



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Institut National Polytechnique de Toulouse*
Discipline ou spécialité : *Signal, Image, Acoustique et Optimisation*

Présentée et soutenue par *Jean-Rémi De Boer*
Le *12 janvier 2010*

Titre : *Capteurs MEMS : Optimisation des méthodes de traitement capteurs, de navigation et d'hybridation*

JURY

M. André Ferrari.....Rapporteurs
M. Guillaume Gelle.....
M. Jacques Blanc-Talon.....Examineurs
M. Bertrand Lesot.....
Mme Corinne Mailhes.....
M. Vincent Calmettes.....Encadrants
M. Jean-Yves Tourneret.....

Ecole doctorale : *Mathématiques, Informatique et Télécommunications de Toulouse*
Unité de recherche : *Institut de Recherche en Informatique de Toulouse*
Directeur(s) de Thèse : *Jean-Yves Tourneret*
Rapporteurs :

Capteurs MEMS : Optimisation des méthodes de
traitement capteurs, de navigation et
d'hybridation

J.-R. De Boer

25 janvier 2010

Remerciements

Je remercie André Ferrari et Guillaume Gelle pour avoir accepté de rapporter ce manuscrit ainsi que pour leurs commentaires sur ce travail. J'adresse également mes remerciements à Corinne Mailhes pour avoir accepté de présider son second jury de thèse, après s'être exercée sur mon ancienne binôme N7ienne.

Je remercie Jean-Yves Tourneret de m'avoir encadré durant ces trois années, de m'avoir fait confiance en laissant libre cours à mes digressions tout en gardant un œil attentif à la bonne conduite de cette thèse et, finalement, de m'avoir laissé travailler sur les réseaux de neurones, sa nouvelle passion !

Je remercie Vincent Calmettes pour tout ce qu'il a apporté à cette thèse, aussi bien pour son encadrement scientifique que pour ses encouragements à publier, sans oublier les campagnes de mesures mouvementées.

Je remercie Bertrand Lesot pour toutes les fois où sa grande connaissance du domaine de la navigation m'a permis de garder le cap.

Je remercie la DGA qui a financé cette thèse et, plus particulièrement, Gilles Clerc-Renaud et Jacques Blanc-Talon pour avoir suivi mes travaux.

Je remercie tous mes collègues du labo (IRIT et TéSA) pour ces trois années passées en leur compagnie et pour tous les souvenirs que j'y ai gagnés. Pour être sûr de n'oublier personne, voici seulement les premières lettres des prénoms de toutes les personnes que j'ai rencontrées : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.

Je remercie mes parents de m'avoir poussé vers la voie des études et mon frère de m'en avoir montré le chemin. Sans eux, je ne serais pas celui que je suis aujourd'hui. Je remercie également ma famille au sens élargi du terme (belle-famille, amis et filleules inclus !) pour leur soutien sans faille et pour tous les bons moments passés et à venir.

Et surtout, je remercie Stéphanie, mon épouse, sans qui rien n'aurait la même saveur. Te savoir à mes côtés est ma plus belle réussite.

Table des matières

1	Introduction	3
2	Fondamentaux de la navigation	9
2.1	Introduction	9
2.2	Navigation à l'estime	10
2.2.1	Définition des repères nécessaires	10
2.2.2	Plate-forme inertielle	12
2.2.3	Équations de la navigation inertielle	14
2.3	Radionavigation	16
2.3.1	Principe du positionnement par satellite	17
2.3.2	Global Positioning System (GPS)	19
2.3.3	GLObal NAvigation Satellite System (GLONASS)	20
2.3.4	Systèmes en cours de développement	21
2.4	Navigation hybride	22
2.4.1	Architectures d'hybridation	22
2.4.2	Modèle d'état	23
2.4.3	Méthode de résolution	28
2.5	Conclusion	31
3	Évaluation des capteurs MEMS	35
3.1	Introduction	35
3.2	Caractérisation des capteurs MEMS	36
3.2.1	Cas de l'accéléromètre	36
3.2.2	Cas du gyromètre	38
3.3	Performances en navigation inertielle	40
3.3.1	Accéléromètres seuls	40
3.3.2	Gyromètres seuls	40
3.3.3	Accéléromètres et gyromètres	42
3.4	Performances en navigation hybride	42
3.4.1	Perte partielle des signaux GNSS	42
3.4.2	Perte totale des signaux GNSS	45
3.5	Conclusion	46

4	Inversion du modèle des capteurs MEMS	49
4.1	Introduction	49
4.2	Calibrage	50
4.2.1	Cas de l'accéléromètre	50
4.2.2	Cas du gyromètre	51
4.3	Inversion du modèle	52
4.3.1	Cas de l'accéléromètre	53
4.3.2	Cas du gyromètre	53
4.4	Résultats de simulations	54
4.4.1	Calibrage	54
4.4.2	Inversion du modèle	61
4.4.3	Navigation	63
4.5	Conclusion	66
5	Correction par réseau de neurones	69
5.1	Introduction	69
5.2	Bases des réseaux de neurones	70
5.2.1	La cellule élémentaire : le neurone	70
5.2.2	Structure du réseau de neurones	71
5.2.3	Algorithmes d'apprentissage	73
5.3	Application à la navigation	74
5.3.1	Correction de la position uniquement	75
5.3.2	Correction des différentes estimées inertielles	76
5.4	Résultats de simulations	79
5.4.1	Cas simple de trajectoires invariantes dans le temps	81
5.4.2	Cas de trajectoires évoluant dans le temps	82
5.4.3	Corrections des différentes estimées inertielles	88
5.5	Conclusion	97
6	Conclusion	101
A	Modélisation complète des MEMS	115
B	Variance d'Allan	119
C	Rétropropagation du gradient	123

Table des figures

2.1	Repères nécessaires à la navigation inertielle	11
2.2	Angles d'Euler	13
2.3	Plate-forme inertielle classique en "strapdown"	17
2.4	Calcul de la position par trilatération	18
2.5	Architecture d'hybridation lâche	23
2.6	Architecture d'hybridation serrée	24
2.7	Architecture d'hybridation très serrée	24
3.1	Vue d'ensemble des accéléromètres de technologie MEMS	37
3.2	Variance d'Allan des trois accéléromètres étudiés	37
3.3	Vue d'ensemble des gyromètres de technologie MEMS	39
3.4	Variance d'Allan des trois gyromètres étudiés	39
3.5	Erreur de position liée aux perturbations accélérométriques [m]	41
3.6	Erreur de position liée aux perturbations gyrométriques [m]	41
3.7	Erreur de position liée aux perturbations accélérométriques et gyrométriques [m]	43
3.8	Constellation des satellites en vue du récepteur GNSS	44
3.9	Erreur de position liée à une perte partielle des signaux GNSS dans une navigation hybride [m]	45
3.10	Erreur de position liée à une perte totale des signaux GNSS dans une navigation hybride [m]	46
4.1	Erreur quadratique moyenne de l'estimation de $\delta_{Ba} [g^2]$	55
4.2	Erreur quadratique moyenne de l'estimation de δ_{SFa}	55
4.3	Erreur quadratique moyenne de l'estimation de $\delta_{MAa}^u [rad^2]$	56
4.4	Erreur quadratique moyenne de l'estimation de $\delta_{MAa}^v [rad^2]$	57
4.5	Erreur quadratique moyenne de l'estimation de $\delta_{MAa}^w [rad^2]$	57
4.6	Erreur quadratique moyenne de l'estimation de $\delta_{NL2a} [1/g^2]$	58
4.7	Erreur quadratique moyenne de l'estimation de $\delta_{Bg} [(\circ/s)^2]$	59
4.8	Erreur quadratique moyenne de l'estimation de δ_{SFg}	59
4.9	Erreur quadratique moyenne de l'estimation de $\delta_{MAg}^p [rad^2]$	60
4.10	Erreur quadratique moyenne de l'estimation de $\delta_{MAg}^q [rad^2]$	60
4.11	Erreur quadratique moyenne de l'estimation de $\delta_{MAg}^r [rad^2]$	61
4.12	Erreur quadratique moyenne de l'estimation de $\delta_{Kg}^p [(\circ/s/g)^2]$	62

4.13	Erreur quadratique moyenne de l'estimation de δ_{Kg}^q [$(^\circ/s/g)^2$] . . .	62
4.14	Erreur quadratique moyenne de l'estimation de δ_{Kg}^r [$(^\circ/s/g)^2$] . . .	63
4.15	Erreur quadratique moyenne de l'estimation de Γ [g^2]	64
4.16	Erreur quadratique moyenne de l'estimation de Ω [$(^\circ/s)^2$]	64
4.17	Erreur en position (biais +/- écart-type) dans le cas nominal [m]	65
4.18	Erreur en position (biais +/- écart-type) dans le cas nominal [m] (zoom sur la zone d'intérêt)	66
4.19	Erreur en position (biais +/- écart-type) en cas de perte du signal GNSS [m]	67
4.20	Erreur en position (biais +/- écart-type) en cas de perte du signal GNSS [m] (zoom sur la zone d'intérêt)	67
5.1	Cellule élémentaire générique	71
5.2	Structure interne d'un réseau de neurones	72
5.3	Phase d'apprentissage supervisé d'un réseau de neurones	72
5.4	Phase de généralisation d'un réseau de neurones	73
5.5	Architecture en phase d'apprentissage (correction de la position uniquement)	77
5.6	Construction de la base d'apprentissage	78
5.7	Architecture en phase de restitution (correction de la position uniquement)	78
5.8	Architecture en phase d'apprentissage (correction des estimées inertielles)	80
5.9	Architecture en phase de restitution (correction des estimées iner- tielles)	80
5.10	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'un point fixe	81
5.11	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire à vitesse constante	82
5.12	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire à accélération constante	83
5.13	Trajectoire simulée (Ligne droite suivie d'un virage)	84
5.14	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une ligne droite suivie d'un virage	84
5.15	Trajectoire simulée (Virage suivi d'une ligne droite)	85
5.16	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'un virage suivi d'une ligne droite	85
5.17	Trajectoire simulée (Succession de deux virages)	86
5.18	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une succession de deux virages	87
5.19	Vitesse du mouvement simulé (Vitesse constante suivie d'une accélération constante)	87
5.20	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire à vitesse constante suivie d'une accélération constante	88

5.21	Vitesse du mouvement simulé (Accélération constante suivie d'une vitesse constante)	89
5.22	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire à accélération constante suivie d'une vitesse constante	89
5.23	Vitesse du mouvement simulé (Succession de deux accélérations constantes)	90
5.24	Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire composée de deux accélérations constantes	90
5.25	Racine carrée de l'erreur quadratique moyenne de position avec 10 neurones par couche	92
5.26	Racine carrée de l'erreur quadratique moyenne de vitesse avec 10 neurones par couche	92
5.27	Racine carrée de l'erreur quadratique moyenne d'attitude avec 10 neurones par couche	93
5.28	Racine carrée de l'erreur quadratique moyenne de position avec 15 neurones par couche	93
5.29	Racine carrée de l'erreur quadratique moyenne de vitesse avec 15 neurones par couche	94
5.30	Racine carrée de l'erreur quadratique moyenne d'attitude avec 15 neurones par couche	94
5.31	Racine carrée de l'erreur quadratique moyenne de position avec 20 neurones par couche	95
5.32	Racine carrée de l'erreur quadratique moyenne de vitesse avec 20 neurones par couche	96
5.33	Racine carrée de l'erreur quadratique moyenne d'attitude avec 20 neurones par couche	96
5.34	Racine carrée de l'erreur quadratique moyenne de position avec un réseau dédié	97
5.35	Racine carrée de l'erreur quadratique moyenne de vitesse avec un réseau dédié	98
5.36	Racine carrée de l'erreur quadratique moyenne d'attitude avec un réseau dédié	98
A.1	Vue d'ensemble des erreurs liées aux systèmes inertiels [GWA07]	115
A.2	Description des erreurs d'alignement dans les systèmes inertiels [Bri71]	117
B.1	Variance d'Allan d'un bruit blanc	120
B.2	Variance d'Allan d'une marche aléatoire	122
B.3	Variance d'Allan d'un processus de Gauss Markov d'ordre 1	122

Notations

Repères nécessaires à la navigation

i	repère inertiel
e	repère géocentrique
n	repère de navigation
m	repère de mobile

Modélisation des capteurs MEMS

S	Sorties du capteur
δ_B	Biais
δ_{SF}	Facteurs d'échelles
δ_{MA}	Erreurs d'alignement
δ_{NLa}	Erreurs de non-linéarité de l'accéléromètre
δ_{Kg}	Sensibilité du gyromètre aux accélérations
v	Bruit additif du capteur

Description du mouvement du mobile

v_a^b	vitesse par rapport au repère a projetée dans le repère b
Ω_{ab}	vecteur instantané de rotation du repère a au repère b
R_{a2b}	matrice de passage du repère a au repère b
$\left[\frac{d(\cdot)}{dt} \right]_a$	dérivée par rapport au repère a
(λ, ϕ, h)	latitude, longitude et altitude
(φ, θ, ψ)	angles d'Euler

Algorithmes d'hybridation

$X(t)$	Vecteur d'état à l'instant t
$P(t)$	Matrice de covariance de l'état à l'instant t
$Y(t)$	Vecteur de mesure à l'instant t
$S(t)$	matrice de covariance de l'état à l'instant t
$(\cdot)_k$	k^{ieme} échantillon du processus discrétisé
$(\cdot)_{k k-1}$	Prédiction du processus à l'instant k sachant $k-1$
$(\cdot)_{k k}$	Estimation du processus à l'instant k sachant k

Acronymes

CDMA	Code Division Multiple Acces
FDMA	Frequency Division Multiple Acces
EKF	Extended Kalman Filter
FK	Filtre de Kalman
FP	Filtre Particulaire
GLONASS	russian GLObal NAvigation Satellite System
GEO	Geostationary Earth Orbit
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile communications
ILS	Instrument Landing System
INS	Inertial Navigation System
LORAN	LOng RAnge Navigation
MAP	Maximum A Posteriori
MC	Moindres Carrés
MEMS	Micro Electro Mechanical System
MEO	Medium Earth Orbit
MLP	Multi-Layer Perceptron
MMSE	Minimum Mean-Square Error
PPS	Precise Positioning System
RBF	Radial Basis Function
RN	Réseau de Neurones
SA	Selective Availability
SPS	Standard Positioning System
UKF	Unscented Kalman Filter
WSS	Wheel Speed Sensor

Introduction

Chapitre 1

Introduction

Contexte de la thèse

Depuis que l'humanité a eu besoin de se déplacer, connaître sa position a toujours été pour elle un enjeu. Dans ce but, elle a montré son habilité en concevant divers savoir-faire jusqu'à en faire, dans certains cas, une forme d'art. Ainsi, des méthodes ancestrales aux techniques modernes, des procédés les plus rudimentaires aux technologies les plus évoluées, l'humanité n'a eu de cesse de perfectionner sa connaissance de son positionnement. Initialement aidés uniquement de ses sens (essentiellement, la vue), l'être humain s'est rapidement vu assisté de capteurs lui permettant d'affiner la précision de son estimation.

La science de la navigation regroupe les différentes méthodes en deux grandes familles. La première est la navigation à l'estime, qui à l'aide de la connaissance de la vitesse, de l'orientation et de la position précédente du mobile va permettre de déterminer la position présente. L'évolution au cours du temps, des capteurs mesurant ces grandeurs physiques, a permis d'améliorer l'estimation du positionnement. Il faut cependant noter que l'ensemble de ces techniques souffrent d'un inconvénient majeur, les erreurs de mesure réalisées sur la vitesse et l'orientation vont se cumuler et donc conduire à des erreurs, en terme de position, croissantes avec le temps. Pour limiter ce phénomène, il est nécessaire de posséder régulièrement et directement une estimation du positionnement. Ce rôle est donné à la seconde famille de méthodes de navigation, qui en mesurant la distance et/ou l'orientation du mobile par rapport à un certain nombre de points de référence va permettre de déterminer sa position relative. Les technologies ont là aussi évolué avec le temps, permettant l'obtention d'une précision toujours plus grande.

Parmi ces différentes méthodes, deux d'entre elles ont été plus particulièrement l'objet d'un grand nombre de travaux. Dans la famille des méthodes de navigation à l'estime, la navigation inertielle (INS pour l'anglais Inertial Navigation System) se distingue par son utilisation de capteurs de mouvement (accéléromètres) et de rotation (gyromètres) pour déterminer la vitesse et l'orien-

tation du mobile. Initialement conçue durant la seconde guerre mondiale dans le fonctionnement des missiles, son utilisation ne se limite plus aujourd'hui aux activités militaires et est devenu indispensable à l'astronautique et l'aéronautique. Appartenant à la seconde grande famille de méthodes de navigation, la radionavigation, et plus particulièrement la radionavigation par satellite (GNSS pour l'anglais Global Navigation Satellite System), est basée sur l'utilisation de balises radio en tant que points de références. L'utilisation de ces deux techniques de manière combinée, appelée hybridation GNSS/INS, permet alors d'obtenir des résultats de navigation précis.

Un grand nombre d'études ont été réalisées au sujet de cette hybridation GNSS/INS. Le problème est alors communément posé sous forme d'un système dynamique et peut plus précisément s'exprimer selon une représentation d'état où les équations d'état (resp. les équations d'observation) sont obtenues à partir de la navigation inertielle (resp. la navigation par satellite). Ce problème est alors résolu par un estimateur récursif, provenant d'un filtre de Kalman ou d'un filtre particulière.

Problématique

Pour certaines applications grand public, une centrale inertielle traditionnelle équipée d'accéléromètres (piézoélectrique, à poutre vibrante, ...) et de gyromètres (gyrolaser, gyromètre à fibre optique, ...) peut se révéler trop encombrante ou trop onéreuse. Dans ce cas, la technologie des microsystèmes électromécaniques (MEMS pour l'anglais Micro Electro Mechanical Systems) peut s'avérer efficace pour réduire l'encombrement et le coût. Apparus dans les années 1970, ces systèmes sont basés sur l'alliance de l'électronique et de la mécanique à l'échelle micrométrique. Différents types de capteurs ou d'actionneurs (injecteurs pour imprimantes à jet d'encre, micro-miroirs qui définissent les pixels de certains modèles de vidéoprojecteurs, capteurs de pression, ...) relèvent de la catégorie des MEMS mais seuls les accéléromètres et les gyromètres seront exploités dans cette étude.

Malheureusement, la précision des mesures accélérométriques et gyrométriques fournies par les MEMS ne permet pas d'égaliser celle obtenue à l'aide d'une centrale inertielle traditionnelle. Il est donc nécessaire de mettre en œuvre des techniques supplémentaires pour compenser la qualité des mesures, accroissant d'autant la précision de la localisation par le biais de la plate-forme inertielle, ou/et pour réaliser directement une meilleure estimation de la position. De plus, dans le cadre de la navigation hybride, il faut différencier deux cas distincts :

- Le cas nominal, durant lequel les deux systèmes (GNSS et MEMS) sont disponibles et qui va permettre l'amélioration des sorties accélérométriques et gyrométriques à l'aide des données GNSS.
- Le cas d'une perte de signal GNSS, où seule l'intégration des mesures issues des MEMS à travers la plate-forme fournit une estimation de la position.

Dans ce manuscrit, ces deux cas feront l'objet d'études spécifiques mettant en

œuvre, respectivement, des méthodes de calibration en ligne et des méthodes de corrections neuronales également en ligne.

Organisation du manuscrit

Le document est organisé en deux grandes parties.

La première partie présente les fondamentaux de la navigation nécessaires à la bonne compréhension du manuscrit. Les principes des navigations inertielle, par satellite et hybride, ainsi que leurs méthodes de résolution, y sont détaillés. Cette étude bibliographique permet également de préciser le contexte de la thèse.

La seconde partie comprend les travaux réalisés au cours de la thèse et est organisée autour de trois axes, chacun faisant l'objet d'un chapitre. Une évaluation des capteurs MEMS est proposée dans le chapitre 3 aussi bien en terme de mesures que de navigation. Le chapitre 4 s'intéresse plus particulièrement au cas nominal (présence de signal GNSS) durant lequel l'objectif est d'estimer, au plus proche, l'accélération et la vitesse angulaire à partir des sorties issues des capteurs MEMS. Cette inversion du modèle MEMS est réalisée en deux étapes, le calibrage et l'inversion proprement dite du modèle. Dans le chapitre 5, l'objet de l'étude est l'amélioration de l'estimation de la position dans le cas de la perte de signal GNSS. L'idée retenue est de corriger le système de navigation à l'aide de réseaux de neurones.

Étude bibliographique

Chapitre 2

Fondamentaux de la navigation

Sommaire

2.1	Introduction	9
2.2	Navigation à l'estime	10
2.2.1	Définition des repères nécessaires	10
2.2.2	Plate-forme inertielle	12
2.2.3	Équations de la navigation inertielle	14
2.3	Radionavigation	16
2.3.1	Principe du positionnement par satellite	17
2.3.2	Global Positioning System (GPS)	19
2.3.3	GLOBAL NAVIGATION Satellite System (GLONASS)	20
2.3.4	Systèmes en cours de développement	21
2.4	Navigation hybride	22
2.4.1	Architectures d'hybridation	22
2.4.2	Modèle d'état	23
2.4.3	Méthode de résolution	28
2.5	Conclusion	31

2.1 Introduction

Savoir se situer est un problème que l'homme a toujours voulu résoudre et c'est pourquoi un grand nombre de techniques ont été développées en ce sens. La navigation est la science qui regroupe l'ensemble de ces techniques permettant de connaître la position d'un mobile ainsi que toutes autres informations concernant le déplacement de ce mobile dans l'espace. L'objet de ce premier chapitre est de présenter ces différentes méthodes, des plus anciennes au plus

récentes, et de détailler celles qui ont été plus particulièrement utilisées durant cette thèse dédiée au développement d’algorithmes spécifiques au cas de la navigation hybride GNSS/MEMS. Ces méthodes peuvent se séparer en deux grandes familles. La première (Navigation à l’estime [Ayl01]) cherche à estimer la position présente du mobile connaissant sa position à l’instant précédent, son orientation ainsi que sa vitesse. La seconde a pour but d’estimer la distance et/ou l’orientation du mobile à un ou plusieurs points de références dont la position est connue et ainsi de déterminer sa position dans l’espace. Ces points de références peuvent être naturels (Astronavigation ou navigation céleste [Bro02], non détaillée par la suite) ou artificiels (Radionavigation [FB98, Sec. 5],[GWA07, Sec. 2.3]) comme des balises, des satellites . . .

2.2 Navigation à l’estime

Largement utilisée pour des applications nautiques et plus tard aéronautiques, la navigation à l’estime est inspirée du comportement de tout être vivant lorsqu’il se déplace. En effet, si un individu veut se rendre d’un point A à un point B, il va tout d’abord déterminer la direction qu’il doit suivre puis va avancer un certain temps à une vitesse donnée. En admettant maintenant que le point B n’est pas connu, la problématique reste la même, l’individu choisit son orientation, sa vitesse, il se déplace pendant un temps donné et arrive alors à un nouveau point dont il peut calculer la position.

Il existe plusieurs manières de déterminer la vitesse ou l’orientation mais, de nos jours, c’est bien souvent par le biais d’une centrale inertielle que cette tâche est réalisée. Dans ces conditions, la navigation à l’estime est alors appelée “Navigation inertielle” ou INS (Inertial Navigation System) [FB98, Sec. 6], [GWA07, Sec. 9], [Kin98] dont la principale difficulté réside dans les changements de repère nécessaires pour rendre les données exploitables par l’utilisateur.

2.2.1 Définition des repères nécessaires

Repère inertielle L’origine du repère inertielle $\mathbf{i} = (X_i, Y_i, Z_i)$ (Fig. 2.1) est le centre de gravité de la Terre. Ces axes (X_i, Y_i, Z_i) ne tournent pas avec la Terre et pointent vers des étoiles suffisamment lointaines pour sembler fixes à un observateur terrestre. Z_i est l’axe de rotation de la Terre, X_i pointe vers l’équinoxe vernal et Y_i vient compléter le trièdre direct.

Repère géocentrique L’origine du repère géocentrique $\mathbf{e} = (X_e, Y_e, Z_e)$ (Fig. 2.1) est le centre de gravité de la Terre et ces axes tournent en accord avec la rotation terrestre, ce qui le rend fixe par rapport à la Terre. Par définition, les repères \mathbf{i} et \mathbf{e} se déduisent l’un de l’autre par une rotation liée à la rotation de la Terre de vitesse angulaire Ω_{ie} . Z_e est alors toujours l’axe de rotation de la Terre, X_e pointe vers l’intersection de l’équateur avec le méridien de Greenwich et Y_e vient compléter le trièdre direct. Pour des applications submétriques, il faut noter que le décalage des pôles magnétique et géographique serait à prendre

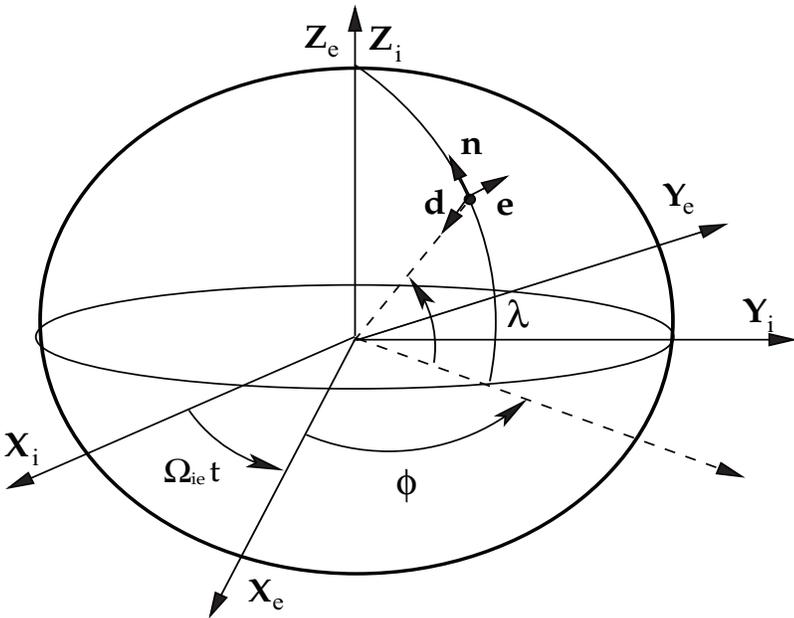


FIGURE 2.1 – Repères nécessaires à la navigation inertielle

en compte, ce qui ne sera pas le cas dans le reste de l'étude. La relation pour passer du repère géocentrique au repère inertiel est alors :

$$R_{e2i} = \begin{bmatrix} \cos(\Omega_{iet}) & -\sin(\Omega_{iet}) & 0 \\ \sin(\Omega_{iet}) & \cos(\Omega_{iet}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Dans ce repère, il est possible d'écrire un même point selon différentes coordonnées. Ainsi il est intéressant de définir la position en coordonnées ellipsoïdales par sa latitude λ , sa longitude ϕ et son altitude h , le lien entre les coordonnées ellipsoïdales (λ, ϕ, h) et cartésiennes (X_e, Y_e, Z_e) pouvant s'écrire sous la forme :

$$\left. \begin{aligned} X_e &= (N+h) \cos \lambda \cos \phi, \\ Y_e &= (N+h) \cos \lambda \sin \phi, \\ Z_e &= (N+h) \sin \lambda, \end{aligned} \right\} \begin{array}{l} \text{où } N = a\sqrt{1-e^2}/(1-e^2 \sin^2 \lambda), \text{ avec } a \text{ et } e \\ \text{étant respectivement le demi-grand axe et} \\ \text{l'excentricité de l'ellipsoïde terrestre.} \end{array} \quad (2.2)$$

Dans le cas d'applications submétriques, le rayon moyen de la Terre N ne peut être directement utilisé dans les équations 2.2, ce qui ferait intervenir les rayons de courbure en longitude et en latitude.

Repère de navigation L'origine du repère de navigation $\mathbf{n} = (n, e, d)$ (Fig. 2.1) est le centre de gravité du mobile. n pointe vers le nord géodésique, d est orthogonal à l'ellipsoïde de référence représentant la Terre et en direction du sol et e vient compléter le trièdre direct (ce qui le fait pointer globalement à l'est). La relation pour passer du repère géocentrique au repère de navigation est alors :

$$R_{e2n} = \begin{bmatrix} -\sin \lambda \cos \phi & -\sin \lambda \sin \phi & \cos \lambda \\ -\sin \phi & \cos \phi & 0 \\ -\cos \lambda \cos \phi & -\cos \lambda \sin \phi & -\sin \lambda \end{bmatrix} \quad (2.3)$$

Repère du mobile L'origine du repère du mobile $\mathbf{m} = (X_m, Y_m, Z_m)$ est le centre de gravité du mobile. Ce repère est solidaire du mobile et ces axes tendent à coïncider avec les axes de la plate-forme inertielle. Son orientation par rapport au repère de navigation \vec{n} est décrite par les angles d'Euler (φ, θ, ψ) (Fig. 2.2) représentant respectivement le roulis, le tangage et le lacet. La relation pour passer du repère de navigation au repère du mobile est alors :

$$R_{n2m} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \varphi \sin \psi + \cos \psi \sin \theta \sin \varphi & \cos \varphi \cos \psi + \sin \psi \sin \theta \sin \varphi & \cos \theta \sin \varphi \\ \sin \varphi \sin \psi + \cos \psi \sin \theta \cos \varphi & -\sin \varphi \cos \psi + \sin \psi \sin \theta \cos \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (2.4)$$

2.2.2 Plate-forme inertielle

Comme énoncé précédemment, une centrale (ou plate-forme) inertielle a pour but de délivrer une mesure de la vitesse et de l'orientation du mobile afin d'en calculer sa position. Pour ce faire, elle est composée de trois accéléromètres et

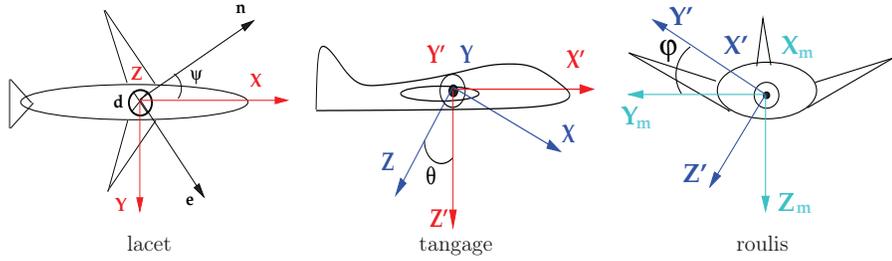


FIGURE 2.2 – Angles d'Euler

de trois gyromètres permettant de mesurer l'accélération et la vitesse angulaire du mobile dans l'espace. Ces deux grandeurs sont en effet observables grâce à la seconde loi de Newton (2.5) et respectivement grâce aux effets d'entraînement de repère ou à l'effet Sagnac.

$$\sum \vec{F}_i = m\vec{a} \text{ où } \begin{cases} \vec{F}_i & \text{désigne les forces extérieures exercées sur l'objet,} \\ m & \text{est la masse de l'objet,} \\ \vec{a} & \text{correspond à l'accélération du centre d'inertie.} \end{cases} \quad (2.5)$$

Il suffit alors d'intégrer dans le temps ces deux observations pour estimer la vitesse et l'orientation du mobile.

Accéléromètre Il existe un grand nombre de technologies permettant de mesurer une accélération pouvant se répartir deux grandes familles : les accéléromètres non asservis et les accéléromètres à asservissement. Il est cependant intéressant de noter que, dans les deux cas, l'erreur de mesure peut s'écrire sous la forme :

$$S_a = (I - \delta_{SFa} - \delta_{MAa})(\Gamma - \delta_{Ba} - \delta_{NL a} - v_a) \quad (2.6)$$

$$\text{où } \begin{cases} \Gamma & \text{est le vecteur accélérations } \in \mathbb{R}^3, \\ S_a & \text{est le vecteur sorties accélérométriques } \in \mathbb{R}^3, \\ \delta_{Ba} & \text{est le vecteur biais accélérométriques } \in \mathbb{R}^3, \\ \delta_{SFa} & \text{est la matrice diagonale } 3 \times 3 \text{ des erreurs de facteurs d'échelle,} \\ \delta_{NL a} & \text{est le vecteur contenant les erreurs de non-linéarité } \in \mathbb{R}^3, \\ \delta_{MAa} & \text{est la matrice hors-diagonale } 3 \times 3 \text{ des erreurs d'alignement,} \\ v_a & \text{est le vecteur des bruits accélérométriques } \in \mathbb{R}^3. \end{cases}$$

Gyromètre De même que pour les accéléromètres, différentes technologies sont disponibles pour mesurer une vitesse angulaire. Les gyromètres mécaniques utilisent les principes de conservation du moment cinétique et de la force de Coriolis tandis que les gyromètres optiques recourent à l'effet Sagnac. L'erreur de mesure peut s'écrire de manière générale :

$$S_g = (I - \delta_{SFg} - \delta_{MAg})(\Omega - \delta_{Bg} - \delta_{Kg} - v_g) \quad (2.7)$$

$$\text{où } \begin{cases} \Omega & \text{est le vecteur vitesses angulaires } \in \mathbb{R}^3, \\ S_g & \text{est le vecteur sorties gyrométriques } \in \mathbb{R}^3, \\ \delta_{Bg} & \text{est le vecteur biais gyrométriques } \in \mathbb{R}^3, \\ \delta_{SFg} & \text{est la matrice diagonale } 3 \times 3 \text{ des erreurs de facteurs d'échelle,} \\ \delta_{MAg} & \text{est la matrice hors-diagonale } 3 \times 3 \text{ des erreurs d'alignement,} \\ \delta_{Kg} & \text{est la matrice } 3 \times 3 \text{ de sensibilité aux accélérations,} \\ v_g & \text{est le vecteur des bruits gyrométriques } \in \mathbb{R}^3. \end{cases}$$

Cette écriture des modèles a été détaillée dans [FB98, Sec. 6.5] et sera utilisé dans la suite de ce manuscrit. Cependant, il est bon de savoir qu'il existe également des modèles plus complets (A).

Structure de la plate-forme L'utilisation de ces capteurs peut se faire à travers deux structures fondamentalement différentes :

- Les Systèmes à plate-forme stabilisée (“gimbaled” ou “floated” systems) ont l'avantage d'isoler les capteurs de certaines perturbations et permettent ainsi d'obtenir des résultats plus précis en terme de navigation. Cependant, leur prix et leur encombrement ne leur permettent pas d'être utilisés aussi bien pour des applications grand public que dans les cas de forte miniaturisation.
- Les Systèmes inertiels à composants liés (“strapdown” systems) nécessitent un coût calculatoire plus important mais permettent d'éliminer la stabilisation de la plate-forme, ce qui diminue leur coût et leur taille. À capteurs équivalents, les résultats de navigation de ces systèmes seront moins précis que ceux obtenus avec la structure précédente.

2.2.3 Équations de la navigation inertielle

Ces équations relient les mesures accélérométriques et gyrométriques aux estimées inertielles (position, vitesse, attitude). Pour des raisons de commodité, la position sera exprimée en coordonnées ellipsoïdales, la vitesse du mobile dans le repère géocentrique sera projetée dans le repère de navigation et l'attitude sera décrite par les angles d'Euler.

Équation de position Par définition, on peut écrire la vitesse dans le repère géocentrique, projetée dans le repère de navigation

$$v_e^n = \left[\frac{dr^n}{dt} \right]_e \text{ avec } r^n = \begin{pmatrix} 0 \\ 0 \\ -(N+h) \end{pmatrix}.$$

Cette vitesse v_e^n peut alors également s'exprimer sous la forme

$$v_e^n = -\frac{dh}{dt}d - (N+h) \left[\frac{dd}{dt} \right]_e, \text{ où } \left[\frac{dd}{dt} \right]_e = \begin{pmatrix} -\frac{d\lambda}{dt} \\ -\frac{d\phi}{dt} \cos \lambda \\ 0 \end{pmatrix}$$

d'après les formules de passage du repère géocentrique au repère de navigation (2.3). On peut alors en déduire une nouvelle écriture de la vitesse $v_{\mathbf{e}}^{\mathbf{n}}$

$$v_{\mathbf{e}}^{\mathbf{n}} = \begin{pmatrix} (N+h) \frac{d\lambda}{dt} \\ (N+h) \frac{d\phi}{dt} \cos \lambda \\ -\frac{dh}{dt} \end{pmatrix},$$

ce qui nous donne finalement l'équation de position

$$\frac{d}{dt} \begin{pmatrix} \lambda \\ \phi \\ h \end{pmatrix} = \begin{pmatrix} \frac{1}{N+h} & 0 & 0 \\ 0 & \frac{1}{(N+h) \cos \lambda} & 0 \\ 0 & 0 & -1 \end{pmatrix} v_{\mathbf{e}}^{\mathbf{n}} \quad (2.8)$$

Équation de vitesse Par définition, on peut écrire la vitesse du mobile dans le repère géocentrique

$$v_{\mathbf{e}} = \left[\frac{dr}{dt} \right]_{\mathbf{e}}.$$

Cette vitesse $v_{\mathbf{e}}$ peut également s'exprimer à l'aide du théorème de Coriolis

$$v_{\mathbf{e}} = \left[\frac{dr}{dt} \right]_{\mathbf{i}} + \Omega_{\mathbf{ei}} \wedge r.$$

On peut alors déduire l'expression de la dérivée de la vitesse $v_{\mathbf{e}}$ par rapport au repère de navigation

$$\begin{aligned} \left[\frac{dv_{\mathbf{e}}}{dt} \right]_{\mathbf{n}} &= \frac{d}{dt} \left[\left[\frac{dr}{dt} \right]_{\mathbf{i}} + \Omega_{\mathbf{ei}} \wedge r \right]_{\mathbf{i}} + \Omega_{\mathbf{ni}} \wedge \left[\left[\frac{dr}{dt} \right]_{\mathbf{i}} + \Omega_{\mathbf{ei}} \wedge r \right] \\ &= \left[\frac{d^2 r}{dt^2} \right]_{\mathbf{i}} + (\Omega_{\mathbf{ei}} + \Omega_{\mathbf{ni}}) \wedge \left[\frac{dr}{dt} \right]_{\mathbf{i}} + \Omega_{\mathbf{ni}} \wedge \Omega_{\mathbf{ei}} \wedge r \\ &= \left[\frac{d^2 r}{dt^2} \right]_{\mathbf{i}} + (\Omega_{\mathbf{ei}} + \Omega_{\mathbf{ni}}) \wedge (v_{\mathbf{e}} - \Omega_{\mathbf{ei}} \wedge r) + \Omega_{\mathbf{ni}} \wedge \Omega_{\mathbf{ei}} \wedge r \\ &= \left[\frac{d^2 r}{dt^2} \right]_{\mathbf{i}} - (\Omega_{\mathbf{en}} + 2\Omega_{\mathbf{ie}}) \wedge v_{\mathbf{e}} - \Omega_{\mathbf{ei}} \wedge \Omega_{\mathbf{ei}} \wedge r. \end{aligned}$$

La force f_a qui peut être mesurée par les accéléromètres est l'accélération inertielle non gravitationnelle

$$f_a = \left[\frac{d^2 r}{dt^2} \right]_{\mathbf{i}} - g(r) - \Omega_{\mathbf{ei}} \wedge \Omega_{\mathbf{ei}} \wedge r,$$

ce qui nous permet d'écrire l'équation de vitesse en projetant le résultat dans le repère de navigation

$$\frac{dv_{\mathbf{e}}^{\mathbf{n}}}{dt} = R_{m2n} f_a + g(r)^{\mathbf{n}} - a_c, \quad (2.9)$$

où $\begin{cases} R_{m2n} \text{ est la matrice de passage définie en (2.4),} \\ g(r) \text{ est la gravité estimée en la position } r, \\ a_c = (\Omega_{\mathbf{en}}^{\mathbf{n}} + 2\Omega_{\mathbf{ie}}^{\mathbf{n}}) \wedge v_{\mathbf{e}}^{\mathbf{n}}, \text{ est l'accélération de Coriolis,} \end{cases}$

Équation d'attitude Par définition, les angles d'Euler (φ, θ, ψ) caractérisent le passage du repère de navigation au repère du mobile dans le cas d'une structure "strapdown". Il existe donc une relation entre le vecteur instantané de rotation Ω_{nm}^m et $(\frac{d\varphi}{dt}, \frac{d\theta}{dt}, \frac{d\psi}{dt})$. Considérant les rotations de ce changement de repères (Fig. 2.2) de manière indépendante, il est aisé de voir que la vitesse angulaire de la rotation autour de X_m (resp. Y' et Z) est simplement $\frac{d\varphi}{dt}$ (resp. $\frac{d\theta}{dt}$ et $\frac{d\psi}{dt}$). En projetant dans le repère du mobile ces résultats, on obtient alors

$$\Omega_{nm}^m = \begin{pmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \varphi & \sin \varphi \cos \theta \\ 0 & -\sin \varphi & \cos \varphi \cos \theta \end{pmatrix} \begin{pmatrix} \frac{d\varphi}{dt} \\ \frac{d\theta}{dt} \\ \frac{d\psi}{dt} \end{pmatrix},$$

ce qui nous permet d'écrire l'équation d'attitude

$$\frac{d}{dt} \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \frac{\sin \varphi}{\cos \theta} & \frac{\cos \varphi}{\cos \theta} \end{pmatrix} \Omega_{nm}^m. \quad (2.10)$$

Il est intéressant de noter que Ω_{nm}^m ne correspond pas directement à la mesure des gyromètres Ω_{im}^m mais qu'ils sont cependant liés par

$$\begin{aligned} \Omega_{nm}^m &= \Omega_{im}^m - R_{n2m} \Omega_{in}^n \\ &= \Omega_{im}^m - R_{n2m} (\Omega_{ie}^n + \Omega_{en}^n) \\ &= \Omega_{im}^m - R_{n2m} (R_{e2n} \Omega_{ie}^e + \Omega_{en}^n). \end{aligned}$$

où tous les termes de l'égalité sont connus.

