliographie

- [67] A.L. Satin and R.L.Gates. Evaluation of parity equations for gyro failure detection and isolation. *Journal of Guidance, Control and Dynamics*, 1 :14 – 20, 1978.
- [68] Michael J. Schulte and Earl E. Swartzlander, Jr. Hardware design and arithmetic algorithms for a variable-precision, interval arithmetic coprocessor. In *Proceedings of the 12th Symposium on Computer Arithmetic*, ARITH '95, Bath, England, UK, 1995.
- [69] Michael J. Schulte and Earl E. Swartzlander, Jr. A family of variable-precision interval arithmetic processors. *IEEE Transactions on Computers*, 49 :387 – 397, 2000.
- [70] Berkeley sensor & actuator center (<http://www.bsac.eecs.berkeley.edu/>). Gyrometer.
- [71] M.D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41 :439 – 517, 1993.
- [72] G. M. Siouris, Guanrong Chen, and Jianrong Wang. Tracking an incoming ballistic missile using an extended interval kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 33 :232 – 240, 1997.
- [73] A. Tanwani, J. Galdun, J.M. Thiriet, S. Lesecq, and S. Gentil. Experimental Networked Embedded Minidrone. Part I : Consideration of faults. In *Proceeding of ECC 2007*, Kos, Greece, 2007.
- [74] T.Csendes and D.Ratz. Subdivision direction selection in interval methods for global optimization. *SIAM Journal on Numerical Analysis*, 34 :922 – 938, 1997.
- [75] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis. Part I : Quantitative model-based methods. *Computers & chemical engineering*, 27 :293 – 311, 2003.
- [76] V.G.Héctor-Moisés. *Contribution à l'Estimation d'Etat à Horizon Glissant par Méthodes Ensemblistes : Applications à la Surveillance et Détection des Dysfonctionnements sur des Bioprocédés*. PhD thesis, Université Joshep Fourier de Grenoble, 2002.
- [77] K. Vladik, Hung T. Nguyen, P. D. Gracaliz, and B. Benjamin. A new differential formalism for interval-valued functions and its potential use in detecting 1-d landscape features. In *The international conference on information technology Intech'03*, pages 491 – 498, Chiang Mai, Thailan, 2003.
- [78] V.Lemesle and J.L.Gouzé. Hybrid bounded error observers for uncertain bioreactor models. *Bioprocess And Biosystems Engineering*, 27 :311 – 318, 2005.

- [79] Waltz and L.David. Generating semantic descriptions from drawings of scenes with shadows. *The Psychology of Computer Vision*, McGraw-Hill, New York, NY, pages 19 – 91, 1975.
- [80] G. Whaba. A least squares estimate of spacecraft attitude. *SIAM Review*, 7, 1965.
- [81] wikipedia (<http://en.wikipedia.org/wiki/Accelerometer>). Accelerometer.
- [82] wikipedia (<http://en.wikipedia.org/wiki/Gyroscope>). Rate gyro.
- [83] wikipedia (<http://en.wikipedia.org/wiki/Magnetometer>). Magnetometers.
- [84] A.S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12 :601 – 611, 1976.
- [85] M. Witczak. *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems : From analytical to soft computing approaches*. Springer, 2007.

Résumé : L'estimation de l'attitude (ou orientation) est un problème récurrent de nombreuses applications allant de la robotique aérienne ou sous-marine en passant par des applications médicales (surveillance de patients, réhabilitation), mais aussi jeux vidéo, etc. L'objectif de cette thèse est d'évaluer l'apport des approches ensemblistes dans le cadre de l'estimation de l'attitude à partir de données issues de triaxes accéléromètres (A), magnétomètres (M) et gyromètres (G). Dans un premier temps, on s'intéresse aux mouvements « quasi-statiques » et l'estimation de l'attitude est réalisée à partir de mesures AM. On aborde ensuite le cas des mouvements dynamiques, en considérant l'ensemble des mesures AGM. Le problème du choix de la paramétrisation de l'attitude a été abordé et on a comparé les résultats obtenus et le temps calcul pour des modélisations avec les angles de Cardan et le quaternion unitaire. Les algorithmes développés ont été validés en simulation et avec des données réelles. Les résultats ont été comparés avec ceux fournis par des algorithmes de l'état de l'art, par exemple SIVIA. La deuxième partie du manuscrit est consacrée à au diagnostic des capteurs de la centrale inertielle avec des approches ensemblistes. Les algorithmes développés dans la première partie du travail sont adaptés afin de pouvoir détecter et localiser un défaut dans l'ensemble des capteurs considérés.

Mots clefs : Estimation d'attitude, Estimation ensembliste, diagnostic ensembliste, FDI, intervalle, centrale inertielle, IMU, angles de Cardan, angles d'Euler, quaternion unitaire.

Abstract: Attitude estimation is one of the prominent problem encountered in various application areas such as Aerial and submarine robotics, bio-medical applications (elderly people monitoring, rehabilitation) but also, video game and augmented reality. The main objective of this PhD is to assess the capabilities of set-membership estimation in the field of attitude estimation when triaxes accelerometer (A) magnetometer (M) and rate gyros (G) are used. Quasi-static movements are first considered. In this case AM measurements are taken into account. Then the dynamic case is considered with AGM measurement taken into account in the set-membership estimation algorithm. The problem of attitude parametrisation is also studied as it will have a strong influence on the computational time. The algorithms proposed during this work have been validated with simulated and real data. The second part of the report deals with Fault Detection and Isolation based upon set-membership approaches. The algorithms that have been developed in the first part of this work have been adapted to cope with diagnosis of a faulty sensor within the Inertial Measurement Unit.

Keywords : Attitude estimation, Set-membership estimation, Set-membership diagnosis, FDI, Interval analysis, Inertial Measurement Unit, IMU, Cardan, Euler, Quaternion.