

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 PRÉSENTATION DU SUJET DE RECHERCHE.....	3
1.1 Mise en contexte	3
1.2 Motivation et défis	7
1.3 Objectifs du projet.....	9
1.4 Méthodologie de développement	10
1.4.1 Phase 1 (Septembre 2010 – Janvier 2011)	10
1.4.2 Phase 2 (Février 2011 – Janvier 2012).....	11
1.4.3 Phase 3 (Février 2012 à aujourd’hui).....	12
CHAPITRE 2 REVUE DE LA LITTÉRATURE.....	17
2.1 Capture de mouvements	17
2.1.1 Capture optique.....	18
2.1.2 Capture mécanique.....	19
2.1.3 Capture magnétique	20
2.1.4 Capture de mouvements par inertie	20
2.1.5 Autres technologies de capture	22
2.2 Navigation à l’intérieur des bâtiments	22
2.2.1 Systèmes de type LPS (Sources externes)	23
2.2.2 Systèmes de type PDR.....	24
2.2.3 Solutions hybride : PDR / LPS	26
2.3 Capteurs inertiels à faible coût.....	27
2.3.1 Caractérisation des erreurs de nature déterministe	28
2.3.2 Modélisation des erreurs de nature déterministe	28
2.3.3 Caractérisation et modélisation des erreurs de nature stochastique.....	33
2.4 Algorithmes de navigation par inertie.....	35
2.4.1 Algorithmes pour la capture de mouvements	35
2.4.2 Algorithmes pour la navigation à l’intérieur des bâtiments.....	36
2.5 Analyse des systèmes existants utilisant une technologie semblable d’IMU	38
2.6 Synthèse de la revue de la littérature.....	40
CHAPITRE 3 PRINCIPES DE LA NAVIGATION INERTIELLE	43
3.1 Concepts en navigation	43
3.1.1 Repères de référence	44
3.1.2 Représentation et propagation de l’attitude du système	46
3.1.3 Représentation de la vitesse et de la position en coordonnées ECEF.....	51
3.1.4 Représentation de la gestuelle (BVH).....	53
3.1.5 Représentation du déplacement pédestre (KML)	54
3.2 IMU et capteurs inertiels de type MEMS	55
3.2.1 Vue d’ensemble d’un IMU	55

3.2.2	Accéléromètres	56
3.2.3	Gyroscopes.....	57
3.2.4	Magnétomètres.....	57
3.2.5	Fusion des capteurs.....	58
3.3	Sources d'erreurs des capteurs inertiels	59
3.3.1	Erreurs de type déterministe	60
3.3.2	Erreurs de type stochastique	64
3.4	Modèles de propagation de l'erreur	65
3.5	Modèles de référence	67
3.6	Filtre de Kalman.....	69
3.7	Synthèse des principes de la navigation inertielle.....	73
CHAPITRE 4 CALIBRATION DES CAPTEURS INERTIELS.....		75
4.1	Présentation du matériel d'étalonnage	75
4.2	Présentation des IMUs utilisés pour les tests	79
4.3	Correction des erreurs de type déterministe.....	80
4.3.1	Méthode multi-positions	82
4.3.2	Détection des états stationnaires.....	84
4.3.3	Calibration du biais initial des magnétomètres.....	86
4.3.4	Calibration initiale des capteurs.....	87
4.3.5	Calibration continue des capteurs	90
4.3.6	Démonstration du besoin de la calibration.....	91
4.3.7	Analyse des performances et discussion.....	96
4.4	Caractérisation des erreurs de type stochastique.....	103
4.4.1	Application de la variance d'Allan	103
4.4.2	Analyse des résultats et discussion	104
CHAPITRE 5 ALGORITHMES D'ATTITUDE ET DE POSITIONNEMENT		109
5.1	Développement de l'algorithme AHRS	109
5.1.1	Alignement initial de l'IMU	110
5.1.2	Correction de la position des IMUs avec les boîtiers	113
5.1.3	Modèle dynamique de mise à jour de l'attitude.....	114
5.1.4	Ajout des modèles de connaissances	114
5.1.5	Modélisation du filtre EKF pour corriger la dérive de la solution.....	115
5.1.6	Analyse des performances et discussion.....	118
5.2	Développement de l'algorithme INS.....	122
5.2.1	Ajout de l'intégration de l'accélération et de la vitesse.....	124
5.2.2	Ajout des modèles de connaissances	124
5.2.3	Modélisation du filtre EKF pour corriger la dérive de la solution.....	125
5.2.4	Discussion.....	128
5.3	Développement de méthodes de soutien.....	128

CHAPITRE 6	RÉALISATION DU PROTOTYPE IBNAV	131
6.1	Conception du micro-logiciel des IMUs	132
6.1.1	Objectifs de la conception.....	132
6.1.2	Architecture du micro-logiciel des IMUs	134
6.1.3	Fonctionnalités du micro-logiciel des IMUs.....	143
6.1.4	Résultats et possibilités supplémentaires	147
6.2	Développement du système de communication	148
6.2.1	Objectifs du développement	148
6.2.2	Architecture du système de communication	149
6.3	Réalisation de l'architecture matérielle.....	151
6.3.1	Objectifs de la réalisation.....	152
6.3.2	Composition de l'architecture.....	153
6.4	Analyse du système ibNav (Coût, poids et consommation)	155
CHAPITRE 7	RÉALISATION DE LA PLATEFORME D'ÉTUDES IPAD.....	157
7.1	Objectifs de la réalisation.....	157
7.2	Implémentation des interfaces graphiques utilisateurs	159
7.2.1	Interface principale	159
7.2.2	Interface pour les données brutes.....	163
7.2.3	Interface pour les algorithmes.....	165
7.2.4	Interface pour la capture de mouvements et la localisation à l'intérieur des bâtiments.....	165
7.3	Outils intégrés	166
7.3.1	Mise en place de la communication UDP / TCP	166
7.3.2	Mise en place d'une base de données	167
7.3.3	Réalisation d'une interface 2D / 3D utilisant la librairie OpenGL puis Unity3D.....	168
7.3.4	Réalisation d'une librairie pour la création de graphiques	169
7.3.5	Mise en place de diverses API.....	170
CHAPITRE 8	SYSTÈME GLOBAL ET TRANSFERT TECHNOLOGIQUE.....	171
8.1	Synthèse et rendu du projet ibNav	171
8.1.1	Tableaux synthétique des développements réalisés lors de la Phase 3	172
8.1.2	Faiblesses et forces.....	177
8.2	Transfert technologique vers VTADS, MEDOC et d'autres projets	178
8.2.1	Le Micro-iBB, une création à partir d'ibNav.....	179
8.2.2	Le logiciel iOS ibNav 2 vers iMetrik Show Case	183
CONCLUSION	187
RECOMMANDATIONS	191
ANNEXE I	RENDU À LA FIN DE LA PHASE 1 DU PROJET IBNAV	195
ANNEXE II	RAISONS DE L'IMPLÉMENTATION DE LA PHASE 3.....	197

ANNEXE III PHOTOS DE SYSTÈMES PERMETTANT LA CAPTURE DE MOUVEMENTS	201
ANNEXE IV CONVERSION COORDONNÉES CARTÉSIENNES ET GÉODÉSIQUES	203
ANNEXE V INFORMATIONS TECHNIQUES SUR LES CENTRALES INERTIELLES UTILISÉES.....	205
ANNEXE VI ERREURS DÉTERMINISTES POUR CHAQUE IMU DU PROJET IBNAV	207
ANNEXE VII ERREURS STOCHASTIQUES POUR CHAQUE IMU DU PROJET IBNAV	209
ANNEXE VIII TABLEAU DES LIMITATIONS DES ARTICULATIONS DU CORPS HUMAIN	211
ANNEXE IX POSTER DE PRÉSENTATION DU PROJET IBNAV	213
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES.....	215

LISTE DES TABLEAUX

	Page
Tableau 2.1	Avantages et inconvénients pour la capture optique.....18
Tableau 2.2	Avantages et inconvénients pour la capture mécanique19
Tableau 2.3	Avantages et inconvénients pour la capture magnétique20
Tableau 2.4	Avantages et inconvénients pour la capture inertielle21
Tableau 4.1	Liste des IMUs utilisés pour tester les algorithmes de calibration80
Tableau 4.2	Configuration du scénario pour la méthode multi-positions.....82
Tableau 4.3	Valeurs initiales des composantes de la matrice P.....88
Tableau 4.4	Influence de la température sur les mesures90
Tableau 4.5	Différences d'erreurs entre les IMU STEVAL-MKI062V2 du projet ibNav95
Tableau 4.6	Coefficients de bruit blanc calculés suite à l'analyse d'Allan Variance..106
Tableau 5.1	Erreur RMS statique et dynamique des angles d'Euler (en degrés)122
Tableau 5.2	Calcul de l'attitude - Hiérarchie entre les IMUs.....128
Tableau 6.1	Répartition du coût du premier prototype ibNav155
Tableau 8.1	Synthèse sur la conception du micro-logiciel des IMU172
Tableau 8.2	Synthèse sur le développement du système de communication173
Tableau 8.3	Synthèse sur la réalisation de l'architecture matérielle.....174
Tableau 8.4	Synthèse sur la conception du logiciel pour iPad175
Tableau 8.5	Synthèse sur l'implémentation des algorithmes inertiels.....176
Tableau 8.6	Faiblesses et forces du projet ibNav177
Tableau 8.7	Logo des différents projets basés sur les développements du projet ibNav179

LISTE DES FIGURES

		Page
Figure 0.1	Logo du projet ibNav	1
Figure 1.1	Répartition des différentes sections du projet ibNav	12
Figure 1.2	Méthodologie de développement pour le projet ibNav.....	13
Figure 2.1	Méthode de calibration à 4 positions	30
Figure 2.2	Méthode de calibration à 18 positions	30
Figure 2.3	Résultats d'une analyse des erreurs stochastique par la variance d'Allan (échelle logarithmique).....	34
Figure 3.1	Relation entre les différents repères orthogonaux et lien avec l'attitude et la position du mobile.....	43
Figure 3.2	Représentation des différents repères terrestres.....	44
Figure 3.3	Représentation de la transformation entre les coordonnées géodésiques et cartésiennes ECEF.....	45
Figure 3.4	Extrait et représentation du fichier BVH du projet ibNav	54
Figure 3.5	L'IMU iNemoV2 et son diagramme simplifié.....	55
Figure 3.6	Schéma du fonctionnement d'un accéléromètre masse-ressort de type MEMS.....	56
Figure 3.7	Schéma du fonctionnement d'un gyroscope masse-ressort de type MEMS.....	57
Figure 3.8	Schéma du fonctionnement d'un magnétomètre AMR de type MEMS....	58
Figure 3.9	Schéma d'une fusion des capteurs pour un algorithme AHRS.....	59
Figure 3.10	Principales erreurs entre les valeurs mesurées et réels d'un capteur inertiel et d'un magnétomètre	60
Figure 3.11	Représentation des deux types d'erreurs causant la distorsion du champ magnétique	63

Figure 4.1	Photo du robot de calibration Lynxmotion AL5D équipé avec des IMUs.....	75
Figure 4.2	Liste des fonctions et imprime écran de l'interface principale.....	76
Figure 4.3	Liste des fonctions et imprime écran de l'interface pour le contrôle manuelle.....	77
Figure 4.4	Liste des fonctions et imprime écran de l'interface pour le contrôle automatique.....	78
Figure 4.5	Schéma de l'algorithme de calibration initial du projet ibNav.....	81
Figure 4.6	Schéma de l'algorithme de calibration continu du projet ibNav.....	81
Figure 4.7	Schéma de l'algorithme de détection des états stationnaires.....	85
Figure 4.8	Résultats de l'algorithme de détection des états stationnaires.....	86
Figure 4.9	Norme des accélérations suivant différentes orientations.....	92
Figure 4.10	Norme de la vitesse angulaire suivant différentes orientations.....	93
Figure 4.11	Norme du champ magnétique suivant différentes orientations.....	93
Figure 4.12	Champ magnétique mesuré comparé à la référence théorique.....	94
Figure 4.13	Déviations de l'accélération par rapport à la température interne.....	95
Figure 4.14	Résultat de la calibration pour l'accéléromètre de l'IMU maître du projet ibNav.....	96
Figure 4.15	Résultat de la calibration pour le gyroscope de l'IMU maître du projet ibNav.....	97
Figure 4.16	Résultat de la calibration EKF pour le magnétomètre du projet ibNav.....	98
Figure 4.17	Résultat de la calibration EKF + Lissage pour le magnétomètre du projet ibNav.....	98
Figure 4.18	Résultat de la calibration pour le magnétomètre de l'IMU maître du projet ibNav.....	99
Figure 4.19	Données du magnétomètre durant le fonctionnement du système sans CC.....	100

Figure 4.20	Données du magnétomètre durant le fonctionnement du système avec CC.....	100
Figure 4.21	Correction des erreurs déterministes de l'accéléromètre à l'aide de la température	101
Figure 4.22	Prototype de plateforme de calibration des magnétomètres à faible coût.....	102
Figure 4.23	Analyse de la variance d'Allan pour les accéléromètres de 4 IMUs à faible coût.....	105
Figure 4.24	Analyse de la variance d'Allan pour les gyroscopes de 4 IMUs à faible coût.....	105
Figure 4.25	Analyse de la variance d'Allan pour les magnétomètres de 4 IMUs à faible coût.....	106
Figure 5.1	Schéma de l'algorithme AHRS du projet ibNav.....	109
Figure 5.2	Prototype de plateforme de calibration des erreurs de mauvais alignement.....	113
Figure 5.3	Dérive de l'angle Phi (Roulis) avant et après la calibration	119
Figure 5.4	Dérive de l'angle Phi (Roulis) avant et après la correction de la dérive.....	120
Figure 5.5	Estimation de l'angle d'Euler Theta calculés par la méthode AHRS.....	121
Figure 5.6	Erreurs sur l'angle d'Euler Theta.....	121
Figure 5.7	Schéma de l'algorithme INS du projet ibNav.....	123
Figure 5.8	Interface du logiciel ibNav sur iPad représentant une carte 2D du laboratoire	130
Figure 6.1	Architecture globale du système ibNav.	131
Figure 6.2	Architecture du micro logiciel du projet ibNav version 1.14	134
Figure 6.3	Processus global du micro-logiciel des IMU	139
Figure 6.4	Processus concernant l'initialisation du microcontrôleur.....	140
Figure 6.5	Processus concernant la communication pour l'IMU « maître ».....	141
Figure 6.6	Processus concernant la communication pour les IMUs « esclave ».....	142

Figure 6.7	Processus concernant la communication avec l'afficheur LCD	143
Figure 6.8	Schéma du système de communication mis en place	149
Figure 6.9	Trame pour l'IMU « maître » et les IMUs « esclave avant l'initialisation	150
Figure 6.10	Trame de l'IMU « maître » après l'initialisation	150
Figure 6.11	Trame des IMUs « esclaves » après l'initialisation (données brutes).....	150
Figure 6.12	Trame des IMUs « esclaves » après l'initialisation (données calculées)	150
Figure 6.13	Trame de transfert de l'IMU « maître » vers l'iPad.....	151
Figure 6.14	Architecture matérielle du système ibNav composé de 14 IMUs.....	152
Figure 6.15	Photographie du sous-système pour le bras droit	153
Figure 6.16	Schéma de l'architecture matérielle.....	149
Figure 6.17	Photographie des IMUs « esclave » et de l'IMU « maître ».....	155
Figure 7.1	Fonctionnalités devant être incluses dans le GUI pour iPad.....	157
Figure 7.2	Barre de navigation du logiciel iOS.....	159
Figure 7.3	Interface principale et à propos du logiciel ibNav sur iPad	160
Figure 7.4	Schéma des différentes sections de l'interface principale	161
Figure 7.5	Exemple de l'interface permettant de sélectionner un enregistrement.....	162
Figure 7.6	Interface pour les données brutes du logiciel ibNav sur iPad.....	164
Figure 7.7	Version initiale et en cours du développement du modèle pour la capture de mouvements.....	165
Figure 7.8	Schéma SQL de la base de données SQL du projet ibNav	167
Figure 7.9	Modèle 3D du projet ibNav développé sous 3Ds Max	169
Figure 8.1	Projet ibNav en fonctionnement avec les 14 IMUs dans la salle de sport de l'ÉTS	171
Figure 8.2	Architecture globale du Micro-iBB VTADS	181