Un schéma fonctionnel de la plate-forme suivant les équations de la navigation inertielle décrites en (2.8), (2.9) et (2.10) peut alors être réalisé (Fig. 2.3).

2.3 Radionavigation

L'astronavigation permet de calculer la localisation d'un mobile à partir de l'observation de différents corps célestes utilisés comme "balises" de positionnement. La radionavigation est le prolongement technologique de cette navigation céleste où les ondes lumineuses ont été remplacées par les ondes radioélectriques et les corps célestes par des balises artificielles (stations au sol, satellites, ...). Ceci permet à la radionavigation d'avoir comme avantage majeur par rapport à la navigation céleste de ne plus être soumis aux conditions de visibilité pour pouvoir être utilisée.

Le principe de cette méthode réside en le calcul de la distance entre des positions connues et le mobile dont il faut déterminer la position. Contrairement aux méthodes de navigation à l'estime, la radionavigation nécessite la mise en oeuvre d'une infrastructure conséquente de balises. Il existe deux types différents de systèmes de radionavigation : les systèmes à base terrestre (LORAN, ILS, ..., non détaillés par la suite) et les systèmes de positionnement par satellites (GNSS).

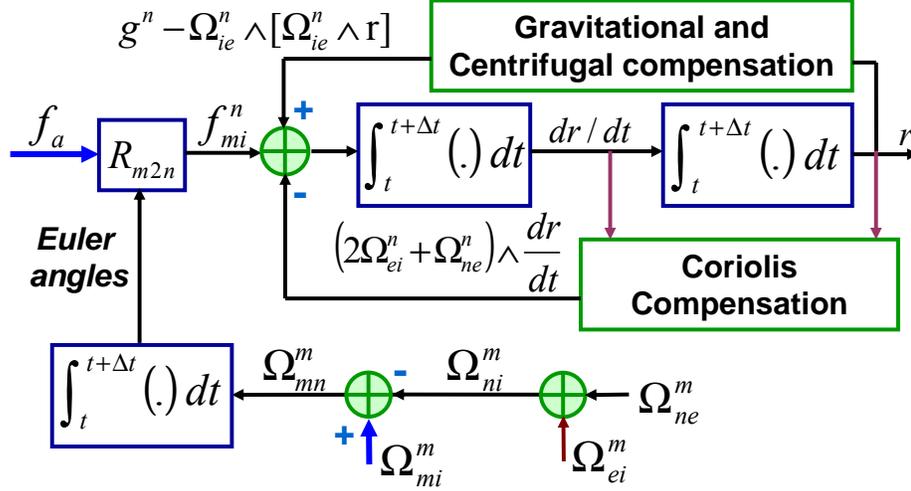


FIGURE 2.3 – Plate-forme inertielle classique en “strapdown”

2.3.1 Principe du positionnement par satellite

Les systèmes GNSS (Global Navigation Satellite System) délivrent à tout utilisateur muni d’un récepteur adéquat des informations précises de position, vitesse et temps. Pour ce faire, le récepteur effectue des mesures du temps écoulé entre l’émission des signaux radioélectriques par les satellites et la réception par l’antenne du récepteur. Connaissant alors la vitesse de propagation des ondes radio, il est possible de déterminer la distance séparant le satellite émetteur du récepteur GNSS. Trois mesures sont théoriquement nécessaires à un positionnement dans l’espace. Le mobile est alors localisé à l’une des intersections des trois sphères (Fig. 2.4) de centre la position des satellites et de rayons les mesures effectuées par la démarche précédemment décrite, l’autre intersection étant aberrante du point de vue de la solution recherchée.

Il faut cependant remarquer que l’horloge du récepteur n’est pas synchronisée aux horloges des satellites, ce qui va perturber la mesure du temps de propagation. La mesure de la distance satellite/récepteur est affectée par cette erreur d’horloge et par conséquent cette mesure est appelée pseudo-distance et peut être modélisée par

$$\rho^i = R^i + c(\Delta\tau_R - \Delta\tau_{S^i}) + c\Delta a + w^i, \quad (2.11)$$

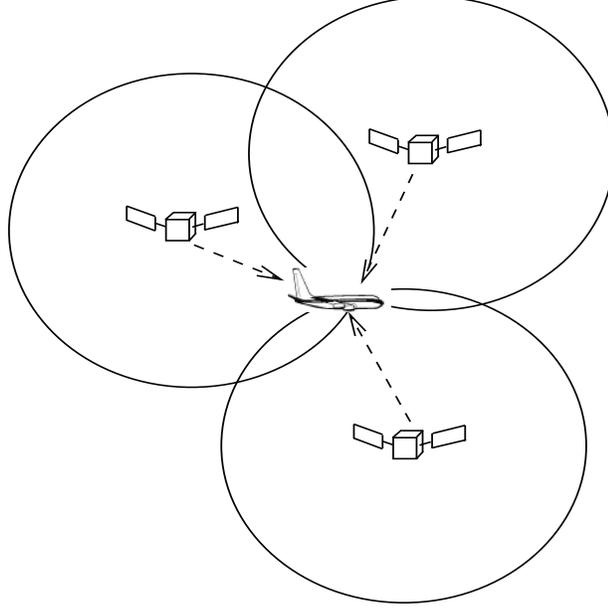


FIGURE 2.4 – Calcul de la position par trilatération

$$\text{où } \left\{ \begin{array}{l} R^i \text{ est la distance entre le récepteur et le } i\text{-ème satellite,} \\ c \text{ est la célérité,} \\ \Delta\tau_R, \text{ est l'erreur d'horloge du récepteur par rapport au temps de} \\ \text{référence GNSS,} \\ \Delta\tau_{S^i} \text{ est l'erreur d'horloge du } i\text{-ème satellite par rapport au} \\ \text{temps de référence GNSS,} \\ \Delta a \text{ sont les retards induits par la traversée de l'atmosphère,} \\ w^i \text{ est une erreur résiduelle.} \end{array} \right.$$

De plus, la distance entre le i -ème satellite et le récepteur peut s'écrire

$$R^i = \sqrt{(X - X^i)^2 + (Y - Y^i)^2 + (Z - Z^i)^2}, \quad (2.12)$$

$$\text{où } \left\{ \begin{array}{l} (X, Y, Z) \text{ est la position du récepteur dans le repère géocentrique,} \\ (X^i, Y^i, Z^i) \text{ est la position du } i\text{-ème satellite dans ce même repère.} \end{array} \right.$$

Dans ces conditions, l'intersection des trois sphères n'est plus un point unique mais devient un domaine et il sera nécessaire de posséder une mesure supplémentaire pour lever l'incertitude. En pratique, cela revient à estimer l'erreur d'horloge récepteur car il existe des modèles d'erreur pour évaluer la dérive de l'horloge satellite par rapport au temps de référence GNSS. Le problème ainsi posé ((2.11) et (2.12)) revient alors approximativement à la résolution d'un système d'au minimum quatre équations à quatre inconnues.

2.3.2 Global Positioning System (GPS)

Le GPS est le second système de navigation par satellite développé par les États-Unis, après Transit. Son développement commença dans les années 70 mais le système ne fut pleinement opérationnel que dans les années 90 lorsque la constellation initialement prévue fut atteinte.

Infrastructure du GPS Le système GPS est composé de 3 parties distinctes, appelées encore segments :

- **Le segment spatial** est constitué actuellement d'une constellation de 31 satellites (24 initialement prévus). Ces satellites évoluent sur six plans orbitaux ayant une inclinaison d'environ 55° , deux plans orbitaux étant séparés de 60° . Ces orbites ont été choisies pour garantir qu'au moins six satellites soient toujours en vue de n'importe quel point du globe terrestre. Les satellites suivent une orbite quasi circulaire de rayon orbital moyen 26560 km qu'ils parcourent en un demi jour sidéral, soit en 11h 58min 02s.
- **Le segment de contrôle** assure la surveillance de la constellation GPS à travers cinq stations réparties dans le monde (Hawaii, Kwajalein, Ascension Island, Diego Garcia, et Colorado Springs). Les données collectées sont alors centralisées à la station maîtresse de Colorado Springs, où les corrections à apporter aux satellites (éphémérides, paramètres d'horloge) sont calculées. Les mises à jour sont alors transmises aux satellites via les différentes stations de contrôle.
- **Le segment utilisateurs** regroupe l'ensemble des récepteurs GPS qui réalisent le calcul de positionnement en temps réel. Le GPS propose deux types de services (trois initialement prévus) : le standard positioning system (SPS) accessible à tous et le precise positioning system (PPS) réservé à des utilisateurs habilités par le ministère de la défense américaine. Le troisième type de service consistait en la dégradation volontaire du signal GPS, connu sous le nom de selective availability (SA). Ce service fut interrompu le 1^{er} Mai 2000.

Signal GPS Chaque satellite émet continûment un message de navigation à 50 bit/s qui contient la date d'émission, la position du satellite et la correction d'horloge, informations nécessaires à la résolution du problème de navigation. Ce message de navigation est rendu robuste aux perturbations extérieures grâce à l'utilisation d'un code pseudo-aléatoire et la séquence résultante module la porteuse adéquat. Le signal GPS ainsi réalisé est alors émis sur ces deux porteuses en bande L , dénotées L_1 et L_2 :

$$s_{L_1}(t) = \sqrt{2A}d(t)c(t) \cos(2\pi f_1 t + \theta_1) + \sqrt{A}d(t)p(t) \sin(2\pi f_1 t + \theta_1), \quad (2.13)$$

$$s_{L_2}(t) = \sqrt{\frac{A}{2}}d(t)p(t) \sin(2\pi f_2 t + \theta_2), \quad (2.14)$$

$$\text{où } \left\{ \begin{array}{l} A \quad \text{est la puissance moyenne du signal émis,} \\ d(t) \quad \text{est le message de navigation,} \\ c(t) \quad \text{est le signal correspondant au code pseudo-aléatoire } C/A, \\ p(t) \quad \text{est le signal correspondant au code pseudo-aléatoire } P, \\ f_1 \quad \text{est fréquence de la porteuse } L_1 (= 1575,42\text{Mhz}), \\ f_2 \quad \text{est fréquence de la porteuse } L_2 (= 1227,6\text{Mhz}). \end{array} \right.$$

Il faut noter que chaque satellite utilise des codes pseudo-aléatoires orthogonaux les uns par rapport aux autres, ce qui aura la particularité de donner de faibles intercorrélations entre deux différents codes. Cette technique de transmission permet à tous les satellites GPS de pouvoir émettre simultanément sans interférences et porte le nom de CDMA (code division multiple acces).

2.3.3 GLObal NAVigation Satellite System (GLONASS)

Le GLONASS est le premier système de navigation par satellite développé par l'URSS, puis par la Russie. Son développement commença dans les années 70 avec pour but d'être pleinement opérationnel dans les années 90. La chute de l'URSS a entraîné son déclin mais, en 2001, la Russie s'est engagée à restaurer le système.

Infrastructure du GLONASS Le système GLONASS est composé lui aussi de 3 segments distincts :

- **Le segment spatial** est constitué actuellement d'une constellation de 20 satellites, dont 19 sont opérationnels (24 prévus pour fin 2009). Ces satellites évoluent sur trois plans orbitaux ayant une inclinaison d'environ 65° , deux plans orbitaux étant séparé de 120° . Ces orbites ont été choisies pour garantir qu'au moins cinq satellites soient toujours en vue de n'importe quel point du globe terrestre. Les satellites suivent une orbite quasi circulaire de rayon orbital moyen 25510 km qu'ils parcourent en 11h 15min 44s.
- **Le segment de contrôle** assure la surveillance de la constellation GLONASS à travers cinq stations réparties en ex-URSS : Saint-Pétersbourg, Ternopil, Ienisseïsk, Komsomolsk-sur-l'Amour et Krasnoznamensk (station de contrôle principale).
- **Le segment utilisateurs** réunit l'ensemble des récepteurs utilisant les signaux GLONASS. Il propose deux types de service, l'un accessible à tous les utilisateurs et l'autre réservé aux utilisateurs ayant droit. Contrairement au GPS, aucun système de dégradation volontaire (SA) du signal n'a été mis en place.

Signal GLONASS Chaque satellite émet continûment un message de navigation à 50 bit/s contenant les informations nécessaires à la résolution du problème de navigation. Ce message de navigation est rendu robuste aux perturbations extérieures grâce à un code réalisant un étalement de spectre et la séquence résultante module des porteuses légèrement différentes pour chaque

satellite. Le signal ainsi réalisé est alors émis sur ces deux porteuses en bande L , dénotées L_1 et L_2 :

$$s_{L_1}(t, k) = \sqrt{2A}d(t)c(t) \cos(2\pi(f_1 + \frac{9k}{16})t + \theta_1) + \sqrt{2A}d(t)p(t) \sin(2\pi(f_1 + \frac{9k}{16})t + \theta_1), \quad (2.15)$$

$$s_{L_2}(t, k) = \sqrt{2A}d(t)p(t) \sin(2\pi(f_2 + \frac{7k}{16})t + \theta_2), \quad (2.16)$$

$$\text{où } \begin{cases} A & \text{est la puissance moyenne du signal émis,} \\ d(t) & \text{est le message de navigation,} \\ c(t) & \text{est le signal correspondant au code } C/A, \\ p(t) & \text{est le signal correspondant au code } P, \\ f_1 & \text{est fréquence de la porteuse } L_1 (= 1602\text{Mhz}), \\ f_2 & \text{est fréquence de la porteuse } L_2 (= 1246\text{Mhz}), \\ k & \text{est le numéro du satellite.} \end{cases}$$

Il faut noter que chaque satellite utilise le même code d'étalement de spectre. Ce sont les différentes porteuses utilisées qui vont assurer la faible intercorrélacion entre deux différents signaux. Cette technique de transmission permet à tous les satellites GLONASS de pouvoir émettre simultanément sans interférences et porte le nom de FDMA (frequency division multiple acces).

2.3.4 Systèmes en cours de développement

De nouveaux systèmes de positionnement par satellite sont aujourd'hui à l'étude pour s'affranchir de la dépendance au GPS ou au GLONASS, tous deux sous contrôle militaire.

Galileo C'est le futur GNSS européen qui devrait être pleinement opérationnel en 2013 (en phase de test depuis 2004) et qui sera sous contrôle strictement civil, contrairement aux autres systèmes existants.

Le système Galileo sera composé de 4 segments : le segment spatial (composé de trente satellites placés sur trois orbites circulaires de rayon orbital moyen 29 600 kilomètres), le segment sol de contrôle (chargé du contrôle des satellites), le segment sol de mission (chargé de créer le message de navigation diffusé par les satellites et de gérer l'intégrité du système) et le segment de test des utilisateurs (chargé de valider les performances des récepteurs en environnement réel).

Les satellites Galileo émettront sur trois différentes bandes de fréquence et permettront donc de proposer cinq différents services : le service ouvert (gratuit et accessible à tous), le service commercial (payant mais accessible à tous), le service de sûreté de la vie (sécurisé, intègre et certifiable, réservé aux applications critiques sur le plan de la sécurité de la vie), le service public réglementé (système antibrouillage, réservé aux missions de service public dépendantes de la précision) et le service de recherche et secours (localisation des balises Cospas-Sarsat).

Compass Aussi connu sous le nom de Beidou-2, Compass est le projet chinois de GNSS indépendant. A l’instar de ces homologues, ce système utiliserait des satellites (30 ?) en orbite terrestre moyenne (MEO), mais inclurait également cinq satellites géostationnaires (GEO). La transmission du message de navigation se ferait, comme dans le cas du GPS ou de Galileo, à l’aide de CDMA.

2.4 Navigation hybride

Le terme de navigation hybride s’adapte à tous les types de navigation utilisant plusieurs systèmes de nature différente pouvant réaliser l’estimation de la position de manière indépendante (GNSS, INS, WSS (Wheel Speed Sensor), ...). Dans ces conditions, le système hybride combine les propriétés et fonctionnements des autres systèmes. Comme cela a été expliqué précédemment, GNSS et INS reposent sur des techniques de positionnement très différentes et complémentaires. En effet, l’erreur de localisation induite par le GNSS est bornée dans le temps mais soumise aux pertes ou dégradations du signal tandis que celle induite par l’INS est complètement indépendante de l’environnement mais dérive au cours du temps. L’hybridation GNSS/INS va alors permettre de compenser les défauts des deux systèmes indépendants et sera donc particulièrement bénéfique. Dans le cadre de cette thèse, seul ce couplage GNSS/INS sera détaillé.

2.4.1 Architectures d’hybridation

L’architecture d’hybridation va permettre de fixer les entrées du filtre réalisant cette hybridation. Il existe classiquement trois types d’architecture (lâche, serrée et très serrée) avec chacune ses avantages et ses inconvénients qui seront détaillés par la suite. Il est intéressant de noter qu’il faut également déterminer les sorties du filtre d’hybridation. L’approche la plus intuitive consisterait en l’estimation des paramètres cinématiques du mobile (position, vitesse, attitude) à l’aide des deux systèmes GNSS et INS. Cette approche comporte un grand défaut : en cas de forte dynamique du mobile, les sorties du filtre devraient être calculées à haute fréquence, ce qui impose un temps de réponse du filtre d’hybridation relativement faible. Par conséquent, il est bien plus souvent utilisé l’erreur de navigation inertielle (en position, vitesse et attitude) comme sortie du filtre, ce qui réduit considérablement les dynamiques mises en oeuvre. Les différentes architectures d’hybridation présentées par la suite le sont dans ce cas.

Hybridation lâche Ce premier schéma d’intégration combine, dans son filtre d’hybridation, les estimées cinématiques (position, vitesse, attitude) obtenues par le GNSS seul et par l’INS seul (Fig. 2.5). C’est la solution la plus directe et la plus simple à implémenter car les données issues des deux différents systèmes sont de même nature. Cependant, les performances de ce type d’intégration sont bien souvent décevantes dans des environnements où les satellites GNSS ne sont pas visibles (directement ou non) [FB98, Sec. 7.3].

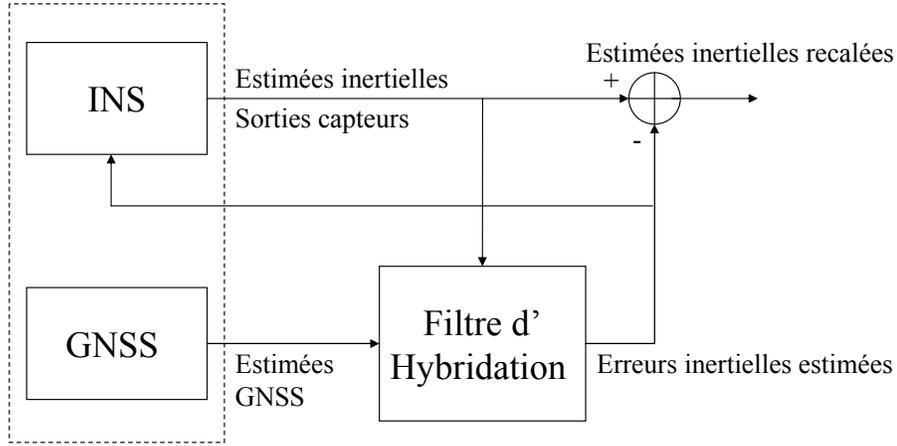


FIGURE 2.5 – Architecture d’hybridation lâche

Hybridation serrée Contrairement à l’architecture précédente, l’hybridation serrée met en oeuvre directement les mesures de pseudo-distances GNSS (Fig. 2.6). En combinant ces dernières avec les estimées cinématiques inertielles dans un filtre légèrement plus complexe, il est alors possible d’intégrer les deux systèmes. Cette solution est souvent préférable car plus précise, notamment lorsque moins de quatre satellites GNSS sont en vue, ce qui ne permet pas de recalculer l’INS dans le cas de l’hybridation lâche.

Hybridation très serrée Cette dernière structure a pour but d’intégrer des informations en provenance d’un autre système de positionnement (Fig. 2.7) pour améliorer l’estimation des mesures GNSS et ainsi rendre le système de navigation plus robuste. Le système résultant est complètement imbriqué, dans le sens où l’INS va permettre de mieux estimer les mesures GNSS, qui à leur tour vont recalculer l’INS à travers le filtre d’hybridation [Gir05, Sec. 6.4].

2.4.2 Modèle d’état

Dans la section 2.4.1, il a été beaucoup question d’hybridation et il convient maintenant de détailler les mathématiques qui lui sont associées. Quelle que soit l’architecture retenue, il faut tout d’abord noter que le problème peut se formaliser de la même manière. En effet, un certain nombre d’observations sont disponibles pour décrire un phénomène évoluant dans le temps, qui lui, n’est pas directement observable et ceci peut donc se formuler sous la forme d’un modèle d’état :

$$\begin{cases} \dot{X}(t) = f(t, X(t), U(t)), \\ Y(t) = h(t, X(t), U(t)), \end{cases} \quad (2.17)$$

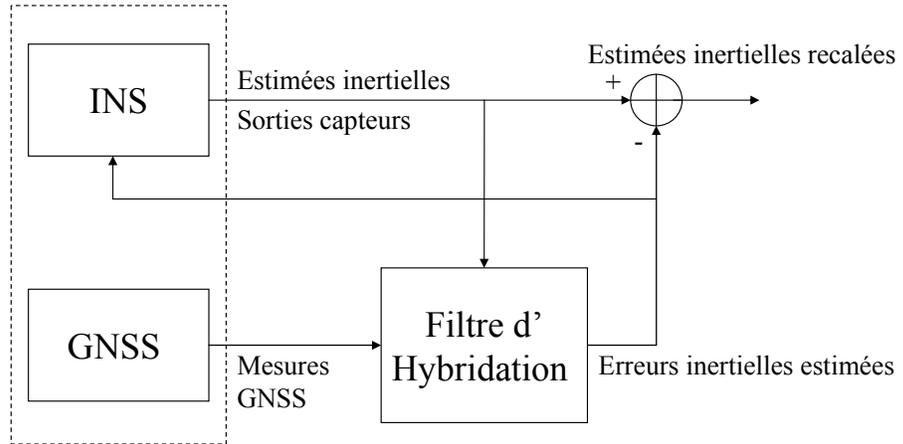


FIGURE 2.6 – Architecture d'hybridation serrée

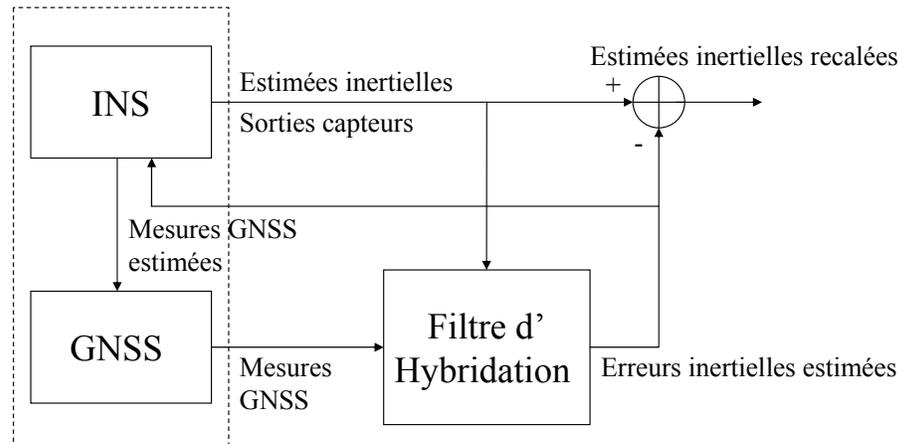


FIGURE 2.7 – Architecture d'hybridation très serrée

$$\text{où} \begin{cases} t & \text{est le temps,} \\ Y(t) & \text{est le vecteur de mesure,} \\ X(t) & \text{est le vecteur d'état,} \\ U(t) & \text{est le vecteur de commande,} \\ f \text{ et } h & \text{sont des fonctions non linéaires.} \end{cases}$$

Vecteur d'état Comme énoncé précédemment, la sortie du filtre d'hybridation, qui n'est rien d'autre que le vecteur d'état, contient les erreurs de navigation inertielle (en position, vitesse et attitude). Afin d'obtenir une meilleure précision, il est également nécessaire de suivre les perturbations liées aux systèmes (INS et GNSS). Pour l'INS, les modèles accélérométrique (2.6) et gyrométrique (2.7) font intervenir un grand nombre de sources d'erreur dont la principale composante est un biais évoluant dans le temps et qu'il faut donc estimer. Dans cette étude, les autres perturbations inertielles, comme par exemple le facteur d'échelle, ne sont pas modélisées dans le vecteur d'état mais auraient pu être également suivies. Pour le GNSS, il a été vu que le problème posé par le modèle de la pseudo-distance (2.11) ne pouvait être résolu qu'avec la connaissance du biais d'horloge récepteur qu'il faut donc également estimer. Le vecteur d'état peut alors finalement s'écrire :

$$x(t) = \begin{pmatrix} \delta_p(t) \\ \delta_v(t) \\ \delta_\rho(t) \\ b_a(t) \\ b_g(t) \\ b_{GNSS}(t) \\ \dot{b}_{GNSS}(t) \end{pmatrix}, \quad (2.18)$$

$$\text{où} \begin{cases} \delta_p(t) \in \mathbb{R}^3 & \text{est l'erreur inertielle en position,} \\ \delta_v(t) \in \mathbb{R}^3 & \text{est l'erreur inertielle en vitesse,} \\ \delta_\rho(t) \in \mathbb{R}^3 & \text{est l'erreur inertielle en attitude,} \\ b_a(t) \in \mathbb{R}^3 & \text{est le biais accélérométrique,} \\ b_g(t) \in \mathbb{R}^3 & \text{est le biais gyrométrique,} \\ b_{GNSS}(t) \in \mathbb{R} & \text{est le biais d'horloge du récepteur GNSS,} \\ \dot{b}_{GNSS}(t) \in \mathbb{R} & \text{est la dérive d'horloge du récepteur GNSS.} \end{cases}$$

Vecteur de mesure Dans la section 2.4.1, l'entrée du filtre d'hybridation, qui est équivalente au vecteur de mesure, a été définie de plusieurs manières :

- Dans le cas d'une hybridation lâche, il s'agit des estimées cinématiques à l'aide du GNSS seul.
- Pour une hybridation serrée, le vecteur de mesure contient directement les mesures de pseudo-distances.
- Lorsqu'il s'agit d'une architecture très serrée, les mesures de pseudo-distances ou les sorties des corrélateurs du récepteur GNSS peuvent être utilisées.

Équation d'état Il a été montré précédemment à travers les équations de la navigation inertielle (2.8), (2.9) et (2.10) qu'il était possible d'écrire le problème de navigation sous la forme

$$\frac{d}{dt} \begin{pmatrix} p(t) \\ v(t) \\ \rho(t) \end{pmatrix} = f(p(t), v(t), \rho(t)).$$

Les équations d'erreurs inertielles en position $\delta_p(t) = p(t) - \hat{p}(t)$, en vitesse $\delta_v(t) = v(t) - \hat{v}(t)$ et en attitude $\delta_\rho(t) = \rho(t) - \hat{\rho}(t)$ sont alors obtenues en différenciant cette équation au premier ordre

$$\frac{d}{dt} \begin{pmatrix} \delta_p(t) \\ \delta_v(t) \\ \delta_\rho(t) \end{pmatrix} = \frac{\partial f}{\partial p(t)} \delta_p(t) + \frac{\partial f}{\partial v(t)} \delta_v(t) + \frac{\partial f}{\partial \rho(t)} \delta_\rho(t).$$

- **L'équation d'erreur de position** se déduit simplement en développant l'équation de position inertielle (2.8), désignée ici par f_{pos} , autour des valeurs estimées par l'INS

$$\frac{d}{dt} \delta_p(t) = \frac{d}{dt} \begin{pmatrix} \delta_\lambda \\ \delta_\phi \\ \delta_h \end{pmatrix} = \frac{\partial f_{pos}}{\partial p(t)} \delta_p(t) + \frac{\partial f_{pos}}{\partial v(t)} \delta_v(t) + \frac{\partial f_{pos}}{\partial \rho(t)} \delta_\rho(t),$$

ce qui donne finalement

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} \delta_\lambda \\ \delta_\phi \\ \delta_h \end{pmatrix} &= \begin{pmatrix} 0 & 0 & -\frac{v_n}{(N+h)^2} \\ \frac{v_e \sin \hat{\lambda}}{N+h \cos(\hat{\lambda})^2} & 0 & -\frac{v_e}{(N+h)^2 \cos \hat{\lambda}} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_\lambda \\ \delta_\phi \\ \delta_h \end{pmatrix} \\ &+ \begin{pmatrix} \frac{1}{N+h} & 0 & 0 \\ 0 & \frac{1}{(N+h) \cos \hat{\lambda}} & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \delta_{v_n} \\ \delta_{v_e} \\ \delta_{v_d} \end{pmatrix}. \end{aligned} \quad (2.19)$$

- **L'équation d'erreur de vitesse**, comme précédemment, est obtenue à partir de l'équation de vitesse inertielle f_{vit} (2.9). Il faut cependant prendre en compte que le vecteur d'état choisi contient les biais accélérométriques et ceux-ci vont intervenir dans l'équation d'erreur de vitesse à travers les mesures accélérométriques

$$\begin{aligned} \frac{d}{dt} \delta_v(t) &= \frac{d}{dt} \begin{pmatrix} \delta_{v_n} \\ \delta_{v_e} \\ \delta_{v_d} \end{pmatrix} = \frac{\partial f_{vit}}{\partial p(t)} \delta_p(t) + \frac{\partial f_{vit}}{\partial v(t)} \delta_v(t) + \frac{\partial f_{vit}}{\partial \rho(t)} \delta_\rho(t) + \frac{\partial f_{vit}}{\partial b_a(t)} b_a(t) \\ &= \left(\frac{\partial g(p)}{\partial p} \right)^n - \left(\frac{\partial \Omega_{\mathbf{en}}^n}{\partial p} + 2 \frac{\partial \Omega_{\mathbf{ie}}^n}{\partial p} \right) \wedge \hat{v}_{\mathbf{e}}^n(t) \delta_p(t) + \hat{R}_{m2n}(t) b_a(t) \\ &\quad - \left(\frac{\partial \Omega_{\mathbf{en}}^n}{\partial v} \wedge \hat{v}_{\mathbf{e}}^n(t) + (\hat{\Omega}_{\mathbf{en}}^n + 2 \hat{\Omega}_{\mathbf{ie}}^n) \wedge \frac{\partial v_{\mathbf{e}}^n}{\partial v} \right) \delta_v(t) + \frac{\partial R_{m2n}(t)}{\partial \rho} f_a(t) \delta_\rho(t). \end{aligned}$$

Sous une forme légèrement simplifiée, le résultat est alors

$$\begin{aligned} \frac{d}{dt} \delta_v(t) &= \delta_{R_{m2n}}(t) f_a(t) + \hat{R}_{m2n}(t) b_a(t) + \delta_{g^n}(t) \\ &\quad - (\Omega_{\mathbf{en}}^n + 2 \Omega_{\mathbf{ie}}^n) \wedge \delta_v(t) - (\delta \Omega_{\mathbf{en}}^n + 2 \delta \Omega_{\mathbf{ie}}^n) \wedge v_{\mathbf{e}}^n(t), \end{aligned} \quad (2.20)$$

$$\text{où} \begin{cases} \delta_{R_{m2n}}(t) = \frac{\partial R_{m2n}(t)}{\partial \rho} \delta_\rho(t), \\ \delta_{g^n}(t) = \frac{\partial g(p)^n}{\partial p} \delta_p(t), \\ \delta_{\Omega_{\text{en}}^n} = \frac{\partial \Omega_{\text{en}}^n}{\partial p} \delta_p(t) + \frac{\partial \Omega_{\text{en}}^n}{\partial v} \delta_v(t), \\ \delta_{\Omega_{\text{ie}}^n} = \frac{\partial \Omega_{\text{ie}}^n}{\partial p} \delta_p(t). \end{cases}$$

- **L'équation d'erreur d'attitude** est calculée à son tour à partir de l'équation d'attitude inertielle (2.10), notée ici f_{att} . De même que pour l'équation d'erreur de vitesse, il faut prendre en compte que les biais gyrométriques vont participer à cette équation du fait de la présence des mesures gyrométriques

$$\frac{d}{dt} \delta_\rho(t) = \frac{d}{dt} \begin{pmatrix} \delta_\phi \\ \delta_\theta \\ \delta_\psi \end{pmatrix} = \frac{\partial f_{att}}{\partial p(t)} \delta_p(t) + \frac{\partial f_{att}}{\partial v(t)} \delta_v(t) + \frac{\partial f_{att}}{\partial \rho(t)} \delta_\rho(t) + \frac{\partial f_{att}}{\partial b_g(t)} b_g(t)$$

ce qui permet d'obtenir

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} \delta_\phi \\ \delta_\theta \\ \delta_\psi \end{pmatrix} &= \begin{pmatrix} \omega_{ie} \sin(\hat{\lambda}) & 0 & \frac{\dot{v}_e}{(N+\hat{h})^2} \\ 0 & 0 & \frac{-\dot{v}_n}{(N+\hat{h})^2} \\ \omega_{ie} \cos(\hat{\lambda}) + \frac{v_e}{(N+\hat{h}) \cos(\hat{\lambda})^2} & 0 & -\frac{\dot{v}_e \tan \hat{\lambda}}{(N+\hat{h})^2} \end{pmatrix} \begin{pmatrix} \delta_\lambda \\ \delta_\phi \\ \delta_h \end{pmatrix} \\ &+ \begin{pmatrix} 0 & \frac{-1}{N+\hat{h}} & 0 \\ \frac{1}{N+\hat{h}} & 0 & 0 \\ 0 & \frac{\tan \hat{\lambda}}{N+\hat{h}} & 0 \end{pmatrix} \begin{pmatrix} \delta_{v_n} \\ \delta_{v_e} \\ \delta_{v_d} \end{pmatrix} - \Omega_{in}^n \delta_\rho(t) + R_{m2n} b_g(t). \end{aligned} \quad (2.21)$$

- **L'équation d'erreur d'horloge du récepteur GNSS** est classiquement représentée à l'aide d'un modèle d'ordre 2 permettant de décrire le comportement du biais et de la dérive du temps mesurés par l'horloge du récepteur GNSS

$$\frac{d}{dt} \begin{pmatrix} b_{GNSS}(t) \\ \dot{b}_{GNSS}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} b_{GNSS}(t) \\ \dot{b}_{GNSS}(t) \end{pmatrix} + \begin{pmatrix} w_b(t) \\ w_{\dot{b}}(t) \end{pmatrix} \quad (2.22)$$

$$\text{où} \begin{cases} b_{GNSS}(t) & \text{est le biais d'horloge du récepteur GNSS,} \\ \dot{b}_{GNSS}(t) & \text{est la dérive d'horloge du récepteur GNSS,} \\ w_b(t) & \sim \mathcal{N}(0, \sigma_b^2), \\ w_{\dot{b}}(t) & \sim \mathcal{N}(0, \sigma_{\dot{b}}^2). \end{cases}$$

- **Les équations du biais accélérométrique et gyrométrique** sont habituellement modélisées à l'aide de processus de Markov d'ordre 1, pouvant s'écrire dans le cas de l'accéléromètre

$$\frac{d}{dt} b_a(t) = -\frac{1}{\tau_a} b_a(t) + w_a(t), \quad (2.23)$$

et dans le cas du gyromètre

$$\frac{d}{dt} b_g(t) = -\frac{1}{\tau_g} b_g(t) + w_g(t), \quad (2.24)$$

$$\text{où} \begin{cases} b_a(t) & \text{est le biais accélérométrique,} \\ b_g(t) & \text{est le biais gyrométrique,} \\ \tau_a & \text{est le temps de corrélation accélérométrique,} \\ \tau_g & \text{est le temps de corrélation gyrométrique,} \\ w_a(t) & \sim \mathcal{N}(0, \sigma_{b_a}^2), \\ w_g(t) & \sim \mathcal{N}(0, \sigma_{b_g}^2). \end{cases}$$

L'équation d'état complète est alors obtenue en regroupant les différentes équations définies précédemment (2.19, 2.20, 2.21, 2.22, 2.23 et 2.24) en une seule écriture.

Équation de mesure Selon le type d'architecture choisie, le vecteur de mesure et par conséquent l'équation de mesure vont avoir différentes écritures possibles.

- Dans le cas de **P'hybridation lâche**, la mesure est la position estimée seulement par le GNSS. L'équation de mesure alors obtenue peut se mettre sous la forme

$$Y(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} (N + \hat{h}(t) + \delta_h) \cos(\hat{\lambda}(t) + \delta_\lambda) \cos(\hat{\phi}(t) + \delta_\phi) \\ (N + \hat{h}(t) + \delta_h) \cos(\hat{\lambda}(t) + \delta_\lambda) \sin(\hat{\phi}(t) + \delta_\phi) \\ (N(1 - e^2) + \hat{h}(t) + \delta_h) \sin(\hat{\lambda}(t) + \delta_\lambda) \end{pmatrix}. \quad (2.25)$$

- Dans le cas **des hybridations serrée et très serrée**, ce sont directement les pseudo-distances GNSS qui servent de mesures. L'équation de mesure résultante est donc légèrement plus compliquée car elle doit prendre en compte le passage de ces pseudo-distances GNSS à l'estimation de la position

$$h_1(x(t), y(t), z(t)) = \begin{pmatrix} \sqrt{(x(t) - x_{S_1})^2 + (y(t) - y_{S_1})^2 + (z(t) - z_{S_1})^2} \\ \vdots \\ \sqrt{(x(t) - x_{S_{N_S}})^2 + (y(t) - y_{S_{N_S}})^2 + (z(t) - z_{S_{N_S}})^2} \end{pmatrix}. \quad (2.26)$$

$$\text{où} \begin{cases} (x(t), y(t), z(t)) & \text{sont les coordonnées cartésiennes du mobile dans } \mathbf{e}, \\ (x_{S_1}, y_{S_1}, z_{S_1}) \\ \vdots \\ (x_{S_{N_S}}, y_{S_{N_S}}, z_{S_{N_S}}) \end{cases} \text{ sont les coordonnées cartésiennes des } N_S \text{ satellites en vue dans } \mathbf{e}.$$

L'équation de mesure est alors la composition des équations (2.25) et (2.26), notées respectivement ici $h_2(\delta_\lambda, \delta_\phi, \delta_h)$ et $h_1(x(t), y(t), z(t))$ à laquelle il faut ajouter le biais dû à l'horloge du récepteur GNSS $b_{GNSS}(t)$ ainsi qu'un bruit de mesure $w_{GNSS}(t)$ détaillé en (2.11)

$$Y(t) = h_1 \circ h_2(\delta_\lambda, \delta_\phi, \delta_h) + b_{GNSS}(t) + w_{GNSS}(t). \quad (2.27)$$

2.4.3 Méthode de résolution

La représentation d'état ainsi obtenue peut être résolue par différentes méthodes de filtrage. Le principe est d'utiliser le maximum d'informations disponibles pour

estimer l'état du système. À l'instant t , toute l'information est contenue dans la distribution conditionnelle de l'état connaissant toutes les mesures collectées $P[X(t)|Y(1:t)]$. Il faut donc estimer au cours du temps cette loi pour calculer la valeur de l'état à l'instant t à l'aide d'un estimateur (MMSE, MAP, ...). Ceci est réalisé en deux étapes :

- **La prédiction** peut s'écrire grâce à la relation de Chapman-Kolmogorov

$$p[X(t)|Y(1:t-1)] = \int_{X(t)} p[X(t)|X(t-1)]p[X(t-1)|Y(1:t-1)]dX. \quad (2.28)$$

- **La correction**, grâce à la loi de Bayes, peut être estimé par

$$p[X(t)|Y(1:t)] = \frac{p[Y(t)|X(t)]p[X(t)|Y(1:t-1)]}{p[Y(t)|Y(1:t-1)]}. \quad (2.29)$$

Le calcul analytique de cette solution peut s'avérer compliqué dans le cas général mais il existe cependant un cas simple pour lequel le résultat optimal peut être calculé et qui sera l'objet du prochain paragraphe. Dans les autres cas, l'absence de solution analytique au problème d'estimation implique alors de recourir à des méthodes sous-optimales.

Filtre de Kalman (FK) Dans le cas de systèmes linéaires gaussiens, la représentation d'état discrète peut s'écrire simplement (2.30) où $v_k \sim \mathcal{N}(0, Q_k)$ et $w_k \sim \mathcal{N}(0, R_k)$ sont respectivement les bruits d'état et de mesure.

$$\begin{cases} X_k = F_k X_{k-1} + v_k, \\ Y_k = H_k X_k + w_k, \end{cases} \quad (2.30)$$

Dans ces conditions, il a été démontré que $p(X_k|Y_{1:k-1}) \sim \mathcal{N}(X_{k|k-1}, P_{k|k-1})$, $p(Y_k|Y_{1:k-1}) \sim \mathcal{N}(Y_{k|k-1}, S_k)$ et $p(X_k|Y_{1:k}) \sim \mathcal{N}(X_{k|k}, P_{k|k})$. Ces lois sont donc complètement connues à partir de la connaissance de leurs deux premiers moments, qui peuvent être calculés analytiquement au cours du temps. L'algorithme qui en découle est appelé Filtre de Kalman [Kal60], [KB61] et est décrit par

$$\left\{ \begin{array}{l} X_{k|k-1} = F_k X_{k-1|k-1}, \\ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \\ Y_{k|k-1} = H_k X_{k|k-1}, \\ S_k = H_k P_{k|k-1} H_k^T + R_k, \end{array} \right\} \quad \text{Prédiction}$$

$$\left\{ \begin{array}{l} K_k = P_{k|k-1} H_k^T S_k^{-1}, \\ X_{k|k} = X_{k|k-1} + K_k (Y_k - Y_{k|k-1}), \\ P_{k|k} = (I - K_k H_k) P_{k|k-1}. \end{array} \right\} \quad \text{Correction}$$

Filtre de Kalman Étendu (EKF) Dans le cas de systèmes non-linéaires, la première idée pour résoudre le problème est de linéariser le modèle. L'algorithme du filtre de Kalman étendu propose alors de remplacer les équations

non-linéaires par leur développement de Taylor au premier ordre autour de l'estimation la plus récente de l'état. De même que dans le filtre de Kalman, l'hypothèse que les bruits soient gaussiens permet alors la résolution du problème. On a classiquement ([Jaz70] et [AM05])

$$\begin{cases} X_k = f_k(X_{k-1}) + v_k \approx F_k(X_{k-1} - X_{k-1|k-1}) + f_k(X_{k-1|k-1}) + v_k, \\ Y_k = h_k(X_k) + w_k \approx H_k(X_k - X_{k|k-1}) + h_k(X_{k|k-1}) + w_k, \end{cases} \quad (2.31)$$

$$\text{où } \begin{cases} F_k = \frac{\partial f_k}{\partial X_{k-1}} \Big|_{X_{k-1|k-1}}, \\ H_k = \frac{\partial h_k}{\partial X_k} \Big|_{X_{k|k-1}}. \end{cases}$$

Ce nouvel algorithme peut alors s'écrire sous une forme proche de celle du filtre de Kalman

$$\left\{ \begin{array}{l} X_{k|k-1} = f_k(X_{k-1|k-1}), \\ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \\ Y_{k|k-1} = h_k(X_{k|k-1}), \\ S_{k|k} = H_k P_{k|k-1} H_k^T + R_k, \\ K_k = P_{k|k-1} H_k^T S_{k|k}^{-1}, \\ X_{k|k} = X_{k|k-1} + K_k (Y_k - Y_{k|k-1}), \\ P_{k|k} = (I - K_k H_k) P_{k|k-1}. \end{array} \right. \quad \left. \begin{array}{l} \text{Prédiction} \\ \\ \\ \\ \text{Correction} \end{array} \right.$$

Filtre de Kalman “Unscented” (UKF) Contrairement à l'EKF, l'UKF ne fait pas d'approximation des fonctions non-linéaires mais utilise le véritable modèle d'état. Son but sera alors d'estimer directement la loi de distribution de l'état connaissant l'ensemble des observations. Ceci est réalisé à l'aide de la transformation sans parfum (appelée “Unscented transform” (UT) en anglais [JU96], [JU97]) qui permet de calculer les moments d'ordre 1 et 2 d'une variable aléatoire qui est passée dans une fonction non-linéaire.

En se plaçant dans le cas simple de $y = g(x)$ où \bar{x} et P_x sont le vecteur moyenne et la matrice de covariance du vecteur aléatoire x de dimension N_x , l'UT va tout d'abord déterminer $2N_x + 1$ échantillons pondérés $\{w_i, \chi_i\}$ décrivant complètement \bar{x} et P_x de la manière suivante :

$$\begin{cases} \chi_0 = \bar{x} & w_0 = \frac{\kappa}{N_x + \kappa} & i = 0 \\ \chi_i = \bar{x} + (\sqrt{(N_x + \kappa)P_x})_i & w_i = \frac{1}{2(N_x + \kappa)} & i = 1, \dots, N_x \\ \chi_i = \bar{x} - (\sqrt{(N_x + \kappa)P_x})_i & w_i = \frac{1}{2(N_x + \kappa)} & i = N_x + 1, \dots, 2N_x \end{cases}$$

où κ est un paramètre de pondération et $(\sqrt{(N_x + \kappa)P_x})_i$ est la i -ème colonne de la racine carré de la matrice de covariance pondérée. En propageant ces échantillons à travers la fonction non-linéaire, il est alors possible d'obtenir une estimation des moments d'ordre 1 et 2 qui permet de décrire la loi a posteriori de l'état à l'ordre 2.

$$\begin{cases} \gamma_i = g(\chi_i) & i = 0, \dots, 2N_x \\ \bar{y} \approx \sum_{i=0}^{2N_x} w_i \gamma_i \\ P_y \approx \sum_{i=0}^{2N_x} w_i (\gamma_i - \bar{y})(\gamma_i - \bar{y})^T \\ P_{xy} \approx \sum_{i=0}^{2N_x} w_i (\chi_i - \bar{x})(\gamma_i - \bar{y})^T \end{cases}$$

Appliquée au cas de la navigation, l'UT va permettre de définir un nouveau système

$$\begin{cases} X_k^a = f_k(X_{k-1}^a), \\ Y_k = h_k(X_k), \end{cases} \quad \text{où } X_k^a = \begin{pmatrix} X_k \\ v_k \\ w_k \end{pmatrix}, \quad (2.32)$$

résolu par l'algorithme de l'UKF

$$\left. \begin{aligned} \chi_{k-1}^a &= \begin{bmatrix} X_{k-1|k-1} \\ X_{k-1|k-1} + \sqrt{(N_{X^a} + \kappa)P_{k-1|k-1}^a} \\ X_{k-1|k-1} - \sqrt{(N_{X^a} + \kappa)P_{k-1|k-1}^a} \end{bmatrix}, \\ \chi_{k|k-1}^X &= f_k(\chi_{k-1|k-1}), \\ X_{k|k-1} &= \sum_{i=0}^{2N_{X^a}} w_i \chi_{i,k|k-1}^X, \\ P_{k|k-1} &= \sum_{i=0}^{2N_{X^a}} w_i (\chi_{i,k|k-1}^X - X_{k|k-1})(\chi_{i,k|k-1}^X - X_{k|k-1})^T, \\ \gamma_{k|k-1} &= h_k(\chi_{k|k-1}), \\ Y_{k|k-1} &= \sum_{i=0}^{2N_{X^a}} w_i \chi_{i,k|k-1}^\gamma, \\ S_k &= \sum_{i=0}^{2N_{X^a}} w_i (\gamma_{i,k|k-1} - Y_{k|k-1})(\gamma_{i,k|k-1} - Y_{k|k-1})^T, \\ T_k &= \sum_{i=0}^{2N_{X^a}} w_i (\chi_{i,k|k-1}^X - X_{k|k-1})(\gamma_{i,k|k-1} - Y_{k|k-1})^T, \\ K_k &= T_k H_k^T S_k^{-1}, \\ X_{k|k} &= X_{k|k-1} + K_k (Y_k - Y_{k|k-1}), \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T. \end{aligned} \right\} \begin{array}{l} \text{Prédiction} \\ \text{Correction} \end{array}$$

Filtre particulaire (FP) Là aussi, le but est d'estimer directement la loi a posteriori de l'état connaissant l'ensemble des observations. Cette distribution est approchée par une loi discrète dont les valeurs possibles sont appelées particules et dont les probabilités sont appelées poids. Classiquement, un filtre particulaire [Dou98], [DGA00], [DDG01], procède en deux étapes : l'échantillonnage pondéré qui propage récursivement les particules et le rééchantillonnage qui sélectionne les particules les plus pertinentes. La démarche est alors la suivante :

- Initialisation des N particules par un tirage aléatoire sur la loi *a priori* de l'état,
- Prédiction de chaque particule grâce à la loi associée à l'équation d'état,
- Évaluation et normalisation des poids d'importance associés à chaque particule,
- Sélection/rejet des particules de poids d'importance élevé/faible,
- Estimation de l'état à l'aide de la loi de distribution ainsi estimée.

2.5 Conclusion

Ce chapitre a permis de présenter différentes techniques de navigation ainsi que les méthodes de résolution qu'il faut mettre en oeuvre pour obtenir une information de positionnement. Une description plus détaillée a été apportée pour les techniques utilisées dans le cadre des travaux menés durant cette thèse. Ainsi, la navigation inertielle et la radionavigation par satellite ont été plus particulièrement expliquées, ce qui a alors permis d'introduire la notion de navigation hybride dans le cas spécifique GNSS/INS. Présentée dans un cadre général,

cette navigation hybride pourra alors être utilisée dans le cas qui a été plus particulièrement étudié durant cette thèse, la navigation hybride GNSS/MEMS et qui sera détaillée dans les prochains chapitres. Pour la suite de ce manuscrit, l'hybridation sera toujours réalisée dans le cadre d'une architecture serrée et résolue à l'aide d'un filtre de Kalman étendu.

Travaux effectués

Chapitre 3

Évaluation des capteurs MEMS

Sommaire

3.1	Introduction	35
3.2	Caractérisation des capteurs MEMS	36
3.2.1	Cas de l'accéléromètre	36
3.2.2	Cas du gyromètre	38
3.3	Performances en navigation inertielle	40
3.3.1	Accéléromètres seuls	40
3.3.2	Gyromètres seuls	40
3.3.3	Accéléromètres et gyromètres	42
3.4	Performances en navigation hybride	42
3.4.1	Perte partielle des signaux GNSS	42
3.4.2	Perte totale des signaux GNSS	45
3.5	Conclusion	46

3.1 Introduction

De nombreuses études ont été menées au sujet des méthodes présentées dans le chapitre précédent concernant une hybridation GNSS/INS ([SNES05], [WMM⁺06], [ESNSN06], [NNAES06], ...). Récemment, une nouvelle génération de capteurs inertiels appelés MEMS (pour l'anglais Micro Electro Mechanical Systems) est devenue disponible à un coût relativement faible et permet d'envisager l'utilisation de système de navigation hybride dans de nouvelles applications (véhicules terrestres, piétons, ...). L'inconvénient majeur de ces nouveaux capteurs réside dans la qualité des mesures qu'ils délivrent. En effet, les perturbations qui l'affectent sont bien plus importantes que celles observées pour des accéléromètres et des gyromètres traditionnels ([Sta06], [Pet06], [SO06],

[FWB05], ...). L'enjeu de ce chapitre est d'obtenir une évaluation des performances de MEMS existants (ou non) mettant en relief la précision accessible par leur utilisation. Ainsi, une analyse des modèles de bruit à l'aide de la variance d'Allan est ici proposée pour différents capteurs et est suivie d'une étude des performances en terme de navigation aussi bien inertielle qu'hybride à l'aide de simulations de Monte Carlo (100 itérations). Ces dernières sont réalisées dans des conditions plus ou moins favorables (cas nominal, perte de signal GNSS partielle ou totale).

3.2 Caractérisation des capteurs MEMS

Dans cette analyse, trois catégories de capteurs MEMS de qualités différentes ont été étudiées à l'aide de la variance d'Allan (Annexe B). La Classe 1 fait référence à des systèmes qui comprennent le capteur et l'électronique nécessaire à son utilisation sur un même circuit intégré. Ce type de capteurs, proposé par des fondeurs de silicium comme Analog Devices ou STMicroelectronics, permet d'obtenir des dispositifs de taille très réduite et de faible consommation. Les performances peuvent se voir augmentées pour d'autres capteurs, mais ceci s'obtient, en règle générale, au détriment de l'encombrement et du prix. C'est le cas des gyromètres de Melexis et des accéléromètres de Colibrys qui représentent un bon compromis coût/performances et qui seront référencés, dans cette étude, comme la Classe 2. Finalement, une dernière catégorie de capteurs est proposée qui, considérant les améliorations apportées à la technologie MEMS ces dernières années, devrait faire son apparition dans le futur (Classe 3).

3.2.1 Cas de l'accéléromètre

Les trois classes de capteurs choisies dans la technologie des accéléromètres de type MEMS permettent une bonne représentation sur l'ensemble de la gamme (Fig. 3.1). Dans l'équation (2.6), il a été vu que la mesure accélérométrique S_a est entachée d'un certain nombre de perturbations. Une écriture légèrement simplifiée peut alors être donnée par

$$S_a = (I - \delta_{SFa} - \delta_{MAa})\Gamma - \delta_{Ba} - \delta_{NLa} - v_a,$$

qui, sachant que le biais δ_{Ba} est en fait la composition d'un biais δ_{Ba}^b , d'une marche aléatoire δ_{Ba}^{ma} et d'un processus de Gauss-Markov à l'ordre 1 δ_{Ba}^{gm} et que le capteur est soumis à une accélération nulle ($\Gamma = 0$), peut également s'écrire

$$S_a = -\delta_{Ba}^b - \delta_{Ba}^{ma} - \delta_{Ba}^{gm} - v_a. \quad (3.1)$$

$$\text{où } \begin{cases} v_a \sim \mathcal{N}(0, \sigma_a^2) \text{ de densité spectrale de puissance } Q_a, \\ \delta_{Ba}^{ma}(k) = \delta_{Ba}^{ma}(k-1) + v_{a,ma} \text{ avec } v_{a,ma} \sim \mathcal{N}(0, \sigma_{a,ma}^2), \\ \delta_{Ba}^{gm} \text{ est tel que son autocorrélation } R_{\delta_{Ba}^{gm}}(\tau) = \sigma_{a,gm}^2 e^{-\frac{|\tau|}{\tau_{a,gm}}} \text{ et de} \\ \text{densité spectrale de puissance } K_{a,gm}. \end{cases}$$

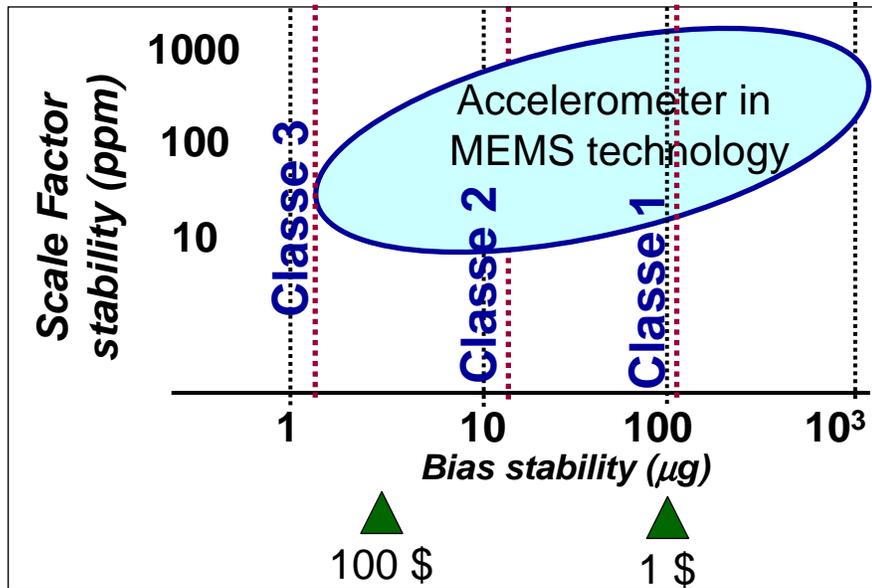


FIGURE 3.1 – Vue d'ensemble des accéléromètres de technologie MEMS

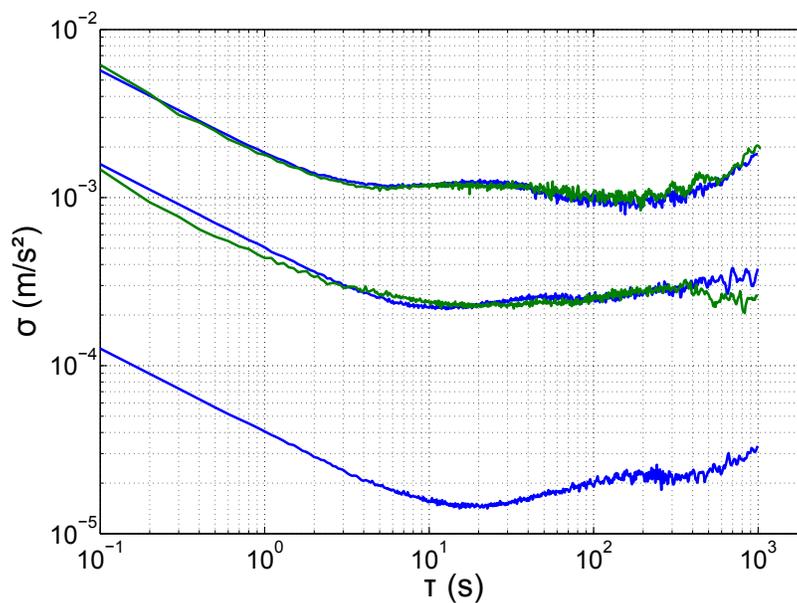


FIGURE 3.2 – Variance d'Allan des trois accéléromètres étudiés

Pour caractériser les paramètres de ce modèle, la variance d'Allan (Annexe B) est alors représentée pour les trois classes d'accéléromètres sur la figure 3.2 (Classe 1 en haut, Classe 2 au milieu et Classe 3 en bas). Les paramètres alors obtenus sont indiqués dans le tableau 3.1. Il est intéressant de noter que les courbes vertes sont obtenues à partir des données expérimentales (uniquement pour les Classes 1 et 2) tandis que les courbes bleues sont issues des données simulées à l'aide du modèle accélérométrique (Eq. 3.1) avec les valeurs estimées des paramètres.

Catégorie	$Q_a[m.s^{-2}/\sqrt{Hz}]$	$\sigma_{a,ma}[m.s^{-2}]$	$K_{a,gm}[m.s^{-2}/s/\sqrt{Hz}]$	$\tau_{a,gm}[s]$
Classe 1	1.8×10^{-3}	1×10^{-4}	1.85×10^{-3}	10
Classe 2	5×10^{-4}	2×10^{-5}	3.5×10^{-4}	30
Classe 3	4×10^{-5}	1×10^{-6}	3.5×10^{-5}	80

TABLE 3.1 – Paramètres issus de la modélisation pour les trois accéléromètres

3.2.2 Cas du gyromètre

De la même manière que dans le cas des accéléromètres, les trois gyromètres choisis pour cette étude permettent d'avoir une bonne représentativité des systèmes MEMS existants (Fig. 3.3). Le modèle gyrométrique (2.7) permettant de décrire l'ensemble des perturbations affectant la mesure gyrométrique S_g peut se simplifier et s'écrit alors

$$S_g = (I - \delta_{SFg} - \delta_{MAg})\Omega - \delta_{Bg} - \delta_{Kg} - v_g.$$

Le biais δ_{Bg} est, ici aussi, le résultat de la somme de trois différents termes : un biais δ_{Bg}^b , une marche aléatoire δ_{Bg}^{ma} et un processus de Gauss-Markov à l'ordre 1 δ_{Bg}^{gm} . En considérant un capteur au repos, la vitesse angulaire est proche de 0 ($\Omega \approx 0$), ce qui permet d'obtenir

$$S_g = -\delta_{Bg}^b - \delta_{Bg}^{ma} - \delta_{Bg}^{gm} - v_g. \quad (3.2)$$

$$\text{où } \begin{cases} v_g \sim \mathcal{N}(0, \sigma_g^2) \text{ de densité spectrale de puissance } Q_g, \\ \delta_{Bg}^{ma}(k) = \delta_{Bg}^{ma}(k-1) + v_{g,ma} \text{ avec } v_{g,ma} \sim \mathcal{N}(0, \sigma_{g,ma}^2), \\ \delta_{Bg}^{gm} \text{ est tel que son autocorrélation } R_{\delta_{Bg}^{gm}}(\tau) = \sigma_{g,gm}^2 e^{-\frac{|\tau|}{\tau_{g,gm}}} \text{ et de} \\ \text{densité spectrale de puissance } K_{g,gm}. \end{cases}$$

Pour caractériser les paramètres de ce modèle, la variance d'Allan est également représentée pour les trois classes de gyromètres sur la figure 3.4 (Classe 1 en haut, Classe 2 au milieu et Classe 3 en bas). Les paramètres alors obtenus sont indiqués dans le tableau 3.2. Il est intéressant de noter que, dans le cas du gyromètre de Classe 2, le processus de Gauss-Markov est négligeable et n'a donc pas pu être estimé. A l'instar de l'étude concernant les accéléromètres, les courbes vertes sont obtenues à partir des données expérimentales (uniquement

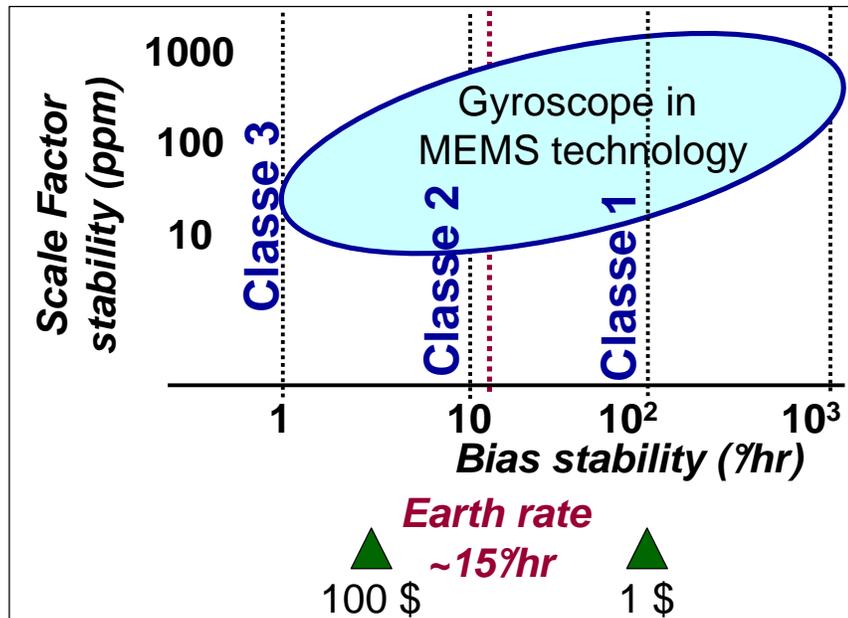


FIGURE 3.3 – Vue d'ensemble des gyromètres de technologie MEMS

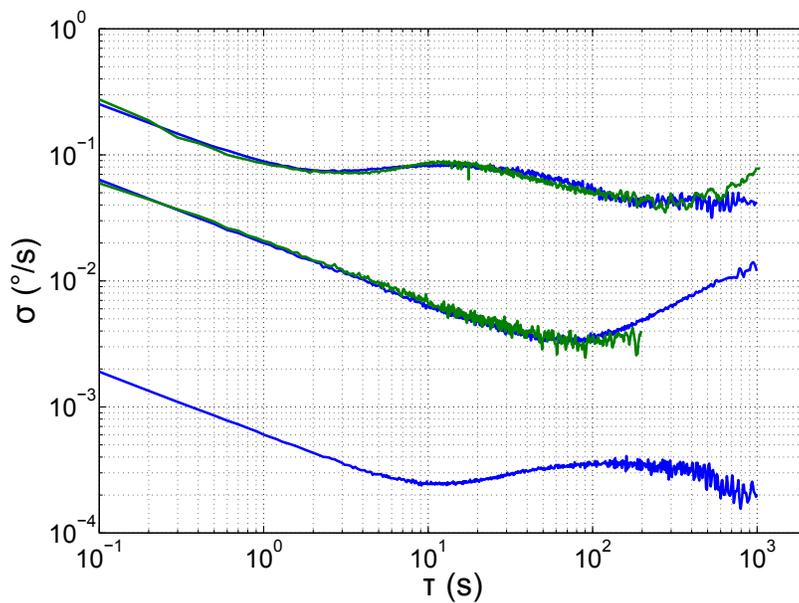


FIGURE 3.4 – Variance d'Allan des trois gyromètres étudiés

pour les Classes 1 et 2) tandis que les courbes bleues sont issues des données simulées à l'aide du modèle gyrométrique (Eq. 3.2) avec les valeurs estimées des paramètres.

3.3 Performances en navigation inertielle

Maintenant que les accéléromètres et les gyromètres ont été correctement caractérisés en terme de modélisation des capteurs, un second point d'intérêt est de connaître la précision qu'ils permettraient d'obtenir au cours d'une navigation inertielle seule. Le but, ici, n'est pas de réaliser réellement une navigation à l'aide des capteurs MEMS uniquement, mais plutôt de continuer la caractérisation des capteurs, cette fois-ci, en terme de navigation. Les accéléromètres et les gyromètres seront tout d'abord étudiés de manière séparée puis une analyse conjointe des capteurs de même classe permettra de simuler, à proprement parler, une navigation inertielle. Les résultats représentés, au cours de ces études, sont les écart-types calculés à l'aide de simulations de Monte Carlo (100 itérations).

3.3.1 Accéléromètres seuls

Afin de ne se soucier que de l'apport des perturbations des accéléromètres, les mesures gyrométriques seront, ici, considérées parfaites ($S_g = \Omega$). A partir de ces mesures, bruitées pour les accéléromètres et non bruitées pour les gyromètres, la plate-forme inertielle (Fig. 2.3) va produire une estimation de la position ne dépendant que de la modélisation accélérométrique. Sur la figure 3.5, la Classe 1 est représentée en bleu, la Classe 2 en vert et la Classe 3 en cyan. En dessous d'une durée de 40s, la précision obtenue est submétrique pour les trois différents capteurs. De même, si la durée n'excède pas 100s, on observe une dérive de moins de 10m pour la Classe 1 et d'approximativement 1m (resp. 0.1m) pour la Classe 2 (resp. Classe 3). Il est important de noter que l'erreur augmente quadratiquement avec le temps, ce qui est dû à la double intégration dans le temps des mesures accélérométriques.

3.3.2 Gyromètres seuls

Ce sont maintenant les mesures accélérométriques qui seront considérées parfaites ($S_a = \Gamma$) afin que seul l'apport des perturbations des gyromètres soit considéré. La plate-forme inertielle (Fig. 2.3) va alors à nouveau estimer

Catégorie	$Q_g [^\circ \cdot s^{-1} / \sqrt{Hz}]$	$\sigma_{g,ma} [^\circ \cdot s^{-1}]$	$K_{g,gm} [^\circ \cdot s^{-1} / s / \sqrt{Hz}]$	$\tau_{g,gm} [s]$
Classe 1	8×10^{-2}	3×10^{-3}	1.3×10^{-1}	7
Classe 2	2×10^{-2}	5×10^{-4}	—	—
Classe 3	6×10^{-4}	1×10^{-5}	6×10^{-4}	80

TABLE 3.2 – Paramètres issus de la modélisation pour les trois gyromètres

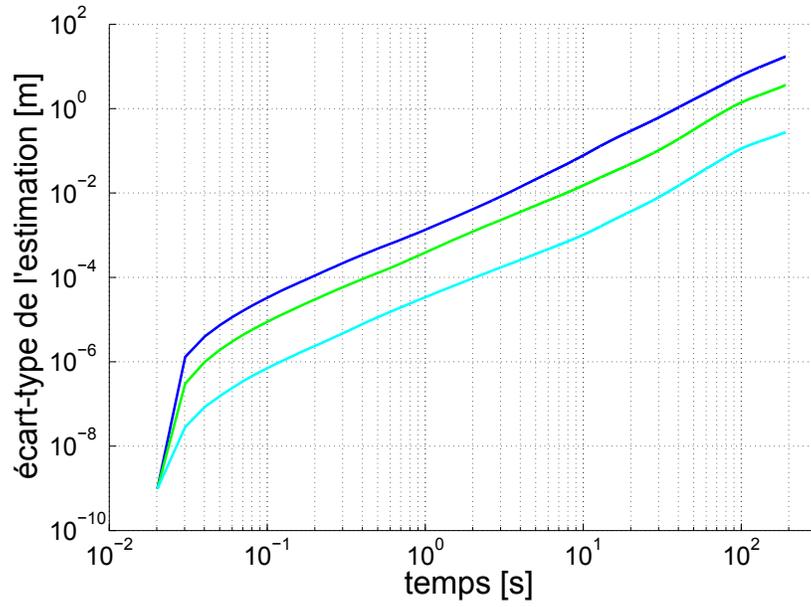


FIGURE 3.5 – Erreur de position liée aux perturbations accélérométriques [m]

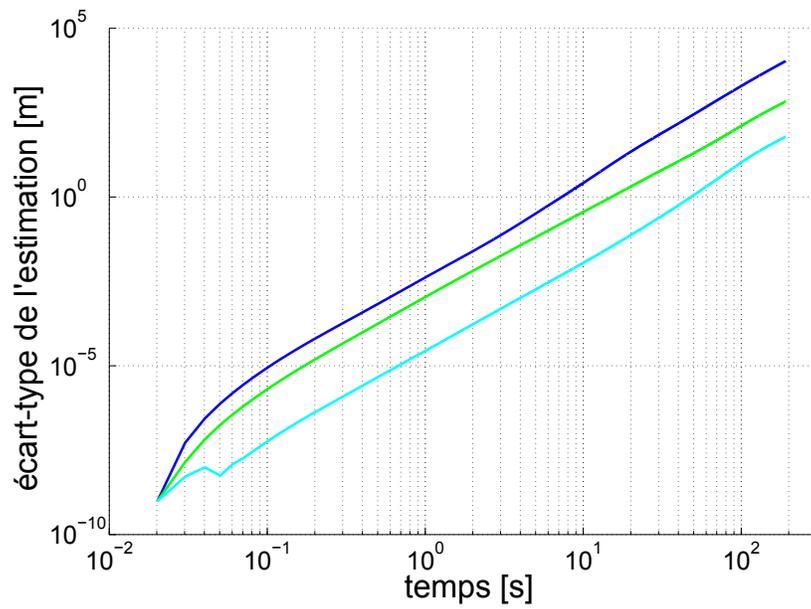


FIGURE 3.6 – Erreur de position liée aux perturbations gyrométriques [m]

la position ne dépendant que de la modélisation gyrométrique. De même que précédemment, sur la figure 3.6, la Classe 1 est représentée en bleu, la Classe 2 en vert et la Classe 3 en cyan. Ici, et suivant la même logique, comme les mesures gyrométriques sont intégrées trois fois, l'erreur augmente cubiquement avec le temps. La précision submétrique n'est donc plus conservée qu'en deçà d'une durée de 7s pour la Classe 1, 15s pour la Classe 2 et 50s pour la Classe 3. Au terme de 100s, les dérives observées sont bien plus grandes que lors de l'étude des accéléromètres (10km pour la Classe 1, 1km pour la classe 2 et 100m pour la Classe 3).

3.3.3 Accéléromètres et gyromètres

Les contributions respectives étant désormais connues, une navigation inertielle complète est réalisée. Les deux modèles MEMS, accélérométrique (Eq. 2.6) et gyrométrique (Eq. 2.7), vont venir, ici, impacter les performances du système de navigation. Les résultats sont présentés pour les trois catégories de capteurs (Fig. 3.7, en bleu pour la Classe 1, en vert pour la Classe 2 et en cyan pour la Classe 3). Les résultats sont conformes à nos attentes : si la durée est inférieure à 1s, nous retrouvons les performances des accéléromètres tandis que si elle est supérieure, ce sont les erreurs d'estimation liées aux gyromètres qui sont observées. Comme énoncé précédemment, le but de cette étude n'était pas de réaliser une navigation inertielle basée sur des MEMS, mais plutôt de caractériser ces capteurs en terme de navigation.

3.4 Performances en navigation hybride

L'enjeu principal de la navigation inertielle dans un système hybride est de pallier les défauts de service GNSS (perte de signal, brouillage, multi-trajet, ...). Ainsi, cette analyse des performances de la navigation hybride GNSS/MEMS a été menée dans de telles conditions. Plus précisément, le cas des pertes de signaux GNSS (partielle ou totale) a été traité ici. Durant le cas nominal, six satellites GNSS (Fig. 3.8) sont en vue du récepteur pour estimer la position et recalibrer le système de navigation inertielle. Le cas de la perte partielle ne met plus en œuvre que les trois satellites les moins éloignés du zénith du récepteur GNSS (Fig. 3.8, dans le cercle vert) alors que la perte totale ne fait intervenir aucun satellite. L'hybridation est, dans tous les cas, réalisée à travers une architecture serrée et résolue à l'aide d'un EKF, tous deux présentés dans le chapitre précédent. Les résultats représentés sont, ici encore, les écart-types calculés à l'aide de simulations de Monte Carlo (100 itérations).

3.4.1 Perte partielle des signaux GNSS

Le cas étudié ici, est celui d'une perte partielle des signaux GNSS intervenant à la moitié de la simulation et durant jusqu'à la fin. Sur la figure 3.9, l'erreur d'estimation de la position est représentée en bleu pour la Classe 1, en vert pour

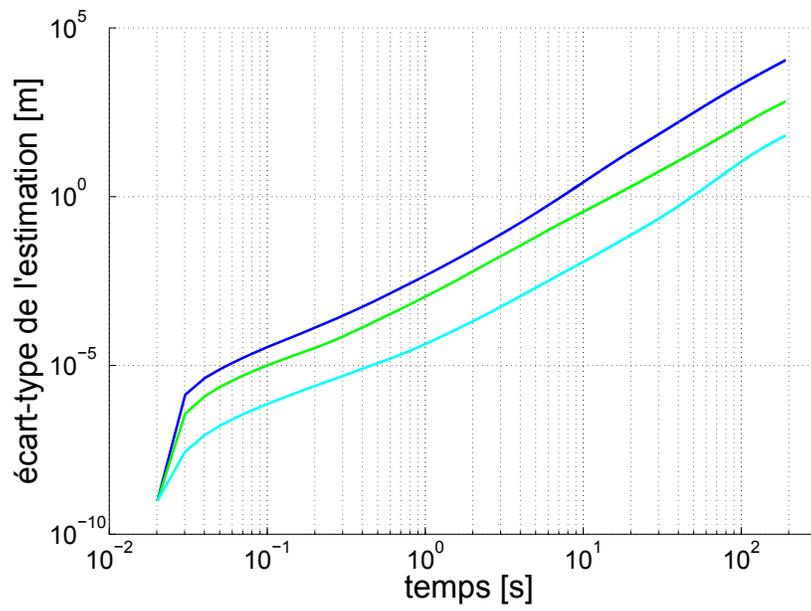


FIGURE 3.7 – Erreur de position liée aux perturbations accélérométriques et gyrométriques [m]

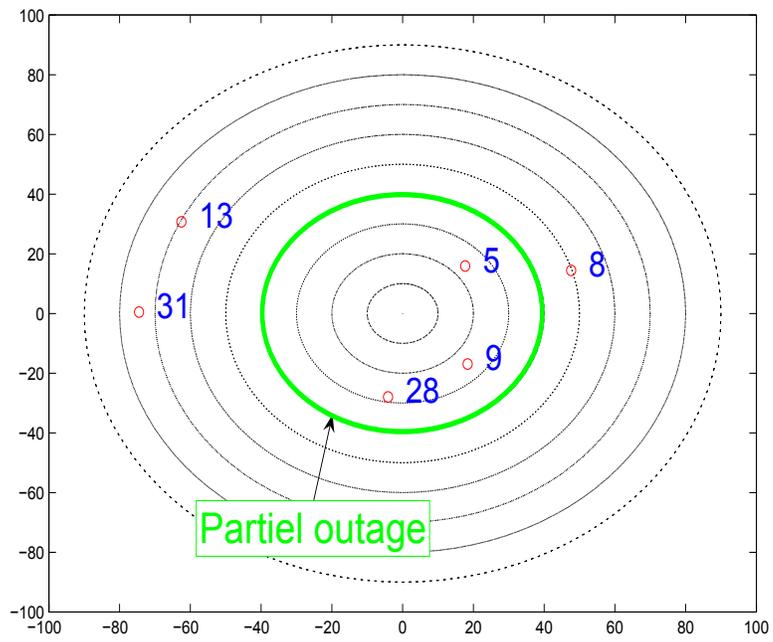


FIGURE 3.8 – Constellation des satellites en vue du récepteur GNSS

la Classe 2 et en cyan pour la Classe 3. Durant le cas nominal, les performances des système de navigation hybride sont de l'ordre de $5m$ pour la Classe 1, $4m$ pour la Classe 2 et $2.5m$ pour la Classe 3. Il est également possible d'observer que le temps de convergence vers ce cas nominal diffère en fonction de la catégorie de capteurs utilisée ($9.5s$ pour la Classe 1, $13s$ pour la Classe 2 et $28s$ pour la Classe 3). Durant la perte partielle des signaux GNSS, les résultats convergent vers de nouvelles valeurs d'erreurs. La Classe 1 (resp. Classe 2) converge en $14s$ (resp. $25s$) vers une précision d'approximativement $15m$ (resp. $11m$) tandis que la Classe 3 tend à converger vers $8m$ au bout de $80s$. Pour toutes les catégories de capteurs, un point important est que les performances durant une perte partielle de signaux GNSS (trois satellites disponibles) ne divergent pas avec le temps et restent relativement bonnes.