Figure 8.3	Interfaces sur iPad du logiciel iVave (Partie 1)	182
Figure 8.4	Interfaces sur iPad du logiciel iVave (Partie 2)	183
Figure 8.5	Interfaces sur iPad et PC du logiciel iMetrik Show Case	184
Figure 8.6	Photographie du stand incluant l'iMetrik Show Case.....	185

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AHRS	Attitude and Heading Reference System (Centrale d'attitude et de cap)
ADC	Analog to Digital Converter (Convertisseur analogique-numérique)
ASF	Acclaim Skeleton File
AMC	Acclaim Motion Capture Data
AMR	Anisotropic magneto-resistive (Magnétorésistance anisotrope)
AOA	Angle of Arrival (Angle d'arrivée)
AP	Access Point (Point d'accès)
AR	Autoregressive (Autorégressif)
AVAR	Allan Variance (Variance d'Allan)
BIM	Building Information Model (Modèle de connaissance sur le bâtiment)
BVA	BioVision Animation
BVH	BioVision Hierarchie
CGRF	Canadian Geomagnetic Reference Field (Référence du champ magnétique Canadien)
DCM	Direction Cosine Matrix (Matrice en cosinus de direction)
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique des hôtes)
DMA	Direct Memory Access (Accès direct à la mémoire)
DOF	Degree Of Freedom (Degré de liberté)
ECEF	Earth Centered Earth Fixed (Repère fixe terrestre)
ECI	Earth Centered Inertial (Repère inertiel terrestre)
EKF	Extended Kalman Filter (Filtre de Kalman étendu)
FEKF	Fast Extended Kalman Filter (Filtre de Kalman étendu rapide)
GD	Gradient Descent (Descente de Gradient)
GM	Gauss-Markov
GNSS	Global Navigation Satellite System (Géolocalisation et navigation par un système de satellites)
GPS	Global Positioning System (Système de positionnement global)
GPIO	General Purpose Input/Output (Lignes d'entrée / sortie)
GSM	Global System for Mobile Communications (Système mondial pour

	communication avec les mobiles)
GUI	Graphical User Interface (Interface graphique utilisateur)
HDR	Heuristic Drift Reduction (Réduction de la Dérive Heuristique)
I2C	Inter Integrated Circuit (Interface de circuit intégré)
IEZ	INS + EKF + ZUPT
ibNav	Indoor Body Navigation
IGRF	International Geomagnetic Reference Field (Référence du champ magnétique international)
IMC	Inertial Motion Capture (Capture de mouvements par inertie)
IMM	Inertial Motion Module (Module de mouvement inertiel)
IMU	Inertial Measurement Unit (Centrale inertielle)
INS	Inertial Navigation System (Système de navigation inertielle)
JTAG	Join Test Action Group
KML	Keyhole Markup Language
LACIME	LABoratoire de Communication et d'Intégration de la MicroÉlectronique
LASSENA	LABoratoire des technologies Spatiales, Systèmes Embarqués, Navigation et Avionique
LED	Light Emitting diode (Diode électroluminescente)
LPS	Local Positioning Systems (Systèmes de positionnement local)
LSB	Least Significant Bit (Bit de poids faible)
MCU	Microcontroller Unit (Microcontrôleur)
MEMS	MicroElectroMechanical System (Système microélectromécanique)
NVIC	Nested Vectored Interrupt Controller (Contrôleur d'interruptions)
PDR	Pedestrian Dead-Reckoning (Navigation pedestre à l'estime)
PnP	Plug and Play (Branche et utilise)
PSD	Power Spectral Density (Densité spectrale de puissance)
RCC	Real-Time Clock Controller (Contrôleur d'horloge et de reset)
RF	Radio Frequency (Radiofréquence)
RFID	Radio Frequency Identification (Radio-identification)
RIOS	Robotic Arm Interactive Operating System
RMS	Root Mean Square (Valeur efficace)
RS232	Norme standard d'un bus de communication de type série

RSS	Received-Signal Strength (La puissance du signal reçu)
RX	Receive (Réception)
RTOF	Roundtrip Time Of Flight (Temps d'aller retour en vol)
SLAM	Simultaneous Localization and Mapping (Localisation et cartographie simultanées)
SoOP	Signal of Opportunity (Signal d'opportunité)
SPI	Serial Peripheral Interface (Interface Série pour Périphériques)
SSC-32	Type de contrôleur pour servomoteurs
SWD	Serial Wire Debug
TCP	Transmission Control Protocol (Protocole de Contrôle de Transmission)
TIM	Timers
TOA	Time Of Arrival (Temps d'arrivé)
TX	Transmit (Transmission)
TDOA	Time Difference Of Arrival (Différence des temps d'arrivées)
TWI	Voir I2C
UART	Universal Asynchronous Receiver Transmitter (Émetteur-récepteur asynchrone universel)
UDP	User Datagram Protocol (Protocole de Datagramme Utilisateur)
UWB	Ultra Wideband (Ultra Large Bande)
WLAN	Wireless local-area network (Réseau local sans-fil)
WWDG	Window Watchdog (Chien de garde de fenêtre)
ZARU	Zero Angular Rate Update (Détection de la vitesse angulaire nulle)
ZUPT	Zero Velocity Update (Détection de la vitesse nulle)
μC	Microcontroller (Microcontrôleur)

INTRODUCTION

La technologie principale qui a permis la naissance du projet ibNav (Figure 1.1) est le MEMS (Système microélectromécanique) pour les capteurs inertiels. Ils sont utilisés de nos jours dans de nombreux dispositifs et leur nombre continue de s'accroître au cours du temps. On en retrouve essentiellement dans certains téléphones intelligents, appareils photos, consoles de jeu, disques durs ainsi que dans certaines voitures modernes (pour le déclenchement de l'airbag). Cette technologie suscite également un intérêt important au niveau de la recherche pour la navigation puisqu'elle présente un faible coût, un faible encombrement et une autonomie au niveau du dispositif.

Le projet consiste à réaliser un prototype pourvu de trois types de capteurs inertiels MEMS à faible coût : accéléromètre, magnétomètre et gyroscope. L'objectif étant d'étudier cette technologie et d'obtenir un système effectuant de la capture de mouvements « Motion Capture » et de navigation à l'intérieur des bâtiments « Indoor Pedestrian Navigation ». À cet effet, un prototype composé de 14 IMUs (Centrales inertiels) sera réalisé. Un IMU étant un système composé de capteurs inertiels et d'un μ C (Microcontrôleur). L'avantage d'utiliser cette technologie est qu'elle peut être déployée dans n'importe quel environnement, sans nécessiter l'installation d'une infrastructure spécifique et la transmission de signaux vers/depuis une source externe. Cependant de nombreux défis doivent être relevés afin de rendre fonctionnel ce prototype.



Figure 0.1 Logo du projet ibNav
« Indoor body Navigation »

L'utilisation de la technologie MEMS à faible coût ne peut pas être utilisée directement. Le premier défi est donc la réalisation d'algorithmes performants de calibration et de navigation afin de corriger les erreurs présentes dans les capteurs de manière efficace. Les autres défis résultent de la réalisation du système ibNav, c'est-à-dire la conception d'un micro-logiciel pour les IMUs, d'une architecture matérielle et d'une interface graphique utilisateur sur une tablette graphique (iPad) liées par un système de communication performant permettant le fonctionnement en temps réel. De plus, le projet étant destiné à la recherche et au développement, l'interface graphique sera pourvue d'une plateforme d'études afin de permettre de comparer, d'analyser et d'améliorer les différents algorithmes inertiels.

Le mémoire est composé de 8 chapitres, le premier présente le projet de recherche avec la problématique, les défis, les objectifs et la méthodologie de recherche. Le chapitre 2 fait l'état de l'art des deux types de navigation et des technologies utilisées dans le projet afin de vérifier la viabilité des différentes stratégies adoptées. Le chapitre 3 est une introduction aux principes de navigation, permettant au lecteur de comprendre le fonctionnement des capteurs inertiels et la suite du mémoire. Le chapitre 4 est consacré à l'algorithme de calibration et le chapitre 5 aux algorithmes de navigation, développés pour le projet ibNav. Le chapitre 6 reprend la réalisation du prototype avec la conception du micro-logiciel des IMUs, le développement du système de communication et la réalisation de l'architecture matérielle. Le chapitre 7 concerne la réalisation de l'interface graphique utilisateur sur l'iPad. Finalement le chapitre 8 est une synthèse du système global et conclut sur le transfert technologique qui a été effectué, essentiellement avec la réalisation du Micro-iBB, soit l'intégration des technologies développées pour ibNav au projet VTADS.

CHAPITRE 1

PRÉSENTATION DU SUJET DE RECHERCHE

Le projet ibNav est abordé dans ce chapitre afin d'en connaître le contexte actuel ayant mené à sa réalisation, son utilité du point de vue universitaire et industriel fait aussi l'objet de cette partie. Par la suite, la problématique de recherche ainsi que les défis devant être relevés sont énumérés. Finalement, les objectifs à atteindre pour répondre à la spécification du projet sont listés et la méthodologie de développement choisie est formalisée.

1.1 Mise en contexte

Le projet ibNav signifiant indoor body Navigation consiste à réaliser un prototype permettant d'effectuer de la capture de mouvements et de la navigation à l'intérieur des bâtiments en utilisant des capteurs inertiels à faible coût sans l'utilisation d'aucun autre système externe tel que le GPS. Ce projet a vu le jour en septembre 2009 au sein du laboratoire LACIME, puis il s'est poursuivi au sein du laboratoire LASSENA en 2012. Ce sont des unités de recherche accréditées de l'ÉTS située dans la ville de Montréal au Québec.

Afin de comprendre l'intérêt du projet ainsi que la raison justifiant sa naissance en septembre 2009, il est intéressant d'étudier brièvement l'historique des méthodes de capture de mouvements, de navigation à l'intérieur des bâtiments ainsi que l'évolution des capteurs inertiels.

La capture de mouvements est une technique permettant l'animation de personnages virtuels à partir de mouvements effectués par un individu ou un robot. À l'heure actuelle, elle est utilisée dans plusieurs applications, telles que : l'animation, le médical, la simulation, le sport et le divertissement. Cette technique a vu le jour au début des années 80 avec l'effervescence de la télévision et continue d'évoluer par l'utilisation de nouvelles techniques et de technologies de plus en plus précises (Sturman, 1999; Menache, 2010) :

- la première méthode pouvant être catégorisée comme de la capture de mouvements a été développée par Tom Calvert en 1982 et consiste à utiliser un exosquelette doté d'électrogoniomètres potentiométriques afin d'obtenir l'angle de flexion du genou. Par la suite en 1988, l'entreprise Jim Henson Production crée l'ordinateur Waldo C. Graphic. Ce système améliore le précédent afin de l'utiliser pour l'animation de personnages. Il sert encore de nos jours mais plutôt dans le cadre de la robotique puisqu'il n'est pas très adapté aux mouvements du corps humain par la gêne que le système occasionne sur les mouvements de l'utilisateur. Cette méthode est communément appelée la capture mécanique;
- en 1983, Delle Maxwell et Carol Ginsberg développent au MIT un système appelé Graphical Marionnette, un des premiers systèmes de capture optique. Il est constitué de deux caméras et de DEL placées sur chaque articulation du corps. À cette époque, la technologie limite le rendu en temps réel (soit le nombre de capteurs pouvant être utilisés ainsi que le nombre de positions par seconde pouvant être sauvegardé). C'est en 1989 que le système est repris et amélioré par la société Motion Analysis afin d'obtenir un système convenable, permettant d'obtenir des mouvements réalistes pour l'animation d'un personnage. Les problèmes essentiels restant présents avec cette technologie, c'est premièrement la nécessité d'une infrastructure spécifique et deuxièmement le suivi des points, puisque les marqueurs peuvent être cachés par certains mouvements et nécessitent de ce fait beaucoup de post-traitement. De nos jours, c'est cependant la technologie la plus utilisée puisqu'elle permet d'obtenir des résultats précis, ne nuit pas aux mouvements naturels du corps et a fait ses preuves au cours du temps dans le domaine de la capture de mouvements. Cette méthode est la capture optique;
- en 1992, la troisième technologie utilisée est l'utilisation de capteurs électromagnétiques par la société Colossal Picture. Cette technologie présente l'avantage de ne pas avoir les problèmes de la capture optique et de pouvoir fonctionner en temps réel sans avoir besoin de post-traitement. Cette méthode est la capture magnétique;

- depuis les années 2000, de nombreux ouvrages concernant une nouvelle méthode de navigation corporelle sont apparus suite aux récentes avancées sur les systèmes de technologie MEMS (Système microélectromécanique) permettant le développement de capteurs inertiels à faible coût et de faible envergure, tels que les accéléromètres, les gyroscopes et les magnétomètres combinés formant des IMUs (Centrales inertielle). Cette méthode s'appelle IMC, soit la capture inertielle de mouvements;
- note : D'autres technologies existent, essentiellement des technologies hybrides mais elles restent du domaine de la recherche et ne sont pas commercialisées.

De son côté, la navigation intérieure permet à un individu (ou à un robot) de pouvoir connaître sa position et son déplacement à l'intérieur d'un bâtiment ou dans certains cas de permettre à un autre individu ou à un dispositif de connaître ces informations. L'individu en question effectue des déplacements pédestres. Elle est utilisée dans plusieurs applications, telles que : Le divertissement, l'entraînement d'urgence, le repérage dans les centres commerciaux, le marketing, la visite des musées, la surveillance des enfants, etc.

Ce type de navigation reste principalement du domaine de la recherche. Depuis les années 2000, on retrouve essentiellement comme technologies permettant d'effectuer de la navigation à l'intérieur des bâtiments, les systèmes radio utilisant le réseau GSM, le réseau internet sans-fil (Wi-Fi) ainsi que le réseau GPS simulé à l'intérieur. Cependant, la précision et l'intensité des signaux sont bien souvent trop faibles pour être efficace. Pour aider ces technologies, il est possible d'installer d'autres émetteurs dans le bâtiment mais ceux-ci nécessitent la mise en place d'une infrastructure conséquente et coûteuse. On retrouve aussi d'autres technologies RF pouvant permettre cette technique mais nécessitant également l'installation d'une infrastructure dans le bâtiment. Finalement, la seule méthode permettant au système d'être autonome est l'utilisation d'IMUs même s'ils doivent être utilisés avec une méthodologie particulière. En effet, la position dérivera de plusieurs mètres en quelques minutes avec un algorithme INS standard. De plus, ils sont hybridés avec d'autres technologies permettant d'améliorer leur efficacité et la détermination précise de la position initiale (Le, 2009; Menache, 2010; Bird et Arden, 2011).

Les capteurs inertiels de type MEMS n'ont fait leur apparition sur le marché qu'à partir des années 1990. Au fil du temps, cette technologie continue de s'améliorer et son prix à diminuer. Mais bien qu'elle offre de très bonnes performances par rapport à son encombrement et à son coût, elle reste tout de même limitée lorsqu'une autonomie complète est requise. Cependant, comme nous le verrons dans ce mémoire, il est possible d'améliorer considérablement les performances en identifiant, modélisant et en corrigeant plusieurs des erreurs présentes dans ces capteurs inertiels ainsi qu'en utilisant des algorithmes de navigation performants. (Giroux, 2004; Lavoie, 2012).

Ce bref historique confirme la tendance à utiliser des IMUs pour combiner deux besoins, soient la capture de mouvements et la navigation à l'intérieur des bâtiments puisque s'il est possible d'obtenir des performances adéquates, cette solution offrirait de nombreux avantages par rapport aux autres technologies présentées. Ces dernières posent une limite au niveau matériel (le besoin d'une infrastructure externe, la non adaptation à la morphologie humaine, etc.), des performances limitées et un coût important. De plus, la technologie inertielle n'émettant ou nécessitant aucun signal, permet un système entièrement autonome et robuste ne dépendant pas des conditions extérieures et de son environnement. Le principal inconvénient de l'utilisation de cette technologie est qu'elle nécessite la conception de méthodes corrigeant les différentes erreurs des capteurs inertiels, ceci afin de limiter la dérive de la solution et en permettre l'utilisation dans ces deux types de navigation (intérieure et corporelle).

1.2 Motivation et défis

Le développement du projet ibNav exploite les toutes dernières technologies dans le domaine de la microélectronique faisant leurs apparitions sur le marché depuis quelques années. La motivation résulte essentiellement dans le fait de pouvoir faire fonctionner la technologie des IMU-MEMS bas de gamme en compensant les inconvénients afin d'obtenir un système utilisant tous les avantages que la technologie inertielle peut offrir en navigation. On réalisera de ce fait un système global à faible coût dont les performances se rapprochent le plus possible de celles des systèmes actuels très dispendieux. ibNav offre de surcroît des fonctionnalités actuellement non présentes dans un prototype réellement utilisable. Il pourra également servir à l'étude du comportement des capteurs inertiels dans diverses situations, scénarios ainsi que pour l'analyse, l'amélioration et la comparaison de différents algorithmes inertiels. Par sa diversité au niveau des fonctionnalités, le projet en plus d'être une solution efficace pour l'étude des technologies inertielles, se révèle intéressant pour trois secteurs distincts :

- le secteur du divertissement (cinéma, animation et jeux vidéo) pour la capture de mouvements. Les solutions similaires commercialisées ont un coût très élevé et cet investissement n'est pas envisageable pour beaucoup d'entreprises. Concernant l'animation, une technologie moins onéreuse est très avantageuse puisqu'elle permet de réduire le coût en éliminant le besoin de dessiner chaque image à la main. La technologie est indépendante du lieu de son utilisation, car l'individu peut se déplacer partout dans un bâtiment. On pourrait même penser à de nouveaux types de jeu qui serait joué à la grandeur d'un édifice où la navigation intérieure serait nécessaire avec un casque de réalité augmentée;
- le secteur de la défense, sécurité (simulation et réalité augmentée) pour les deux types de navigation (capture de mouvements et navigation intérieure). Il peut permettre lors de simulations d'attaque d'améliorer la position de combat ou les manœuvres à l'intérieur des bâtiments par l'observation des différents mouvements. Puisqu'il utilise uniquement les technologies inertielles, le système peut être vu comme furtif permettant de se repérer

dans un environnement où il serait impossible d'utiliser une méthode de repérage utilisant des signaux externes;

- le secteur du sport pour les deux types de navigation (capture de mouvements et navigation intérieure), les mouvements effectués par l'individu peuvent être analysés afin de lui permettre de voir sa performance et de l'améliorer en lui apportant une nouvelle solution de mouvements associée à une puissance spécifique qui serait plus efficace.

Les défis qui en découlent sont premièrement qu'il faut pouvoir réaliser un système robuste en utilisant du matériel à faible coût. De ce fait, de nombreuses contraintes matériels viennent s'ajouter au projet et nécessitent un développement délicat au niveau du micro-logiciel des IMUs, de l'architecture matérielle et du logiciel offrant une interface graphique utilisateur. Nous verrons dans le CHAPITRE 2 que les chercheurs utilisant des IMUs similaires aux nôtres, se limitent à 2 ou 3 IMUs maximums, avec uniquement l'utilisation de données en post-traitement. Cette limitation du nombre d'IMUs permet de réduire les nombreux problèmes matériels en découlant. L'élaboration d'un système complet requiert également de nombreuses connaissances que ce soit en programmation embarquée, en électronique, en télécommunication et en navigation.

L'autre défi et non le moindre est de pouvoir utiliser ces capteurs de façon autonome sans l'utilisation de source externe. Nous verrons dans le CHAPITRE 2 que la tendance est la mise en place d'une solution hybride afin d'apporter une correction de la position à certains moments spécifiques pour corriger les erreurs des capteurs inertiels. Cependant, les recherches se multiplient sur l'utilisation des IMUs seules pour les deux types de navigation qui nous intéressent, le défi va donc être de concevoir des algorithmes performants permettant ainsi de maximiser les performances de notre système afin d'offrir des résultats se rapprochant le plus possible des systèmes haut de gamme.

1.3 Objectifs du projet

Comme introduit, le développement du projet ibNav doit permettre de réaliser deux types de navigation : corporelle (capture de mouvements) et à l'intérieur des bâtiments. Pour ce faire, l'objectif global est de réaliser un prototype fonctionnel composé de 14 IMUs incorporés dans un vêtement et placés entre chaque articulation du corps. De plus, une interface graphique (GUI) doit recueillir les données brutes de ces IMUs afin de les traiter et de permettre l'interaction utilisateur. Plus précisément, les objectifs du projet sont :

- la conception d'un micro-logiciel performant pour les IMUs, stable et offrant toute une gamme de fonctionnalités utiles afin de répondre aux spécifications du projet;
- le développement d'un système de communication global permettant la récupération sans erreur de toutes les données utiles des IMUs;
- la réalisation d'une architecture matérielle robuste, flexible et fiable;
- la conception du logiciel pour l'interface graphique qui est dans notre cas l'iPad. Il doit intégrer en plus des interfaces correspondantes aux deux types de navigation, des fonctionnalités d'études des données inertielles provenant des capteurs à faible coût;
- le développement et l'implémentation des algorithmes inertiels. Plus précisément, les algorithmes se divisent en trois grandes catégories :
 1. Les algorithmes de calibration permettant de corriger les erreurs connues pouvant être modélisées des capteurs inertiels;
 2. Les algorithmes pour la capture de mouvements, soit un modèle AHRS (Centrale d'attitude et de cap) / EKF (Filtre de Kalman étendu) incluant des modèles de connaissances afin d'obtenir une méthodologie appelée IMC;
 3. Les algorithmes pour la navigation à l'intérieur des bâtiments, soit INS (Système de navigation inertielle) / EKF incluant également des modèles de connaissances afin d'obtenir une méthodologie appelée PDR (Navigation pedestre à l'estime).
- finalement, une fois le prototype fonctionnel, une analyse des performances sera effectuée sur les deux types de navigation et les résultats seront discutés.

Remarque : L'utilisation d'un filtre de Kalman étendu fait partie des objectifs initiaux du projet.

1.4 Méthodologie de développement

Le projet ibNav s'est déroulé en 3 phases, la première phase et une partie de la phase 2 se sont déroulées avant le commencement de la maîtrise (stage d'étudiants), elles sont décrites brièvement dans cette section afin d'apporter une vue global du projet et de permettre de comprendre son évolution ainsi que les choix de la méthodologie qui a suivie.

1.4.1 Phase 1 (Septembre 2010 – Janvier 2011)

Le début du projet a commencé avec une équipe de quatre membres s'occupant chacun d'une partie spécifique par rapport à sa formation :

- Philippe Lavoie essentiellement pour la partie concernant les algorithmes inertiels;
- Jean Lambert pour le développement du logiciel de l'iPad;
- Hugo Laurent Beaucage pour la communication sans-fil;
- moi-même pour le micro-logiciel des IMU, le système de communication filaire, l'architecture matérielle pour le robot de calibration, le GUI (Interface graphique utilisateur) PC ainsi que le GUI pour le robot permettant la calibration des IMU.

Le rendu à la fin de la phase 1 était un système composé de 4 IMUs liés à un bras robotique pouvant fonctionner en temps réel à l'aide d'un GUI permettant la réception des données inertielles et la configuration du robot de calibration. Les résultats sont affichés par la suite sous forme de graphiques à l'aide du logiciel Matlab. Le système est fonctionnel mais seul l'algorithme de type AHRS est fonctionnel (sans l'EKF, c'est-à-dire sans méthode corrigeant la dérive de la solution de navigation). On note également que la vitesse de transmission maximum de l'UART (Émetteur-récepteur asynchrone universel) est trop faible pour supporter la transmission des données qui proviendraient de 14 IMUs, que l'architecture hardware est adéquate pour le robot mais poserait des problèmes pour l'architecture corps humain à cause des connecteurs et des câbles utilisés.

Remarque : Durant cette phase, le logiciel pour l'iPad ainsi que le module sans-fil n'étaient pas encore opérationnels. En ANNEXE I, une liste de mon développement et un schéma de mon rendu après la phase 1 est présent.

1.4.2 Phase 2 (Février 2011 – Janvier 2012)

Durant cette phase, le logiciel pour l'iPad développé par Jean Lambert est rendu. Il permet de récupérer les données des IMUs en mode UDP (Protocole de Datagramme Utilisateur). Ces données sont les résultats provenant de l'IMC. Il permet d'afficher les valeurs directement sous la forme d'un squelette en fil de fer 2D. Il est également possible d'enregistrer ces données. (Lambert, 2011).

Dans cette phase également, le module sans-fil a été mis en place. Il permet de récupérer les données en provenance de la liaison filaire UART afin de les transmettre par liaisons sans-fil. La liaison est un réseau Ad-hoc non sécurisé fonctionnant en UDP.

Dans un même temps, une nouvelle équipe d'étudiants stagiaires est constituée de février à septembre 2011. Leurs apports dans le projet ibNav sont les suivants :

- mise en place d'une nouvelle architecture matérielle;
- ajout des algorithmes inertiels AHRS/EKF avec une méthode de calibration des capteurs inertiels par EKF.

Lors de la reprise du projet, ma première tâche consistait à mettre en place une batterie pour l'alimentation autonome du prototype et à valider les travaux réalisés sur le plan matériel et logiciel. A ce stade, le projet ne fonctionnait pas, le GUI sur l'iPad recevait des données mais celles-ci étaient faussées et il n'était pas possible de connecter plus de 7 IMUs. Suite aux recherches, de nombreuses erreurs ont été trouvées et les plus importantes sont détaillées en ANNEXE II. Dans un même temps, des solutions ont été appliquées afin de rendre fonctionnel le projet, cependant malgré ces corrections les résultats obtenus n'étaient pas satisfaisants. De plus, certaines erreurs nécessitaient la conception d'un nouveau système, essentiellement pour

l'architecture matérielle. Finalement, il est important de noter qu'à ce stade de nombreuses fonctionnalités sont manquantes et qu'il faudra les ajouter pour respecter les spécifications du projet ibNav. En conséquence, une nouvelle phase de développement a été débutée et certaines parties ont été reprises du début.

1.4.3 Phase 3 (Février 2012 à aujourd'hui)

A ce stade, j'ai repris le projet dans son intégralité pour la phase 3. Cette nouvelle phase consiste à réaliser le prototype, permettant la capture de mouvements et la navigation à l'intérieur des bâtiments, performant, robuste et intégrant toutes les fonctionnalités respectant les spécifications du projet ibNav. Le projet est réparti en 5 sections de développement différentes (Figure 1.1).

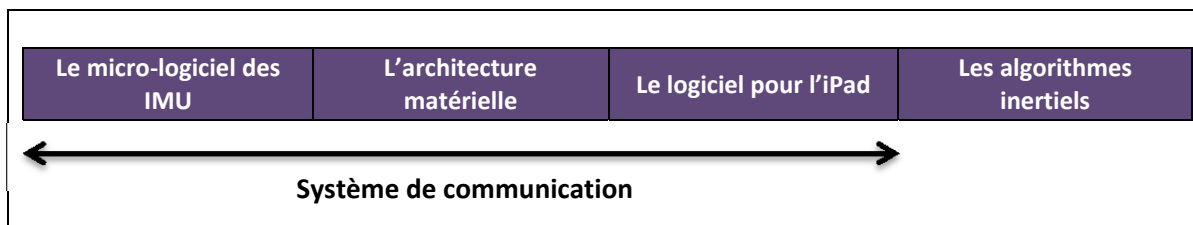


Figure 1.1 Répartition des différentes sections du projet ibNav

Afin de réaliser cette nouvelle phase de développement, chaque section du projet est reprise et une première analyse est effectuée afin de prendre en compte les changements à effectuer ainsi que les fonctions récupérables. Par la suite, une nouvelle base fonctionnelle est créée et chaque fonctionnalité à implémenter suit une nouvelle méthodologie afin de s'assurer qu'elle ne comporte aucune erreur (Figure 1.2).

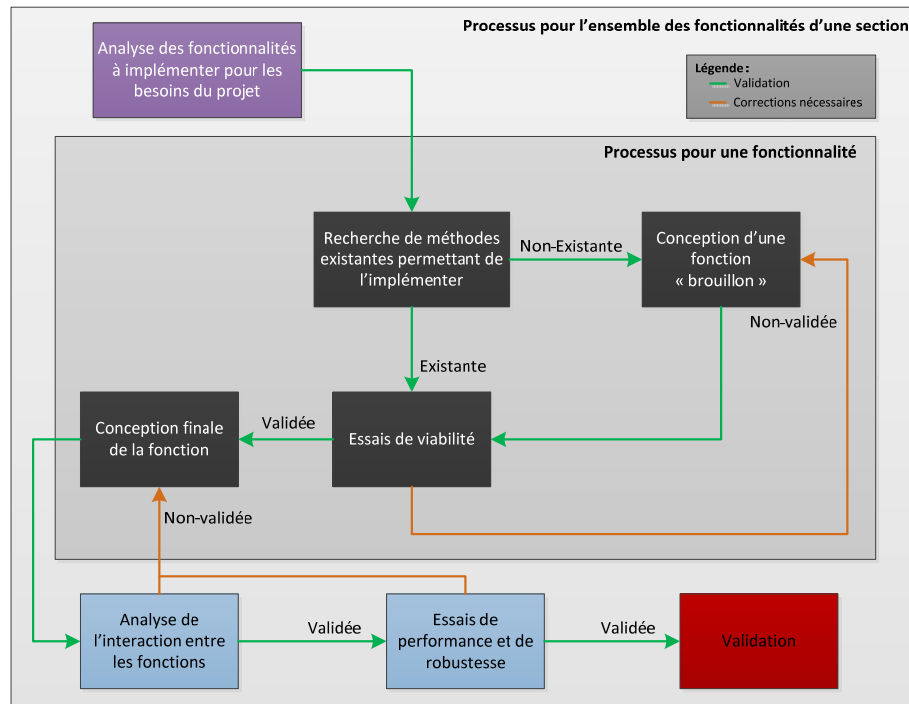


Figure 1.2 Méthodologie de développement pour le projet ibNav

Les différentes tâches à réaliser pour la phase 3 du projet ibNav sont globalement :

Le micro-logiciel des IMU :

Il doit être modifié afin d'offrir une communication globale SPI et UART plus performante tout en offrant de nouvelles fonctionnalités telles que la communication bidirectionnel pour l'UART et la connexion/déconnexion d'IMUs durant le fonctionnement du système pour le SPI. Les données du capteur de température interne doivent être récupérées. La gestion des exceptions doit être prise en compte afin de prévenir les problèmes au niveau de la communication. La consommation de chaque IMUs doit idéalement être plus faible ainsi que la demande de ressources afin de permettre la récupération des données brutes et des données calculées (après application des algorithmes inertiels).

Le système de communication :

Il doit être robuste afin de permettre la récupération des données brutes des capteurs inertiels au niveau de l'iPad. Contrairement à l'utilisation des données calculées, l'utilisation des

données brutes est sensible aux erreurs puisqu'elles sont par la suite appliquées sur des algorithmes inertiels en post-traitement. Il nous faut donc un système avec un nombre de trames manquantes négligeables tendant vers zéro. De plus, le système de communication sans-fil doit pouvoir fonctionner en deux modes différents. Le premier étant une liaison directe entre le système et l'iPad, permettant ainsi de se déplacer où l'on veut dans le bâtiment et le deuxième est une liaison passant par un point d'accès (AP) permettant d'avoir l'internet et de pouvoir ainsi transmettre les données pour du post-traitement.

L'architecture matérielle :

Elle doit être constituée de câbles flexibles afin que le système soit incorporé dans un vêtement et ne nuise pas aux mouvements de l'utilisateur. La répartition énergétique doit être faite de façon étoilée afin que l'intensité circulant dans l'architecture ne nuise pas au système de communication. Le même principe doit également être adopté pour l'horloge de la communication SPI. Des régulateurs de tensions et des optocoupleurs doivent être ajoutés. Un blindage sur les câbles serait également intéressant à mettre en place. De plus, il faudrait mettre des connecteurs sur chaque boîtier contenant un IMUs afin d'éviter des fils cassant au niveau des soudures dû aux mouvements de l'utilisateur. Ces connecteurs permettront également, en cas de panne d'un IMUs, de pouvoir le remplacer facilement et efficacement. Il faut également prévoir un connecteur/câble permettant la reconfiguration de l'IMUs en cas de besoin (mise à jour ou panne du système). Finalement, il faudrait ajouter un écran LCD afin d'avoir un visuel directement sur le bon fonctionnement du système.