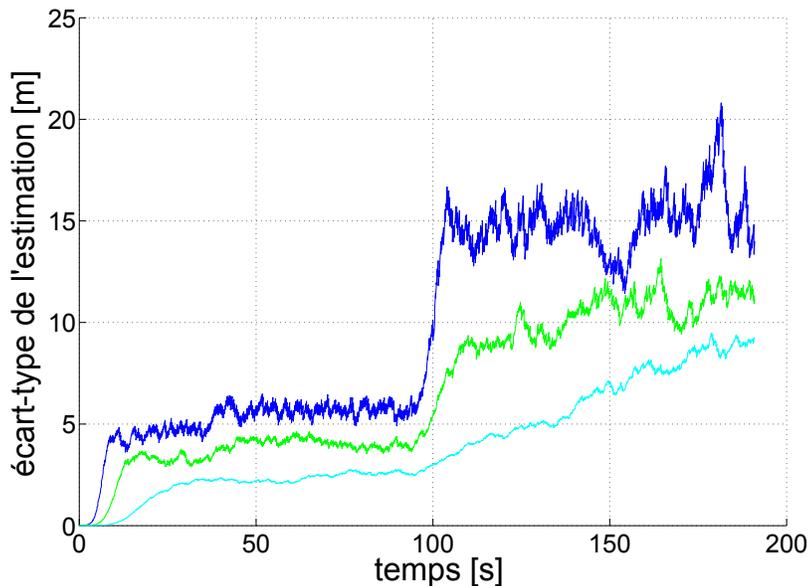


FIGURE 3.9 – Erreur de position liée à une perte partielle des signaux GNSS dans une navigation hybride [m]

3.4.2 Perte totale des signaux GNSS

La perte totale des signaux GNSS est le cas le plus défavorable de la navigation hybride. En effet, dans ces conditions, le système de navigation n'obtient des informations que de la centrale inertielle et ne sera aucunement recalé par le GNSS. Les résultats sont présentés pour les trois catégories de capteurs (Fig. 3.10, en bleu pour la Classe 1, en vert pour la Classe 2 et en cyan pour la

Classe 3). Dans cette étude, une perte totale de signaux GNSS de 30s de durée a été simulée. Au terme de ces 30s, l'erreur d'estimation de la position a divergé jusqu'à 450m pour la Classe 1, 50m pour la Classe 2 et 10m pour la Classe 3. C'est véritablement durant cette perte totale des signaux GNSS que la qualité des capteurs MEMS va impacter les performances en terme de navigation. Dans le cas nominal, il faut noter que les résultats observés précédemment (Fig. 3.9) peuvent l'être à nouveau ici (entre 5 et 6m pour la Classe 1, 4m pour la Classe 2 et entre 2.5 et 3m pour la Classe 3).

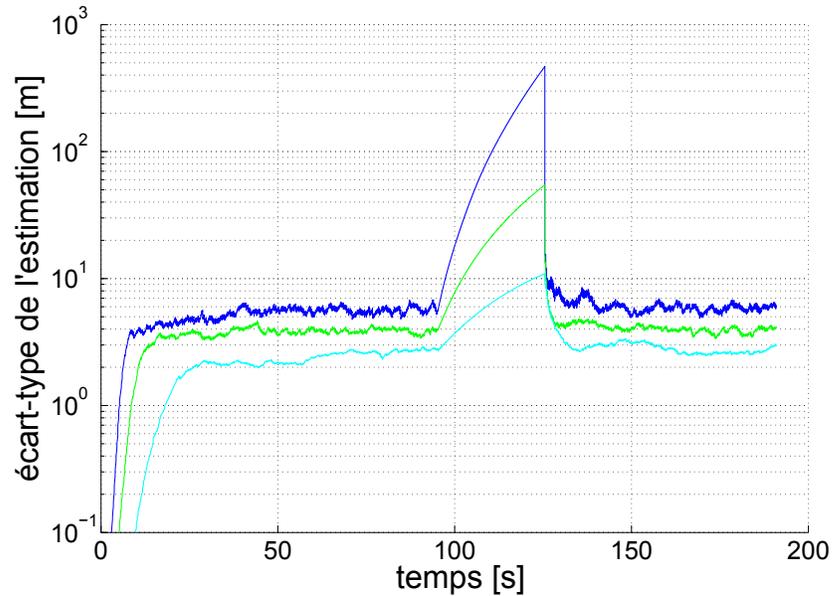


FIGURE 3.10 – Erreur de position liée à une perte totale des signaux GNSS dans une navigation hybride [m]

3.5 Conclusion

Ce chapitre a permis de caractériser les capteurs accélérométriques et gyrométriques de technologie MEMS, et ce, aussi bien en terme de mesures que de navigation. L'étude a montré que le système inertiel possède une grande influence sur les performances de la navigation dans le cas d'une perte totale des signaux GNSS [PDGC08]. Dans le cas nominal, voire dans celui d'une perte partielle des signaux GNSS, l'amélioration des mesures issues des capteurs MEMS n'a qu'un effet relatif, la précision étant en grande partie apportée par le GNSS. Les méthodes qui seront présentées dans les prochains chapitres auront donc pour vocation, d'une part d'améliorer les mesures des capteurs MEMS (Chap.

4) et d'autre part de définir des algorithmes de navigation spécifiques aux MEMS (Chap. 5). Pour la suite de ce manuscrit et pour des raisons de clarté, seuls les capteurs de Classe 1, les plus susceptibles d'applications grand public (encombrant moindre et coût modique), seront désormais étudiés.

Chapitre 4

Inversion du modèle des capteurs MEMS

Sommaire

4.1	Introduction	49
4.2	Calibrage	50
4.2.1	Cas de l'accéléromètre	50
4.2.2	Cas du gyromètre	51
4.3	Inversion du modèle	52
4.3.1	Cas de l'accéléromètre	53
4.3.2	Cas du gyromètre	53
4.4	Résultats de simulations	54
4.4.1	Calibrage	54
4.4.2	Inversion du modèle	61
4.4.3	Navigation	63
4.5	Conclusion	66

4.1 Introduction

Il a été vu précédemment que la relation entrée/sortie d'un accéléromètre (resp. gyromètre) pouvait être décrite par l'équation (2.6) (resp. (2.7)). Dans le cas des capteurs inertiels conventionnels, la plupart des termes intervenants dans ces relations sont négligeables et seuls les biais et les bruits sont nécessaires pour décrire correctement le phénomène physique. Ceci ne peut plus être considéré exact dans le cadre de la navigation hybride GNSS/MEMS et il sera alors nécessaire de mettre en oeuvre des techniques spécifiques adaptées aux comportements de ces capteurs. Une approche classique dans la littérature ([SBS99], [Nas03], [ESN04], [Chi04], [SAG⁺07] et [AH05]) est de considérer le problème sous forme d'un problème de débruitage qui peut se résoudre à l'aide d'une

décomposition en ondelettes. L'idée retenue ici a été d'utiliser la connaissance disponible des modèles de capteurs afin d'estimer ses entrées (i.e. accélérations et vitesses angulaires) en fonction de ses sorties (i.e. sorties accélérométriques et gyrométriques). Cette méthode se décompose en deux étapes, le calibrage et l'inversion proprement dite du modèle, qui seront détaillées par la suite dans ce chapitre.

4.2 Calibrage

Le calibrage est la première des deux étapes définies en vue d'inverser le modèle de capteur. Durant cette première phase, le but est d'estimer la non-linéarité et les paramètres intrinsèques des capteurs afin de mieux caractériser ces derniers. D'un point de vue général, la modélisation d'un capteur MEMS peut être formulée par :

$$S = \Phi(\Gamma, \Omega, \theta), \quad (4.1)$$

$$\text{où } \begin{cases} S & \text{est le vecteur des sorties du capteur MEMS } \in \mathbb{R}^3, \\ \Gamma & \text{est le vecteur des accélérations } \in \mathbb{R}^3, \\ \Omega & \text{est le vecteur des vitesses angulaires } \in \mathbb{R}^3, \\ \theta & \text{contient les paramètres intrinsèques du capteur MEMS,} \\ \Phi & \text{est la fonction non-linéaire caractérisant le capteur MEMS.} \end{cases}$$

Le calibrage revient alors à estimer θ et Φ à l'aide d'une relation $(\hat{\theta}, \hat{\Phi}) = f(\hat{\Gamma}, \hat{\Omega}, S)$ où $\hat{\Gamma}$ (resp. $\hat{\Omega}$) est le vecteur accélération (resp. vitesse angulaire) déterminé à partir des mesures GNSS.

4.2.1 Cas de l'accéléromètre

Cette équation très générale peut être décrite par la relation (2.6) lorsque l'intérêt est porté uniquement sur l'accéléromètre. Dans cette écriture, la fonction non-linéaire $\Phi(\cdot)$ est alors scindée en deux parties, l'une linéaire et l'autre non-linéaire. En exprimant la partie non-linéaire sous forme polynomiale, on obtient

$$S_a = \underbrace{-\delta_{Ba}}_{\text{ordre 0}} + \underbrace{(I - \delta_{SFa} - \delta_{MAa})}_{\text{ordre 1}} \Gamma - \underbrace{\delta_{NL2a}}_{\text{ordre 2}} \Gamma^2 - \dots - \underbrace{\delta_{NLja}}_{\text{ordre } j} \Gamma^j - \dots - \underbrace{v_a}_{\text{bruit}},$$

ce qui donne en 3 dimensions si les ordres supérieurs à 2 sont négligés

$$\begin{pmatrix} S_a^u \\ S_a^v \\ S_a^w \end{pmatrix} = \begin{pmatrix} 1 - \delta_{SFa}^u & \delta_{MAa}^{uv} & \delta_{MAa}^{vw} \\ \delta_{MAa}^{vw} & 1 - \delta_{SFa}^v & \delta_{MAa}^{vu} \\ \delta_{MAa}^{vu} & \delta_{MAa}^{wu} & 1 - \delta_{SFa}^w \end{pmatrix} \begin{pmatrix} \Gamma^u \\ \Gamma^v \\ \Gamma^w \end{pmatrix} - \begin{pmatrix} \delta_{Ba}^u \\ \delta_{Ba}^v \\ \delta_{Ba}^w \end{pmatrix} \\ - \begin{pmatrix} \delta_{NL2a}^u & 0 & 0 \\ 0 & \delta_{NL2a}^v & 0 \\ 0 & 0 & \delta_{NL2a}^w \end{pmatrix} \begin{pmatrix} (\Gamma^u)^2 \\ (\Gamma^v)^2 \\ (\Gamma^w)^2 \end{pmatrix} - \begin{pmatrix} v_a^u \\ v_a^v \\ v_a^w \end{pmatrix},$$

où u, v et w sont les trois axes de l'accéléromètre définis dans [FB98].

Il suffit alors d'écrire cette nouvelle relation sous une forme légèrement différente pour chacun des trois axes, ce qui donne pour l'axe u

$$\begin{pmatrix} S_{a,1}^u \\ \vdots \\ S_{a,N}^u \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w & (\Gamma_1^u)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w & (\Gamma_N^u)^2 \end{pmatrix} \begin{pmatrix} -\delta_{Ba}^u \\ 1 - \delta_{SFa}^u \\ \delta_{MAa}^{uw} \\ \delta_{MAa}^{uv} \\ -\delta_{NL2a}^u \end{pmatrix} - \begin{pmatrix} v_{a,1}^u \\ \vdots \\ v_{a,N}^u \end{pmatrix}, \quad (4.2)$$

pour l'axe v

$$\begin{pmatrix} S_{a,1}^v \\ \vdots \\ S_{a,N}^v \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w & (\Gamma_1^v)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w & (\Gamma_N^v)^2 \end{pmatrix} \begin{pmatrix} -\delta_{Ba}^v \\ \delta_{MAa}^{vw} \\ 1 - \delta_{SFa}^v \\ \delta_{MAa}^{vu} \\ -\delta_{NL2a}^v \end{pmatrix} - \begin{pmatrix} v_{a,1}^v \\ \vdots \\ v_{a,N}^v \end{pmatrix}, \quad (4.3)$$

et, finalement, pour l'axe w

$$\begin{pmatrix} S_{a,1}^w \\ \vdots \\ S_{a,N}^w \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w & (\Gamma_1^w)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w & (\Gamma_N^w)^2 \end{pmatrix} \begin{pmatrix} -\delta_{Ba}^w \\ \delta_{MAa}^{wv} \\ \delta_{MAa}^{wu} \\ 1 - \delta_{SFa}^w \\ -\delta_{NL2a}^w \end{pmatrix} - \begin{pmatrix} v_{a,1}^w \\ \vdots \\ v_{a,N}^w \end{pmatrix}. \quad (4.4)$$

Ces trois équations (4.2), (4.3) et (4.4) peuvent alors être résolues à l'aide de l'estimateur des moindres carrés (MC).

4.2.2 Cas du gyromètre

De même que pour l'accéléromètre, l'équation très générale (4.1) peut être précisée par (2.7) dans le cas du gyromètre. Contrairement aux capteurs accélérométriques, la modélisation gyrométrique ne prend pas en compte de phénomènes non-linéaires. Cependant, une certaine sensibilité aux accélérations et vibrations va venir s'ajouter à la sortie du capteur, ce qui donne en 3 dimensions

$$\begin{pmatrix} S_g^p \\ S_g^q \\ S_g^r \end{pmatrix} = \begin{pmatrix} 1 - \delta_{SFg}^p & \delta_{MAg}^{pr} & \delta_{MAg}^{pq} \\ \delta_{MAg}^{qr} & 1 - \delta_{SFg}^q & \delta_{MAg}^{qp} \\ \delta_{MAg}^{rq} & \delta_{MAg}^{rp} & 1 - \delta_{SFg}^r \end{pmatrix} \begin{pmatrix} \Omega^p \\ \Omega^q \\ \Omega^r \end{pmatrix} - \begin{pmatrix} \delta_{Bg}^p \\ \delta_{Bg}^q \\ \delta_{Bg}^r \end{pmatrix} \\ - \begin{pmatrix} \delta_{Kg}^{pu} & \delta_{Kg}^{pv} & \delta_{Kg}^{pw} \\ \delta_{Kg}^{qu} & \delta_{Kg}^{qv} & \delta_{Kg}^{qw} \\ \delta_{Kg}^{ru} & \delta_{Kg}^{rv} & \delta_{Kg}^{rw} \end{pmatrix} \begin{pmatrix} \Gamma^u \\ \Gamma^v \\ \Gamma^w \end{pmatrix} - \begin{pmatrix} v_g^p \\ v_g^q \\ v_g^r \end{pmatrix},$$

où p, q et r sont les trois axes du gyromètre définis dans [FB98].

Sous une forme plus propice à la résolution par la méthode des moindres

carrés, on obtient alors pour l'axe p

$$\begin{pmatrix} S_{g,1}^p \\ \vdots \\ S_{g,N}^p \end{pmatrix} = \begin{pmatrix} 1 & \Omega_1^p & \Omega_1^q & \Omega_1^r & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Omega_N^p & \Omega_N^q & \Omega_N^r & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w \end{pmatrix} \begin{pmatrix} -\delta_{Bg}^p \\ 1 - \delta_{SFg}^p \\ \delta_{MAg}^{pr} \\ \delta_{MAg}^{pq} \\ -\delta_{Kg}^{pu} \\ -\delta_{Kg}^{pv} \\ -\delta_{Kg}^{pw} \end{pmatrix} - \begin{pmatrix} v_{g,1}^p \\ \vdots \\ v_{g,N}^p \end{pmatrix}, \quad (4.5)$$

pour l'axe q

$$\begin{pmatrix} S_{g,1}^q \\ \vdots \\ S_{g,N}^q \end{pmatrix} = \begin{pmatrix} 1 & \Omega_1^p & \Omega_1^q & \Omega_1^r & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Omega_N^p & \Omega_N^q & \Omega_N^r & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w \end{pmatrix} \begin{pmatrix} -\delta_{Bg}^q \\ \delta_{MAg}^{qr} \\ 1 - \delta_{SFg}^q \\ \delta_{MAg}^{qp} \\ -\delta_{Kg}^{qu} \\ -\delta_{Kg}^{qv} \\ -\delta_{Kg}^{qw} \end{pmatrix} - \begin{pmatrix} v_{g,1}^q \\ \vdots \\ v_{g,N}^q \end{pmatrix}, \quad (4.6)$$

et, finalement, pour l'axe r

$$\begin{pmatrix} S_{g,1}^r \\ \vdots \\ S_{g,N}^r \end{pmatrix} = \begin{pmatrix} 1 & \Omega_1^p & \Omega_1^q & \Omega_1^r & \Gamma_1^u & \Gamma_1^v & \Gamma_1^w \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Omega_N^p & \Omega_N^q & \Omega_N^r & \Gamma_N^u & \Gamma_N^v & \Gamma_N^w \end{pmatrix} \begin{pmatrix} -\delta_{Bg}^r \\ \delta_{MAg}^{rq} \\ \delta_{MAg}^{rp} \\ 1 - \delta_{SFg}^r \\ -\delta_{Kg}^{ru} \\ -\delta_{Kg}^{rv} \\ -\delta_{Kg}^{rw} \end{pmatrix} - \begin{pmatrix} v_{g,1}^r \\ \vdots \\ v_{g,N}^r \end{pmatrix}, \quad (4.7)$$

4.3 Inversion du modèle

Possédant désormais une estimation $\hat{\theta}$ et $\hat{\Phi}$ des paramètres et non-linéarités du modèle accélérométrique (resp. gyrométrique), il reste l'inversion proprement dite du modèle à réaliser. C'est le but de la seconde étape qui sera présentée par la suite pour le cas des accéléromètres ainsi que pour celui des gyromètres. En référence à l'équation (4.1), cela revient alors à estimer Γ et Ω par $(\Gamma, \Omega) = \hat{\Phi}^{-1}(S, \hat{\theta})$ où $\hat{\theta}$ et $\hat{\Phi}$ sont décrits à l'aide des estimations réalisées durant l'étape précédente.

Pour accomplir cette seconde phase, il faut mettre en oeuvre des techniques récursives permettant l'inversion de systèmes non-linéaires. L'idée retenue dans cette étude a été de construire un modèle dynamique de l'accélération et de la vitesse angulaire permettant alors sa résolution à l'aide de méthodes de filtrage classiquement utilisées dans le domaine de la navigation (EKF, UKF, FP, ...).

4.3.1 Cas de l'accéléromètre

Pour construire un modèle dynamique de l'accélération Γ , il faut prendre une hypothèse sur son évolution dans le temps. L'idée est de choisir cette hypothèse de manière à pouvoir représenter le maximum de trajectoires possibles. Ainsi, notre choix a été de modéliser l'accélération par une marche aléatoire (4.8), ce qui permet de décrire un mouvement très général. La variance du bruit d'état $v_{\Gamma,k}$ devra être ajustée en fonction de la nature du mouvement du véhicule.

$$\begin{cases} \Gamma_k = \Gamma_{k-1} + v_{\Gamma,k}, \\ S_{a,k} = \hat{\Phi}_a(\Gamma_k, \hat{\theta}_a) + v_{a,k}. \end{cases} \quad (4.8)$$

Comme énoncé précédemment, la modélisation dynamique est résolue par une méthode de filtrage classique type EKF dont le déroulement peut être décrit comme suit

$$\left. \begin{array}{l} \Gamma_{k|k-1} = \Gamma_{k-1|k-1}, \\ P_{\Gamma,k|k-1} = P_{\Gamma,k-1|k-1} + Q_{\Gamma,k}, \\ S_{a,k|k-1} = \hat{\Phi}_a(\Gamma_{k|k-1}, \hat{\theta}_a), \\ P_{S_{a,k|k}} = H_{a,k} P_{\Gamma,k|k-1} H_{a,k}^T + R_{S_{a,k}}, \\ K_k = P_{\Gamma,k|k-1} H_{a,k}^T P_{S_{a,k|k}}^{-1}, \\ \Gamma_{k|k} = \Gamma_{k|k-1} + K_k (S_{a,k} - S_{a,k|k-1}), \\ P_{\Gamma,k|k} = (I - K_k H_{a,k}) P_{\Gamma,k|k-1}, \end{array} \right\} \begin{array}{l} \text{Prédiction} \\ \text{Correction} \end{array}$$

$$\text{où } \begin{cases} P_{\Gamma} = \text{Var}(\Gamma), \\ P_{S_a} = \text{Var}(S_a), \\ v_{\Gamma,k} \sim \mathcal{N}(0, Q_{\Gamma,k}), \\ v_{a,k} \sim \mathcal{N}(0, R_{S_{a,k}}), \\ H_{a,k} = \frac{\partial \hat{\Phi}_a}{\partial \Gamma_k} |_{\Gamma_{k|k-1}}. \end{cases}$$

4.3.2 Cas du gyromètre

De la même manière que dans le cas de l'accéléromètre, il est nécessaire d'établir une modélisation dynamique de la vitesse angulaire permettant de décrire le cas général d'un mouvement. Ceci est obtenu par une marche aléatoire (4.9) de bruit $v_{\Omega,k}$ dont la variance est fixée en fonction de la trajectoire du véhicule considérée.

$$\begin{cases} \Omega_k = \Omega_{k-1} + v_{\Omega,k}, \\ S_{g,k} = \hat{\Phi}_g(\Omega_k, \Gamma_{k|k}, \hat{\theta}_g) + v_{g,k}. \end{cases} \quad (4.9)$$

Ici aussi, la résolution sera réalisée à l'aide d'un filtre EKF dont l'algorithme

est le suivant

$$\left. \begin{aligned} \Omega_{k|k-1} &= \Omega_{k-1|k-1}, \\ P_{\Omega,k|k-1} &= P_{\Omega,k-1|k-1} + Q_{\Omega,k}, \\ S_{g,k|k-1} &= \hat{\Phi}_g(\Omega_{k|k-1}, \Gamma_{k|k}, \hat{\theta}_g), \\ P_{S_g,k|k} &= H_{g,k} P_{\Omega,k|k-1} H_{g,k}^T + R_{S_g,k}, \\ K_k &= P_{\Omega,k|k-1} H_{g,k}^T P_{S_g,k|k}^{-1}, \\ \Omega_{k|k} &= \Omega_{k|k-1} + K_k (S_{g,k} - S_{g,k|k-1}), \\ P_{\Omega,k|k} &= (I - K_k H_{g,k}) P_{\Omega,k|k-1}, \end{aligned} \right\} \begin{array}{l} \text{Prédiction} \\ \text{Correction} \end{array}$$

$$\text{où } \left\{ \begin{array}{l} P_{\Omega} = \text{Var}(\Omega), \\ P_{S_g} = \text{Var}(S_g), \\ v_{\Omega,k} \sim \mathcal{N}(0, Q_{\Omega,k}), \\ v_{g,k} \sim \mathcal{N}(0, R_{S_g,k}), \\ H_{g,k} = \frac{\partial \hat{\Phi}_g}{\partial \Omega_k} |_{\Omega_{k|k-1}}. \end{array} \right.$$

4.4 Résultats de simulations

Maintenant que la théorie des méthodes d'inversion utilisées dans cette étude a été détaillée, il est intéressant de présenter quelques résultats obtenus aussi bien en terme d'amélioration des mesures accélérométriques et gyrométriques (calibrage et inversion du modèle) que de navigation.

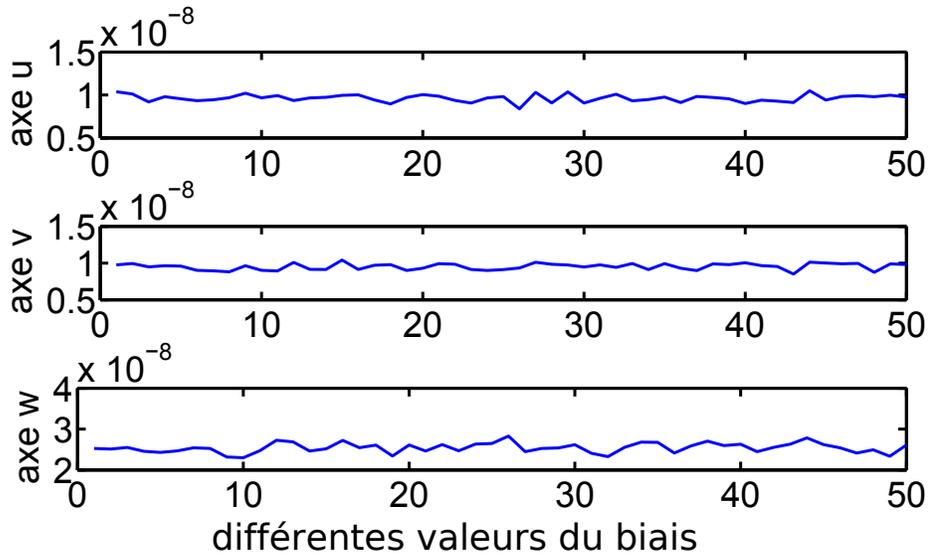
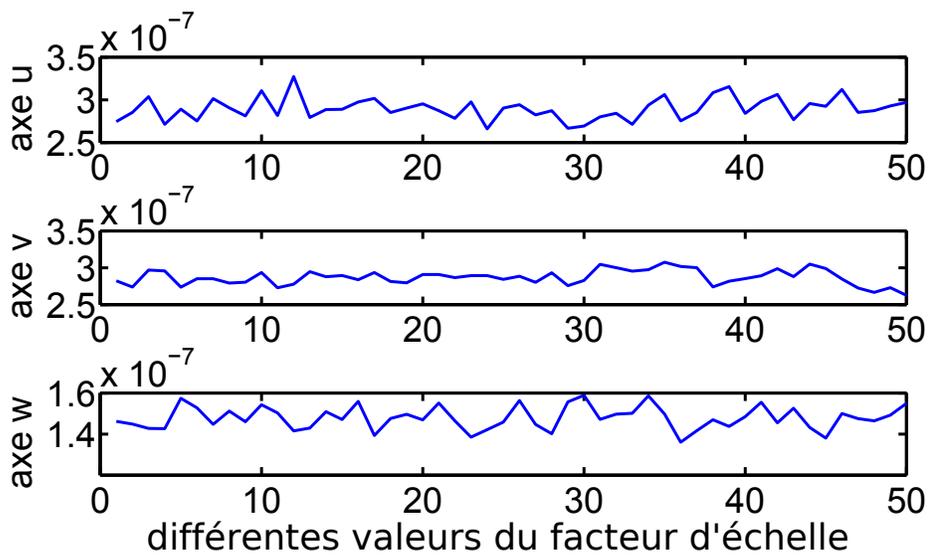
4.4.1 Calibrage

Le point d'intérêt est ici de déterminer la précision de l'estimation des paramètres $\hat{\theta}$ calculée à partir de la démarche décrite précédemment dans la section 4.2. Une étude statistique a donc été réalisée à l'aide de 1000 réalisations de Monte Carlo pour 50 valeurs différentes de θ et a permis de calculer l'erreur quadratique moyenne d'estimation. Il faut alors comparer cette erreur à l'ordre de grandeur des paramètres afin de pouvoir conclure sur l'exactitude de la méthode de calibrage proposée dans le cadre de cette étude.

Cas de l'accéléromètre Il est bon de rappeler que pour l'accéléromètre, le vecteur paramètre θ_a est composé de δ_{Ba} , δ_{SFa} , δ_{MAa} et δ_{NL2a} (Fig. 4.1 à 4.6).

L'erreur quadratique moyenne d'estimation des biais accélérométriques (Fig. 4.1) selon les 3 axes (u , v et resp. w) est proche de $1 \times 10^{-8} g^2$ (resp. $2.5 \times 10^{-8} g^2$). En comparant ces résultats à la variance du biais initial ($\approx 1 \times 10^{-4} g^2$), il apparaît que l'erreur commise par l'estimation est d'environ 1% de la valeur du paramètre. L'estimation de δ_{Ba} est donc très précise grâce à cette méthode.

Concernant les facteurs d'échelles δ_{SFa} , les résultats sont de l'ordre de 2.8×10^{-7} (resp. 1.5×10^{-7}) pour les axes u et v (resp. w). Suivant le même raisonnement que précédemment, il est aisé de voir que la précision de cette estimation sera inférieure à celle obtenue pour le biais.

FIGURE 4.1 – Erreur quadratique moyenne de l'estimation de δ_{Ba} [g^2]FIGURE 4.2 – Erreur quadratique moyenne de l'estimation de δ_{SF_a}

Les erreurs d'alignement δ_{MAa} sont décrites par une matrice hors-diagonale permettant de caractériser les éventuelles erreurs de calage des trois axes de l'accéléromètre u, v et w les uns par rapport aux autres. Un axe ne pouvant être mal calé par rapport à lui-même, les termes $\delta_{MAa}^{uu}, \delta_{MAa}^{vv}$ et δ_{MAa}^{ww} n'existent pas et aucune estimation n'est calculée (erreur d'estimation nulle sur les figures 4.3 à 4.5). Pour les termes dont l'erreur quadratique moyenne d'estimation est calculée, le résultat est compris entre $1 \times 10^{-8} rad^2$ et $4 \times 10^{-8} rad^2$. En comparant ces résultats à la variance des erreurs d'alignement initiales ($\approx 2.5 \times 10^{-3} rad^2$), il apparaît que l'erreur commise par l'estimation est inférieure à 1% de la valeur du paramètre. L'estimation de δ_{MAa} est donc très précise grâce à cette méthode.

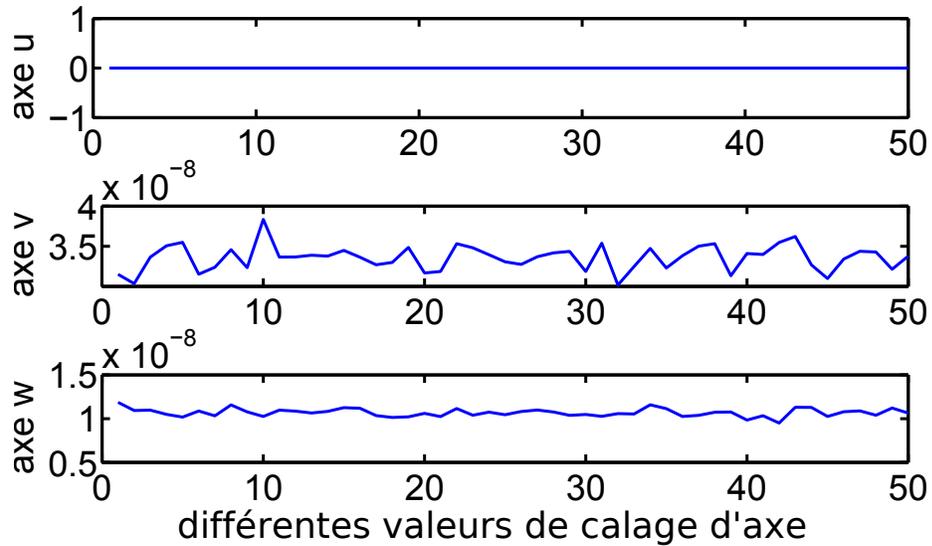
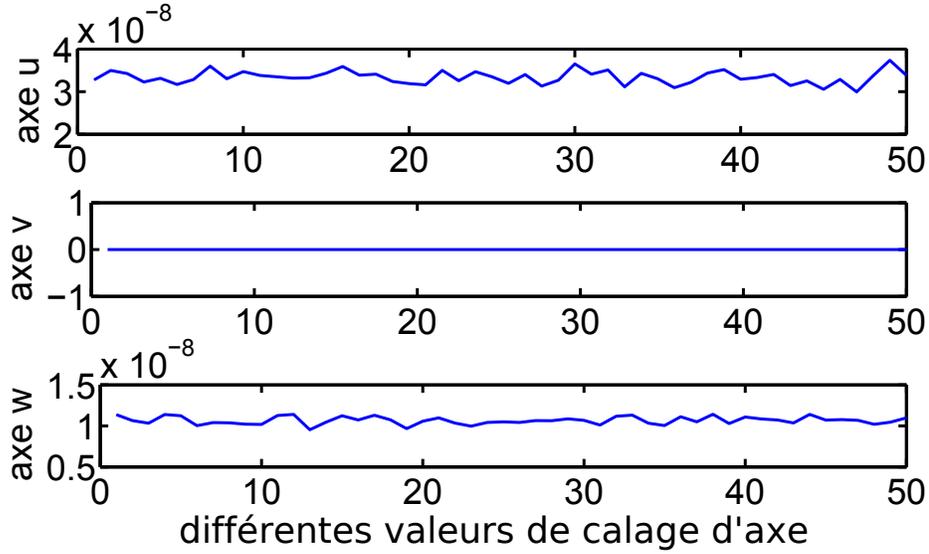
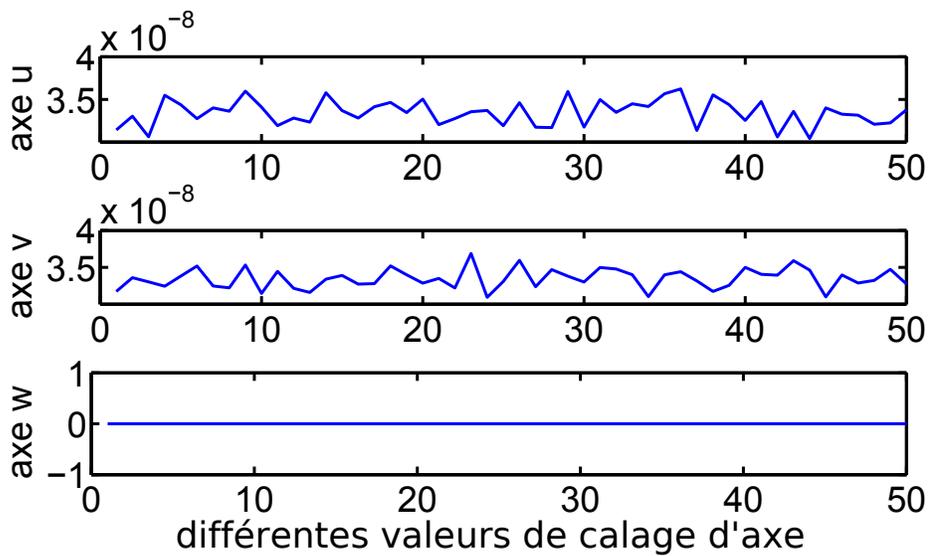


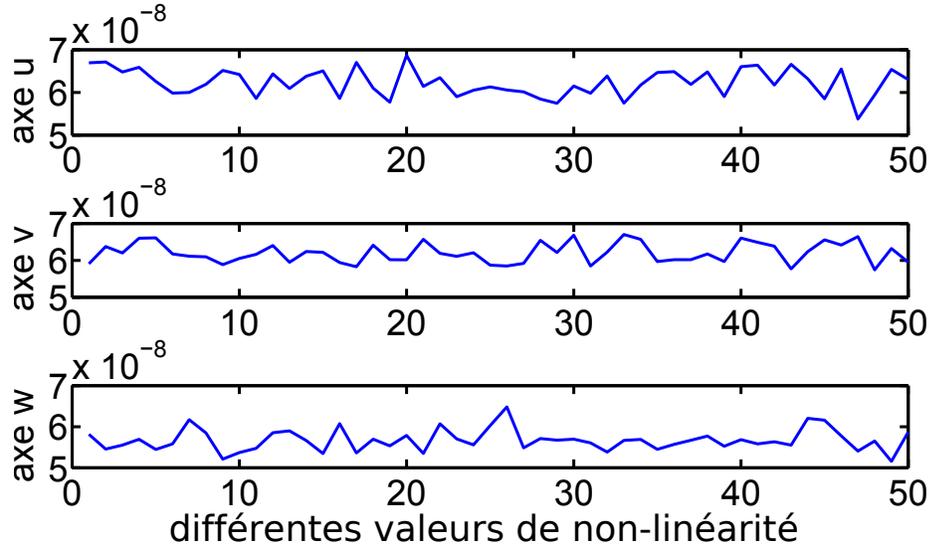
FIGURE 4.3 – Erreur quadratique moyenne de l'estimation de $\delta_{MAa}^u [rad^2]$

L'erreur quadratique moyenne d'estimation de la non-linéarité δ_{NL2a} de l'accéléromètre (Fig. 4.6) selon les 3 axes (u, v et resp. w) est proche de $6.5 \times 10^{-8} g^{-2}$ (resp. $5.5 \times 10^{-8} g^{-2}$). Suivant le même raisonnement que précédemment, il est aisé de voir que la précision de cette estimation sera inférieure à celle obtenue pour le biais ou pour les erreurs d'alignement.

Cas du gyromètre Le vecteur paramètre θ_g , composé de $\delta_{Bg}, \delta_{SFg}, \delta_{MAg}$ et δ_{Kg} , est estimé à l'aide de la méthode des moindres carrés. Les résultats sont représentés sur les figures 4.7 à 4.14.

L'erreur quadratique moyenne d'estimation des biais gyrométriques (Fig. 4.7) selon les 3 axes (p, q et r) est proche de $0.5 \times 10^{-5} (^\circ/s)^2$. En comparant ces résultats à la variance du biais initial ($\approx 5 \times 10^{-3} (^\circ/s)^2$), il apparaît que

FIGURE 4.4 – Erreur quadratique moyenne de l'estimation de δ_{MAa}^u [rad^2]FIGURE 4.5 – Erreur quadratique moyenne de l'estimation de δ_{MAa}^w [rad^2]

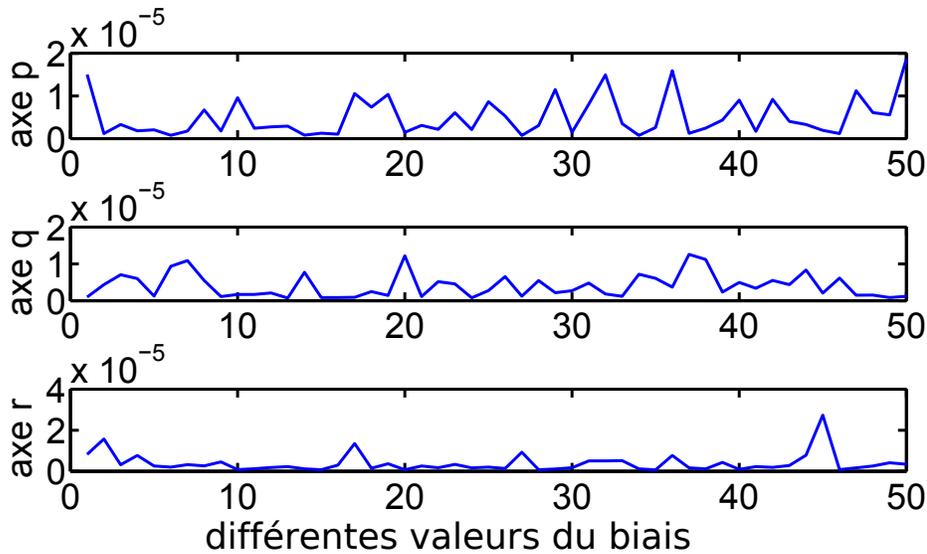
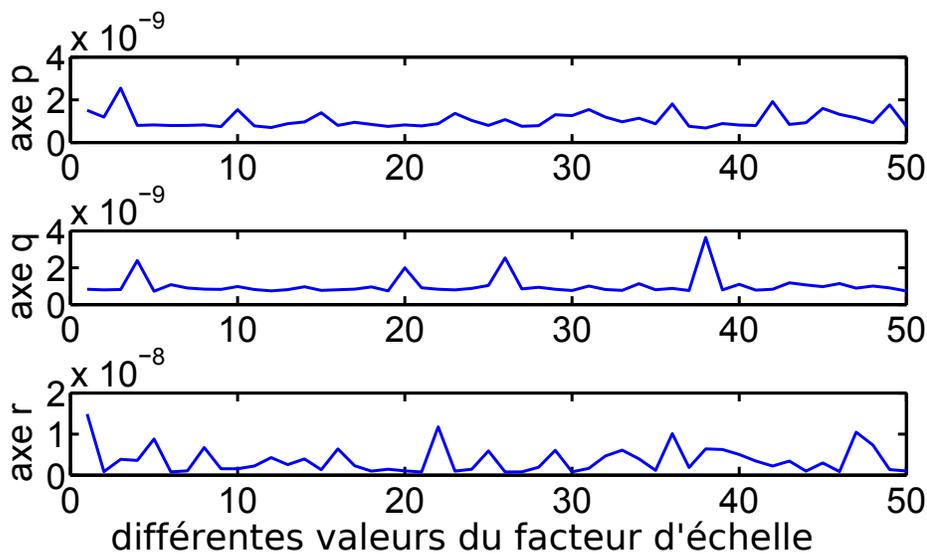
FIGURE 4.6 – Erreur quadratique moyenne de l'estimation de δ_{NL2a} [$1/g^2$]

l'erreur commise par l'estimation est d'environ 3% de la valeur du paramètre. L'estimation de δ_{Bg} est donc très précise grâce à cette méthode.