Le logiciel pour l'iPad :

Il doit présenter de nouvelles fonctionnalités, que ce soit pour la capture de mouvements ou la navigation à l'intérieur des bâtiments. Il doit également être plus ergonomique et mettre l'accent sur l'esthétique. Elle nécessite de fonctionner sous 3 modes de traitements différents suivant le choix de l'utilisateur :

1. En mode connecté « maître » où les algorithmes sont appliqués au niveau de l'iPad en mode temps réel;

2. En mode connecté « esclave » où les algorithmes sont appliqués soit sur l'iPad « maître », soit au niveau des IMUs en mode temps réel;
3. En mode post-traitement.

Elle doit intégrer une interface pour la localisation des bâtiments et améliorer l'interface pour la capture de mouvements afin de pouvoir visualiser correctement les mouvements du corps en 3D. Puisque c'est également une plateforme permettant l'étude des capteurs inertiels et de ses algorithmes, une interface permettant la visualisation des données brutes et calculées sous forme de graphique sera réalisée ainsi qu'une autre interface permettant l'analyse et la modification des algorithmes.

Les algorithmes inertiels :

Ils doivent être vérifiés dans un premier temps en post-traitement sous Microsoft Visual C++ et Matlab. Cette première étape concernera les algorithmes existants soit l'algorithme de calibration avec EKF et l'algorithme de navigation AHRS/EKF avec ses modèles de connaissances associées. Des améliorations et modifications importantes seront sûrement nécessaires. Dans un deuxième temps, l'algorithme pour la navigation à l'intérieur des bâtiments doit être réalisé soit INS/EKF avec ses modèles de connaissances associées et la méthodologie PDR. Il faudrait également effectuer une dernière fois l'analyse des erreurs stochastiques et déterministes pour chacun des IMUs. Finalement, le robot développé lors de la phase 1 permettra de vérifier la performance des algorithmes inertiels.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

La capture de mouvements ainsi que la navigation à l'intérieur des bâtiments sont effectuées par des techniques existantes depuis de nombreuses années, mais ce n'est seulement qu'à partir des années 2000 que la technologie des IMU-MEMS a commencé à être utilisée pour ces deux types de navigation. Ces capteurs lorsqu'ils sont bas de gamme présentent cependant de nombreuses erreurs faisant dériver la solution, il est donc important de modéliser ces erreurs afin de les corriger. Les algorithmes de navigation par inertie sont de plus en plus nombreux, il est intéressant d'identifier ceux répondant le plus à nos besoins. Finalement, n'étant pas le premier à utiliser ce type de capteurs, les travaux des autres doivent être étudiés afin de porter un regard critique sur les performances possibles et la réalisation de notre objectif.

2.1 Capture de mouvements

La capture de mouvements est classée suivant 3 catégories qui permettent de distinguer les différents emplacements de la source de capture et des marqueurs présents dans l'environnement d'acquisition (Perales, 2001 ; Menache, 2010):

1. Le système dit « Outside-In » classe les systèmes où la source est placée sur le corps et les capteurs sont à l'extérieur (par exemple, les systèmes optiques);
2. Le système dit « Inside-Out » classe les systèmes où les capteurs sont placés sur le corps et les sources sont à l'extérieur (par exemple, la capture électromagnétique);
3. Le système dit « Inside-In » sont les systèmes où les capteurs et les sources sont placés sur le corps humain (par exemple, les systèmes mécaniques et inertiels).

Notre projet utilisant la capture inertielle, nous nous intéresserons d'avantage à ce dernier type de système dit « Inside-In », mais il est tout de même important de passer en revue les autres possibilités existantes afin d'effectuer de la capture de mouvements. La revue de littérature suivante est écrite à l'aide des travaux de : Vlastic et al., 2007 ; Wong, 2007 ; Zhiqiang et al., 2009 ; Menache, 2010.

2.1.1 Capture optique

A l'heure actuelle, c'est la principale technologie utilisée par les industries du divertissement pour effectuer de la capture de mouvements. Cette technologie est répartie en deux catégories :

- pour la catégorie passive, un réseau composé d'au moins 2 caméras est utilisé afin de suivre les mouvements des marqueurs réfléchissants;
- pour la catégorie active, des cellules photosensibles sont utilisées afin de suivre les mouvements LED.

Remarque : Une solution où les caméras sont montées sur l'individu existe également et permet la capture des mouvements de la main, du bras et des points fixés par le porteur (Yokokohji et al., 2002).

Tableau 2.1 Avantages et inconvénients pour la capture optique

Capture optique	
Avantages	Désavantages
<ul style="list-style-type: none"> • un nombre important de marqueurs peuvent être suivis (<i>exemple 128</i>). • le système est dépourvu de câble (les mouvements de l'utilisateur ne sont pas affectés par cette contrainte). • la précision du système est importante. • le nombre d'échantillons par secondes recueillis est important (entre 30 et 2000). • les capteurs sont flexibles. 	<ul style="list-style-type: none"> • la capture doit se faire dans une pièce spécialement adaptée, dotée du matériel approprié (un studio d'enregistrement). Le système est donc peu portable. • le système est très dispendieux et de ce fait cette solution n'est pas envisageable pour beaucoup d'industriels. • la capture peut se révéler impossible lorsqu'un marqueur est occulté pendant trop de temps. • pour des animations très réalistes, la période de post-production peut être très longue, de l'ordre de plusieurs jours.

Les principales industries proposant cette technologie sont Vicon, Codamotion, PhaseScape et Qualisys. Un exemple du système optique de la société PhaseScape est en ANNEXE III.

Conclusion : Cette technologie ne peut pas être utilisée pour notre domaine d'application puisqu'elle ne peut être compatible avec la navigation à l'intérieur des bâtiments et demanderait une infrastructure adaptée, trop lourde à gérer, dispendieuse et non universelle.

2.1.2 Capture mécanique

La capture mécanique aussi dite prosthétique utilise un exosquelette constitué de potentiomètres placés sur le corps humain. Quand l'objet à capter bouge, l'exosquelette suit le mouvement de l'objet et ainsi l'angle de flexion est capturé.

Tableau 2.2 Avantages et inconvénients pour la capture mécanique

Capture mécanique	
Avantages	Désavantages
<ul style="list-style-type: none"> • il offre une très grande précision de mesure (le système mesure directement l'angle de l'articulation, la position n'est pas estimée et les résultats ne sont pas influencés par l'environnement extérieur). • il présente un faible coût par rapport aux systèmes optiques et magnétiques. • le taux d'échantillonnage peut être assez élevé puisque peu de processus sont nécessaires pour extraire les données. • il est portable et simple à utiliser. 	<ul style="list-style-type: none"> • il est peu adapté aux mouvements humains et de ce fait, utilisé plutôt en robotique. Certains mouvements sont difficiles à effectuer. • certaines actions sont limitées et ne peuvent être réalisées tel que le saut. Il n'est pas possible d'en déduire la position absolue par rapport à la position relative dans ce système. • il n'est pas possible de suivre le mouvement d'objets ne présentant pas d'articulations. • les capteurs sont fixés.

Les principales industries proposant cette technologie sont Measurand et Metamotion. Un exemple du système mécanique de la société Metamotion est en ANNEXE III.

Conclusion : Comme la capture optique, cette technologie ne peut être utilisée pour la navigation à l'intérieur des bâtiments puisqu'il n'est pas possible d'en déduire sa position absolue et de ce fait, son déplacement dans un environnement.

2.1.3 Capture magnétique

La capture magnétique permet de détecter la position et l'orientation d'un individu en utilisant des capteurs magnétiques (bien souvent une bobine de fils électriques) mesurant le champ magnétique (différence de potentiel) généré par un transmetteur magnétique.

Tableau 2.3 Avantages et inconvénients pour la capture magnétique

Capture magnétique	
Avantages	Désavantages
<ul style="list-style-type: none"> • les données des capteurs sont traitées en temps réel, elles sont peu interprétées et enlèvent de ce fait le besoin de post-traitement. • il présente un coût plus faible que les systèmes de capture optique. • les marqueurs ne peuvent pas être occultés. • il présente un faible encombrement. 	<ul style="list-style-type: none"> • la présence d'objets métalliques dans l'environnement peut influencer les résultats. • le champ magnétique ne peut avoir une étendue importante, de ce fait l'environnement d'acquisition est limité. • les câbles présents dans le système peuvent causer des contraintes pour l'utilisateur. • les capteurs sont fixés.

La principale industrie proposant cette technologie est Ascension Technologie. Un exemple du système magnétique de la société Ascension Technologie est en ANNEXE III.

Conclusion : Cette technologie présente un environnement de captation restreint et facilement influençable par l'environnement extérieur. Elle ne peut donc être utilisée pour la navigation à l'intérieur des bâtiments.

2.1.4 Capture de mouvements par inertie

Elle permet à l'aide d'IMUs d'acquérir l'attitude (les mouvements relatifs) du corps humain en temps réel. Le système mesure directement la rotation des angles de chaque articulation par l'utilisation de capteurs inertiels qui souvent sont une combinaison d'accéléromètres et de

gyroscopes. On a ainsi l'orientation et la position relative d'un point particulier correspondant à l'emplacement du capteur.

Tableau 2.4 Avantages et inconvénients pour la capture inertielle

Capture de mouvements par inertie	
Avantages	Désavantages
<ul style="list-style-type: none"> • le système présente une très forte portabilité car l'acquisition n'est pas contrainte de s'effectuer à un endroit fixe. • la mesure est directement donnée en 6 ou 9 DOF et de ce fait requiert peu de puissance de calcul. • il est possible de suivre le mouvement d'objet ne disposant pas d'articulations. • le taux d'échantillons est assez élevé (~150 par secondes). • les marqueurs ne peuvent pas être occultés. • les capteurs sont flexibles. 	<ul style="list-style-type: none"> • il n'est pas possible d'obtenir la position absolue directement. • le désavantage majeur est la dérive de la position au fil du temps qui nécessite une calibration initiale méticuleuse et des algorithmes performants. Cette dérive est causée par le fait que la position et l'orientation sont déduites des informations d'accélération et de vitesse angulaire. De ce fait, les erreurs sont accumulées au cours du temps et le système devient instable.

Les principales industries proposant cette technologie sont Xsens, Animazoo, 3DSuit et Measurand. Un exemple du système inertiel de la société Xsens est en ANNEXE III.

Conclusion : Cette technologie répond aux critères pour la navigation à l'intérieur des bâtiments mais elle nécessite l'élaboration d'algorithmes efficaces. La portabilité dont elle fait preuve et la possibilité d'obtenir la position d'un point sans qu'il soit forcément une articulation, en fait une méthode tout à fait adaptée. Elle sera détaillée d'avantage dans la suite du rapport.

2.1.5 Autres technologies de capture

Il existe également d'autres méthodes de capture de mouvements mais plutôt utilisées dans le cadre de la recherche. Ces solutions n'ont pas encore fait preuve d'efficacité par rapport aux technologies précédentes :

- le traitement de l'image est une solution n'utilisant pas de marqueurs mais directement la vidéo pour en déduire les mouvements de l'individu (Yisheng et al., 2005). Cette approche est restée du domaine de la recherche, elle est beaucoup moins précise que les systèmes précédents et nécessite au moins une caméra filmant les mouvements de l'individu. Cependant, elle est plus portable et abordable;
- la capture acoustique ou par fréquence radio utilise le délai d'un signal pour obtenir la localisation du capteur. C'est un système de type « Outside-In » et « Inside-Out » suivant le choix de son concepteur. Cette solution est rarement utilisée à elle seule pour la capture de mouvements, on la retrouve essentiellement combinée avec d'autres technologies;
- les solutions hybrides permettent d'utiliser plusieurs technologies différentes, soit plusieurs solutions de capture de mouvements afin d'obtenir un système plus robuste. Ils sont très nombreux dans la littérature mais pour exemple, on peut citer les systèmes acoustiques et inertiels (Ward et al., 2005), optiques et inertiels (Bachmann, 2000 ; Pons-Moll, 2010), magnétiques et inertiels (Roetenberg, 2006). L'hybridation permet dans ces cas-là de rectifier la dérive de la position et de l'orientation provoquée par l'utilisation de capteurs inertiels. De plus, excepté pour Roetenberg, 2006, ces hybridations font que le système n'est plus « Inside-In » et de ce fait nécessite une infrastructure externe pour fonctionner.

2.2 Navigation à l'intérieur des bâtiments

Suivant l'état de l'art actuel, la navigation à l'intérieur des bâtiments est répartie en trois catégories différentes :

1. Les méthodes dites LPS, soit des solutions nécessitant un réseau complexe de capteurs ayant de bonnes précisions ainsi qu'une bonne couverture;

2. Les méthodes dites IMU, suivant une méthodologie dite PDR, soit des solutions estimant le mouvement d'une personne;
3. Les méthodes hybrides comprenant les deux précédentes dans un modèle d'intégration couplé. L'avantage est qu'elles permettent d'utiliser une infrastructure plus souple, moins onéreuse pour les LPS grâce à l'aide des IMU tout en permettant à ces derniers d'être plus efficaces (Fischer et al., 2008).

2.2.1 Systèmes de type LPS (Sources externes)

Les méthodes de type LPS (Systèmes de positionnement local) sont les principales utilisées mais leurs inconvénients sont qu'elles nécessitent l'installation d'une infrastructure comprenant des balises. Celles-ci peuvent se révéler être un problème conséquent et très coûteux pour des bâtiments de plusieurs étages, de taille importante ou pour un usage privé. De nombreuses applications ainsi que différentes technologies utilisent cette méthode (Vossiek et al., 2003 ; Kolodziej et Hjelm, 2006 ; Retscher et Fu, 2010) :

De nombreuses technologies dans le domaine de la recherche existent pour ce type de navigation, elles sont réparties en 4 types : infrarouge, ultrason, magnétique et radio. Par contre, la solution la plus utilisée et vraiment efficace est la technologie radio. Le principe de calcul de la position est basé sur l'analyse du signal, on retrouve essentiellement dans la littérature les méthodes AOA, RSS, TOA, RTOF et TDOA (voir la section acronyme pour ces termes). Ces technologies sont réparties en deux groupes distincts suivant le rôle de l'unité (sur l'individu) et des balises. De plus, on retrouve deux types d'infrastructures, la première utilisant des systèmes préalablement installés servant à une toute autre utilité que le positionnement (SoOP) et la deuxième nécessitant la mise en place d'une nouvelle infrastructure dédiée au positionnement.

La suite du mémoire n'étant pas consacré à ces technologies, nous verrons sommairement leurs caractéristiques afin de connaître leurs avantages et inconvénients :

- la technologie infrarouge présente l'avantage d'avoir un faible coût et une précision élevée, cependant elle a une portée se limitant à la pièce où l'infrastructure est installée (Want et al., 1992; Aitenbichler et al., 2003);
- la technologie utilisant les ultrasons présente le même avantage que la technologie infrarouge, mais les inconvénients sont plus importants. La précision se limite à plusieurs centimètres et il faut une infrastructure présentant un nombre très important de capteurs pour que ce système se révèle efficace (Hightower et Borriello, 2001; Sanchez et al., 2009). Il s'agit cependant de la technologie privilégiée pour le positionnement sous l'eau;
- la technologie magnétique présente de bonnes précisions, cependant la portée est faible et le coût se révèle plus élevé que les systèmes précédents (Prigge et How, 2002);
- la technologie radio présente une multitude de méthodes ayant des coûts, une précision ainsi qu'une portée très différente de l'une à l'autre. On peut citer la technologie GSM et WLAN (Retscher, 2010) présentant l'avantage d'utiliser une infrastructure déjà existante, la portée est de plusieurs mètres, le coût reste assez faible et malgré le fait que la précision est plus faible que pour les systèmes précédents, elle reste tout de même très acceptable. On entend aussi souvent parler de la méthode Pseudolite-GPS (Edge, 2001) qui consiste à simuler un GPS dans un bâtiment. Cette solution présente une précision et une portée très élevée, cependant elle est parmi les solutions les plus dispendieuses. Finalement, on peut citer les technologies UWB, Bluetooth et RFID (Hallberg et al., 2003, Hui et al., 2007) présentant une faible portée, des coûts variant suivant l'approche utilisée et nécessitant une nouvelle infrastructure.