Concernant les facteurs d'échelles δ_{SFg} , les résultats sont de l'ordre de 0.1×10^{-8} (resp. 0.25×10^{-8}) pour les axes p et q (resp. r). Suivant le même raisonnement que précédemment, il est aisé de voir que la précision de cette estimation sera inférieure à celle obtenue pour le biais (10 à 15% de la valeur du paramètre).

De même que dans le cas de l'accéléromètre, les erreurs d'alignement δ_{MAg} sont décrites par une matrice hors-diagonale permettant de caractériser les éventuelles erreurs de calage des trois axes du gyromètre p, q et r les uns par rapport aux autres. Pour rappel, un axe ne pouvant être mal calé par rapport à lui-même, les termes δ_{MAg}^{pp} , δ_{MAg}^{qq} et δ_{MAg}^{rr} n'existent pas et aucune estimation n'est calculée (erreur d'estimation nulle sur les figures 4.9 à 4.11). Pour les termes dont l'erreur quadratique moyenne d'estimation est calculée, le résultat est de l'ordre de $1 \times 10^{-9} rad^2$. En comparant ces résultats à la variance des erreurs d'alignement initiales ($\approx 2.5 \times 10^{-3} rad^2$), il apparaît que l'erreur commise par l'estimation est inférieure à 1% de la valeur du paramètre. L'estimation de δ_{MAg} est donc très précise grâce à cette méthode.

La sensibilité aux accélérations δ_{Kg} est décrite par une matrice 3×3 reliant l'accélération ressentie sur chaque axe de l'accéléromètre u, v et w à la vitesse angulaire mesurée sur ceux du gyromètre p, q et r . Son erreur quadratique moyenne d'estimation (Fig. 4.12 à 4.14) est comprise entre $2.5 \times 10^{-6} (^\circ/s/g)^2$ et $7.5 \times 10^{-6} (^\circ/s/g)^2$. Suivant le même raisonnement que précédemment, il est

FIGURE 4.7 – Erreur quadratique moyenne de l'estimation de δ_{Bg} [$(^\circ/s)^2$]FIGURE 4.8 – Erreur quadratique moyenne de l'estimation de δ_{SFg}

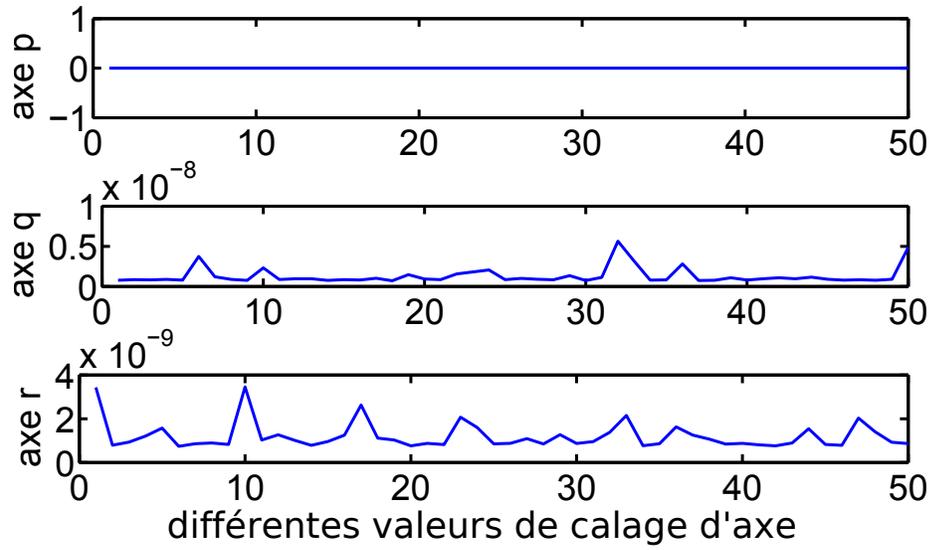


FIGURE 4.9 – Erreur quadratique moyenne de l'estimation de $\delta_{MAg}^p [rad^2]$

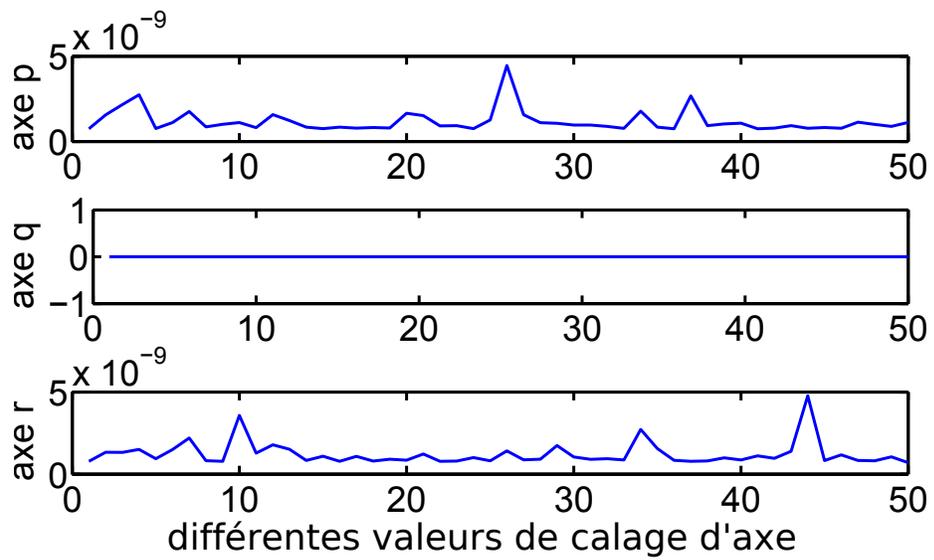


FIGURE 4.10 – Erreur quadratique moyenne de l'estimation de $\delta_{MAg}^q [rad^2]$

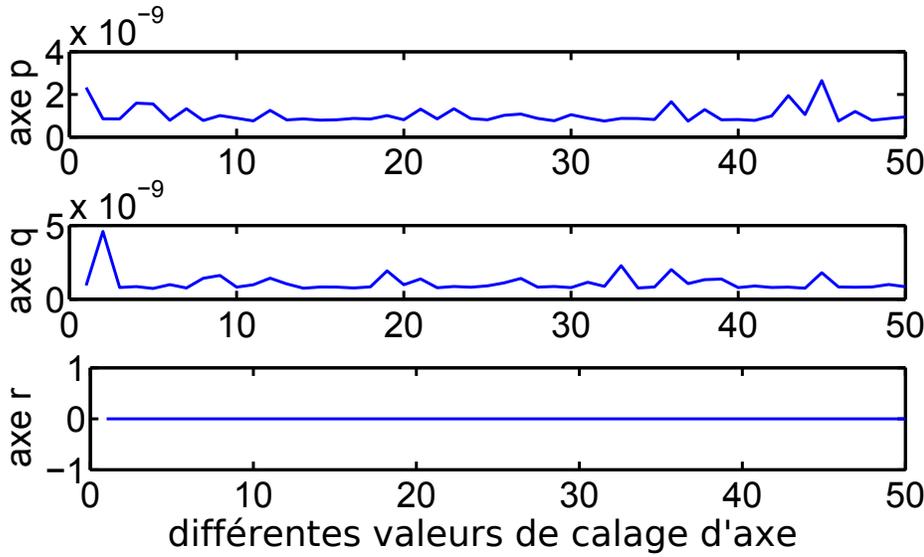


FIGURE 4.11 – Erreur quadratique moyenne de l'estimation de δ_{MA}^r [rad^2]

aisé de voir que la précision de cette estimation sera inférieure à celle obtenue pour le biais ou pour les erreurs d'alignement.

En résumé, notre méthode de calibrage a permis d'obtenir une estimation des paramètres intrinsèques du capteur MEMS. Cette estimation est précise pour les biais δ_B et les erreurs d'alignement δ_{MA} mais reste approximative pour les facteurs d'échelles δ_{SF} , la non-linéarité $\delta_{NL\alpha}$ et la sensibilité aux accélérations δ_{Kg} .

4.4.2 Inversion du modèle

Une fois le vecteur de paramètre θ estimé, il faut inverser le système à proprement parler à l'aide de la démarche précédemment décrite dans la section 4.3. Une estimation de l'accélération et de la vitesse angulaire est alors obtenue à partir des sorties accélérométriques et gyrométriques. C'est cette estimation qui sera caractérisée par son erreur quadratique moyenne calculée à partir de 300 réalisations de Monte Carlo de durée 120s chacune.

Cas de l'accéléromètre Les résultats sont similaires selon les trois axes et sont approximativement de $1 \times 10^{-5} g^2$ (Fig. 4.15). En comparant cette valeur à la variance du bruit accélérométrique $v_{a,k}$, il est aisé de se rendre compte qu'elles sont du même ordre de grandeur. L'estimation de l'accélération ne dépend donc plus que du bruit du système et non plus du modèle. Ce résultat était attendu car la méthodologie proposée permet de s'affranchir des perturbations liées au modèle accélérométrique mais ne prend pas en compte celles du bruit. Pour

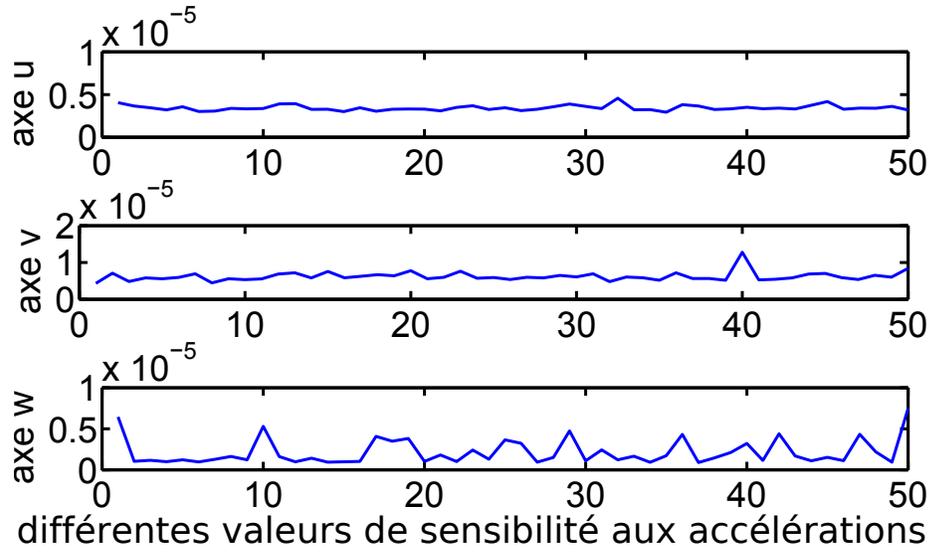


FIGURE 4.12 – Erreur quadratique moyenne de l'estimation de δ_{Kg}^p [$(^\circ/s/g)^2$]

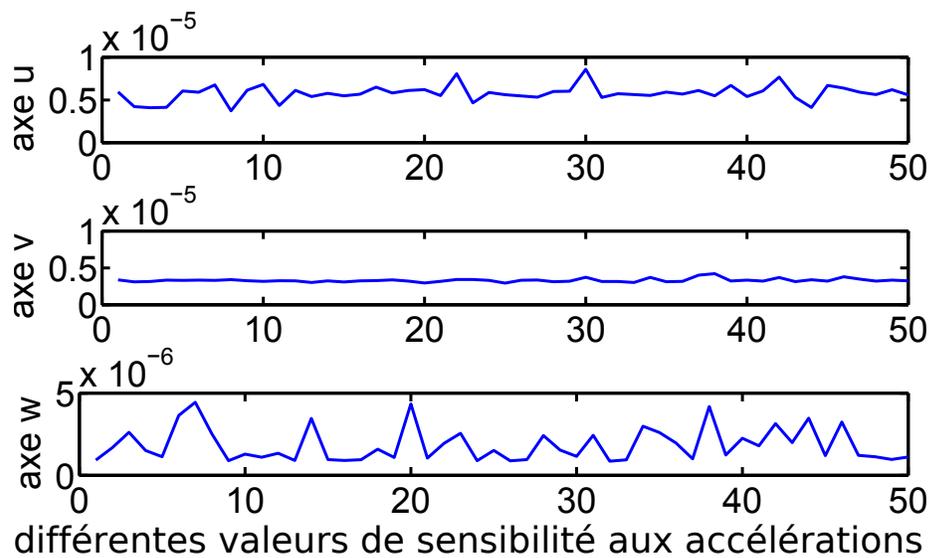


FIGURE 4.13 – Erreur quadratique moyenne de l'estimation de δ_{Kg}^q [$(^\circ/s/g)^2$]

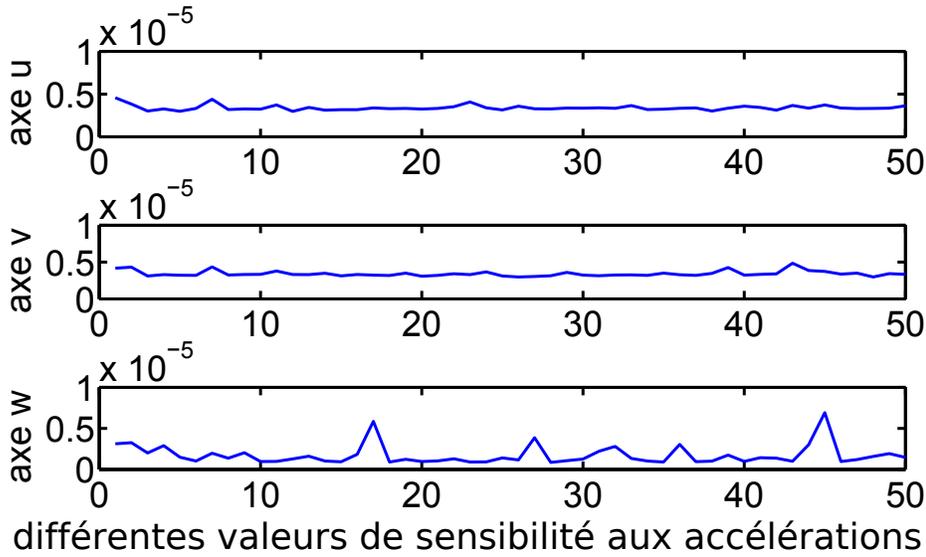


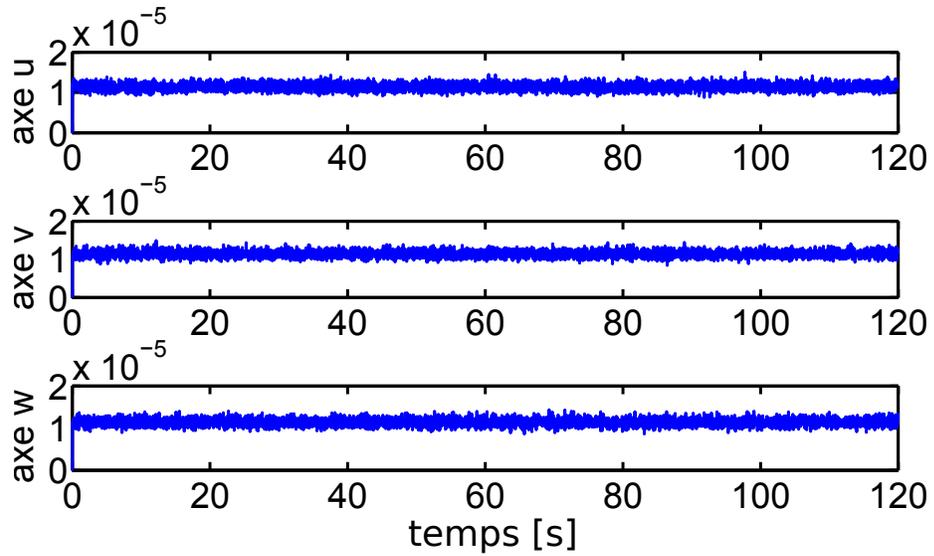
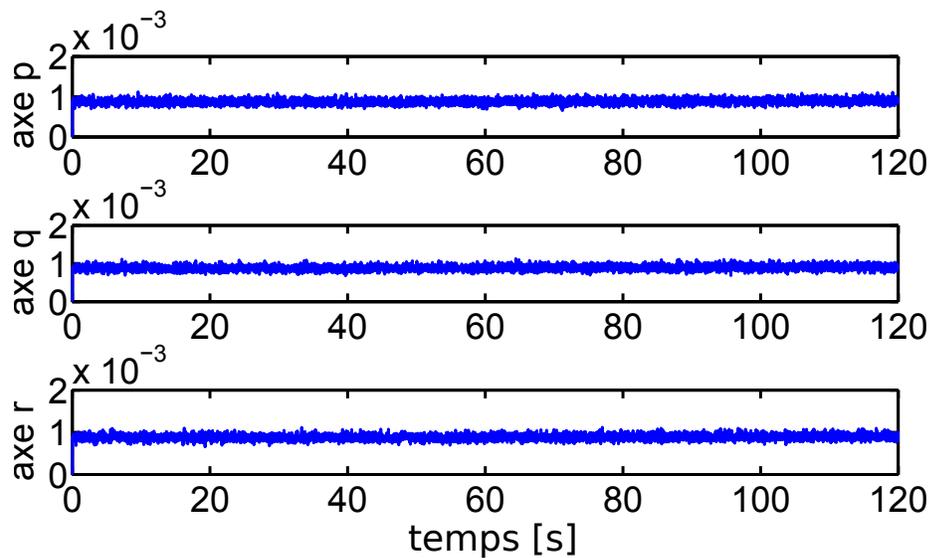
FIGURE 4.14 – Erreur quadratique moyenne de l'estimation de δ_{Kg}^r $[(^\circ/s/g)^2]$

réduire encore cette erreur d'estimation, il faudrait alors contraindre le modèle dynamique (4.8) en imposant une variance au bruit d'état $v_{\Gamma,k}$ moins importante et ainsi faire plus confiance à l'équation d'état.

Cas du gyromètre De manière équivalente, l'erreur quadratique moyenne d'estimation de la vitesse angulaire ($\approx 1 \times 10^{-3} (^\circ/s)^2$, Fig. 4.16) est comparable à la variance du bruit gyrométrique $v_{\Omega,k}$. Ce faisant, le résultat énoncé précédemment dans le cas de l'accéléromètre est aussi valable pour le gyromètre : l'estimation de la vitesse angulaire ne dépend donc plus que du bruit du système et non plus du modèle. De plus, il est toujours possible de réduire l'erreur d'estimation réalisée en diminuant la variance du bruit d'état du modèle dynamique (4.9) comme expliqué dans le cas de l'accéléromètre.

4.4.3 Navigation

Il reste désormais à évaluer la précision de la position hybride estimée, qui est la partie la plus intéressante en terme de navigation. Pour ce faire, le biais et la variance de l'estimateur ont été calculés à travers une étude statistique de 200 réalisations de Monte Carlo sur une durée de 600s. Les résultats issus de la méthode d'inversion du modèle MEMS (en vert) sont alors comparés, d'une part, avec ceux de la navigation hybride sans cette inversion (en bleu) et, d'autre part, avec des résultats de navigation hybride utilisant des centrales inertielles conventionnelles (en rose).

FIGURE 4.15 – Erreur quadratique moyenne de l'estimation de Γ [g^2]FIGURE 4.16 – Erreur quadratique moyenne de l'estimation de Ω [$(^\circ/s)^2$]

Cas nominal Dans un premier temps, l'objectif est de tester la méthodologie proposée sur un cas nominal, i.e., un cas où suffisamment de satellites GNSS sont directement visibles du récepteur et permettent donc l'estimation de la position à l'aide du système GNSS seul. Sur les Fig. 4.17 et 4.18 sont représentés les biais plus ou moins l'écart-type des estimateurs de position dans les 3 cas d'étude décrits précédemment. Il est aisé de voir une différence de comportement durant les 150 premières secondes de la simulation entre l'hybridation GNSS/MEMS avec et sans calibrage. Ceci provient du fait que le calibrage est déjà réalisé au début de la simulation dans le cas MEMS calibré alors que, sans la méthode d'inversion proposée ici, le système d'hybridation aura une phase d'auto-calibrage en début de simulation. De plus, le comportement de l'hybridation GNSS/MEMS calibré va tendre avec le temps vers celui du GNSS/INS alors que la précision du système hybride sans calibrage sera légèrement moins bonne. Cette différence peut notamment s'expliquer par la non prise en compte de toutes les perturbations du modèle MEMS par le filtre d'hybridation utilisé ici.

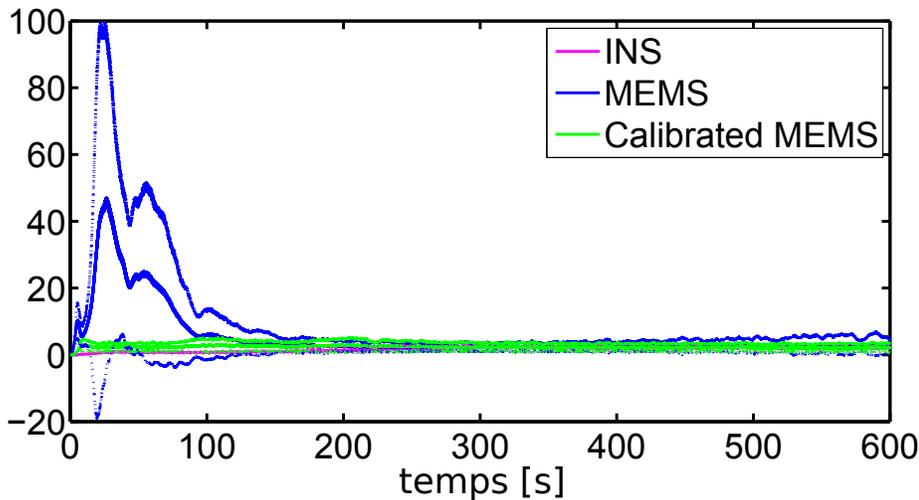


FIGURE 4.17 – Erreur en position (biais +/- écart-type) dans le cas nominal [m]

Perte du signal GNSS Après avoir vu les effets de la méthode d'inversion de modèle dans le cas nominal, il faut maintenant s'intéresser au cas de la perte du signal GNSS, cas durant lequel la centrale inertielle n'est plus recalée et donc particulièrement sensible au comportement des capteurs inertiels. De même que précédemment, les Fig. 4.19 et 4.20 représentent les biais plus ou moins l'écart-type des estimateurs de position dans les différents cas d'études (GNSS/INS, GNSS/MEMS et GNSS/MEMS calibré). Tout d'abord, une première observation permet de constater que les différences de résultats entre les 3 cas d'étude

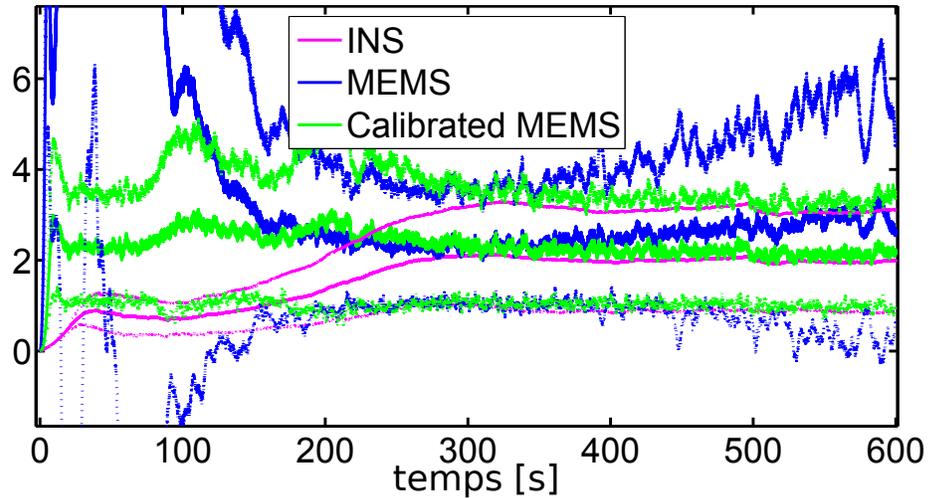


FIGURE 4.18 – Erreur en position (biais +/- écart-type) dans le cas nominal [m] (zoom sur la zone d'intérêt)

du cas nominal sont toujours présentes. De plus, durant la perte du signal GNSS, une amélioration de la précision d'approximativement 15% est obtenue à l'aide de l'inversion du modèle. Ce progrès est toutefois bien insuffisant au vue des résultats de référence de la navigation hybride utilisant des centrales inertielle conventionnelles.

4.5 Conclusion

Ce chapitre a permis de présenter une technique pour résoudre le problème de la navigation hybride GNSS/MEMS et, plus particulièrement, pour améliorer la qualité des mesures issues des capteurs inertiels MEMS. Basée sur des méthodes bien connues (estimation des moindres carrés et filtre de Kalman étendu), son originalité provient plus de sa façon de voir le problème. En effet, l'amélioration des mesures inertielle est classiquement traitée comme un problème de débruitage et, donc, résolu comme tel alors que dans cette étude, l'idée a été de le poser sous forme d'un problème inverse. Les simulations réalisées ont permis de caractériser la méthode proposée et ce, aussi bien, en terme d'estimation des paramètres du modèle MEMS, de l'accélération et de la vitesse angulaire qu'en terme de navigation [DCTL08]. Dans le cas nominal, l'inversion du modèle permet d'égaliser la précision de la navigation GNSS/INS tandis que lors de pertes de signal, bien qu'améliorant l'estimation d'environ 15%, le résultat reste approximatif. Il faudra donc mettre en oeuvre des techniques spécifiques pour traiter cette perte du signal GNSS, ce qui sera l'objet du chapitre suivant.

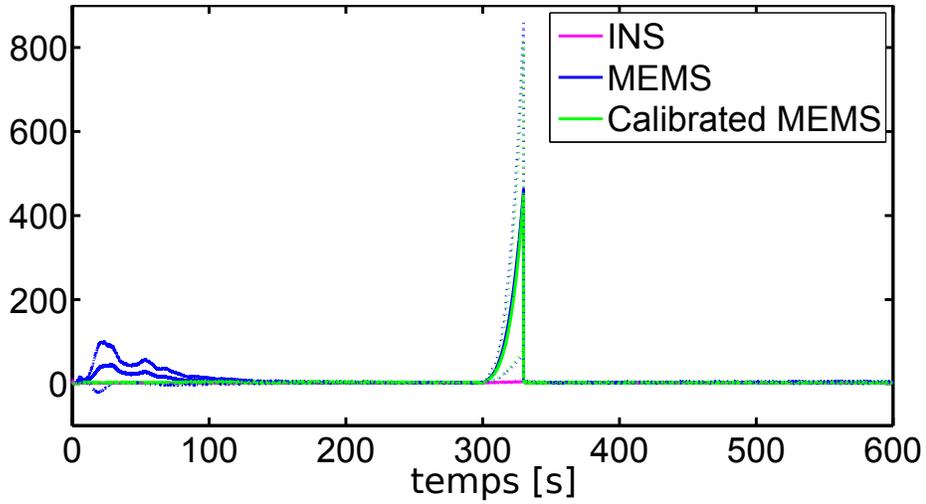


FIGURE 4.19 – Erreur en position (biais +/- écart-type) en cas de perte du signal GNSS [m]

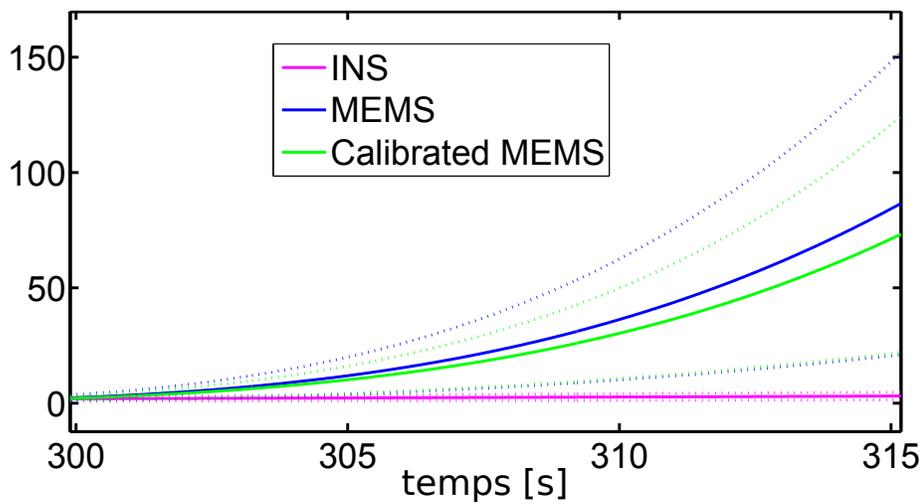


FIGURE 4.20 – Erreur en position (biais +/- écart-type) en cas de perte du signal GNSS [m] (zoom sur la zone d'intérêt)

Chapitre 5

Correction par réseau de neurones

Sommaire

5.1	Introduction	69
5.2	Bases des réseaux de neurones	70
5.2.1	La cellule élémentaire : le neurone	70
5.2.2	Structure du réseau de neurones	71
5.2.3	Algorithmes d'apprentissage	73
5.3	Application à la navigation	74
5.3.1	Correction de la position uniquement	75
5.3.2	Correction des différentes estimées inertielles	76
5.4	Résultats de simulations	79
5.4.1	Cas simple de trajectoires invariantes dans le temps	81
5.4.2	Cas de trajectoires évoluant dans le temps	82
5.4.3	Corrections des différentes estimées inertielles	88
5.5	Conclusion	97

5.1 Introduction

Dans l'optique de réaliser une navigation hybride GNSS/MEMS, il a été montré précédemment que l'information apportée par les capteurs MEMS ne suffisait pas pour obtenir des résultats précis lors de perte de signal GNSS. Dans ce cas, il est donc nécessaire de mettre en oeuvre des méthodes spécifiques pour améliorer l'estimation de positionnement. Cette amélioration ne peut être réalisée qu'en apportant plus d'information au système de navigation. Ceci peut être effectué en fusionnant des données issues de différents capteurs (odomètres [SCT07], [SCKJ07], caméras [WKD09], [SS09], ...) ou systèmes de communications (GSM [HN80], [HMS97], WiFi [HSS03], [DT08], ...), mais ce ne sera pas le

cas de cette étude volontairement limitée à une navigation hybride GNSS/MEMS. Cet apport d'information supplémentaire peut alors être cherché dans le passé du système de navigation en apprenant le comportement de la navigation hybride afin de pouvoir le restituer lors de perte de signal GNSS. C'est dans ce but que des méthodes neuronales seront mises en œuvre et présentées dans ce chapitre. Un certain nombre d'études nécessitant un apprentissage hors-ligne ont été menées en ce sens ([EDER03], [AH04], [ESAHL04], [GSES06], [WWSW06], [AHNES07], [GESS08], ...), l'originalité des travaux réalisés dans cette thèse se contentant d'un apprentissage en ligne. Ainsi, après avoir effectué quelques rappels sur les réseaux de neurones, une description détaillée de leur application à la navigation sera proposée.

5.2 Bases des réseaux de neurones

Le cerveau humain est une structure composée d'un très grand nombre de cellules appelées neurones. Ces neurones sont organisés en un immense réseau dont les fonctionnalités sont aussi diverses que compliquées. Cette structure, ainsi construite, possède les capacités d'apprendre et de mémoriser [Hay99]. Un réseau de neurones artificiel est un modèle du cerveau humain cherchant à avoir ces mêmes capacités.

De façon générale, on situe le début des réseaux de neurones artificiels en 1943 avec les travaux de McCulloch et Pitts [MP43] qui montrent qu'un réseau de neurones discret, sans contrainte de topologie, peut représenter n'importe quelle fonction booléenne et donc émuler un ordinateur. En 1958, Rosenblatt propose le premier algorithme d'apprentissage, qui permet d'ajuster les paramètres d'un neurone [Ros58]. En 1969, Minsky et Papert publient le livre *Perceptrons* [MP69] dans lequel ils utilisent une solide argumentation mathématique pour démontrer les limitations des réseaux de neurones à une seule couche. Ce livre aura une influence telle que la plupart des chercheurs quitteront le champ de recherche sur les réseaux de neurones. En 1982, Hopfield [Hop82] propose des réseaux de neurones associatifs et l'intérêt pour les réseaux de neurones renaît chez les scientifiques. En 1986, Rumelhart, Hinton et Williams publient l'algorithme de la rétropropagation de l'erreur (Annexe C) qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches [RHW86]. À partir de ce moment, la recherche sur les réseaux de neurones connaît un essor fulgurant et les applications commerciales de ce succès académique suivent au cours des années 90.

5.2.1 La cellule élémentaire : le neurone

La cellule élémentaire du réseau ou neurone (Fig. 5.1) possède un ensemble d'entrées qui, après avoir été pondérées, vont être associées par l'intermédiaire d'une fonction de combinaison de type vecteur à scalaire. Ces fonctions sont majoritairement de deux natures différentes :

- Les réseaux de type MLP (Multi-Layer Perceptron) calculent une combinaison linéaire des entrées (c'est-à-dire que la fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques) et appliquent ensuite une non-linéarité φ à cette combinaison linéaire. La sortie d'un réseau de type MLP, représenté sur la figure 5.1, est donc définie par

$$OUT = \varphi \left(\sum_{i=1}^n w_i IN_i \right).$$

Classiquement, des fonctions de type sigmoïde, tangente hyperbolique ou kronecker sont utilisées en tant que fonction d'activation.

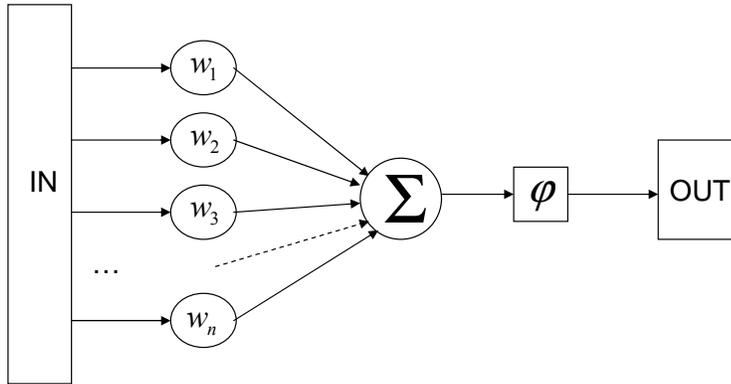


FIGURE 5.1 – Cellule élémentaire générique

- Les réseaux de type RBF (Radial Basis Function) calculent la distance entre les entrées, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrées à travers la fonction de base, classiquement gaussienne.

5.2.2 Structure du réseau de neurones

Un ensemble de neurones va alors être constitué en les interconnectant et donnera naissance au réseau à proprement parler (Fig. 5.2).

Les objectifs de ces réseaux sont multiples (reconnaissance des formes, classification, identification, estimation, ...) mais ils conservent un mode de fonctionnement similaire quelle que soit l'application à laquelle ils sont dédiés. En effet, toute utilisation de réseaux de neurones pourra se décomposer en deux phases :

- une phase d'apprentissage (Fig. 5.3), durant laquelle seront déterminés le nombre de couches, le nombre de neurones par couche et la nature des fonctions composant ces couches. Ces tâches sont réalisées empiriquement car il n'existe pas de manière méthodique rigoureuse pour le faire. Ceci

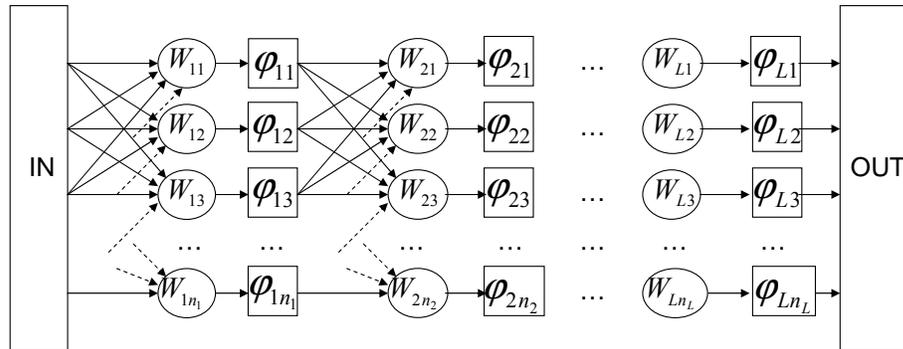


FIGURE 5.2 – Structure interne d'un réseau de neurones

est incontestablement un inconvénient majeur des réseaux de neurones car il oblige à un grand nombre de tentatives avant de trouver le réseau qui convient le mieux au problème. De plus, durant cette phase d'apprentissage, les biais et les poids vont être ajustés au mieux à l'aide d'algorithmes qui seront détaillés ultérieurement.

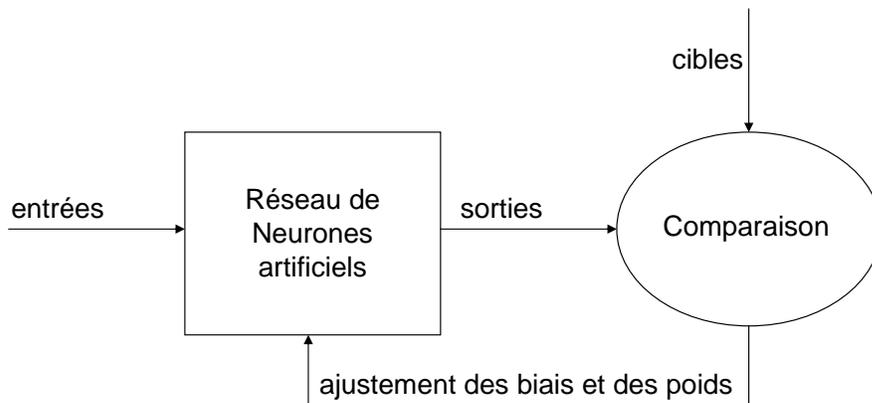


FIGURE 5.3 – Phase d'apprentissage supervisé d'un réseau de neurones

- une phase de généralisation (Fig. 5.4) où le réseau va “réciter” ce qu’il a appris et donc permettre le calcul d’une fonction spécifique, qu’elle soit booléenne ou non.

Il a été démontré, par le passé, qu’un réseau, disposant d’un assez grand nombre de neurones de fonction d’activation non linéaire et, ce, sur une unique couche cachée, permet d’estimer toutes les fonctions possédant un nombre fini de discontinuité [Cyb89]. Ceci est la force des réseaux de neurones car, ainsi, il

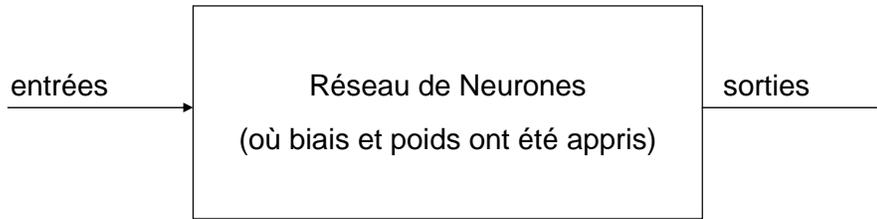


FIGURE 5.4 – Phase de généralisation d’un réseau de neurones

est possible d’estimer des fonctions dont il n’existe pas de modélisation.

5.2.3 Algorithmes d’apprentissage

Ici ne seront détaillées que les méthodes d’apprentissage dites supervisées, cas intéressants du point de vue estimation. Les lois d’apprentissage sont un ensemble de procédés par lesquels les biais et les poids du réseau de neurones vont être modifiés et ce, dans le but de minimiser la différence entre la sortie du réseau et la cible qu’il doit atteindre. Ces algorithmes cherchent donc, en pratique, le minimum d’une fonction coût. Il existe dans la littérature de nombreuses méthodes pour mettre en oeuvre cette minimisation. Les algorithmes de rétropropagation (Annexe C) sont cependant les plus communément utilisés pour effectuer cette minimisation dans le cadre de réseaux de neurones. Ces méthodes se séparent en trois familles distinctes.

La descente en gradient et ses variantes L’idée de la méthode est relativement simple. Il suffit, en effet, de suivre la direction de descente, en un point d’une fonction, définie par l’inverse de son gradient. Il faut alors se déplacer selon cette direction d’une longueur fixe ou variable afin de se diriger vers un minimum local de la fonction. Ainsi, l’algorithme qui en découle est :

INITIALISATION : *choisir le point initial x_0 et mettre t à 1.*
 ÉTAPE t : *calculer $f(x_{t-1})$ et $\nabla f(x_{t-1})$.
 $x_t = x_{t-1} - \gamma_t \nabla f(x_{t-1})$ où γ_t est le pas de descente.
 remplacer t par $t + 1$ et itérer.*

De nombreuses variantes existent telles les descentes en gradient à pas variable, les descentes en gradient conjugué, ...

Les méthodes de Newton ou quasi-Newton L’idée de cette méthode est là encore assez simple. En approximant la fonction f dans le voisinage de x par son développement de Taylor du second ordre, il peut être écrit

$$f(y) \approx f(x) + (y - x)^T \nabla f(x) + \frac{1}{2} (y - x)^T \nabla^2 f(x) (y - x)$$

ce qui, en dérivant $f(y)$ et en posant $\nabla f(y) = 0$, conduit à

$$\nabla^2 f(x)(y - x) = -\nabla f(x),$$

pouvant s'écrire également, si le hessien est inversible,

$$y = x - [\nabla^2 f(x)]^{-1} \nabla f(x).$$

C'est cette dernière équation qui est le coeur des algorithmes itératifs des méthodes de Newton ou quasi-Newton en posant simplement $y = x_t$ et $x = x_{t-1}$. Ici encore, de nombreuses variantes existent. C'est dans le calcul ou plutôt l'approximation du hessien que diffèrent ces formes de méthodes de quasi-Newton.

Les méthodes mixtes Cette troisième classe de méthodes utilise les deux familles de méthodes précédentes. L'idée est bien évidemment de cumuler les avantages pour converger le plus rapidement possible. Pour ce faire, il faut tout d'abord remarquer que les méthodes de quasi-Newton, par leur nature, vont être très performantes lorsque l'évolution de la fonction g créée par le réseau sera linéaire. En effet, dans ces conditions, la fonction coût f à minimiser, et définie ci-dessous, sera quadratique.

$$f(p) = \langle (g(x, p) - y)^2 \rangle \text{ où } \begin{cases} x & \text{sont les entrées du réseau de neurones,} \\ y & \text{sont les cibles qu'il doit atteindre,} \\ p & \text{sont ces biais et ces poids.} \end{cases}$$

L'algorithme de Levenberg Marquardt fait partie de cette famille et peut être décrit par l'association des deux précédentes méthodes en une seule équation.

$$\underbrace{x_t = x_{t-1} - \gamma_t \nabla f(x_{t-1})}_{\searrow} \qquad \underbrace{x_t = x_{t-1} - [\nabla^2 f(x_{t-1})]^{-1} \nabla f(x_{t-1})}_{\swarrow}$$

$$x_t = x_{t-1} - [\nabla^2 f(x_{t-1}) + \lambda I]^{-1} \nabla f(x_{t-1})$$

Ainsi,

- si λ est petit, seule l'approximation quadratique est utilisée.
- si λ est grand, l'équation correspond à une descente en gradient.
- si λ est intermédiaire, l'algorithme est un mélange entre la descente de gradient et l'approche quadratique basée sur l'approximation linéaire de g .

Ce coefficient λ est modifié à chaque itération en fonction de la linéarité de f afin d'augmenter la rapidité de la convergence.

5.3 Application à la navigation

L'idée de cette partie est d'utiliser les méthodes neuronales présentées dans la section précédente au domaine de la navigation. Dans le cas de la navigation GNSS/MEMS, la perte du signal GNSS peut venir détériorer fortement la

précision du positionnement. Afin de lutter contre ce phénomène, il est nécessaire de mettre en oeuvre des techniques spécifiques qui seront, dans cette étude, basées sur les réseaux de neurones. Le but est d'apprendre le comportement du système de navigation hybride lorsque le signal GNSS est disponible puis de restituer cette connaissance lors de pertes de ce signal. Comme il est toujours possible d'accéder au résultat de navigation en survie (navigation sans mesures de pseudo-distances) lors de pertes de signal GNSS, l'approche retenue est d'apprendre la différence entre les deux systèmes, hybride d'une part et en survie d'autre part. Dans un premier temps, l'étude menée s'est contentée d'apporter la correction en terme de position alors que par la suite, la vitesse et l'attitude seront également corrigées.

5.3.1 Correction de la position uniquement

Comme cela a été énoncé précédemment, la mise en oeuvre d'un réseau de neurones se déroule en deux étapes, la phase d'apprentissage et la phase de restitution (appelée aussi phase de restitution). Cette section a pour but de détailler ces deux étapes pour corriger la position d'un véhicule dans le cas de la navigation GNSS/MEMS. Les sorties du réseau de neurones, c'est-à-dire ce qu'il doit apprendre, seront dans ce cas les différences de position entre la solution hybride et la solution en survie selon les trois axes (latitude, longitude et altitude). Il faut choisir les entrées du réseau de manière à ce qu'elles lui apportent de l'information pour permettre l'apprentissage et qu'elles soient disponibles même lorsque le signal GNSS ne l'est pas. Le choix réalisé dans cette étude a été d'apporter le maximum d'information au réseau de neurones. Ainsi, en entrée se trouveront toutes les données relatives au système inertiel en survie :

- Sorties accélérométriques,
- Sorties gyrométriques,
- Positions en survie,
- Vitesses en survie,
- Attitudes en survie.

De plus, le temps écoulé depuis le début de la perte du signal GNSS sera également une entrée du réseau de neurones. Connaissant désormais les entrées et les sorties du réseau de neurones, il est possible de se pencher sur le cas de l'apprentissage.

Phase d'apprentissage Durant cette première étape, le signal GNSS doit être disponible pour pouvoir accéder à l'estimation de la position par le système hybride. Afin d'obtenir également une position estimée en survie, il faut alors simuler des pertes de signal GNSS et intégrer les mesures accélérométriques et gyrométriques dans une autre plateforme inertielle. La Figure 5.5 permet de mieux comprendre le fonctionnement de la méthode proposée :

1. Les mesures issues des capteurs MEMS (accéléromètres et gyromètres) sont utilisées dans deux systèmes de navigation, l'un hybride et l'autre en survie.

2. La différence entre les résultats de ces deux systèmes en termes de positionnement est calculée et placée en sortie du réseau de neurones comme cible à atteindre.
3. Les entrées du réseau de neurones préalablement choisies (sorties accélérométriques et gyrométriques, positions, vitesses et attitudes en survie et temps écoulé depuis le début de la perte de signal GNSS Δ_t) vont alors permettre de réaliser l'apprentissage.

Il est important de noter qu'il est nécessaire de simuler plusieurs pertes de signal GNSS afin de constituer une base d'apprentissage suffisamment conséquente, ce qui permettra un apprentissage du réseau de neurones dans de bonnes conditions. La manière la plus simple de procéder est alors de simuler des pertes de signal GNSS successives (Cas 1, Fig. 5.6). Pour ce faire, il suffit de connaître la durée maximale d'une perte de signal GNSS (T_{out}) qu'il faut pouvoir corriger à l'aide du réseau de neurones et de savoir le nombre de simulations N_{out} qu'il faut réaliser pour obtenir une base d'apprentissage suffisante. Cette première manière de constituer la base d'apprentissage a pour avantage d'être simple à mettre en œuvre mais possède également un inconvénient majeur qui est la durée importante de données nécessaires pour construire la base d'apprentissage ($T_{app} = N_{out}T_{out}$). Pour pallier ce problème, il est possible que les pertes simulées de signal GNSS se chevauchent (Cas 2, Fig. 5.6) afin de créer artificiellement plus de simulations dans une même durée T_{app} ou de réduire considérablement la durée de données nécessaires pour construire la base d'apprentissage ($T'_{app} \ll T_{app}$).

Phase de restitution Durant cette seconde étape, le signal GNSS et, donc, le système de navigation hybride, sont indisponibles mais le réseau de neurones va venir aider le système de navigation en survie. L'architecture proposée en phase de restitution est représentée sur la figure 5.7. Notons que seul le système en survie apparaît et que le réseau de neurones est utilisé, cette fois-ci, en phase de généralisation avec les mêmes entrées que dans la phase d'apprentissage (sorties accélérométriques et gyrométriques, positions, vitesses et attitudes en survie et durée de la perte de signal GNSS Δ_t).

5.3.2 Correction des différentes estimées inertielles

L'idée est, ici, d'utiliser le réseau de neurones pour tenter de corriger, non plus la position uniquement, mais toutes les estimées inertielles que sont la position, la vitesse et l'attitude. De la même manière que précédemment, le réseau de neurones se verra confier la tâche d'apprendre la différence entre la solution hybride et la solution en survie. Ainsi, les sorties seront les différences de position, vitesse et attitude entre la solution hybride et la solution en survie et, ce, en trois dimensions alors que les entrées resteront inchangées (sorties accélérométriques et gyrométriques, positions, vitesses et attitudes en survie et temps écoulé depuis le début de la perte de signal GNSS Δ_t).

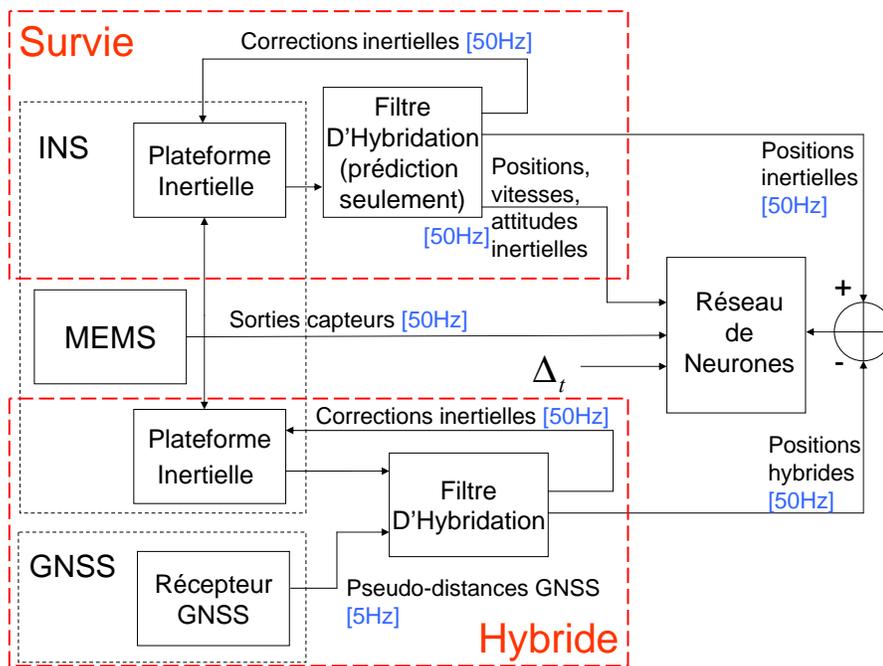


FIGURE 5.5 – Architecture en phase d'apprentissage (correction de la position uniquement)

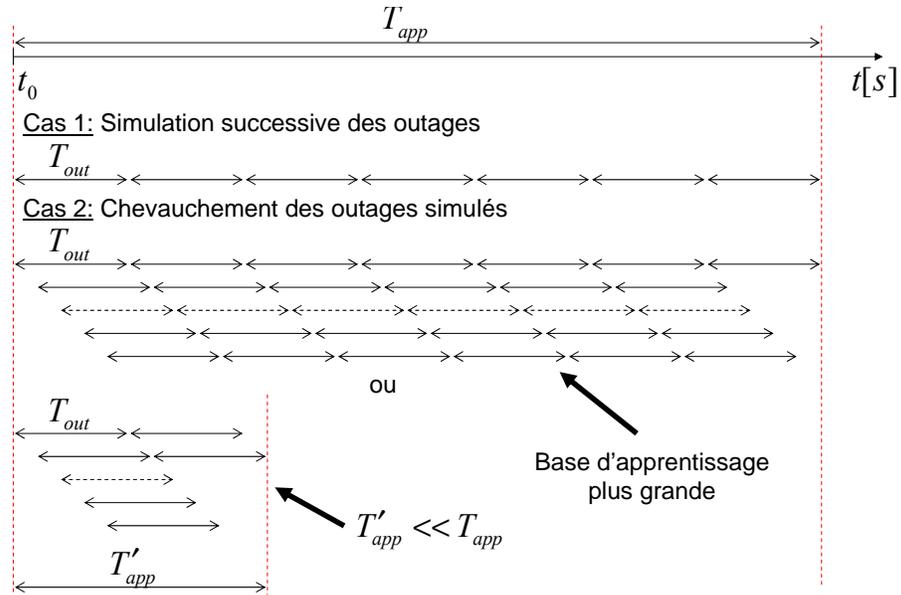


FIGURE 5.6 – Construction de la base d'apprentissage

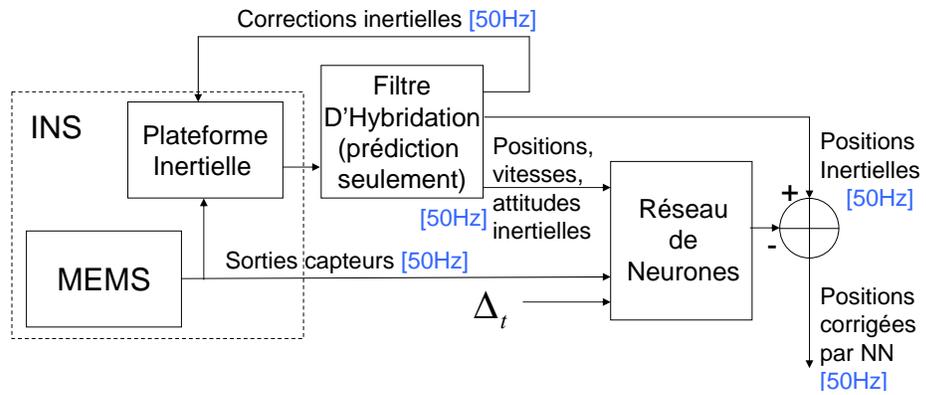


FIGURE 5.7 – Architecture en phase de restitution (correction de la position uniquement)

Phase d'apprentissage De même que précédemment, lorsque le réseau de neurones n'était utilisé que pour corriger la position, le signal GNSS et, donc, le système de navigation hybride doivent être disponibles durant la phase d'apprentissage. La solution en survie est alors simulée dans une autre plateforme inertielle et permet d'obtenir une estimation de la position, de la vitesse et de l'attitude qui serait obtenue en l'absence du signal GNSS. Le fonctionnement de cette nouvelle architecture (Fig. 5.8) est alors très similaire au précédent et peut se décrire sous la forme :

1. Les mesures issues des capteurs MEMS (accéléromètres et gyromètres) sont utilisées dans deux systèmes de navigation, l'un hybride et l'autre en survie.
2. La différence entre les estimées inertielles (positions, vitesses, attitudes) de ces deux systèmes est calculée et placée en sortie du réseau de neurones comme cible à atteindre.
3. Les entrées du réseau de neurones préalablement choisies (sorties accélérométriques et gyrométriques, positions, vitesses et attitudes en survie et temps écoulé depuis le début de la perte de signal GNSS Δ_t) vont alors permettre de réaliser l'apprentissage.

La base d'apprentissage peut là aussi être constituée de deux manières différentes : simulations successives des pertes de signal GNSS ou chevauchement des simulations (Fig. 5.6).

Phase de restitution Durant la phase de restitution et à l'instar de la correction de la position seule, le signal GNSS est indisponible mais le réseau de neurones aide le système de navigation en survie. La nouvelle architecture obtenue en phase de restitution et représentée sur la figure 5.9 est très similaire au cas étudié dans la section 5.3.1.

5.4 Résultats de simulations

Les méthodes exposées précédemment doivent être mises en oeuvre au travers de différents exemples afin d'analyser leurs performances. Dans un premier temps, l'intérêt sera porté sur des trajectoires dont les dynamiques durant la perte de signal GNSS ont déjà été rencontrées dans le passé (cas favorable pour l'apprentissage). Par la suite, des cas moins favorables pour l'apprentissage du réseau de neurones seront également simulés. Ces deux premières séries de tests seront réalisées dans le cas de la correction exclusive de la position. Une dernière étude sera alors menée sur l'utilisation du réseau de neurones pour corriger les différentes estimées inertielles. Dans tous les cas, les résultats présentés sont les racines carrées des erreurs quadratiques moyennes après 100 itérations de Monte Carlo en terme de position et, éventuellement, de vitesse et d'attitude. De plus, la base d'apprentissage sera constituée de 40 pertes de signal GNSS simulées d'une durée de 30s chacune se chevauchant comme expliqué précédemment (Cas 2, Fig. 5.6).

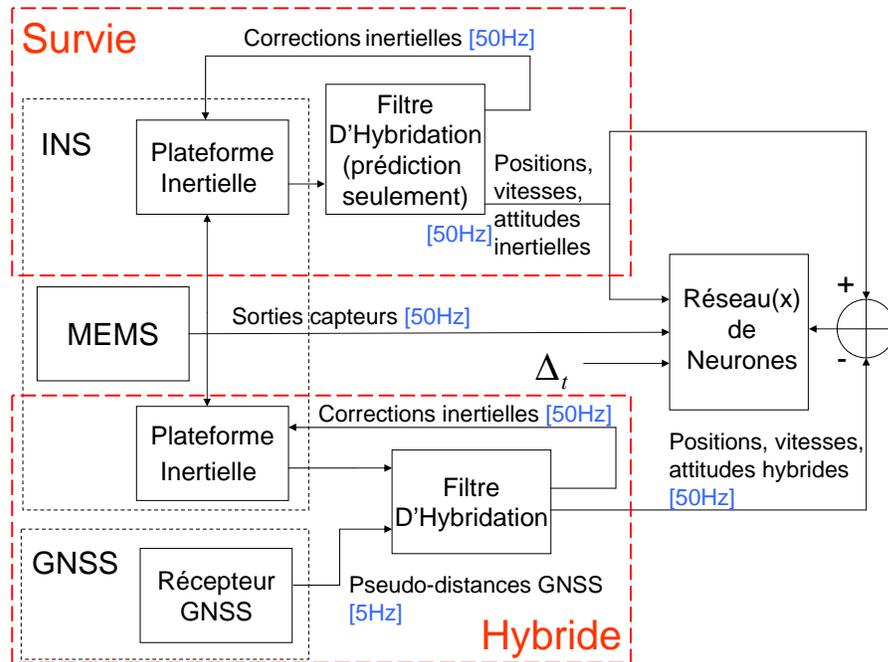


FIGURE 5.8 – Architecture en phase d'apprentissage (correction des estimées inertielles)

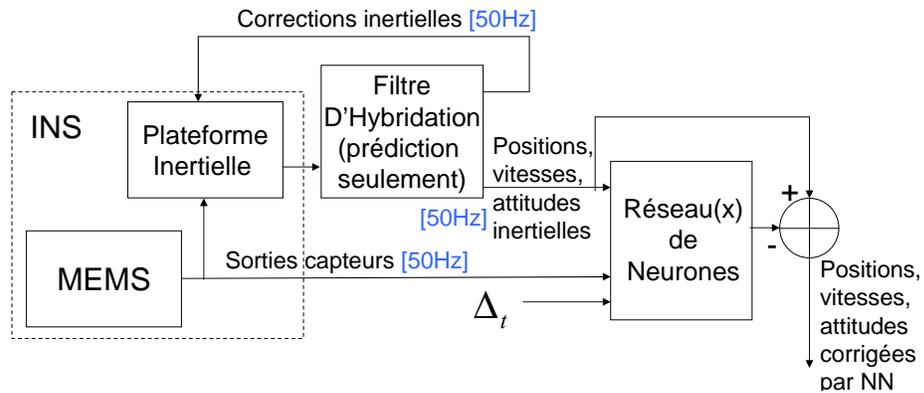


FIGURE 5.9 – Architecture en phase de restitution (correction des estimées inertielles)

5.4.1 Cas simple de trajectoires invariantes dans le temps

Cette première série de trajectoires propose des cas favorables à l'utilisation du réseau de neurones. En effet, la perte de signal GNSS a lieu dans des conditions déjà rencontrées durant la phase d'apprentissage. La phase de restitution devrait donc logiquement permettre au réseau de neurones de venir corriger la dérive du système de navigation en survie, et ceci, pour les différentes trajectoires étudiées (point fixe, vitesse constante, accélération constante).

Point fixe La première étude concerne le cas simple d'un point fixe d'une durée de 600s, la perte du signal GNSS intervenant à la 450^es et durant 30s. Les résultats sont alors représentés autour de la zone d'intérêt (Fig. 5.10) et permettent de constater l'amélioration fournie par le réseau de neurones au système de navigation. En effet, la précision de l'estimation, aussi bien en latitude qu'en longitude, se voit fortement augmentée. Au terme des 30s de perte de signal GNSS, la dérive d'environ 190m (resp. 300m) en latitude (resp. longitude) du système GNSS/MEMS est réduite à seulement 35m (resp. 75m) à l'aide de la correction neuronale.

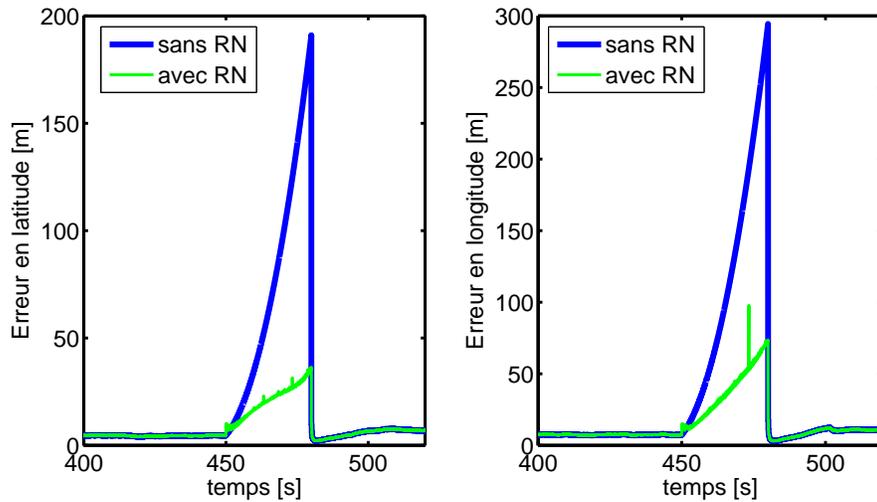


FIGURE 5.10 – Racine carrée de l'erreur quadratique moyenne de position dans le cas d'un point fixe

Vitesse constante Une trajectoire de durée 600s et de vitesse constante ($50ms^{-1}$) est maintenant étudiée. Là encore, les résultats sont présentés autour de la perte de signal GNSS qui se produit toujours à la 450^es (Fig. 5.11). Le réseau de neurones permet à nouveau d'améliorer la précision de l'estimation en latitude et en longitude. Ainsi, la dérive en latitude (resp. longitude) de 120m

(resp. $180m$) obtenue après $30s$ de perte de signal GNSS diminue et est alors égale à $40m$ (resp. $50m$).

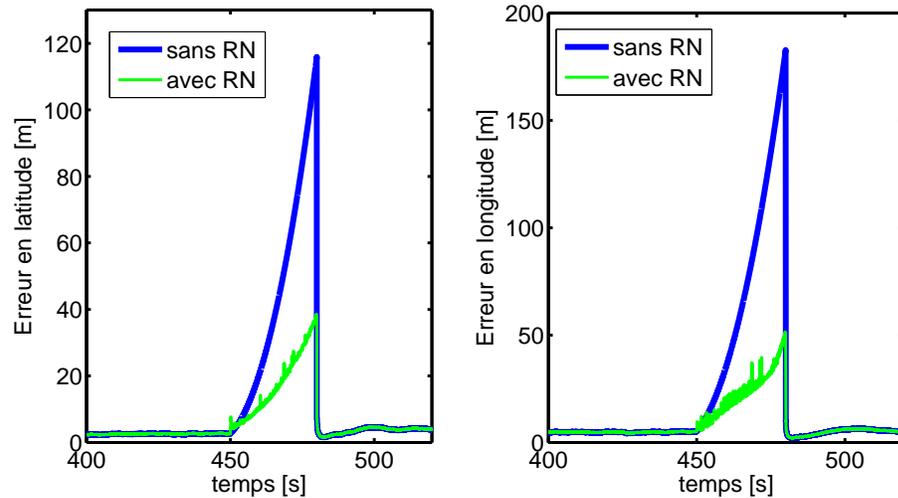


FIGURE 5.11 – Racine carrée de l’erreur quadratique moyenne de position dans le cas d’une trajectoire à vitesse constante

Accélération constante Nous étudions dans cette partie un nouveau cas d’étude concernant une trajectoire à accélération constante ($0.05ms^{-2}$). L’erreur de position durant la perte du signal GNSS est représentée sur la Figure 5.12. La dérive observée au bout de $30s$ de perte de signal GNSS est ramenée de $125m$ à $60m$ en latitude et de $230m$ à $90m$ en longitude grâce à l’utilisation du réseau de neurones. L’amélioration de la précision est donc moins importante que dans les deux cas précédents.

5.4.2 Cas de trajectoires évoluant dans le temps

Cette seconde série de trajectoires propose des cas moins favorables à l’utilisation du réseau de neurones. La perte du signal GNSS intervient désormais dans des conditions que l’apprentissage n’a pas rencontrées. Là encore, plusieurs trajectoires seront étudiées et peuvent se répartir en deux familles : celle où l’accélération est nulle (ligne droite suivie d’un virage, virage suivi d’une ligne droite et succession de deux virages) et celle où la vitesse angulaire est nulle (vitesse constante suivie d’une accélération constante, accélération constante suivie d’une vitesse constante et succession de deux accélérations constantes).

Ligne droite suivie d’un virage Dans cette première étude d’un cas moins favorable à l’usage du réseau de neurones, la trajectoire simulée décrit une ligne

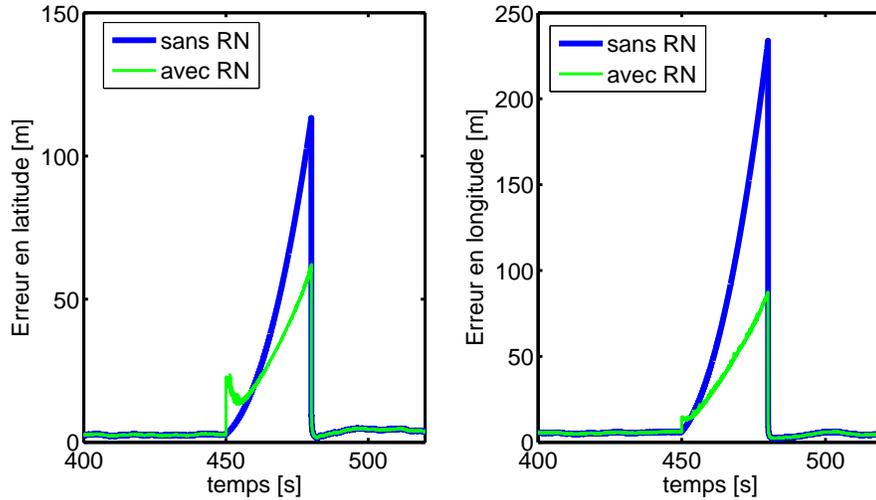


FIGURE 5.12 – Racine carrée de l’erreur quadratique moyenne de position dans le cas d’une trajectoire à accélération constante

droite puis un virage (Fig. 5.13), la phase d’apprentissage se déroulant exclusivement durant la ligne droite tandis que la phase de test n’intervenant que lors du virage. Les résultats sont alors représentés autour de la zone d’intérêt (Fig. 5.14) et permettent de constater l’amélioration fournie par le réseau de neurones au système de navigation. Au terme des 30s de perte de signal GNSS, la dérive d’environ 330m (resp. 225m) en latitude (resp. longitude) du système GNSS/MEMS est réduite à seulement 70m (resp. 175m) à l’aide de la correction neuronale.

Virage suivi d’une ligne droite Un nouveau cas d’étude concernant une trajectoire moins favorable à l’usage du réseau de neurones est ici proposé. La trajectoire est également composée d’une ligne droite et d’un virage mais commence cette fois-ci par ce dernier (Fig. 5.15). La phase d’apprentissage a donc lieu au cours du virage et la perte de signal GNSS permettant de tester le réseau de neurones prend place dans la ligne droite. Le réseau de neurones permet à nouveau d’améliorer la précision de l’estimation en latitude et en longitude (Fig. 5.16). Ainsi, la dérive en latitude (resp. longitude) de 115m (resp. 155m) obtenue après 30s de perte de signal GNSS diminue et est alors égale à 20m (resp. 35m).

Succession de deux virages L’idée est maintenant de simuler deux virages successivement afin de générer une nouvelle trajectoire peu favorable à l’utilisation du réseau de neurones (Fig. 5.17). L’apprentissage du réseau de neurones se déroule durant le premier virage alors que la perte du signal GNSS intervient

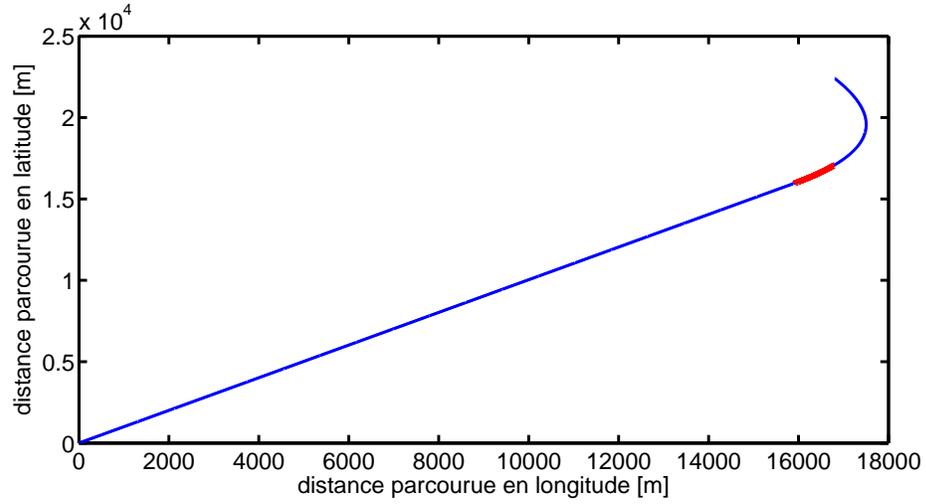


FIGURE 5.13 – Trajectoire simulée (Ligne droite suivie d'un virage)

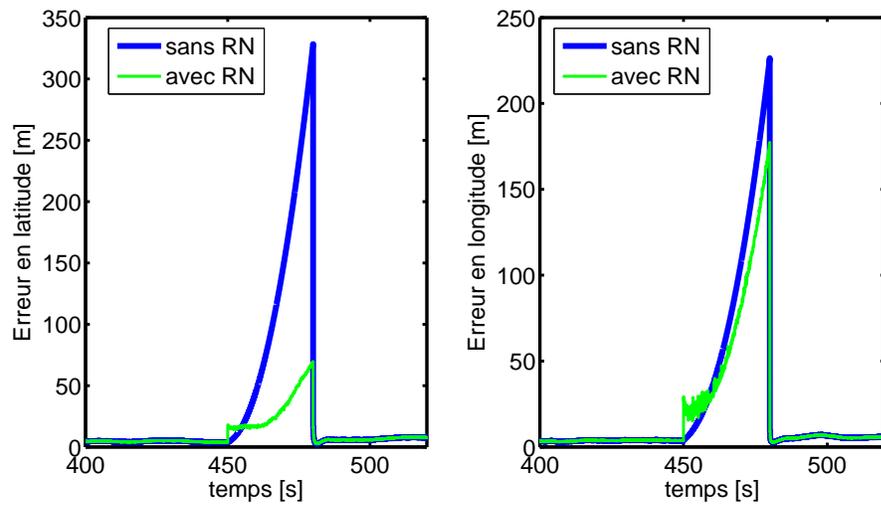


FIGURE 5.14 – Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une ligne droite suivie d'un virage

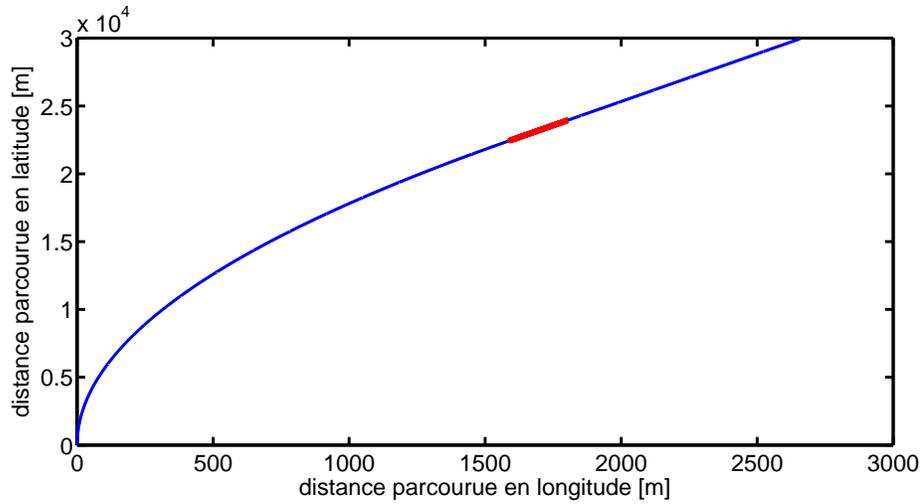


FIGURE 5.15 – Trajectoire simulée (Virage suivi d’une ligne droite)

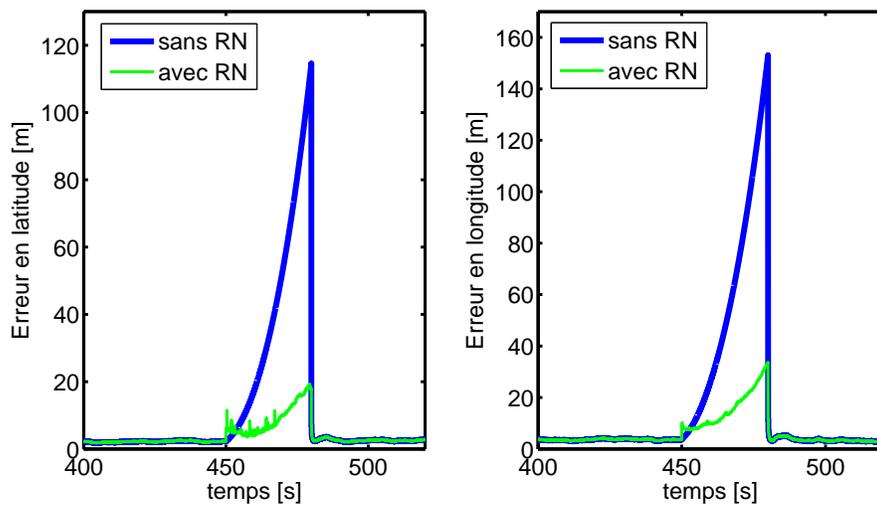


FIGURE 5.16 – Racine carrée de l’erreur quadratique moyenne de position dans le cas d’un virage suivi d’une ligne droite

dans le second. La dérive observée au bout de 30s de perte de signal GNSS est ramenée de 430m à 140m en latitude et de 150m à 120m en longitude grâce à l'utilisation du réseau de neurones (Fig. 5.18). A l'instar de la ligne droite suivie d'un virage, cette étude donne cependant une faible amélioration de la précision en longitude.

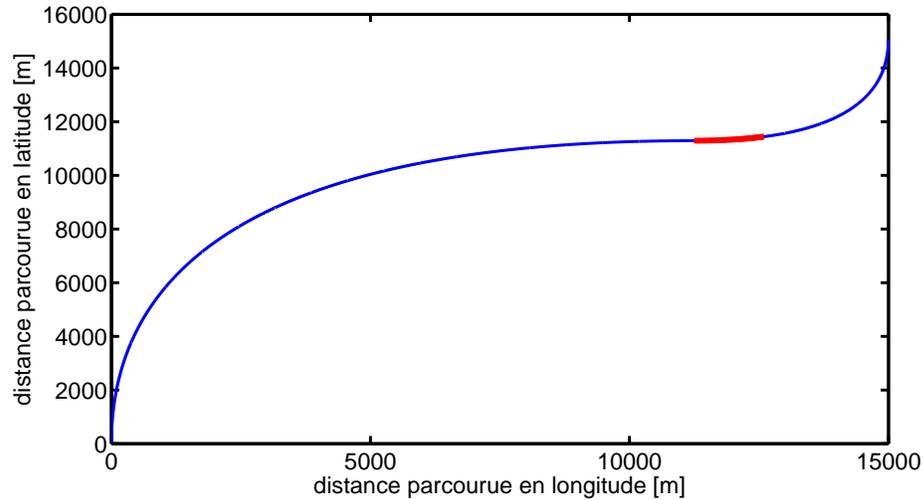


FIGURE 5.17 – Trajectoire simulée (Succession de deux virages)

Vitesse constante suivie d'une accélération constante Cette nouvelle étude se place dans un contexte légèrement différent des précédents. En effet, c'est désormais l'accélération qui sera non nulle et non plus la vitesse angulaire comme dans les cas contenant des virages. Dans cette trajectoire, le mobile suit une ligne droite commençant à vitesse constante (50ms^{-1}) et qui est accéléré constamment (0.1ms^{-2}) par la suite (Fig. 5.19). Les résultats sont alors représentés autour de la zone d'intérêt (Fig. 5.20) et permettent de constater l'amélioration fournie par le réseau de neurones au système de navigation. Au terme des 30s de perte de signal GNSS, la dérive d'environ 160m (resp. 260m) en latitude (resp. longitude) du système GNSS/MEMS est réduite à seulement 50m (resp. 120m) à l'aide de la correction neuronale.