2.2.2 Systèmes de type PDR

Dans le cadre du projet, nous souhaitons utiliser uniquement des IMUs afin d'obtenir un système à faible coût et furtif (n'utilisant pas d'ondes externes), ne pouvant être influencé par l'environnement extérieur (voir des techniques de brouillage) et ne nécessitant pas d'infrastructure (pouvant fonctionner dans un environnement inconnu). Cependant, comparativement à la plupart des méthodes présentées pour les systèmes LPS, cette méthode nous donne une position relative de l'individu et donne la position absolue en partant d'un

point estimé par l'utilisateur (il est également possible de réinitialiser la solution à l'aide de points de repère occasionnels). On a de ce fait les mouvements estimés d'une personne et on est capable de suivre son déplacement, la méthodologie est appelée PDR (Navigation pedestre à l'estime). On retrouve trois systèmes dans la littérature :

1. Les systèmes utilisant un IMU non-fixé sur l'individu (souvent à l'aide d'un téléphone intelligent). On retrouve principalement dans la littérature l'approche appelée « FootPath » (Link, Smith et al. 2011, Pomp et al., 2012) utilisant le magnétomètre pour détecter la direction de l'individu et l'accéléromètre pour détecter les pas. L'algorithme « FootPath » permet d'estimer le trajet effectué par un pas en utilisant la taille et le sexe de la personne. Lorsqu'un pas est détecté, la position de celui-ci est mesurée. Le nombre de pas est également compté et à partir de ces informations l'algorithme estime la position de l'utilisateur;
2. Les systèmes utilisant un IMU fixé sur le pied de l'utilisateur. Cette méthode utilise essentiellement l'accéléromètre et le gyroscope afin d'obtenir l'accélération et la vitesse angulaire. L'algorithme permet de détecter les pas de l'utilisateur et d'estimer son déplacement ainsi que son orientation. Cette méthode offre de meilleures performances que la précédente (Jiménez et al., 2010). D'autres procédures également utilisent l'IMU fixé sur le torse, celle-ci sont un compromis au niveau des performances entre la technique des IMUs non-fixés et des IMUs fixés sur le pied (Ladetto, 2003);
3. Les systèmes utilisant plusieurs IMUs répartis sur le corps. On retrouve celui de Zhang et al., 2012 qui tend à s'en approcher. Ce système est semblable à celui que nous souhaitons réaliser puisqu'il permet d'obtenir les mouvements du corps en plus du trajet effectué par la personne. Les auteurs utilisent cependant uniquement 6 capteurs inertiels de type MEMS et ceux-ci sont haut de gamme. Un seul IMU fixé sur le pied de l'utilisateur est utilisé pour la navigation intérieur. Les erreurs provenant de ces capteurs sont nettement plus faibles que celles provenant des capteurs bas de gamme. Ils n'ont pas de procédure de calibration à effectuer puisque celle-ci est préalablement effectuée par le constructeur.

2.2.3 Solutions hybride : PDR / LPS

Les méthodes les plus populaires et offrant les meilleures performances sont des méthodes hybrides combinant des IMUs et une solution LPS bien souvent radio. Ainsi, la précision du positionnement est accrue et les inconvénients dus à l'utilisation d'IMUs sont réduits. On peut citer pour exemple les 3 plus populaires :

1. La solution IMU/RFID. Dans ce cas, le RFID (Radio-identification) sert de point de contrôle, il permet d'avoir automatiquement la position initiale de l'utilisateur et de diminuer le biais et la dérive engendré par l'utilisation d'IMUs. Les balises RFID pouvant être passive, l'installation peut se faire très rapidement dans un bâtiment, elle est donc flexible, cependant leur nombre peut se révéler important plus le bâtiment est grand (Renaudin et al., 2007);
2. La solution IMU/Ultrason. Les sonars sont répartis sur le chemin de l'utilisateur afin de le détecter, de connaître sa position absolue et de lui offrir une nouvelle solution permettant de corriger les erreurs des IMUs (Fischer et al., 2008);
3. La solution IMU/WLAN. La technique est similaire à la précédente mais elle peut être utilisée en gardant l'infrastructure existante (Lee et al., 2011).

Conclusion : La technologie dite PDR répond parfaitement aux critères que nous cherchons afin d'effectuer de la navigation à l'intérieur des bâtiments. Puisque nous utilisons également des IMUs pour la capture de mouvements, nous avons un IMU installé sur chacun des pieds de l'utilisateur ainsi que sur le torse. De ce fait, nous utiliserons principalement la méthode PDR avec un IMU fixé sur le pied de l'utilisateur puisque celle-ci offre les meilleures performances pour une solution autonome. Cependant, il sera également intéressant d'utiliser l'IMU fixé sur le torse afin de comparer les résultats de ces deux systèmes (l'algorithme de ce deuxième étant similaire à celui utilisé avec un téléphone intelligent).

2.3 Capteurs inertiels à faible coût

Comme dit précédemment, les capteurs inertiels à faible coût de type MEMS sont utilisés dans des applications de plus en plus nombreuses du fait de leur faible encombrement, de leur coût et de leurs performances tout de même convenables pour de nombreuses applications ne nécessitant pas une forte précision (tel que les appareils photos et les consoles de jeux). Ces capteurs sont regroupés sur des systèmes dit IMUs, soit constitués d'accéléromètres, de gyroscopes et bien souvent de magnétomètres et de capteurs de température, le tout relié à un μC . Lorsque ces capteurs fonctionnent dans les trois axes de leur référentiel, on obtient 3 mesures d'accélération, de vitesse angulaire et de champ magnétique. Ces 9 DOF permettent d'obtenir par la suite l'attitude du mobile (AHRS) et la vitesse ainsi que la position (INS).

Les IMUs sont souvent employés en soutien dans le domaine de la navigation puisqu'ils offrent une solution assez précise sur un cours laps de temps et permettent ainsi d'avoir une solution de navigation lorsque le signal GPS est trop faible ou inexistant (par exemple, lorsqu'une voiture rentre dans un tunnel). Malgré qu'une grande partie des recherches effectuées à ce jour consiste à hybrider les IMUs avec une solution externe permettant de corriger leurs erreurs, on retrouve tout de même des recherches portées sur l'utilisation autonome d'IMUs. Une erreur typique inférieure à 1% est souvent obtenue dans la littérature pour les IMUs bas de gamme récents (Jiménez et al., 2010; Yan Li, 2012). Cependant, ces systèmes peuvent fonctionner uniquement durant un faible laps de temps du fait de l'accumulation des erreurs finissant par faire diverger la solution. Cette conséquence s'explique par le fait qu'il n'y a pas de solutions externes venant corriger la position, la vitesse et/ou l'attitude du mobile et qu'il n'est pas possible de modéliser tous les types d'erreurs des capteurs afin de les corriger.

Ces erreurs entraînant la dérive de la solution sont de deux types, on retrouve les erreurs de type déterministe et de type stochastique. Une fois les erreurs caractérisées, il faut pouvoir modéliser une solution afin de les corriger.

2.3.1 Caractérisation des erreurs de nature déterministe

Ce premier type d'erreurs est la source primaire d'erreurs que l'on retrouve dans les systèmes AHRS et INS utilisant un IMU à faible coût. Ces erreurs sont de type non-aléatoire et peuvent dans la plupart des cas être estimées et de ce fait corrigées à l'aide d'une méthode de calibration. A noter qu'au départ, la documentation technique de chaque capteur nous donne sa sensibilité, son biais, son bruit blanc, sa dépendance envers la température, sa non-linéarité, etc. Il est cependant utile de recalculer ces informations car celles-ci sont théoriques et souvent elles ne se révèlent pas très fiables puisqu'en réalité, elle change d'un capteur à l'autre d'une même série. Elles servent cependant de base pour le concepteur.

Les erreurs modélisables que l'on retrouve dans la littérature sont les suivantes (Lavoie 2012, Zeljko, 2012): le biais, le facteur d'échelle, la non-linéarité, le mauvais alignement des axes ainsi que l'influence de la température. Lorsqu'on utilise des magnétomètres, on parle de deux types d'erreurs déterministes créant une distorsion du champ magnétique terrestre (Ladetto, 2003 ; Lavoie 2012) : les erreurs de fer dur et les erreurs de fer doux du champ magnétique. Ces différentes erreurs seront détaillées d'avantage dans le CHAPITRE 3.

2.3.2 Modélisation des erreurs de nature déterministe

Une fois les erreurs caractérisées, il faut les modéliser afin de pouvoir les corriger. La méthode la plus simple pour les capteurs de type MEMS bas de gamme est d'utiliser une méthode multi-positions à 4 positions (Figure 2.1) afin d'obtenir une mesure estimée de la valeur de chaque axe du capteur inertiel. Cette méthode simple doit permettre de corriger le biais, le facteur d'échelle et le mauvais alignement des axes pour l'accéléromètre et le magnétomètre (elle ne permettra pas de corriger le facteur d'échelle et le mauvais alignement des gyroscopes). Cette calibration peut également être améliorée en prenant en compte l'effet de la température sur le biais et le facteur d'échelle. D'après l'auteur (Zeljko, 2012), la

majorité des erreurs peuvent être compensées en réalisant des scénarios statiques et en utilisant les données de la documentation technique. Cette méthode de correction reste très simple et n'est pas optimale.

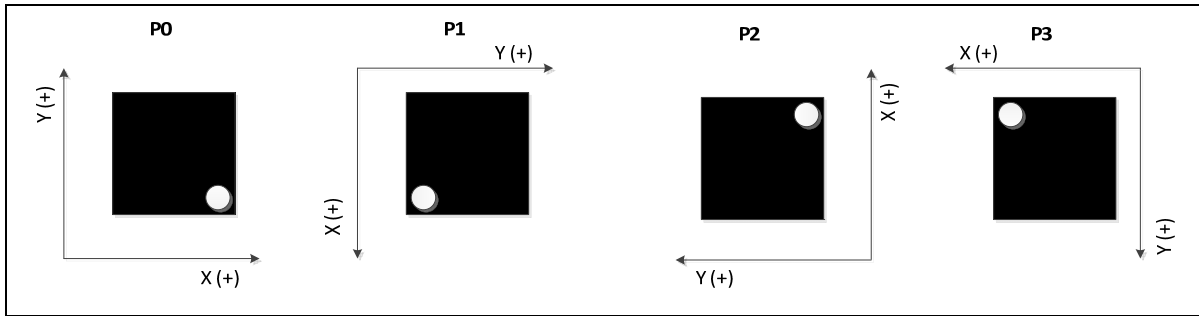


Figure 2.1 Méthode de calibration à 4 positions
Adaptée de Zeljko (2012)

D'autres auteurs suggèrent plutôt d'utiliser une méthode à 18 positions (Figure 2.2), (Shin et al., 2002). Certains systèmes de la littérature requièrent un alignement précis (Syed et al., 2007) mais certains d'entre eux permettent d'utiliser cette méthode avec moins de précision même s'il faut maintenir stationnaires les différents états durant l'acquisition (Dorveaux et al., 2009). Ces méthodes utilisent un vecteur de référence du capteur à calibrer, soit la gravité théorique terrestre pour les accéléromètres, la vitesse de la rotation de la terre pour les gyroscopes et le champ magnétique terrestre pour les magnétomètres.

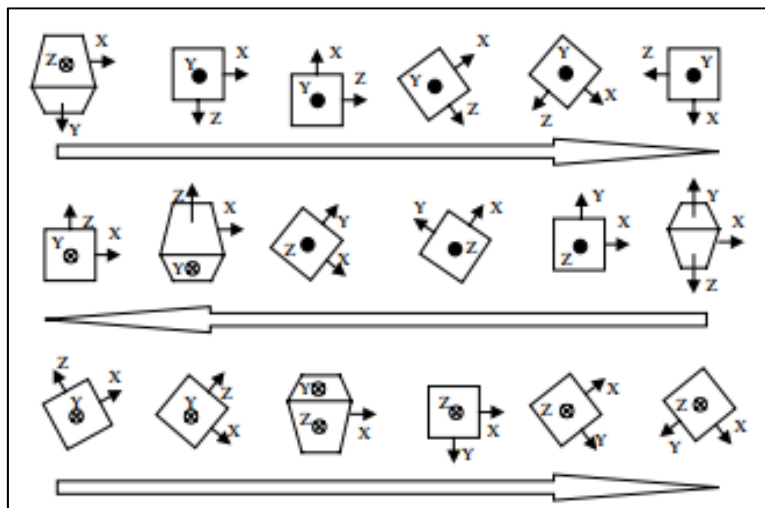


Figure 2.2 Méthode de calibration à 18 positions
Tirée de Syed et al (2007)

D'après plusieurs auteurs, c'est un problème non-linéaire pouvant être résolu par des méthodes itératives telles que les moindres carrés (Dorveaux et al., 2009). D'après Lavoie, 2012, ces méthodes présentent cependant deux problèmes majeurs :

1. Les 18 positions doivent être effectuées de façon très précise;
2. La condition de stationnarité des IMUs doit être respectée en tout temps.

Ce dernier auteur propose de ce fait une solution avec l'utilisation d'un filtre de Kalman EKF réglant ces deux limitations. L'idée d'utiliser ce type de filtre pour la calibration a été introduite par Grewal, 1991. La calibration peut continuer de s'effectuer même si l'état n'est pas stationnaire et il n'est pas nécessaire d'effectuer les 18 positions de façon très précise. Cependant, la méthode mise en place ne permet pas de corriger le facteur d'échelle et le mauvais alignement des axes des gyroscopes, de plus elle ne prend pas en compte l'influence de la température, ni les erreurs de fer doux des magnétomètres. Finalement, cette calibration s'effectue uniquement à l'initialisation.

Afin de corriger le facteur d'échelle et le mauvais alignement des axes des gyroscopes, la méthode de Fong et al., 2008 peut être intéressante à suivre puisqu'elle n'utilise aucun équipement externe dispendieux. Ils utilisent le champ de gravité comme référence afin d'effectuer cette calibration.

Concernant l'influence de la température sur la solution, Bekkeng, 2009 propose une méthode afin d'éliminer cette erreur sur la sensibilité du gyroscope. Pour la caractériser, il utilise une méthode simple comprenant les données du gyroscope obtenues lors de scénarios statiques et d'un capteur de température. La température est par la suite compensée en entrée avant l'utilisation d'un filtre de Kalman EKF permettant de corriger le biais, le facteur d'échelle et le mauvais alignement des axes. Une autre méthode permettant de corriger l'influence de la température à l'aide d'un réseau de neurones procédant durant le fonctionnement du système est présenté par Quinto et al., 2009, elle est cependant plus complexe. D'autres auteurs annoncent que la température suit un modèle non-linéaire et qu'elle se révèle difficile à caractériser.

Concernant la calibration des magnétomètres, Renaudin, 2010 spécifie qu'il est très important de prendre en considération les erreurs de fer doux et de fer dur. Ceux-ci pouvant causer d'importantes erreurs dans la solution de navigation. Dans son cas, il effectue une calibration des magnétomètres en déterminant l'équation de l'ellipsoïde de référence.

D'après Batista, 2011, le biais des capteurs inertiels varie lentement dans le temps et nécessite de ce fait une calibration en ligne, continuant de s'effectuer durant le fonctionnement du système. Vasconcelos et al. 2011 propose une méthode permettant de corriger les erreurs des magnétomètres durant le fonctionnement du système. Cette méthode permet essentiellement de corriger les erreurs de fer doux.

Remarque : De nombreuses méthodes recommandées nécessitent des éléments externes afin d'effectuer une calibration précise. Cependant puisque l'on souhaite conserver le faible coût du projet ibNav, nous ne pouvons pas utiliser de matériels externes dispendieux. On retrouve les méthodes suivantes :

- une méthode utilisant des centrifugeuses de précision pour la calibration précise des accéléromètres (IEEE, 2009);
- une méthode permettant de corriger le facteur d'échelle et le mauvais alignement des axes des gyroscopes à faible coût en utilisant une plaque tournante de précision (Syed et al., 2007 ; Bastista, 2011);
- Renk et al., 2005 utilisent un bras robotisé afin d'effectuer la calibration des accéléromètres et des magnétomètres. Cette méthode est également reprise par Beravs, 2012;
- Bo et al., 2010 considère que la température suit un modèle linéaire et qu'elle influence directement les valeurs de sortie des capteurs venant modifier le facteur d'échelle et le biais. Ils utilisent les résultats d'une chambre thermique et par la suite un algorithme utilisant la méthode des moindres carrés afin de modéliser les erreurs.

Conclusion : Pour le projet ibNav, une méthode multi-positions utilisant un filtre de Kalman EKF sera développée comme base pour la calibration des erreurs déterministes. Elle sera complétée avec une méthode de détection robuste des états stationnaires. Il serait également intéressant de mettre en place la calibration durant le fonctionnement du système afin de corriger plus efficacement les erreurs de fer doux des magnétomètres. Finalement, l'influence de la température sera prise en compte à l'aide d'une méthode simple vue précédemment, considérant les valeurs théoriques de la documentation technique comme référence de l'influence de la température. Un algorithme expérimental basé sur Fong et al., 2008 sera également développé afin de corriger le facteur d'échelle et le mauvais alignement des axes des gyroscopes.

2.3.3 Caractérisation et modélisation des erreurs de nature stochastique

Les erreurs de nature stochastique correspondent à des erreurs aléatoires présentes dans les capteurs inertiels. Ces erreurs sont divisées en 2 catégories :

- **les composantes à haute fréquence** : on retrouve dans cette catégorie le bruit blanc, celui-ci peut facilement être éliminé à l'aide d'un filtre passe-bas ou dans notre cas, à l'intérieur du filtre de Kalman EKF, qui inclut cette correction avec sa matrice de covariance de bruit;
- **les composantes à basse fréquence** : on retrouve l'instabilité du biais, le bruit corrélé, le bruit thermique. Malheureusement, ces erreurs ne peuvent pas facilement être corrigées car elles nécessitent l'utilisation de modèles mathématiques complexes.

El-Sheimy et al., 2008 propose une méthode permettant de caractériser ces différentes erreurs stochastiques. D'après l'auteur, il est préférable de travailler dans le domaine temporel et de ce fait la méthode la plus simple utilisée afin de caractériser les erreurs stochastiques des capteurs bas de gamme est la méthode de la variance d'Allan (AVAR). Cette méthode nous donne l'erreur efficace (RMS) de la dérive aléatoire des capteurs en fonction d'une moyenne de temps. Le résultat est par la suite simple à interpréter et à comprendre (Figure 2.3).

Les autres méthodes utilisées dans cet objectif sont les modèles autorégressifs (AR), le modèle de Gauss-Markov (GM) et la densité spectrale de puissance (PSD). D'après Zhao et al., 2009, la meilleure méthode, par sa simplicité et son efficacité, est la méthode de la variance d'Allan. Cependant comparativement à la méthode PSD, celle-ci peut nécessiter de très longs enregistrements en mode statique (d'une dizaine d'heures) en sachant qu'ils ne doivent pas subir de perturbations. Par exemple, si une personne entre dans le lieu de l'enregistrement, elle peut venir fausser les résultats.

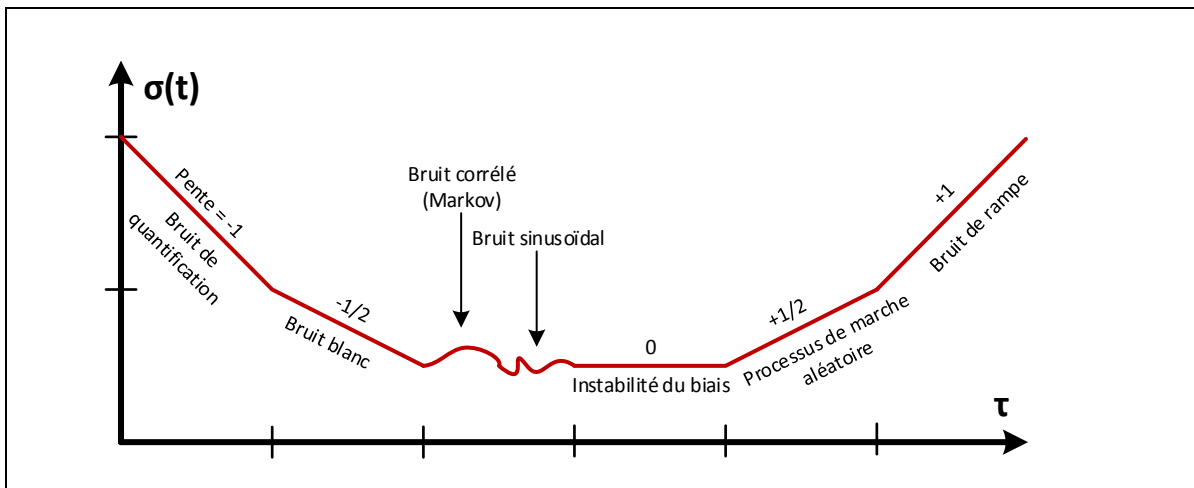


Figure 2.3 Résultats d'une analyse des erreurs stochastiques par la variance d'Allan (échelle logarithmique)
Adaptée de Lavoie (2012)

Concernant la modélisation des erreurs des composantes de basse fréquence, le modèle mathématique principalement utilisé est le processus de Gauss-Markov (Park et al., 2006). Ces modèles ne seront pas étudiés dans le cadre du projet.

Conclusion : Nous utiliserons la méthode de la variance d'Allan afin d'effectuer une étude du bruit blanc présent dans nos différents capteurs. Cette étude sera menée sur chaque capteur de chaque IMU.

2.4 Algorithmes de navigation par inertie

Les algorithmes pour la capture de mouvements sont utilisés afin d'obtenir l'attitude de chaque IMU (modèle AHRS) alors que les algorithmes pour la navigation à l'intérieur des bâtiments permettent d'obtenir la position et la vitesse en plus de l'attitude (modèle INS).

2.4.1 Algorithmes pour la capture de mouvements

L'algorithme le plus fréquemment utilisé que l'on retrouve dans la littérature consiste en une méthodologie AHRS/EKF. Le filtre de Kalman EKF permet d'évaluer l'état d'un modèle non-linéaire à partir d'un modèle contenant plusieurs mesures bruitées. Il est divisé en deux phases, une de prédiction et une de correction. Après la phase de prédiction, le vecteur d'état est mis à jour et les mesures sont corrigées en sortie à l'aide du gain de Kalman.

Xiaoping et al., 2006 et Sabatini et al., 2006 présentent un modèle où les gyroscopes sont utilisés dans la phase de prédiction pendant que les accéléromètres et les magnétomètres sont utilisés pour compenser la dérive de l'intégration de la vitesse angulaire dans la phase de correction. En particulier, les accéléromètres fournissent une référence d'attitude utilisant l'accélération gravitationnelle pendant que les magnétomètres fournissent une référence de cap utilisant le champ magnétique terrestre. Le filtre est basé sur l'estimation du vecteur quaternion (représentant l'orientation du mobile), l'utilisation des quaternions permet d'optimiser les calculs et d'éviter les singularités que l'on retrouverait en utilisant les matrices de rotation DCM. Par la suite, le résultat est affiché sous forme des angles d'Euler, soit le lacet, le tangage et le roulis. La fusion des capteurs que permet cet algorithme permet la compensation des erreurs typiques de chaque capteur.

Les différents modèles étudiés dans la section 2.5 reprennent la même technique afin d'obtenir l'orientation de chaque IMU permettant d'en déduire le mouvement de chaque membre.

Une autre méthode, développée par Madgwick, 2011, est un algorithme AHRS n'utilisant pas le filtre de Kalman mais une méthode de filtrage utilisant l'algorithme de Descente du

Gradient (GD). Cette méthode présente de nombreux avantages pour l'implémentation sur un μC :

- le processus est peu coûteux en calcul;
- il utilise les données des magnétomètres;
- il peut fonctionner avec un faible taux de récupération des données;
- uniquement deux paramètres doivent être ajustés;
- il utilise le quaternion afin d'éliminer les singularités.

Conclusion : Nous utiliserons un modèle équivalent AHRS/EKF pour la capture de mouvements tout en incorporant deux modèles de connaissances permettant de détecter les états stationnaires et quasi-stationnaires de la vitesse angulaire. Nous étudierons l'influence de ces modèles et leur efficacité. La méthode de Madgwick, 2011 sera également implémentée et les résultats seront comparés en terme d'erreur efficace (RMS) et de puissance de calcul.

2.4.2 Algorithmes pour la navigation à l'intérieur des bâtiments

Notre environnement ne dispose d'aucune infrastructure pré-installée et nécessite de ce fait l'utilisation d'un système autonome. La plupart des systèmes autonomes permettent uniquement d'obtenir l'attitude du mobile, lorsque l'on souhaite utiliser un modèle INS afin d'obtenir la vitesse et la position, il y a de nombreux problèmes qui se posent. La méthodologie PDR nécessite de ce fait un modèle de connaissance ZUPT (Détection de la vitesse nulle) afin de contenir les erreurs du système et de permettre une solution de navigation.

Walder et al., 2010, Borensten et al., 2007 et Torrieri et al., 2011 se servent de cette méthode afin d'améliorer le positionnement puisqu'elle peut détecter les différents types de déplacement (marche, course et ramper) tandis que Kwakkel et al., 2007 se sert de cette méthode afin de déterminer la posture du sujet. Skog, 2010 utilise en plus des capteurs de pression afin de fournir des données supplémentaires sur la posture du sujet.

La méthode BIM employée par Walder et al., 2010 permet d'obtenir des informations sur le bâtiment. Cette méthode permet de rectifier la position de l'utilisateur en prenant en compte la carte du bâtiment. En plus de prendre en compte l'architecture, il se fit à la probabilité de déplacement de l'utilisateur dans l'environnement. SPASSOV, 2007 a une méthode similaire prenant en compte les espaces de circulation du bâtiment.

Pomp et al., 2012 présente la méthode SLAM permettant de prendre connaissance d'une carte sans la connaître préalablement, la carte est découverte au fur et à mesure du déplacement de l'utilisateur. Cette méthode serait très intéressante dans notre projet puisque normalement nous ne connaissons pas la carte intérieure du bâtiment, mais elle reste assez complexe à mettre en place lorsqu'on utilise seulement des IMUs.

Vissière et al., 2007 propose une méthode permettant de prendre en compte le champ magnétique puisque celui-ci est facilement perturbé dans un bâtiment du fait de la structure, de l'équipement électrique, des ordinateurs, des téléphones cellulaires, etc. Ces données peuvent être utilisées afin d'obtenir une meilleure estimation de la position. Le système fonctionne ainsi : si le corps bouge, le champ magnétique doit changer en suivant les lois de Maxwell. Si celui-ci ne change pas significativement, alors le corps n'est pas en mouvement. D'après l'auteur, cette méthode permet de déduire des informations précieuses sur la vitesse de l'individu portant les IMUs.

Wang, 2011 présente un algorithme permettant d'utiliser les accéléromètres et les gyroscopes d'un téléphone intelligent. Cet algorithme peut également être repris pour un capteur positionné sur le torse, il est cependant moins performant que les algorithmes utilisant un IMU fixé sur le pied de l'utilisateur.

Finalement, la méthode développée par Jiménez et al. 2010 présente un modèle IEZ (INS-EKF-ZUPT) tout à fait intéressant. En effet, outre le modèle de connaissance ZUPT, elle utilise de nombreux autres modèles de connaissance tels que la détection des états statiques et

dynamiques, ZARU, HDR et Compass. Dans leur étude, ils utilisaient des capteurs MEMS haut de gamme où l'erreur calculée était de 1%. Il sera intéressant de tester cette méthode sur les capteurs MEMS bas de gamme.

Remarque : Majoritairement, les différentes littératures vues ci-dessus maintiennent qu'il faut avoir une méthode externe afin de corriger la position de temps à autre durant le déplacement de l'utilisateur. Dans le cas contraire, la solution finie par diverger. Nous utiliserons les tous derniers développements pour la réalisation de notre méthodologie PDR.

2.5 Analyse des systèmes existants utilisant une technologie semblable d'IMU

Guangyi et al., 2011 développe une méthode utilisant uniquement 4 IMUs, ceux-ci sont dotés d'un module sans fil ZigBee relié à chaque IMU par une liaison SPI mais enlevant de ce fait la contrainte imposée par les fils. Cependant, d'après leurs analyses, leur système n'offre pas de très bonnes performances au niveau du taux de données pouvant être transférées lors de l'utilisation de plusieurs IMUs. (Il n'y a pas de résultats concernant la performance des algorithmes).

Prayudi et Doik, 2012 présente également un système équivalent au nôtre devant pouvoir fonctionner avec 15 IMUs, cependant ils n'en utilisent que 3 dans leur projet, ceux-ci sont placés sur le bras de l'utilisateur. Ils ont cependant une fréquence de données à 100Hz pour les accéléromètres, les gyroscopes et les magnétomètres travaillant dans les 3 axes de leur référentiel. La communication des modules vers le PC s'effectue par liaison UART mais celle-ci monte à plus de 921.6 Kbps, ce qui permet cette fréquence. La communication s'effectue en série (chaque IMU est doté de 2 ports UART). Leurs résultats sont cependant discutables.

Abbate et al, 2011 ont créés leur propre IMM en se basant sur les mêmes IMUs que ceux que nous utilisons. Ils utilisent un système de 10 IMUs communiquant par liaison RS485 à 4.5 Mbits/s. La fréquence de transmission des données est de 100Hz. La méthodologie pour

obtenir l'orientation locale de chaque IMM est également de type AHRS/EKF. Pour leur système, l'erreur sur l'angle de roulis et de tangage est de 0.5° et celui sur l'angle de lacet est de 1.5° en scénario quasi-statique. Ces erreurs sont plus que doublées en scénarios dynamique. Pour eux, ces erreurs proviennent aussi du fait que le microcontrôleur ne travaillant pas en virgule flottante, les algorithmes doivent être optimisés et le nombre de données pouvant être traités est limité.

Abbate et al., 2009 crée un système semblable au précédent, excepté que chaque IMU est doté d'un module sans fil ZigBee pour le transfert des données. Ils ont également une carte MicroSD permettant d'enregistrer les données localement. L'algorithme utilisé est de type AHRS/EKF.

Kap-Ho et al., 2011 crée un système permettant d'apprendre l'équitation afin de donner des informations à l'utilisateur sur sa posture. Le système devrait incorporer jusqu'à 10 IMUs placés dans un vêtement. La transmission vers le PC s'effectue par liaison sans-fil, elle est filaire entre les IMUs. Ils utilisent également une méthode semblable utilisant un filtre de Kalman.

Bakhshi et al., 2011 crée un système composé de 3 IMUs communiquant par liaison sans-fil Bluetooth. Ce système est créé pour la réhabilitation de personnes ayant subis un traumatisme. Aucune information n'est donnée sur les algorithmes, ceux-ci sont effectués par un logiciel externe. Ils utilisent un téléphone intelligent gérant l'enregistrement des données, calculant les mouvements et transférant les résultats sur un serveur internet.

Zhang et al., 2012 est un des seuls systèmes proposant aussi bien la capture de mouvements que la navigation à l'intérieur des bâtiments. Ils utilisent également les quaternions, une méthode AHRS/EKF. Afin d'effectuer la navigation à l'intérieur des bâtiments, ils implémentent une méthodologie ZUPT se servant des données des gyroscopes afin d'obtenir la vitesse et la position. Avec leur système, ils obtiennent la position en 3D des utilisateurs dans le cas de la marche et de la course. Cependant, comme dans les précédents systèmes, les erreurs

sont accumulées au cours du temps et la solution fini par diverger. Dans leur cas, ils prévoient d'ajouter une solution externe RFID à leur système afin de pouvoir réinitialiser la position initiale.

2.6 Synthèse de la revue de la littérature

Les deux premières parties de la revue de la littérature nous permettent d'expliquer nos choix technologiques pour le projet ibNav. En effet, nous voyons que les inconvénients causés par les autres technologies n'utilisant pas de systèmes IMUs ne nous permettent pas de les utiliser efficacement pour effectuer de la capture de mouvements et de la navigation à l'intérieur des bâtiments. L'hybridation n'est également pas envisageable puisqu'elle ne répond pas aux spécifications du projet en nécessitant l'utilisation d'ondes externes. De ce fait, la furtivité du système serait perdue même si le faible coût pourrait demeurer.

La troisième partie est consacrée aux méthodes de calibration des capteurs inertiels à faible coût. Puisque nous souhaitons conserver le faible coût du système, nous n'avons pas à notre disposition une plaque tournante ou une chambre thermique. La première permettrait de calibrer le mauvais alignement des axes et le facteur d'échelle des gyroscopes et le deuxième permettrait de caractériser la température de manière efficace. Cependant, la méthode que nous mettons en place utilisant un filtre EKF et la méthode multi-positions permet de calibrer les erreurs les plus importantes des accéléromètres et des magnétomètres ainsi que le biais des gyroscopes. Un algorithme expérimental sera développé et testé pour le facteur d'échelle et le mauvais alignement des axes des gyroscopes. Concernant la température, une méthode simplifiée et linéaire sera utilisée mais il restera à démontrer son efficacité.