Accélération constante suivie d'une vitesse constante Un nouveau cas d'étude concernant une trajectoire moins favorable à l'usage du réseau de neurones est ici proposé. La trajectoire est également composée d'une ligne droite à vitesse constante (50ms^{-1}) et d'une trajectoire à accélération constante (0.07ms^{-2}) mais commence cette fois-ci par cette dernière (Fig. 5.21). La phase d'apprentissage a donc lieu au cours de l'accélération constante et la perte de signal GNSS permettant de tester le réseau de neurones prend place alors que le mobile est

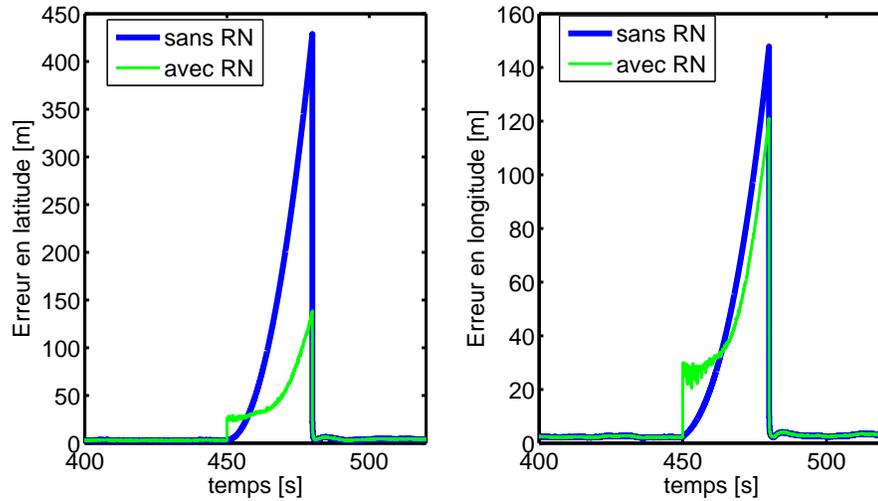


FIGURE 5.18 – Racine carrée de l’erreur quadratique moyenne de position dans le cas d’une succession de deux virages

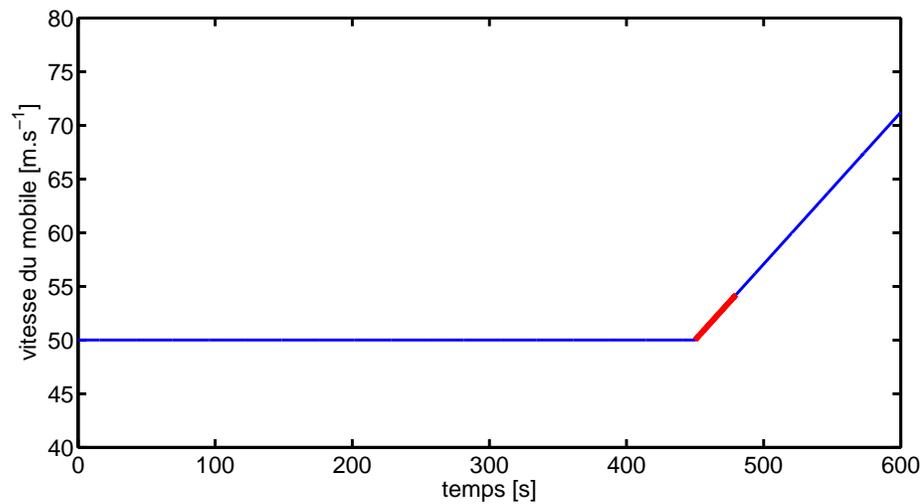


FIGURE 5.19 – Vitesse du mouvement simulé (Vitesse constante suivie d’une accélération constante)

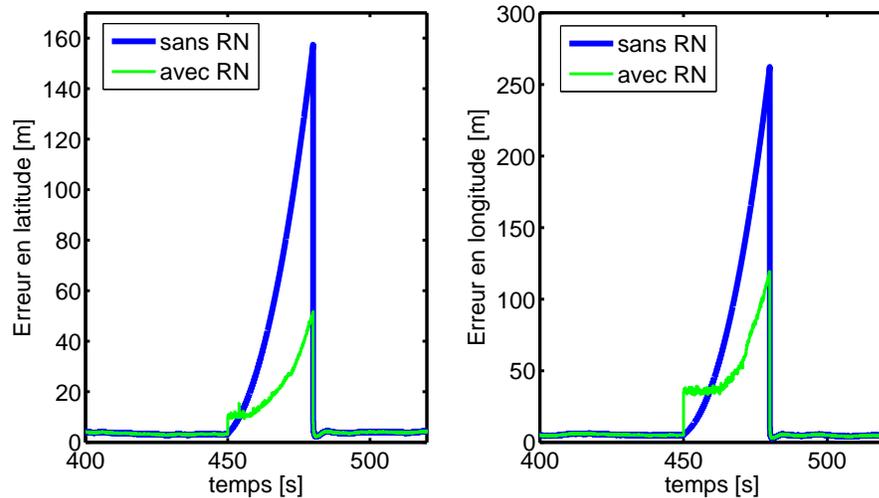


FIGURE 5.20 – Racine carrée de l’erreur quadratique moyenne de position dans le cas d’une trajectoire à vitesse constante suivie d’une accélération constante

à vitesse constante. Le réseau de neurones permet à nouveau d’améliorer la précision de l’estimation en latitude et en longitude (Fig. 5.22). Ainsi, la dérive en latitude (resp. longitude) de $190m$ (resp. $245m$) obtenue après $30s$ de perte de signal GNSS diminue et est alors égale à $70m$ (resp. $95m$).

Succession de deux accélérations constantes L’idée est maintenant de simuler une trajectoire en ligne droite et alternant deux accélérations constantes ($0.015m.s^{-2}$ et $-0.006m.s^{-2}$) afin de toujours générer une nouvelle trajectoire peu favorable à l’utilisation du réseau de neurones (Fig. 5.23). L’apprentissage du réseau de neurones se déroule durant le premier virage alors que la perte du signal GNSS intervient dans le second. La dérive observée au bout de $30s$ de perte de signal GNSS est ramenée de $160m$ à $60m$ en latitude et de $255m$ à $50m$ en longitude grâce à l’utilisation du réseau de neurones (Fig. 5.24). A l’instar de la ligne droite suivie d’un virage, cette étude s’avère donner une faible amélioration de la précision en longitude.

5.4.3 Corrections des différentes estimées inertielles

Cette dernière série de simulations propose de mettre en œuvre la correction, non plus de la position uniquement mais, des différentes estimées inertielles (position, vitesse et attitude). Point qui a peu été évoqué jusqu’ici, l’architecture interne du réseau de neurones est ici au cœur de l’étude. En effet, la correction est calculée en utilisant différents nombres de neurones par couche (10, 15 puis 20) sur deux couches cachées, ou à l’aide de plusieurs réseaux de neurones.

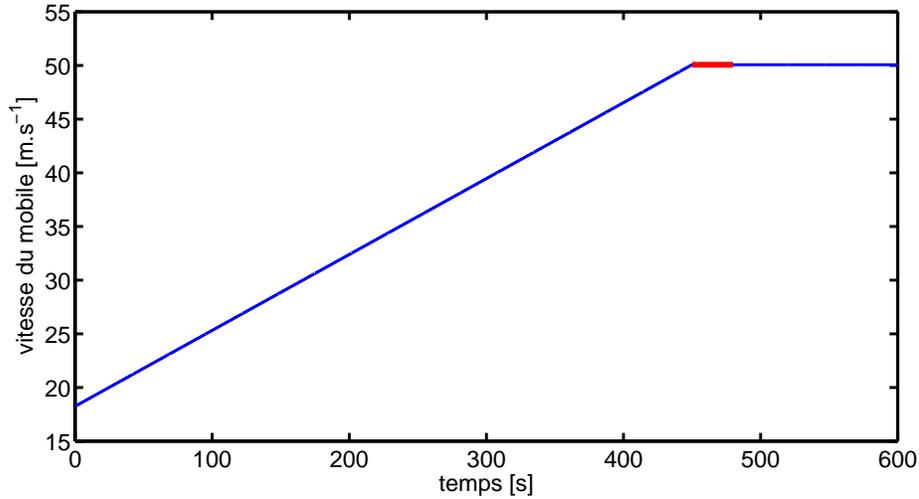


FIGURE 5.21 – Vitesse du mouvement simulé (Accélération constante suivie d'une vitesse constante)

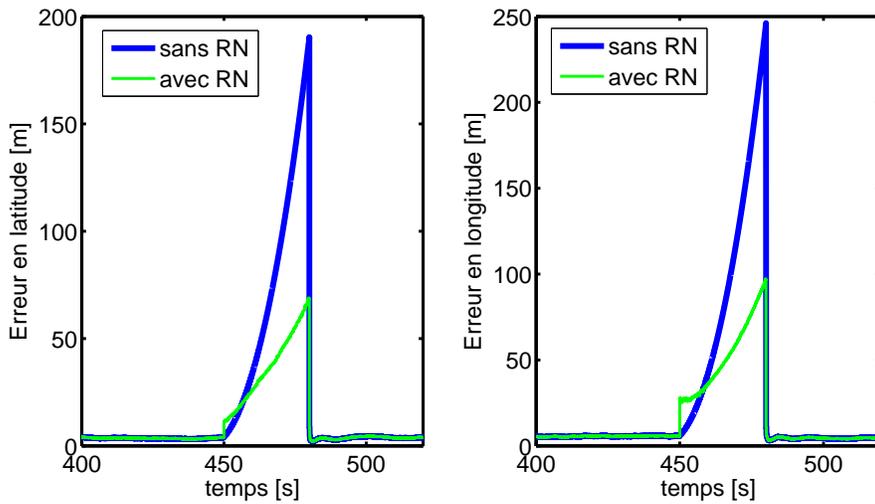


FIGURE 5.22 – Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire à accélération constante suivie d'une vitesse constante

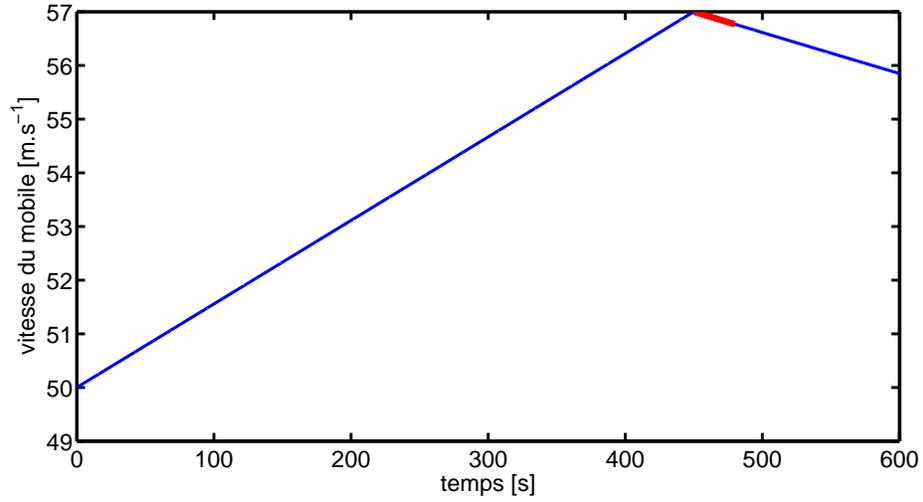


FIGURE 5.23 – Vitesse du mouvement simulé (Succession de deux accélérations constantes)

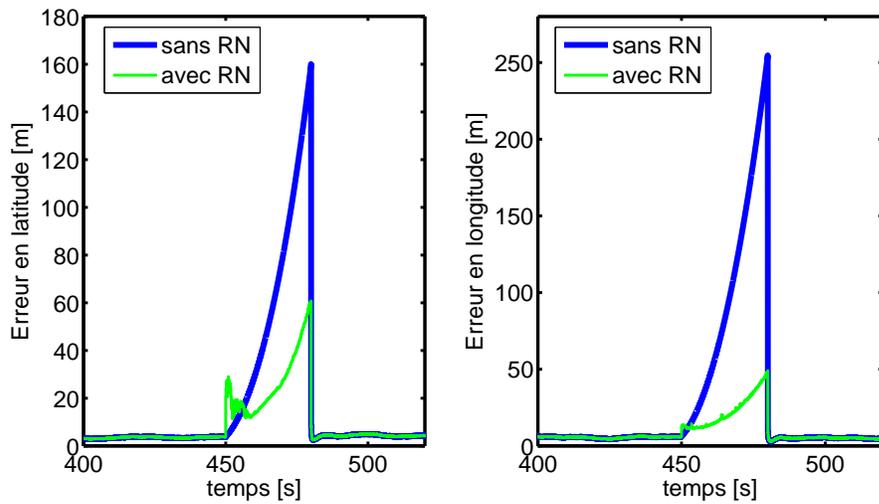


FIGURE 5.24 – Racine carrée de l'erreur quadratique moyenne de position dans le cas d'une trajectoire composée de deux accélérations constantes

Notons ici que toutes ces simulations ont été effectuées dans un cas favorable sur une trajectoire à vitesse constante ($50ms^{-1}$) durant 600s, la perte du signal GNSS intervenant à la 450^es et durant 30s.

Première architecture : 10 neurones par couche Cette première étude utilisant le réseau de neurones pour corriger les différentes estimées inertielles (position, vitesse et attitude) est réalisée à l'aide de la même structure interne du réseau que celle précédemment employée dans le cas de la correction exclusive de la position (dix neurones sur deux couches cachées). Les résultats sont alors représentés autour de la zone d'intérêt (Fig. 5.25, 5.26 et 5.27) et permettent de constater l'amélioration fournie par le réseau de neurones au système de navigation. En terme de position (Fig. 5.25), la précision est améliorée en latitude (resp. longitude), passant de 190m (resp. 195m) à 55m (resp. 90m) au terme des 30s de perte de signal GNSS. Il faut cependant noter la présence d'un biais initial important ($\approx 50m$) dans les deux cas. Ce biais va venir impacter directement les premières secondes de la correction et donc la précision du système de navigation. Ainsi, l'apport en terme de position n'est pas aussi bon que dans le cas de la correction unique de la position. Dans l'idéal, il faudrait réussir à parvenir au même résultat, ce qui sera l'objet des études suivantes. En terme de vitesse (Fig. 5.26), la précision est aussi améliorée selon l'axe n (resp. l'axe e) réduisant la dérive de $12ms^{-1}$ (resp. $8.1ms^{-1}$) à $3.25ms^{-1}$ (resp. $3.8ms^{-1}$). Pour finir, en terme d'attitude (Fig. 5.27), les résultats obtenus avec l'aide du réseau de neurones ne sont pas aussi bons, laissant envisager qu'il n'a pas été capable d'apprendre correctement le comportement demandé. Ceci peut s'expliquer par l'absence de dérive observée en attitude dans le cas d'une perte de signal GNSS. En effet, ce phénomène, présent pour la position et la vitesse, est facilement reconnaissable et donc appréhensible pour le réseau de neurones.

Seconde architecture : 15 neurones par couche Afin de pallier le problème du biais initial important rencontré dans l'étude précédente, nous augmentons le nombre de neurones par couche. Ceci doit permettre d'augmenter la capacité d'apprentissage du réseau et donc de mieux modéliser le phénomène observé lors de pertes de signal GNSS. L'erreur de position est présentée sur la figure 5.28. La dérive observée au bout de 30s de perte de signal GNSS est ramenée de 195m à 112m en latitude et de 330m à 85m en longitude grâce à l'utilisation du réseau de neurones. Bien que légèrement plus faible ($\approx 40m$), le biais initial est toujours présent. Concernant la vitesse (Fig. 5.29), les résultats obtenus avec le réseau de neurones sont toujours plus précis que ceux obtenus sans sa présence. Sur l'axe n (resp. l'axe e), l'erreur quadratique moyenne observée est ramenée de $11.3ms^{-1}$ (resp. $13.4ms^{-1}$) à $4ms^{-1}$ (resp. $3.7ms^{-1}$). A l'instar de l'étude précédente, l'erreur en attitude (Fig. 5.30) n'est pas correctement corrigée par le réseau de neurones.

Troisième architecture : 20 neurones par couche La capacité d'apprentissage d'un réseau comprenant 15 neurones par couche n'étant toujours pas

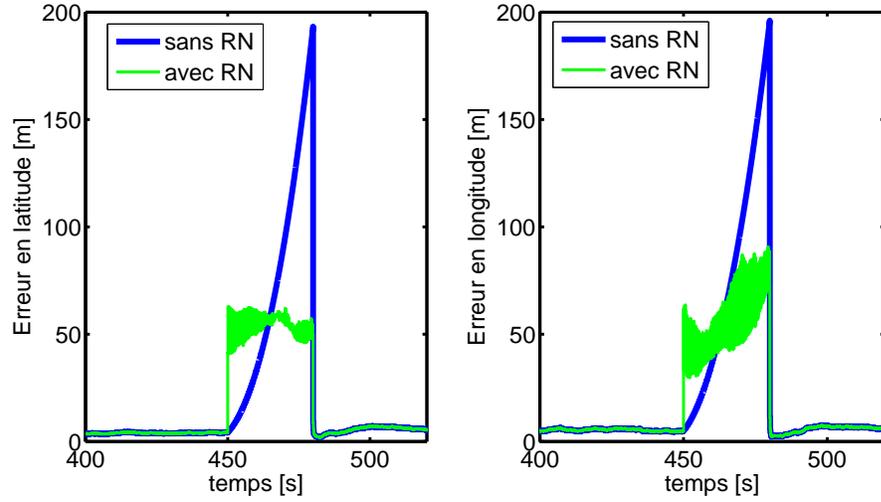


FIGURE 5.25 – Racine carrée de l’erreur quadratique moyenne de position avec 10 neurones par couche

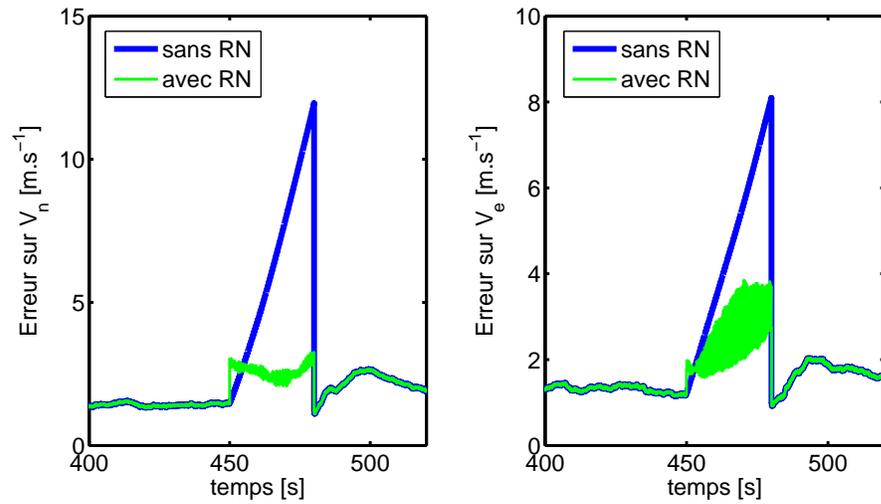


FIGURE 5.26 – Racine carrée de l’erreur quadratique moyenne de vitesse avec 10 neurones par couche

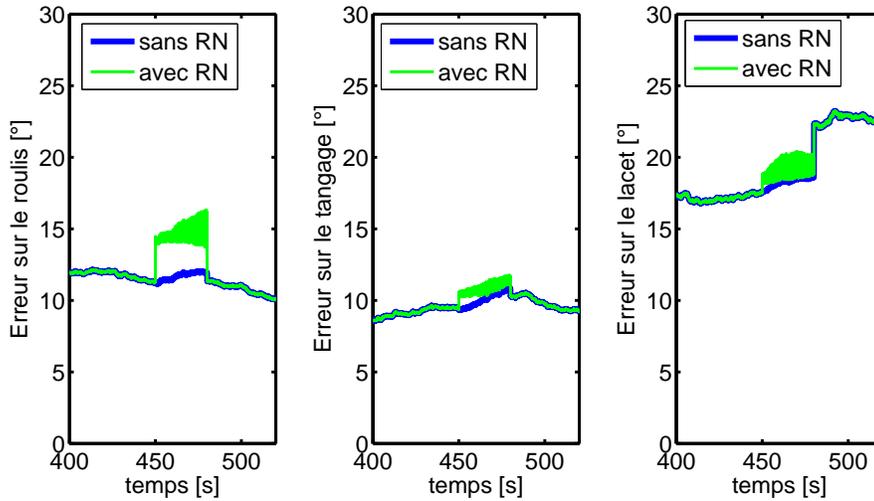


FIGURE 5.27 – Racine carrée de l’erreur quadratique moyenne d’attitude avec 10 neurones par couche

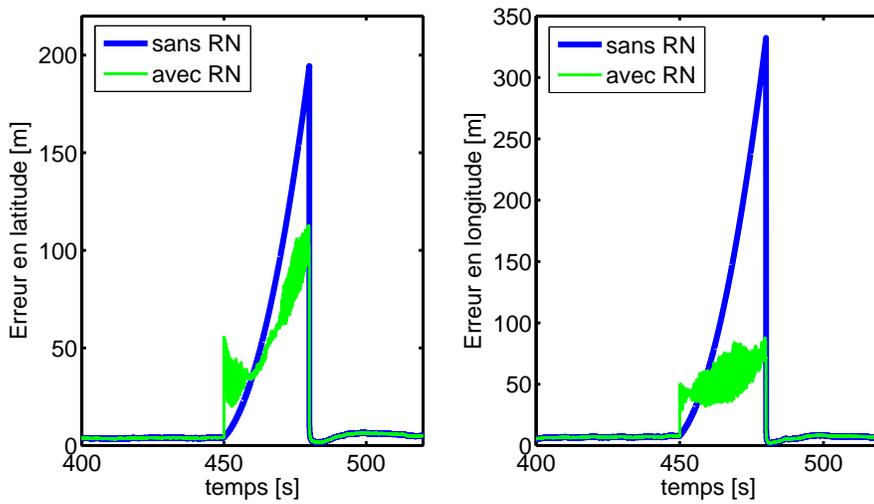


FIGURE 5.28 – Racine carrée de l’erreur quadratique moyenne de position avec 15 neurones par couche

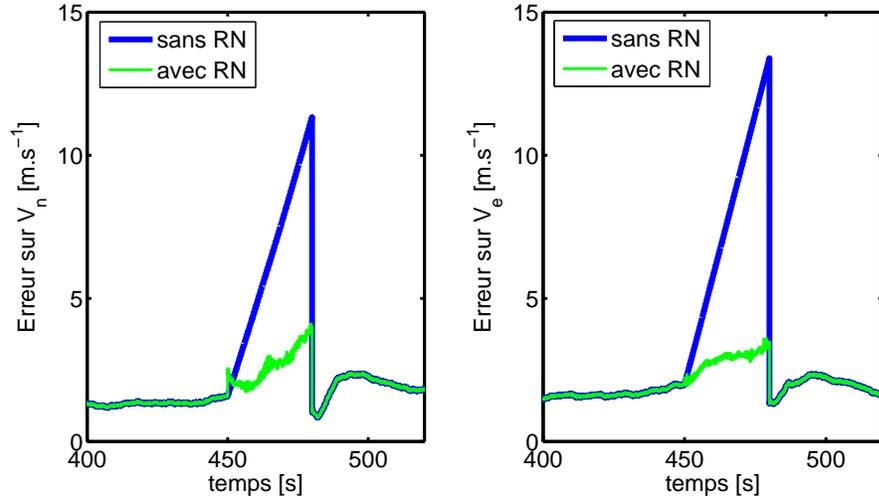


FIGURE 5.29 – Racine carrée de l’erreur quadratique moyenne de vitesse avec 15 neurones par couche

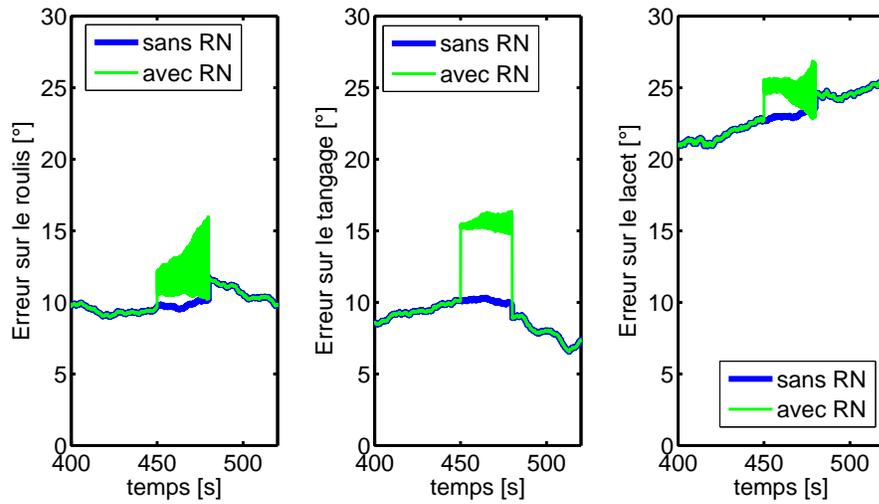


FIGURE 5.30 – Racine carrée de l’erreur quadratique moyenne d’attitude avec 15 neurones par couche

suffisante, le nombre de neurones est à nouveau augmenté passant de 15 à 20. Dans ces conditions, le biais initial constaté (Fig. 5.31) sur la correction de la position semble à nouveau diminuer (20m en latitude et 40m en longitude). Cette diminution n'est cependant pas très rapide et nécessiterait un très grand nombre de neurones par couche pour faire tendre le biais vers 0. La durée d'apprentissage étant directement liée à ce nombre, il est nécessaire de trouver une autre solution pour résoudre ce problème, ce qui sera l'objet de l'étude suivante. Ceci étant dit, les résultats obtenus au cours de cette étude (Fig. 5.31, 5.32 et 5.33) correspondent à ceux qui avait déjà pu être observés au cours des deux précédentes et ne seront donc pas plus détaillés.

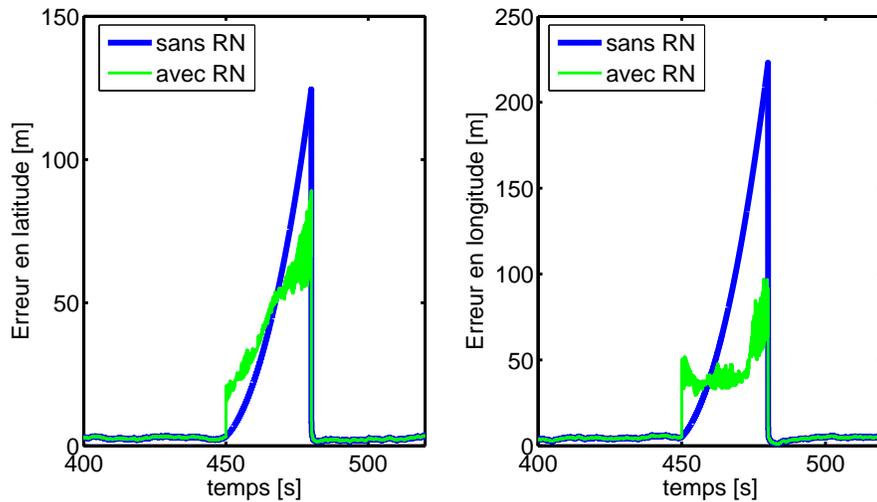


FIGURE 5.31 – Racine carrée de l'erreur quadratique moyenne de position avec 20 neurones par couche

Multi-réseaux Il a été vu dans les études précédentes que la correction des différentes estimées inertielles (position, vitesse et attitude) dans un seul et même réseau de neurones conduisait à des erreurs plus importantes en position que dans le cas de la correction exclusive de la position. L'étude réalisée ici va donc répartir la charge d'apprentissage sur plusieurs réseaux de neurones afin d'approcher au mieux les comportements distincts des différentes estimées inertielles. Nous avons ainsi mis en œuvre trois réseaux de neurones, le premier dédié à la correction de la position, le second dédié à la correction de la vitesse et le dernier dédié à la correction de l'attitude. Les résultats sont alors représentés autour de la zone d'intérêt (Fig. 5.34, 5.35 et 5.36) et permettent de constater l'amélioration fournie par le réseau de neurones au système de navigation. La précision de l'estimation, aussi bien en latitude qu'en longitude, se voit fortement augmentée. Au terme des 30s de perte de signal GNSS, la dérive d'environ 120m

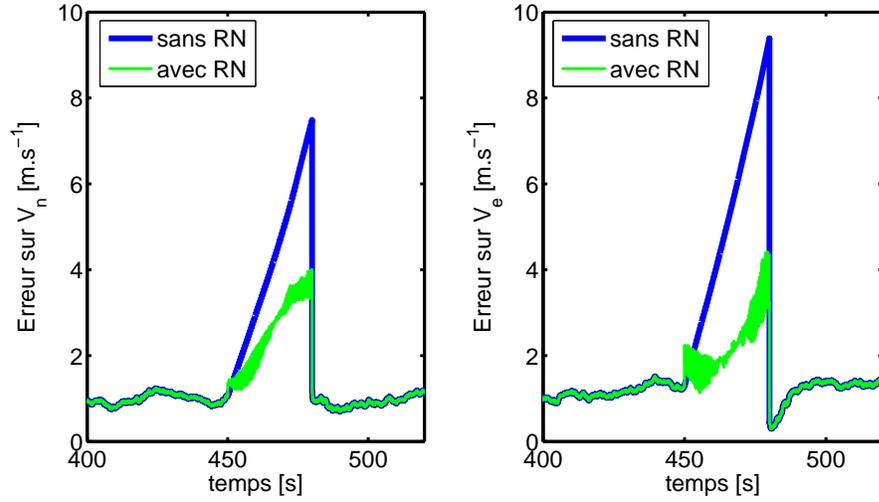


FIGURE 5.32 – Racine carrée de l’erreur quadratique moyenne de vitesse avec 20 neurones par couche

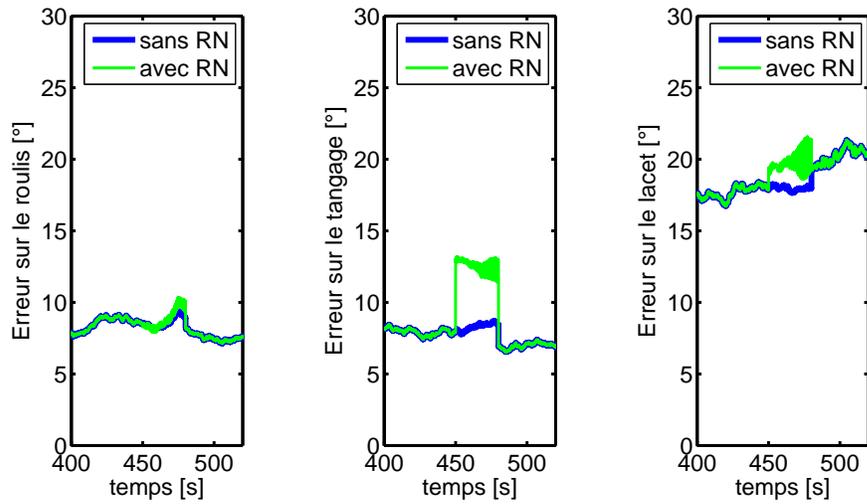


FIGURE 5.33 – Racine carrée de l’erreur quadratique moyenne d’attitude avec 20 neurones par couche

(resp. $200m$) en latitude (resp. longitude) du système GNSS/MEMS est réduite à seulement $35m$ (resp. $65m$) à l'aide de la correction neuronale. Ces résultats correspondent à ceux obtenus dans le cas de la correction unique de la position grâce, notamment, à l'absence de biais initial. En terme de vitesse (Fig. 5.35), la précision est améliorée passant de $7ms^{-1}$ (resp. $8ms^{-1}$) selon l'axe n (resp. l'axe n) à $2.3ms^{-1}$ (resp. $2.2ms^{-1}$). Pour ce qui est de l'attitude (Fig. 5.36), les résultats obtenus avec l'aide du réseau de neurones ne sont toujours pas correctement corrigés, confirmant l'idée que la méthode proposée ne permet pas en l'état d'apprendre correctement le comportement demandé.

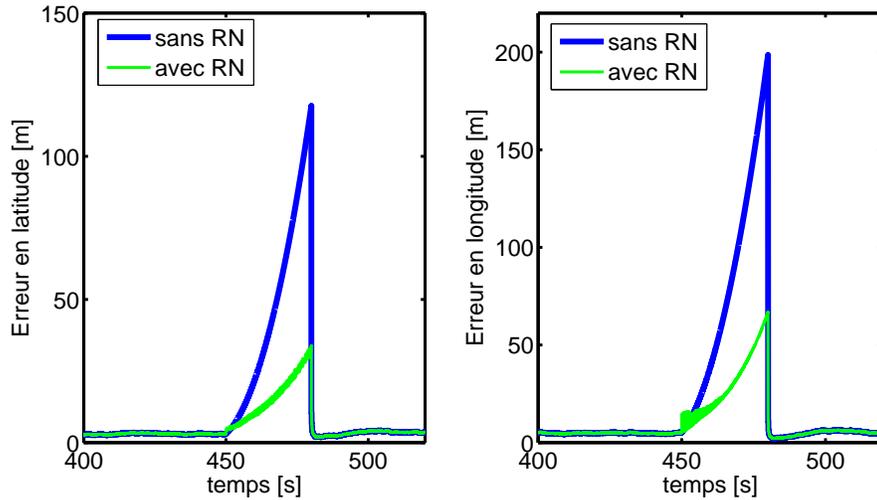


FIGURE 5.34 – Racine carrée de l'erreur quadratique moyenne de position avec un réseau dédié

5.5 Conclusion

Ce chapitre a permis de présenter une mise en œuvre de réseaux de neurones, dans le cas de la navigation hybride GNSS/MEMS, ayant pour objectif d'aider le système de navigation lors de pertes de signal GNSS. L'approche envisagée emploie un réseau de neurones qui utilise le passé de la trajectoire à travers un apprentissage en ligne et ne nécessitant pas d'apprentissage ad hoc [DCTL09]. Différentes structures corrigeant respectivement la position uniquement ainsi que l'ensemble des estimées inertielles (position, vitesse et attitude) ont été proposées et testées à travers un grand nombre de simulations. Les résultats présentés ont permis de valider la méthode proposée en améliorant la précision de l'estimation réalisée sur la position et la vitesse. Malheureusement, ces résultats ne se confirment pas en ce qui concerne l'attitude, qu'il faudrait alors corriger à l'aide d'une autre stratégie.

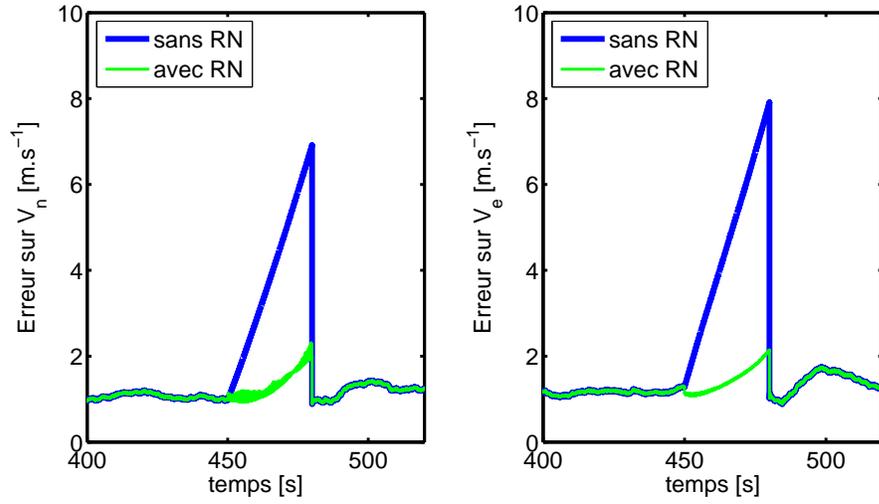


FIGURE 5.35 – Racine carrée de l’erreur quadratique moyenne de vitesse avec un réseau dédié

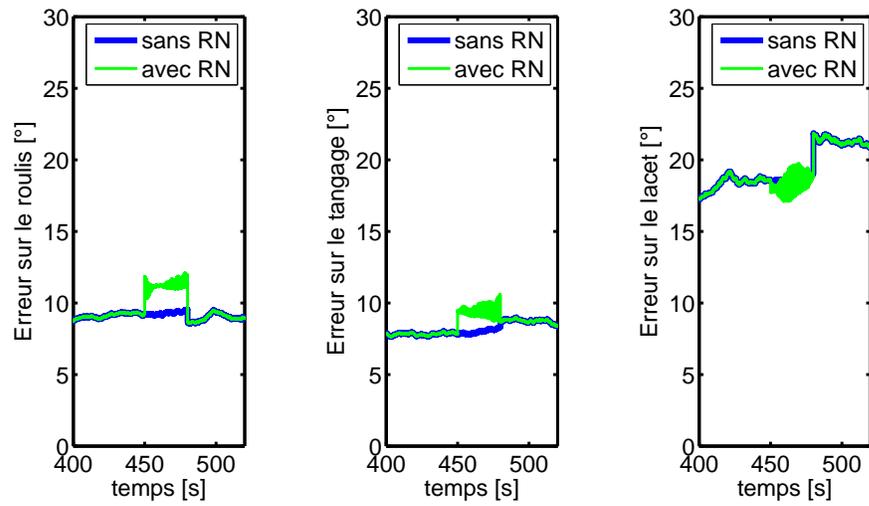


FIGURE 5.36 – Racine carrée de l’erreur quadratique moyenne d’attitude avec un réseau dédié

Conclusion

Chapitre 6

Conclusion

Résumé

Cette thèse a pour objet de proposer des méthodes permettant la réalisation d'une navigation hybride GNSS/MEMS. Pour ce faire, les études menées se décomposent en deux parties distinctes mais visant le même objectif : d'une part, le développement d'un ensemble de traitement capteurs cherchant à améliorer la mesure elle-même et d'autre part, l'optimisation des algorithmes d'hybridation pour les capteurs MEMS.

Le traitement capteur (ou traitement local) consiste en l'estimation de l'accélération vraie (resp. la vitesse angulaire vraie) à partir de la sortie du capteur accélérométrique (resp. gyrométrique). Ce traitement a été réalisé en deux sous-étapes :

- Le calibrage qui consiste en l'identification du système non-linéaire connaissant ses entrées et ses sorties. Les relations entrant en jeu dans le modèle étant linéaires vis-à-vis des paramètres, on peut alors résoudre cette partie du problème par l'estimateur des moindres carrés (après extension du vecteur comprenant les entrées afin qu'il comporte les non linéarités).
- L'inversion du modèle qui a pour but d'estimer les entrées du modèle connaissant ses sorties et effectuant également l'estimation de ses paramètres durant l'étape de calibrage. Après formalisation de ce problème sous forme d'un modèle dynamique, la résolution se fera à l'aide d'algorithmes issus de la théorie du filtre de Kalman ou du filtre particulaire.

Il est intéressant de noter que l'estimation des paramètres du modèle durant l'étape de calibrage ne pourra être que grossière (les dépendances par rapport au temps et à la température n'étant pas prises en compte), le filtre d'hybridation pouvant suivre ces évolutions plus finement.