La quatrième partie concerne la capture de mouvements et la navigation à l'intérieur des bâtiments. Pour la première, un algorithme AHRS/EKF incluant des modèles de connaissances sera développé puisqu'il fait partie des algorithmes les plus efficaces où le plus d'optimisations est possible. L'algorithme utilisant le gradient descendant sera également implémenté puisqu'il est « open source » et permettra de comparer nos résultats.

Pour le deuxième, un modèle INS/EKF basé sur la méthodologie PDR sera développé. Celui-ci utilisera de nombreux modèles de connaissances tel que : HDR, HARU, ZUPT, COMPASS et BIM.

Rapport-Gratuit.com

CHAPITRE 3

PRINCIPES DE LA NAVIGATION INERTIELLE

Afin de comprendre la suite du mémoire, ce chapitre décrit les concepts fondamentaux en navigation, suivi de l'architecture des formats représentant les différentes navigations. Par la suite le fonctionnement d'un IMU et de ses capteurs est expliqué ainsi que les différentes sources d'erreurs. Finalement, ce chapitre reprendra les formules de bases des équations de navigation et des modèles utilisés dans le cadre de ce projet.

3.1 Concepts en navigation

Lorsqu'on utilise un IMU, différents repères orthogonaux sont utilisés afin d'obtenir la position d'un point par rapport à une référence choisie. Il existe des relations entre chaque repère, liées par des matrices de rotation, celles-ci sont d'ailleurs utilisées pour la représentation de l'attitude, de la vitesse et de la position du mobile (Figure 3.1). Puisque nous voulons par la suite utiliser ces données pour de la capture de mouvements et de la navigation à l'intérieur des bâtiments, il est nécessaire de les convertir sous un format standard.

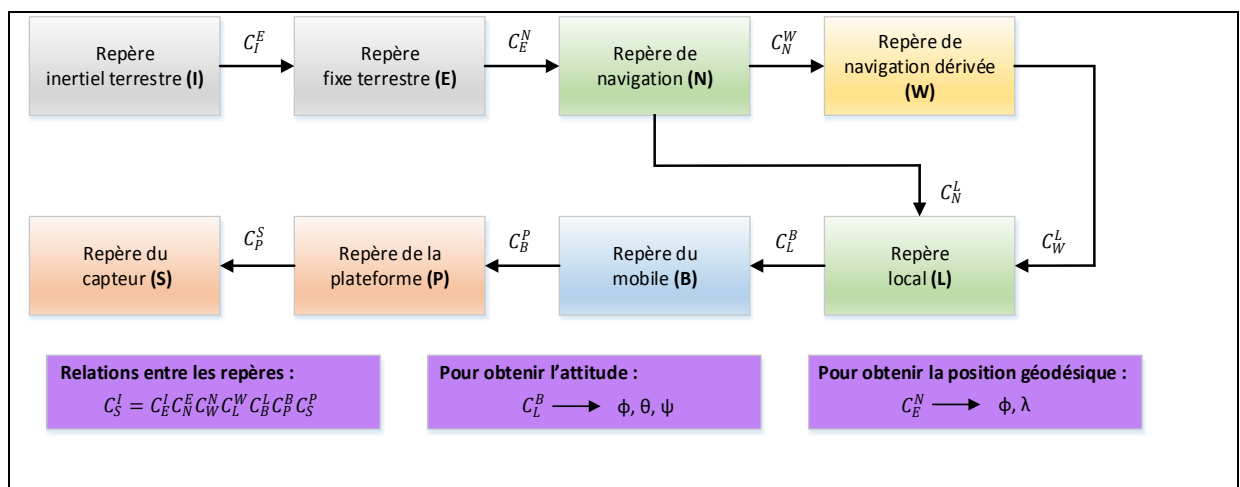


Figure 3.1 Relation entre les différents repères orthogonaux et lien avec l'attitude et la position du mobile

3.1.1 Repères de référence

On dénombre 7 repères principaux qui sont utilisés pour la navigation par inertie (Figure 3.2). La section suivante décrit sommairement chacun de ces repères.

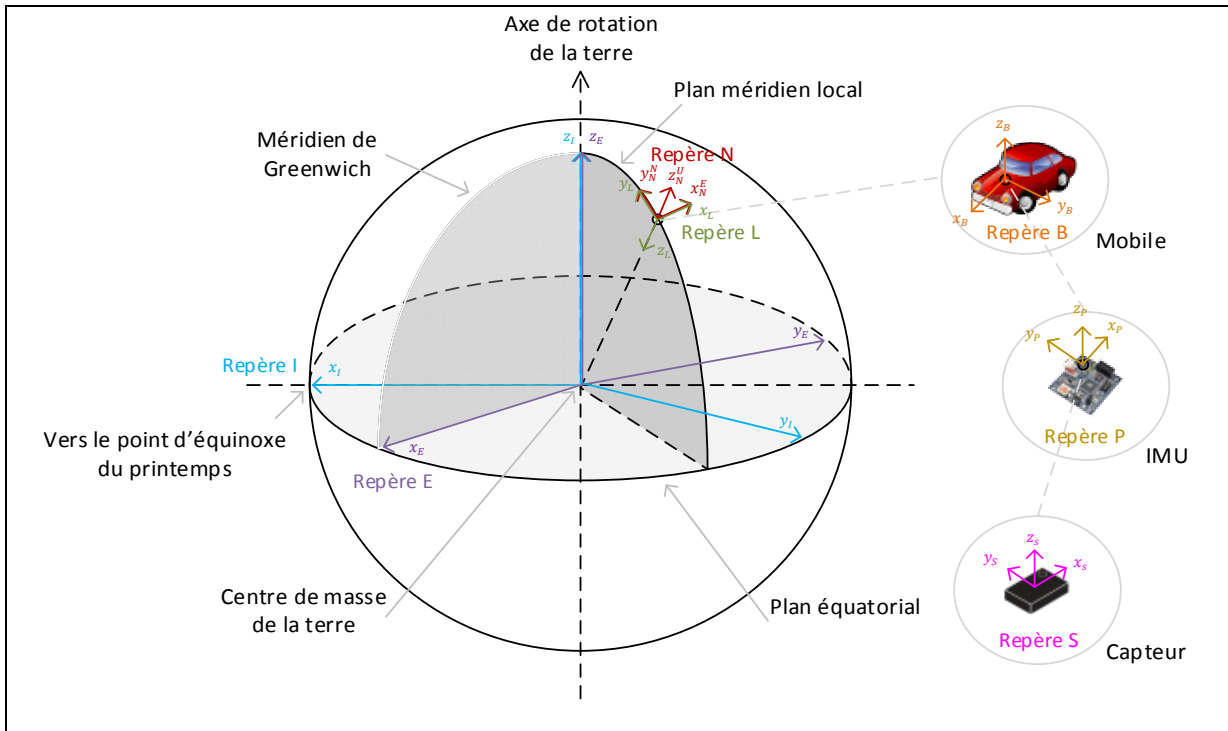


Figure 3.2 Représentation des différents repères terrestres

Le repère inertiel terrestre (ECI - I) :

L'origine de ce repère est le centre de gravité de la terre. En coordonnées cartésiennes, l'axe x_I pointe vers le point d'équinoxe du printemps (ou vernal). Celui-ci correspond au croisement entre l'écliptique (trajectoire du soleil vue de la terre) et l'équateur céleste (projection de l'équateur terrestre). L'axe z_I est l'axe de rotation de la terre et l'axe y_I complète le trièdre. On utilise ce repère car les lois de Newton s'appliquent du fait que les axes restent fixes par rapport à la rotation de la terre. La valeur mesurée par les capteurs inertiels est toujours par rapport à ce repère de navigation.

Le repère fixe terrestre (ECEF - E) :

L'origine du repère est le même que celui du précédent. En coordonnées cartésiennes, l'axe x_E pointe vers l'intersection entre le méridien de Greenwich et le plan équatorial. L'axe z_E est l'axe de rotation de la terre et l'axe y_E complète le trièdre. Ce repère suit la rotation de la terre, il tourne autour de l'axe z_E à la même vitesse que la Terre. On obtient alors la relation entre le repère ECEF et ECI qui est de ce fait une vitesse angulaire constante $\omega_{IE} = 7.292115E^{-5} \text{ rad/s}$. Ce repère est très utilisé en coordonnées géodésiques, c'est-à-dire sous la forme de Latitude (ϕ), Longitude (λ) et d'Altitude ellipsoïdale (h). La conversion réciproque entre le repère géodésique et cartésien peut être facilement réalisée à l'aide d'équations mathématiques (Voir ANNEXE IV). La Figure 3.3 est une représentation de cette transformation.

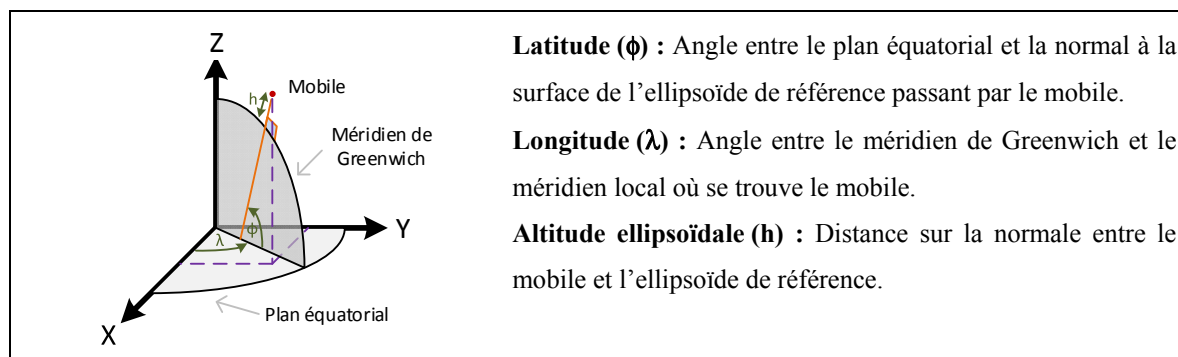


Figure 3.3 Représentation de la transformation entre les coordonnées géodésiques et cartésiennes ECEF

Le repère de navigation (N) :

L'origine du repère est le centre de masse du véhicule. Ces axes pointent respectivement vers le Nord géographique (Nord vrai) y_N^N , l'Est x_N^E et la verticale locale complétant le trièdre z_N^U . Dans le cadre de ce mémoire, le repère est sous la forme ENU (Est, Nord, Haut). **Le repère de navigation dérivé (W)** est un repère intermédiaire où l'axe z_N^U subit une rotation d'angle α afin d'éviter les singularités pouvant être causées par les pôles.

Le repère local (L) :

Ce repère que l'on peut également appeler le repère local inertiel est sensiblement le même que le repère de navigation au niveau des axes, la verticale locale est cependant orientée vers le bas. Il correspond au repère aussi appelé NED (Nord, Est, Bas). L'attitude du mobile est représentée par rapport à ce repère.

Le repère du mobile (B) :

Ce repère a comme origine le centre de masse du mobile. L'axe x_B suit l'axe longitudinal du mobile (roulis), l'axe y_B est orienté vers la droite du mobile (tangage) et l'axe z_B vers le bas de celui-ci (lacet). L'orientation de ce repère par rapport au repère local est d'écrit par les angles d'Euler. Ce repère est utilisé afin d'obtenir l'attitude du mobile à l'aide des capteurs inertiels.

Le repère de la plateforme (P) :

Ce repère est sensiblement le même que le précédent excepté que son origine correspond au centre de masse de l'IMU. Dans le cas idéal, celui-ci correspond au repère du mobile, autrement des erreurs de mauvais alignement doivent être prises en compte.

Le repère du capteur (S) :

Ce repère est similaire au précédent sauf qu'il est directement lié aux valeurs brutes obtenues en sorties du capteur (en digit) avant toutes modifications telles que la prise en compte de la sensibilité. Afin de faire correspondre ce repère avec le repère de la plateforme, des erreurs provenant du mauvais alignement des axes, du facteur d'échelle et du biais sont à déterminer et à corriger.

3.1.2 Représentation et propagation de l'attitude du système

L'attitude du système correspond à une matrice de rotation entre le repère local et le repère du mobile. La convention choisie dans ce mémoire est de représenter le repère local par rapport au repère du mobile, soit la matrice C_B^L . Les éléments de la matrice sont les suivants :

$$C_B^L = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (3.1)$$

Cette matrice de rotation fait référence à une DCM (Matrice en cosinus de direction). En choisissant les axes (x_L, y_L, z_L) pour le repère local et (x_B, y_B, z_B) pour le repère du mobile, on obtient la DCM suivante :

$$C_B^L = \begin{bmatrix} \cos(\alpha_{x_L, x_B}) & \cos(\alpha_{x_L, y_B}) & \cos(\alpha_{x_L, z_B}) \\ \cos(\alpha_{y_L, x_B}) & \cos(\alpha_{y_L, y_B}) & \cos(\alpha_{y_L, z_B}) \\ \cos(\alpha_{z_L, x_B}) & \cos(\alpha_{z_L, y_B}) & \cos(\alpha_{z_L, z_B}) \end{bmatrix} \quad (3.2)$$

où par exemple α_{x_L, x_B} : Angle entre l'axe x du repère local et l'axe x du repère du mobile.

L'attitude du mobile peut être également écrite sous la forme d'angles d'Euler afin d'être plus intuitive pour l'utilisateur. Les angles d'Euler correspondent à trois rotations successives ϕ, θ, ψ , soit l'angle de roulis, de tangage et de lacet sur chaque axe portant le même nom x_B, y_B et z_B . La séquence des rotations varie suivant les articles, dans le cadre de ce mémoire nous utiliserons la plus répandue, soit : La première rotation est autour de l'axe de lacet, puis de l'axe de tangage et finalement de l'axe de roulis :

$$\text{Première rotation : } C_B^{R1} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$\text{Deuxième rotation : } C_{R1}^{R2} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.4)$$

$$\text{Troisième rotation : } C_{R2}^L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.5)$$

On obtient alors :

$$C_B^L = C_{R2}^L C_{R1}^{R2} C_B^{R1} \quad (3.6)$$

$$C_B^L = \begin{bmatrix} \cos(\theta) \cos(\psi) & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos(\theta) \sin(\psi) & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\theta) \cos(\phi) \end{bmatrix} \quad (3.7)$$

où : ϕ, θ, ψ : Angles d'Euler signifiant le roulis, le tangage et le lacet
 C_A^B : Matrice de rotation DCM du repère A vers le repère B

On peut également obtenir les angles d'Euler à partir de cette matrice de rotation, soit :

$$\phi = \text{atan2}(\cos(\theta) \sin(\phi), \cos(\theta) \cos(\phi)) \quad (3.8)$$

$$\theta = \text{atan2}\left(-\sin(\theta), \sqrt{1 - (\sin^2(\theta))}\right) \quad (3.9)$$

$$\psi = \text{atan2}(\cos(\theta) \sin(\psi), \cos(\theta) \cos(\psi)) \quad (3.10)$$

Finalement, l'attitude peut être écrite sous la forme d'un quaternion unitaire. Cette forme est préférable puisqu'elle ne présente aucune singularité (résolution du blocage de cardan) contrairement aux angles d'Euler. De plus, le quaternion permet d'avoir une meilleure précision lors de la propagation de l'attitude du système. Il est cependant moins intuitif pour l'utilisateur puisque ces quatre composantes (un scalaire et un vecteur à trois éléments) n'ont pas de représentation physique.