Les algorithmes d'hybridation (ou traitement global) ont pour but de localiser un mobile dans l'espace connaissant l'estimation de l'accélération et de la vitesse angulaire issue du traitement local ainsi que l'estimation de la position faite par le GPS. Dans la littérature et pour résoudre ce problème, plusieurs filtres ont déjà été utilisés (EKF, UKF, PF, ...). Cette partie peut se

décomposer en deux sous-problèmes qu'il sera possible de résoudre de manières indépendantes :

- Lorsque que les signaux GNSS sont disponibles (cas nominal), le but est d'améliorer les méthodes de navigation hybride GNSS/INS existantes. Dans notre cas, la réflexion a porté sur l'apport de l'ajout d'un état supplémentaire telle la modélisation à l'ordre 2 des biais des capteurs MEMS (non présenté dans ce manuscrit) et sur la fermeture de la boucle de navigation (correction de la centrale inertielle à l'aide des erreurs issues du filtre d'hybridation).
- Dans des scénarii défavorables (multitrajet et masquage des signaux GNSS), la qualité des capteurs MEMS ne permet pas d'obtenir des résultats de navigation satisfaisants. L'idée développée ici consiste alors en la mise en œuvre de méthodes d'apprentissage. Un algorithme basé sur un réseau de neurones a ainsi été développé. Durant les phases où le GNSS est disponible, cet algorithme permet d'apprendre l'erreur commise par la centrale inertielle en mode survie par rapport au résultat de navigation hybride. Le réseau de neurones ainsi appris fournira alors cet élément de correction en cas de perte de l'information GNSS.

Ces différentes méthodes sont alors testées sur différentes trajectoires simulées et des résultats statistiques sont présentés et permettent d'évaluer l'apport respectif de ces techniques, que ce soit en terme de traitement capteur ou de résultat de navigation.

Conclusions

Plusieurs conclusions peuvent être tirées des travaux réalisés dans le cadre de cette thèse. Ces dernières sont présentées, ici, dans l'ordre de lecture du présent manuscrit :

1. L'estimation des paramètres des modèles accélérométrique et gyrométrique à l'aide de la méthode proposée dans cette étude s'avère plus ou moins précise selon la nature du paramètre. Ainsi, l'estimation réalisée sur les biais et les erreurs d'alignement est meilleure que celle effectuée sur les facteurs d'échelles, les erreurs de non-linéarité des accéléromètres et la sensibilité aux accélérations des gyromètres.
2. L'étape d'inversion des modèles MEMS à l'aide d'une résolution d'un modèle dynamique permet d'obtenir une estimation des accélérations et des vitesses angulaires ne dépendant plus des modèles accélérométrique et gyrométrique. Sans contrainte sur l'équation d'état du modèle dynamique, la précision de l'estimation est alors directement liée au bruit additif des capteurs MEMS.
3. Toujours à propos de la méthode d'inversion du modèle, il est possible de réduire encore l'erreur d'estimation des accélérations et des vitesses angulaires en diminuant la variance du bruit d'état du modèle dynamique et ainsi faire plus confiance à l'équation d'état.

4. En terme de navigation et dans le cas nominal (présence du signal GNSS), la méthodologie d'inversion du modèle des capteurs MEMS proposée permet de faire tendre les résultats du système hybride GNSS/MEMS vers ceux obtenus par le système classique GNSS/INS.
5. Durant une perte de signal GNSS, l'inversion du modèle ne permet pas d'améliorer de façon suffisante l'estimation de la position. La robustesse face à ces cas de masquage des satellites, caractéristique essentielle de la navigation GNSS/INS, n'est alors pas retrouvée pour le système hybride GNSS/MEMS. Il est nécessaire de mettre en œuvre des techniques spécifiques pour traiter cette perte de signal GNSS.
6. Le principe d'un apprentissage par réseau de neurones de l'erreur de position à l'aide du passé de la trajectoire a été validé sur la simulation de trajectoires simples et invariantes dans le temps (point fixe, vitesse constante et accélération constante). Il est intéressant de noter que plus la trajectoire est simple, plus l'amélioration est importante.
7. L'étude de trajectoires évoluant dans le temps a permis de montrer que l'apprentissage peut être réalisé dans des conditions différentes de celles de restitution. Ce point s'avère critique pour une mise en œuvre réelle car aucune présupposition ne peut alors être prise sur la trajectoire.
8. La nature de la trajectoire influe sur l'amélioration apportée par le réseau de neurones à l'estimation de la position durant une perte de signal GNSS. En effet, les études menées ont permis de constater que les cas durant lesquels l'accélération évolue (vitesse constante suivie d'accélération constante, accélération constante suivie de vitesse constante et succession de deux accélérations constantes) sont mieux corrigés par le réseau de neurones que les cas durant lesquels la vitesse angulaire évolue (ligne droite suivie d'un virage, virage suivi d'une ligne droite et succession de deux virages).
9. De la même manière que précédemment, le principe d'un apprentissage par réseau de neurones des erreurs de position, de vitesse et d'attitude à l'aide du passé de la trajectoire n'a pu être que partiellement validé. En effet, grâce à la correction apportée par le réseau de neurones, la précision des estimations de la position et de la vitesse est augmentée, ce qui n'est pas le cas de celle concernant l'attitude. Il sera donc nécessaire d'envisager une solution alternative afin que l'estimation de l'attitude puisse être également correctement corrigée.
10. De plus, les résultats obtenus au cours de l'étude menée sur la correction des différentes estimées inertielles prouvent qu'il est préférable d'utiliser des réseaux différents pour la position, la vitesse et l'attitude car l'amélioration apportée est alors sensiblement plus importante.

Ces conclusions, basées sur les résultats de tests réalisés au cours de la thèse, permettent d'imaginer un certain nombre de perspectives, qui font l'objet de la section suivante.

Perspectives et recommandations

Basées sur les travaux réalisés durant cette thèse, plusieurs perspectives et recommandations pour des travaux futurs peuvent être proposées. Comme pour la section précédente, ces dernières sont présentées dans l'ordre de lecture du présent manuscrit :

1. La méthodologie employée pour inverser le modèle des capteurs MEMS utilise une modélisation dynamique pour laquelle il est nécessaire de fixer la variance du bruit d'état. En utilisant la connaissance du passé de la trajectoire, une étude pourrait être réalisée afin que la modélisation dynamique soit la plus fine possible, ce qui permettrait d'éliminer une partie du bruit de mesure des capteurs MEMS.
2. Le traitement capteur peut être réalisé par un grand nombre de méthodes plus ou moins complémentaires (décomposition en ondelettes, décomposition sur une base orthogonale, ...). La particularité du modèle MEMS provenant de son aspect non-linéaire, une étude future pourrait être menée sur l'expression de cette dernière en vue de l'inverser (estimation par fonction noyau, estimation par série orthogonale, ...).
3. L'architecture des réseaux de neurones utilisés dans cette thèse n'a pas fait l'objet d'une optimisation. Une étude exhaustive pourrait alors être réalisée en ce sens, à l'aide de méthodes d'élagage type algorithme de Engelbrecht [Eng01].
4. De même, l'importance du choix de la fonction non-linéaire de chaque neurone, ou fonction d'activation, n'a pas non plus été étudiée avec attention. Si une fonction venait à mieux correspondre au problème d'apprentissage posé au réseau de neurones, les performances de navigation pourraient alors être légèrement améliorées.
5. Toujours concernant l'utilisation de réseaux de neurones pour corriger les erreurs de navigation durant une perte de signal GNSS, une base d'apprentissage optimale, réduisant à la fois le temps nécessaire à l'apprentissage et l'erreur entre la sortie du réseau et la cible à atteindre, pourrait être définie.
6. Bien qu'un peu plus éloignée du sujet de cette thèse, une étude pourrait également être menée en prenant en compte d'autres capteurs que les accéléromètres et les gyromètres de type MEMS. En effet, dans cette catégorie de capteurs bas coût, il existe également des magnétomètres et des baro-altimètres permettant respectivement de limiter les erreurs de cap et d'altitude.

Les travaux supplémentaires recommandés concernant les réseaux de neurones devraient conduire à de légères améliorations des performances qui sont présentées à travers les résultats exposés dans cette thèse.

Bibliographie

Bibliographie

- [AH04] W. Abdel-Hamid. An ANFIS-based modeling of thermal drift of MEMS-based inertial sensors. *Proc. of ION-GNSS*, 2004.
- [AH05] W. Abdel-Hamid. *Accuracy Enhancement of Integrated MEMS-IMU/GPS Systems for Land Vehicular Navigation Applications*. PhD thesis, University of Calgary, 2005.
- [AHNES07] W. Abdel-Hamid, A. Noureldin, and N. El-Sheimy. Adaptive fuzzy prediction of low-cost inertial-based positioning errors. *IEEE Transactions on Fuzzy Systems*, 15(3) :519–529, June 2007.
- [AM05] B. D. O. Anderson and J. B. Moore. *Optimal Filtering (Dover Books on Engineering)*. Dover Publications, 2005.
- [Ayl01] A. Ayliffe. The development of airborne dead reckoning. part I : Before 1940 - finding the wind. *The Journal of Navigation*, 54 :223–233, May 2001.
- [Bri71] K. R. Britting. *Inertial Navigation Systems Analysis*. Wiley-Interscience, New York, first edition, 1971.
- [Bro02] O. S. Brown. *One Day Celestial Navigation for Offshore Sailing*. Washington Book Distributors, Washington, second edition, 2002.
- [Chi04] K.-W. Chiang. *INS/GPS Integration Using Neural Network for Land Vehicular Navigation Applications*. PhD thesis, University of Calgary, 2004.
- [Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4) :303–314, December 1989.
- [DCTL08] J.-R. De Boer, V. Calmettes, J.-Y. Tournet, and B. Lesot. Performance improvement of integrated MEMS/GNSS systems. *Proc. of ION-GNSS*, 2008.
- [DCTL09] J.-R. De Boer, V. Calmettes, J.-Y. Tournet, and B. Lesot. Outage mitigation for GNSS/MEMS navigation using neural networks. *Proc. of EUSIPCO*, 2009.
- [DDG01] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science)*. Springer, first edition, June 2001.

- [DG04] R. Dorobantu and C. Gerlach. Investigation of a navigation-grade rlg simu type inav-rqh. Technical report, Institut für Astronomische und Physikalische Geodäsie, Technische Universität München, 2004.
- [DGA00] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3) :197–208, 2000.
- [Dou98] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical report, Dept. of Engineering, University of Cambridge, 1998.
- [DT08] F. Duvallet and A.D. Tews. WIFI position estimation in industrial environments using Gaussian processes. *Proc. of IROS*, 2008.
- [EDER03] M. El-Diasty and A. El-Rabbany. Adaptive noise reduction model for MEMS-based inertial sensors. *Proc. of ION-NTM*, 2003.
- [Eng01] A. P. Engelbrecht. A new pruning heuristic based on variance analysis of sensitivity information. *IEEE Transactions on Neural Networks*, 12(6) :1386–1399, 2001.
- [ESAHL04] N. El-Sheimy, W. Abdel-Hamid, and G. Lachapelle. An adaptive neuro-fuzzy model for bridging GPS outages in MEMS-IMU/GPS land vehicle navigation. *Proc. of ION-GNSS*, 2004.
- [ESN04] N. El-Sheimy and S. Nassar. Wavelet de-noising for IMU alignment. *IEEE A&E Systems Magazine*, pages 32–39, October 2004.
- [ESNSN06] N. El-Sheimy, S. Nassar, E.-H. Shin, and X. Niu. Analysis of various Kalman filter algorithms with different inertial systems for inertial navigation system/global positioning system integrated systems. *Canadian Aeronautics and Space Journal*, 52(2), June 2006.
- [FB98] J. Farrell and M. Barth. *The Global Positioning System & Inertial Navigation*. McGraw-Hill, New York, 1998.
- [FWB05] W. S. Flenniken, J. H. Wall, and D. M. Bevely. Characterization of various IMU error sources and the effect on navigation performance. *Proc. of of ION-GNSS*, 2005.
- [Gag98] J.-J. Gagnepain. La variance de B. Picinbono. *Traitement du Signal*, 15(6) :477–482, 1998.
- [GESS08] C. Goodall, N. El-Sheimy, and Z. Syed. On-line tuning of an extended Kalman filter for INS/GPS navigation applications. *Proc. of of ION-GNSS*, 2008.
- [Gir05] A. Giremus. *Apports des techniques de filtrage non linéaire pour la navigation avec les systèmes de navigation inertiels et le GPS*. PhD thesis, Ecole Nationale Supérieure de l’Aéronautique et de l’Espace, Université de Toulouse, 2005.

- [GSES06] C. Goodall, Z. Syed, and N. El-Sheimy. Improving INS/GPS navigation accuracy through compensation of Kalman filter errors. *Proc. of IEEE Vehicle and Technology Conference*, 2006.
- [GWA07] M. Grewal, L. Weill, and A. Andrews. *Global Positioning System, Inertial Navigation, and Integration*. John Wiley & Sons, New York, second edition, 2007.
- [Hay99] S. Haykin. *Neural Networks : A Comprehensive Foundation*. Prentice Hall, New York, second edition, 1999.
- [HMS97] M. Hellebrandt, R. Mathar, and M. Scheibenbogen. Estimating position and velocity of mobiles in a cellular radio network. *IEEE Transactions on Vehicular Technology*, 46(1) :65–71, February 1997.
- [HN80] M. Hata and T. Nagatsu. Mobile location using signal strength measurements in a cellular system. *IEEE Transactions on Vehicular Technology*, 29(2) :245–252, May 1980.
- [Hop82] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *PNAS*, 79(8) :2554–2558, april 1982.
- [HSS03] A. Howard, S. Siddiqi, and G. S. Sukhatme. An experimental study of localization using wireless ethernet. *Proc. of FSR*, 2003.
- [Jaz70] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [JU96] S. J. Julier and J. K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Dept. of Engineering Science, University of Oxford, 1996.
- [JU97] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *Proc. of Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 3068 :182–193, 1997.
- [Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82 :35–45, 1960.
- [KB61] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME. Series D, Journal of Basic Engineering*, 83 :95–107, 1961.
- [Kin98] A. D. King. Inertial navigation - forty years of evolution. *GEC Review*, 13(3) :140–149, 1998.
- [MP43] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4) :115–133, December 1943.
- [MP69] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, 1969.

- [Nas03] S. Nassar. *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications*. PhD thesis, University of Calgary, 2003.
- [NNAES06] S. Nassar, X. Niu, P. Aggarwal, and N. El-Sheimy. INS/GPS sensitivity analysis using different Kalman filter approaches. *Proc. of ION-NTM*, 2006.
- [PDGC08] B. Priot, J.-R. De Boer, R. Guidoux, and V. Calmettes. Performance assessment of integrated MEMS/GNSS systems. *Proc. of ENC-GNSS*, 2008.
- [Pet06] S. Pethel. Test and evaluation of high performance micro electro-mechanical system based inertial measurement units. *Proc. of IEEE/ION-PLAN*, 2006.
- [RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323 :533–536, October 1986.
- [Rog00] R. M. Rogers. *Applied Mathematics in integrated Navigation Systems*. American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, USA, 2000.
- [Ros58] F. Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brains. *Psychological Review*, 65 :386–408, June 1958.
- [SAG⁺07] Z.F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy. A new multi-position calibration method for MEMS inertial navigation systems. *Measurement Science and Technology*, 18(7) :1897–1907, July 2007.
- [SBS99] J. Skaloud, A. M. Bruton, and K. P. Schwarz. Detection and filtering of short-term ($1/f^\gamma$) noise in inertial sensors. *Journal of The Institute of Navigation*, 46(2) :97–107, Summer 1999.
- [SCKJ07] M. Spangenberg, V. Calmettes, D. Kubrak, and O. Julien. Optimized low-cost HSGPS/IMU/WSS land vehicle navigation system for urban navigation. *Proc. of ION-GNSS*, 2007.
- [SCT07] M. Spangenberg, V. Calmettes, and J.-Y. Tournier. Fusion of GPS, INS and odometric data for automotive navigation. *Proc. of EUSIPCO*, 2007.
- [SNES05] E.-H. Shin, X. Niu, and N. El-Sheimy. Performance comparison of the extended and the unscented kalman filter for integrated GPS and MEMS-based inertial systems. *Proc. of ION-NTM*, 2005.
- [SO06] R. Sutherland and L. M. Overstreet. Characterization of non-linear error terms for vibrating beam accelerometers. *Proc. of IEEE/ION-PLAN*, 2006.
- [SS09] S. Saeedi and F. Samadzadegan. Vision-aided inertial navigation for pose estimation of aerial vehicles. *Proc. of ION-GNSS*, 2009.

- [Sta06] J.-M. Stauffer. Current capabilities of MEMS capacitive accelerometers in harsh environment. *Proc. of IEEE/ION-PLAN*, 2006.
- [SW00] K. P. Schwarz and M. Wei. INS/GPS integration for geodetic applications : Lecture notes engo 623. Technical report, Dept. of Geomatics Engineering, University of Calgary, 2000.
- [WKD09] J.J. Wang, S. Kodagoda, and G. Dissanayake. Vision aided GPS/INS system for robust land vehicle navigation. *Proc. of ION-GNSS*, 2009.
- [WMM⁺06] J. Wendel, J. Metzger, R. Moenikes, A. Maier, and G. F. Trommer. A performance comparison of tightly coupled GPS/INS navigation systems based on extended and sigma point Kalman filters. *Journal of The Institute of Navigation*, 53(1) :21–31, Spring 2006.
- [WWSW06] J. J. Wang, J. Wang, D. Sinclair, and L. Watts. Designing a neural network for GPS/INS/PL integration. *Proc. of IGNSS Symposium*, 2006.

Annexes

Annexe A

Modélisation complète des MEMS

Dans la littérature, un grand nombre de travaux ont portés sur la modélisation des capteurs accélérométriques et gyrométriques. Basés sur ceux de [FB98], [GWA07], [Rog00] et [SW00], le rapport technique de Dorobantu et Gerlach [DG04] proposent un modèle faisant notamment intervenir la température.

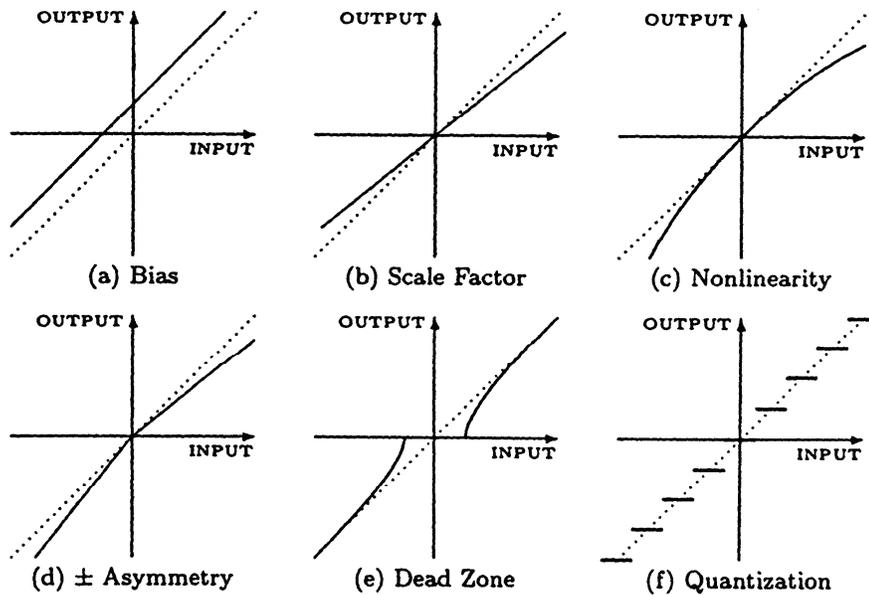


FIGURE A.1 – Vue d'ensemble des erreurs liées aux systèmes inertiels [GWA07]

Erreurs accélérométriques

Pour rappel, l'accéléromètre est modélisé de la manière suivante

$$S_a = (I - \delta_{SFa} - \delta_{MAa})(\Gamma - \delta_{Ba} - \delta_{NLa} - v_a)$$

$$\text{où } \begin{cases} \Gamma & \text{est le vecteur accélérations } \in \mathbb{R}^3, \\ S_a & \text{est le vecteur sorties accélérométriques } \in \mathbb{R}^3, \\ \delta_{Ba} & \text{est le vecteur biais accélérométriques } \in \mathbb{R}^3, \\ \delta_{SFa} & \text{est la matrice diagonale } 3 \times 3 \text{ des erreurs de facteurs d'échelles,} \\ \delta_{NLa} & \text{est le vecteur contenant les erreurs de non-linéarité } \in \mathbb{R}^3, \\ \delta_{MAa} & \text{est la matrice hors-diagonale } 3 \times 3 \text{ des erreurs d'alignements,} \\ v_a & \text{est le vecteur des bruits accélérométriques } \in \mathbb{R}^3. \end{cases}$$

Ces termes peuvent être plus précisément définis par les équations A.1, A.2 et A.3.

Biais δ_{Ba}

$$\delta_{Ba} = \delta_{Ba,biais} + \delta_{Ba,derive} + \delta_{Ba,temp} \quad (\text{A.1})$$

$$\text{où } \begin{cases} \delta_{Ba,biais} & \text{est un biais composé d'un offset constant et d'un offset} \\ & \text{différent à chaque mise sous tension,} \\ \delta_{Ba,derive} & \text{est une dérive composée notamment d'une marche aléatoire,} \\ & \text{d'un processus de Gauss Markov d'ordre 1 et d'un bruit de} \\ & \text{flicker,} \\ \delta_{Ba,temp} & = \sum_{k=1}^4 \delta_{Ba,temp(k)}(T - T_{cal})^k \\ & \text{représente la dépendance du biais } \delta_{Ba} \text{ à la température } T, \\ & T_{cal} \text{ étant la température de calibration.} \end{cases}$$

Facteur d'échelle δ_{SFa}

$$\delta_{SFa} = \delta_{SFa,biais} + \delta_{SFa,derive} + \delta_{SFa,temp} \quad (\text{A.2})$$

$$\text{où } \begin{cases} \delta_{SFa,biais} & \text{est un biais composé d'un offset constant et d'un offset} \\ & \text{différent à chaque mise sous tension,} \\ \delta_{SFa,derive} & \text{est une dérive évoluant avec le temps,} \\ \delta_{SFa,temp} & = \sum_{k=1}^4 \delta_{SFa,temp(k)}(T - T_{cal})^k \\ & \text{représente la dépendance du facteur d'échelle } \delta_{SFa} \text{ à la} \\ & \text{température } T, T_{cal} \text{ étant la température de calibration.} \end{cases}$$

Désalignement δ_{MAa}

$$\delta_{MAa} = \sum_{k=1}^4 \delta_{MAa,temp(k)}(T - T_{cal})^k \quad (\text{A.3})$$

où $\delta_{MAa,temp(k)}$ représente la dépendance du désalignement δ_{MAa} à la température T , T_{cal} étant la température de calibration.

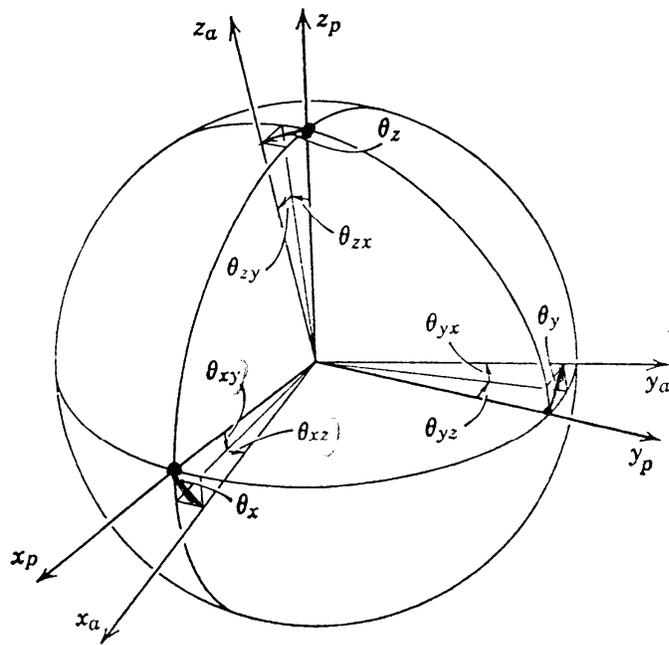


FIGURE A.2 – Description des erreurs d'alignement dans les systèmes inertiels [Bri71]

Erreurs gyrométriques

De même, le modèle mathématique du gyromètre est

$$S_g = (I - \delta_{SFg} - \delta_{MAg})(\Omega - \delta_{Bg} - \delta_{Kg} - v_g)$$

$$\text{où } \begin{cases} \Omega & \text{est le vecteur vitesses angulaires } \in \mathbb{R}^3, \\ S_g & \text{est le vecteur sorties gyrométriques } \in \mathbb{R}^3, \\ \delta_{Bg} & \text{est le vecteur biais gyrométriques } \in \mathbb{R}^3, \\ \delta_{SFg} & \text{est la matrice diagonale } 3 \times 3 \text{ des erreurs de facteurs d'échelles,} \\ \delta_{MAg} & \text{est la matrice hors-diagonale } 3 \times 3 \text{ des erreurs d'alignements,} \\ \delta_{Kg} & \text{est la matrice } 3 \times 3 \text{ de sensibilité aux accélérations,} \\ v_g & \text{est le vecteur des bruits gyrométriques } \in \mathbb{R}^3. \end{cases}$$

Ces termes peuvent être plus précisément définis par les équations A.4 et A.5.

Biais δ_{Bg}

$$\delta_{Bg} = \delta_{Bg,biais} + \delta_{Bg,derive} + \delta_{Bg,temp} \quad (\text{A.4})$$

$$\text{où } \begin{cases} \delta_{Bg,biais} & \text{est un biais composé d'un offset constant et d'un offset} \\ & \text{différent à chaque mise sous tension,} \\ \delta_{Bg,derive} & \text{est une dérive composée notamment d'une marche aléatoire,} \\ & \text{d'un processus de Gauss Markov d'ordre 1 et d'un bruit de} \\ & \text{flicker,} \\ \delta_{Bg,temp} & \text{représente la dépendance du biais } \delta_{Bg} \text{ à la température } T. \end{cases}$$

Facteur d'échelle δ_{SFg}

$$\delta_{SFg} = \delta_{SFg,biais} + \delta_{SFg,derive} + \delta_{SFg,temp} \quad (\text{A.5})$$

$$\text{où } \begin{cases} \delta_{SFg,biais} & \text{est un biais composé d'un offset constant et d'un offset} \\ & \text{différent à chaque mise sous tension,} \\ \delta_{SFg,derive} & \text{est une dérive évoluant avec le temps,} \\ \delta_{SFg,temp} & \text{représente la dépendance du facteur d'échelle } \delta_{SFg} \text{ à la} \\ & \text{température } T. \end{cases}$$

Annexe B

Variance d'Allan

Définition

La variance d'Allan est définie par

$$\sigma_{AL,y}^2(\tau) = \frac{1}{2} \langle (y_{k+1} - \bar{y}_k)^2 \rangle \quad (\text{B.1})$$

$$\text{où } \begin{cases} \bar{y}_k = \frac{1}{\tau} \int_{t_k}^{t_k+\tau} y(t) dt, \\ \langle \bar{y}_k \rangle = \frac{1}{N} \sum_{k=1}^N \bar{y}_k. \end{cases}$$

D'après [Gag98], sous contrainte de stationnarité et d'ergodicité à l'ordre 1 et 2, la variance vraie σ_y^2 peut s'écrire sous la forme

$$\sigma_y^2(\tau) = \langle \bar{y}_k^2 - \bar{y}_k \rangle.$$

Remarquons alors que l'écriture de \bar{y}_k peut être simplifiée et devient alors

$$\begin{aligned} \bar{y}_k &= \frac{1}{\tau} \int_{t_k}^{t_k+\tau} y(t) dt \\ &= \frac{1}{\tau} \int_{-\infty}^{+\infty} \Pi_T(t - t_k - \frac{\tau}{2}) y(t) dt \quad \text{avec } \Pi_T(t) = 1 \text{ si } t \in [-\frac{\tau}{2}, \frac{\tau}{2}], \\ &= \frac{1}{\tau} \int_{-\infty}^{+\infty} \Pi_T(t_k + \frac{\tau}{2} - t) y(t) dt \quad \text{car } \Pi_T(t) \text{ est paire,} \\ &= \frac{1}{\tau} (\Pi_T * y)(t_k + \frac{\tau}{2}), \end{aligned}$$

ce qui donne dans le domaine fréquentiel

$$\bar{Y}(f) = e^{j\pi f\tau} \text{sinc}(\pi f\tau) Y(f),$$

soit finalement en terme de densité spectrale de puissance

$$S_{\bar{Y}}(f) = \text{sinc}^2(\pi f\tau) S_Y(f).$$

La puissance de \bar{y}_k , qui est alors égale à la variance vraie $\sigma_y^2(\tau)$, peut alors s'écrire

$$\sigma_y^2(\tau) = P_{\bar{Y}}(\tau) = \int_0^{+\infty} \text{sinc}^2(\pi f\tau) S_Y(f) df. \quad (\text{B.2})$$

D'après l'équation B.1 et suivant le même raisonnement que précédemment, il est possible d'exprimer la variance d'Allan $\sigma_{AL,y}^2(\tau)$ par une expression proche de B.2 sous la forme

$$\sigma_{AL,y}^2(\tau) = 2 \int_0^{+\infty} \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} S_Y(f) df. \quad (\text{B.3})$$

Bruit Blanc

Un bruit blanc se caractérise par une densité de spectrale de puissance constante ($S_w(f) = h_0$). La variance d'Allan s'écrit alors dans ce cas

$$\begin{aligned} \sigma_{AL,w}^2(\tau) &= 2 \int_0^{+\infty} \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} h_0 df, \\ &= \frac{2h_0}{\pi \tau} \int_0^{+\infty} \frac{\sin^4(u)}{u^2} du, \\ &= \frac{2h_0}{\pi \tau} \frac{\pi}{4}, \end{aligned}$$

soit finalement

$$\sigma_{AL,w}^2(\tau) = \frac{h_0}{2\tau}. \quad (\text{B.4})$$

En échelle logarithmique, $\sigma_{AL,w}^2(\tau)$ sera donc une droite de pente -1 (Fig. B.1, $\log(\sigma_{AL,w}^2(\tau)) = \log(\frac{h_0}{2}) - \log(\tau)$).

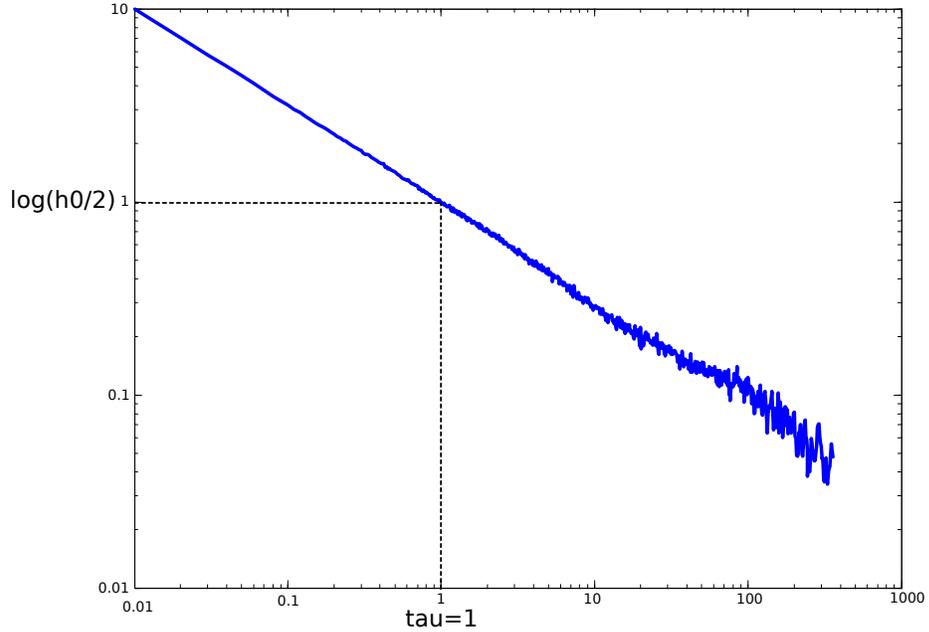


FIGURE B.1 – Variance d'Allan d'un bruit blanc

Bruit de flicker

Un bruit de flicker se caractérise par une densité de spectrale de puissance $S_f(f) = \frac{h_1}{f}$. La variance d'Allan s'écrit alors dans ce cas

$$\begin{aligned}\sigma_{AL,f}^2(\tau) &= 2 \int_0^{+\infty} \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} \frac{h_1}{f} df, \\ &= 2h_1 \int_0^{+\infty} \frac{\sin^4(u)}{u^2} \frac{\pi \tau}{u} \frac{du}{\pi \tau}, \\ &= 2h_1 \int_0^{+\infty} \frac{\sin^4(u)}{u^3} du,\end{aligned}$$

soit finalement

$$\sigma_{AL,f}^2(\tau) = 2h_1 \ln(2). \quad (\text{B.5})$$

En échelle logarithmique, $\sigma_{AL,f}^2(\tau)$ sera donc une droite de pente 0 ($\log(\sigma_{AL,f}^2(\tau)) = \log(2h_1 \ln(2))$).

Marche aléatoire

Une marche aléatoire se caractérise par une densité de spectrale de puissance $S_{ma}(f) = \frac{h_2}{f^2}$. La variance d'Allan s'écrit alors dans ce cas

$$\begin{aligned}\sigma_{AL,ma}^2(\tau) &= 2 \int_0^{+\infty} \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} \frac{h_2}{f^2} df, \\ &= 2h_2 \int_0^{+\infty} \frac{\sin^4(u)}{u^2} \frac{\pi^2 \tau^2}{u^2} \frac{du}{\pi \tau}, \\ &= 2h_2 \pi \tau \int_0^{+\infty} \frac{\sin^4(u)}{u^4} du,\end{aligned}$$

soit finalement

$$\sigma_{AL,ma}^2(\tau) = \frac{2}{3} h_2 \pi^2 \tau. \quad (\text{B.6})$$

En échelle logarithmique, $\sigma_{AL,ma}^2(\tau)$ sera donc une droite de pente 1 (Fig. B.2, $\log(\sigma_{AL,ma}^2(\tau)) = \log(\frac{2}{3} h_2 \pi^2) + \log(\tau)$).

Processus de Gauss Markov d'ordre 1

Un processus de Gauss Markov à l'ordre 1 se caractérise par une auto-corrélation $R_{gm} = \sigma_{gm}^2 e^{-\frac{|\tau|}{\tau_{gm}}}$. La densité de spectrale de puissance $S_{gm}(f)$ s'écrit alors dans ce cas

$$\begin{aligned}\S_{gm}(f) &= TF[R_{gm}], \\ &= TF\left[\sigma_{gm}^2 e^{-\frac{|\tau|}{\tau_{gm}}}\right], \\ &= \frac{2 \frac{\sigma_{gm}^2}{\tau_{gm}}}{\left(\frac{1}{\tau_{gm}}\right)^2 + 4\pi^2 f^2},\end{aligned}$$

ce qui permet alors de calculer la variance d'Allan $\sigma_{AL,gm}^2$

$$\sigma_{AL,gm}^2(\tau) = 4 \frac{\sigma_{gm}^2}{\tau_{gm}} \int_0^{+\infty} \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} \frac{1}{\left(\frac{1}{\tau_{gm}}\right)^2 + 4\pi^2 f^2} df. \quad (\text{B.7})$$

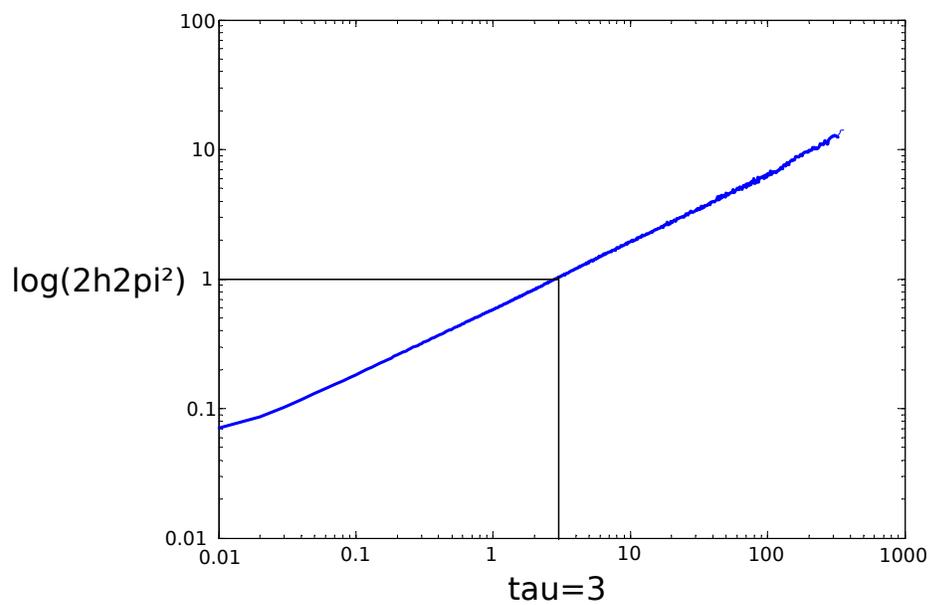


FIGURE B.2 – Variance d'Allan d'une marche aléatoire

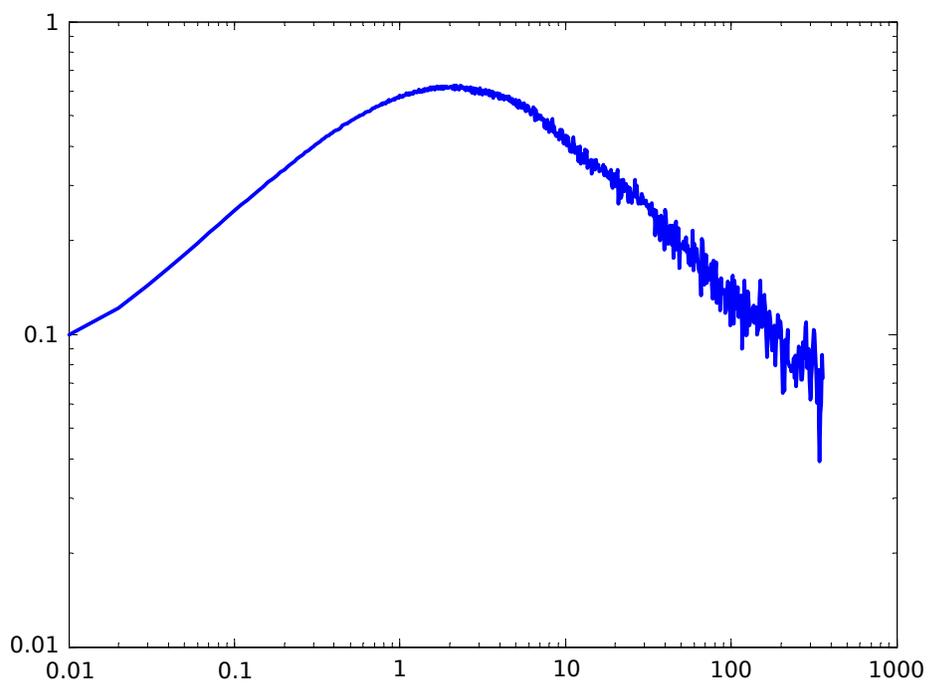


FIGURE B.3 – Variance d'Allan d'un processus de Gauss Markov d'ordre 1

Annexe C

Rétropropagation du gradient

Proposé par Rumelhart, Hinton et Williams en 1986 [RHW86], l'algorithme de rétropropagation du gradient permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches. La démarche est la suivante :

- Initialisation des poids synaptiques,
- Présentation des données d'entrée au réseau de neurones,
- Calcul des sorties du réseau correspondantes,
- Estimation de la proportion de l'erreur se rapportant à chaque entrée,
- Attribution des nouvelles valeurs aux poids synaptiques.

Ci-après, une très bonne explication, en image et pas après pas, d'une itération de cet algorithme trouvé sur <http://galaxy.agh.edu.pl>.