Il peut être représenté par le nombre complexe suivant :

$$q = q_0 + iq_1 + jq_2 + kq_3 \quad (3.11)$$

où : q : Quaternion unitaire
 q_0 : Scalaire du quaternion unitaire
 q_1, q_2, q_3 : Éléments du vecteur du quaternion unitaire

$$q = [q_0, q_1, q_2, q_3]^T \quad (3.12)$$

La relation entre le quaternion et la matrice de rotation DCM est la suivante :

$$C_B^L = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.13)$$

et son inverse :

$$q_0 = \frac{1}{2} \sqrt{1 + c_{11} + c_{22} + c_{33}} \quad (3.14)$$

$$\text{si } q_0 \neq 0 \text{ alors } q_B^L = \begin{bmatrix} q_1 = (c_{32} - c_{23})/4q_0 \\ q_2 = (c_{13} - c_{31})/4q_0 \\ q_3 = (c_{21} - c_{12})/4q_0 \end{bmatrix} \quad (3.15)$$

$$\text{si } q_0 = 0 \text{ alors } q_B^L = \begin{bmatrix} q_1 = \frac{c_{13}c_{12}}{\sqrt{C_{12}^2 C_{13}^2 + C_{12}^2 C_{23}^2 + C_{13}^2 C_{23}^2}} \\ q_2 = \frac{c_{12}c_{23}}{\sqrt{C_{12}^2 C_{13}^2 + C_{12}^2 C_{23}^2 + C_{13}^2 C_{23}^2}} \\ q_3 = \frac{c_{13}c_{23}}{\sqrt{C_{12}^2 C_{13}^2 + C_{12}^2 C_{23}^2 + C_{13}^2 C_{23}^2}} \end{bmatrix} \quad (3.16)$$

Dans le cas d'une singularité (présent dans la matrice de rotation ou des angles d'Euler), ces équations de transformation ne fonctionnent pas. La méthode recommandée est de ce fait d'utiliser le quaternion pour tous les calculs et d'utiliser les angles d'Euler pour afficher le résultat afin d'en avoir une représentation physique. La relation entre le quaternion et les angles d'Euler est la suivante :

$$\phi = \tan^{-1} \left(\frac{2(q_2q_3 - q_0q_1)}{1 - 2(q_1^2 + q_2^2)} \right) \quad (3.17)$$

$$\theta = \sin^{-1}(-2(q_1q_3 + q_0q_2)) \quad (3.18)$$

$$\psi = \tan^{-1} \left(\frac{2(q_1q_2 - q_0q_3)}{1 - 2(q_2^2 + q_3^2)} \right) \quad (3.19)$$

Afin de représenter l'attitude d'un mobile dans le temps, il est important de définir son équation de propagation de base utilisant la vitesse angulaire du gyroscope. La première étape consiste à calculer l'équation de propagation du quaternion unitaire (cette équation ne prend pas en compte la rotation de la terre puisque les capteurs à faible coût ne sont pas assez précis pour pouvoir la mesurer).

$$\dot{q}_B^L(t) = \frac{1}{2} q_B^L(t - \Delta t) \otimes \omega_{IB}^B = \frac{1}{2} \Omega_{IB}^B q_B^L(t - \Delta t) \quad (3.20)$$

$$\Omega_{IB}^B = \begin{bmatrix} 0 & -\omega_{IBx}^B & -\omega_{IBy}^B & -\omega_{IBz}^B \\ \omega_{IBx}^B & 0 & \omega_{IBz}^B & -\omega_{IBy}^B \\ \omega_{IBy}^B & -\omega_{IBz}^B & 0 & \omega_{IBx}^B \\ \omega_{IBz}^B & \omega_{IBy}^B & -\omega_{IBx}^B & 0 \end{bmatrix} \quad (3.21)$$

où :

- ω_{IB}^B : Vecteur de la vitesse angulaire
- Ω_{IB}^B : Réorganisation du vecteur ω_{IB}^B sous la forme d'une matrice 4x4 pour la multiplication du quaternion
- \dot{q}_B^L : Dérivée par rapport au temps du quaternion unitaire

L'équation de propagation de l'attitude s'obtient en intégrant l'équation de propagation du quaternion unitaire (3.20 (dans le cadre de ce mémoire, la méthode d'intégration Runge-Kutta du second ordre est utilisée) :

$$q_B^L(t) = \int \dot{q}_B^L(t) dt \quad (3.22)$$

$$q_B^L(t) = q_B^L(t - \Delta t) + \Delta t \left(\frac{\dot{q}_B^L(t) + \dot{q}_B^L(t - \Delta t)}{2} \right) \quad (3.23)$$

où : Δt : La durée entre l'instant t et $t - \Delta t$

3.1.3 Représentation de la vitesse et de la position en coordonnées ECEF

De base en navigation inertielle, on utilise les données provenant de l'accéléromètre et du gyroscope afin d'obtenir la position et la vitesse du mobile. Les étapes nécessaires sont les suivantes :

1^{er} étape : On calcul la matrice C_B^N à partir de l'attitude du système et de la matrice C_L^N

$$C_B^N = C_L^N C_B^L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} C_B^L \quad (3.24)$$

2^{ème} étape : On obtient la vitesse dans le repère de navigation en intégrant le vecteur de forces spécifiques reporté dans le repère de navigation (3.26). La vitesse initiale est considérée étant nulle. La position s'obtient en intégrant la vitesse (3.28), la position initiale doit être prise en compte afin d'avoir la position absolue et non relative.

$$\dot{V}^N = C_B^N a_{Sf}^B - g^N \text{ ou } \dot{V}^N = C_L^N \left(q_B^L \otimes a_{Sf}^B \otimes q_B^{L^{-1}} \right) - g^N \quad (3.25)$$

$$V^N(t) = V^N(t - \Delta t) + \Delta t \left(\frac{\dot{V}^N(t) + \dot{V}^N(t - \Delta t)}{2} \right) \quad (3.26)$$

$$\dot{P}^N = V^N \quad (3.27)$$

$$P^N = \int V^N dt = P^N(t - \Delta t) + \Delta t \left(\frac{\dot{P}^N(t) + \dot{P}^N(t - \Delta t)}{2} \right) \quad (3.28)$$

où :

- \dot{V}^N : La dérivée de la vitesse
- a_{Sf}^B : Le vecteur des forces d'accélération
- g^N : Le vecteur de gravité terrestre
- Δt : La durée entre l'instant t et $t - \Delta t$
- \dot{P}^N : La dérivée de la position

3^{ème} étape : On va exprimer la position dans le repère géodésique ECEF en utilisant les relations suivantes (On utilise le système géodésique WGS84) :

$$\phi = \int \dot{\phi} dt \text{ avec } \dot{\phi} = \frac{V_{nord}^N}{M + h} \quad (3.29)$$

$$\lambda = \int \dot{\lambda} dt \text{ avec } \dot{\lambda} = \frac{V_{est}^N}{(N + h) \cos \phi} \quad (3.30)$$

$$M = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \phi)^{\frac{3}{2}}} \quad (3.31)$$

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \quad (3.32)$$

$$h = 0 \text{ m} \quad (3.33)$$

$$a = 6378137 \text{ m} \quad (3.34)$$

$$e = 0.081819190842 \quad (3.35)$$

où :	ϕ	: Latitude géodésique
	λ	: Longitude géodésique
	V_{nord}^N	: Vitesse dans le repère de navigation pour l'axe y_N^N (nord vrai)
	V_{est}^N	: Vitesse dans le repère de navigation pour l'axe x_N^E (est)
	M	: Rayon de courbure méridien
	N	: Rayon de courbure normal
	h	: Altitude par rapport à l'ellipsoïde de référence
	a	: Axe semi-majeur
	e	: Coefficient d'excentricité

Note : Le premier calcul de $\dot{\phi}$ et $\dot{\lambda}$ nécessite d'initialiser l'attitude. Afin de résoudre les intégrations, donnant la latitude et la longitude, il faut également prendre en compte la position initiale de la longitude.

4^{ème} étape : Afin d'obtenir la vitesse en coordonnées ECEF cartésiens, il faut changer le vecteur vitesse de repère en calculant la matrice C_N^E à l'aide de la latitude et longitude obtenue en étape 3.

$$C_N^E = \begin{bmatrix} \cos \lambda & -\sin \lambda \sin \phi & \cos \phi \sin \lambda \\ 0 & \cos \phi & \sin \phi \\ -\sin \lambda & \sin \phi \cos \lambda & \cos \phi \cos \lambda \end{bmatrix} \quad (3.36)$$

$$V^E = C_N^E V^N \text{ ou } V^E = q_N^E \otimes V^N \otimes q_N^{E^{-1}} \quad (3.37)$$

3.1.4 Représentation de la gestuelle (BVH)

La modélisation paramétrique du corps humain est représentée sous différents formats, on peut citer essentiellement les formats BVA, BVH et ASF/ACM qui sont associés à différentes compagnies. Nous avons choisi le format BVH pour le projet ibNav du fait de la simplicité de sa structure, du nombre de codes sources disponibles et de sa popularité pour la capture de mouvements.

Un fichier sous format BVH contient deux sections, la première section nommée HIERARCHY permet de définir le squelette du corps humain (puisque le projet est destiné à être porté par un être humain) et la deuxième section nommée MOTION contient chaque trame incluant la position du MASTER et les angles d'Euler dans l'ordre Psi, Theta, Phi pour chaque joint défini dans le squelette (dont le MASTER). Un joint correspond à un point dans l'espace, dans notre cas, un IMU. Dans cette section, il faut également définir le taux de rafraichissement des données.

Sommairement, le squelette est constitué de façon hiérarchique. Il y a toujours deux composantes, OFFSET (qui permet de donner la distance d'un point par rapport à un autre) et CHANNELS qui définit quel type de données est dans MOTION. Avant les différents joints, on trouve le MASTER représentant la position du squelette, celui-ci a toujours son OFFSET à zéro et à 6 composantes dans le CHANNEL (Position X, Y, Z et rotation Z, Y, X).

En plus du MASTER, on a 13 autres IMU (SLAVES) dans le projet ibNav qui sont représentés dans le fichier BVH par des joints, comme on peut voir sur la Figure 3.4. Chaque joint contient un OFFSET correspondant à sa distance par rapport à son point PARENT et il a 3 composantes dans le CHANNEL (rotation Z, Y, X). Finalement, chaque dernier FILS a un OFFSET supplémentaire fournissant une information supplémentaire indiquant la longueur du dernier segment.

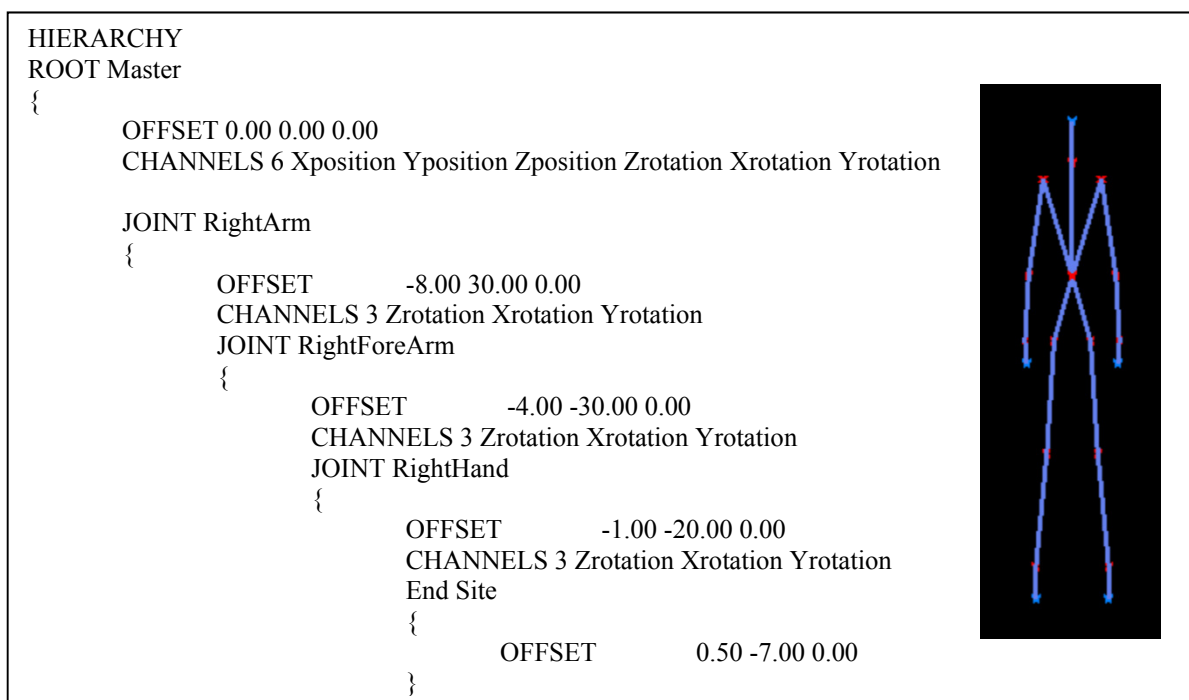


Figure 3.4 Extrait et représentation du fichier BVH du projet ibNav

3.1.5 Représentation du déplacement pédestre (KML)

Dans le même principe que l'utilisation du format BVH pour représenter les mouvements du corps, le format KML a été choisi pour représenter son déplacement sous un format standardisé. Ce format est essentiellement utilisé par Google dans ses applications tel que Google Maps, Google Earth et Google Mes parcours mais aussi par NASA WorldWind, Adobe PhotoShop, AutoCAD, etc. La structure est sous forme de balises, basée sur le XML.

Les informations nécessaires afin de créer le fichier sont :

- la latitude, la longitude et la hauteur du mobile;
- une donnée de temps pour chaque échantillon.

3.2 IMU et capteurs inertiels de type MEMS

Un IMU est une centrale inertielle dotée de différents capteurs de type MEMS tels qu'un accéléromètre, gyroscope et magnétomètre. Il peut parfois être doté également de capteurs de température et de pression. Cette plateforme dispose d'un microcontrôleur afin de pouvoir le configurer et récupérer les données des capteurs. Si celui-ci est assez puissant, il peut permettre également de calculer la position, la vitesse et l'attitude de la plateforme en fusionnant les données des différents capteurs. On a alors un système entièrement autonome intitulé AHRS ou INS.

3.2.1 Vue d'ensemble d'un IMU

Les unités de mesures inertielles utilisées dans le projet ibNav sont référencées sous le nom de STEVAL-MKI062V2 de la société STMicroelectronics. Ils sont également connus sous le nom d'iNemoV2 (Figure 3.5).

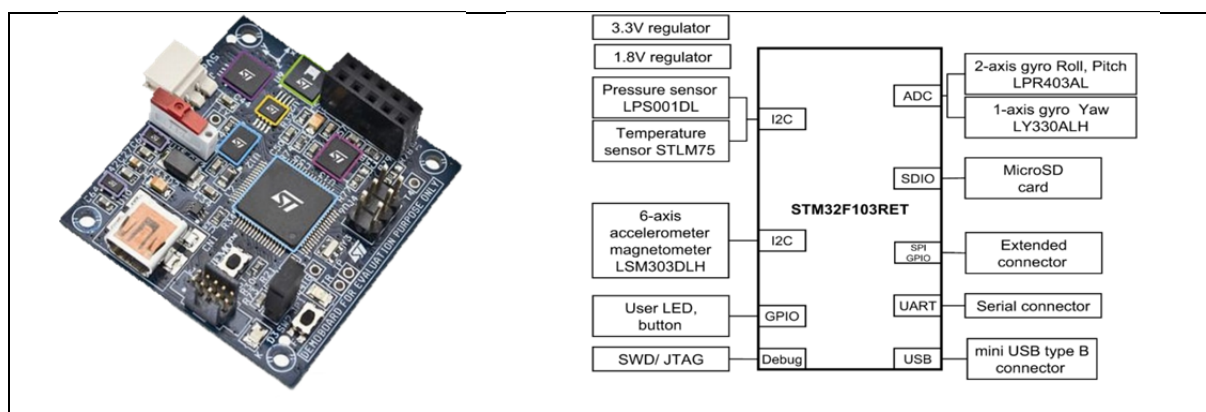


Figure 3.5 L'IMU iNemoV2 et son diagramme simplifié (extrait de fiche technique de STMicroelectronics)

Elles sont dotées d'un accéléromètre, d'un gyroscope et d'un magnétomètre travaillant dans les 3 axes de leurs référentiels. On obtient ainsi un IMU à 9 DOF à moins de 250 \$ CAD. Le microcontrôleur intégré est de type ARM 32 bits Cortex M3 sous le numéro de série : STM32F103RE. Il peut être facilement programmé à l'aide du JTAG/SWD. Le microcontrôleur est également assez puissant pour permettre d'effectuer les calculs pour les algorithmes de calibration et de navigation. Il est doté de plusieurs protocoles de communication USART et SPI. Le capteur de température est également utilisé, cependant le capteur de pression n'est pas utilisé dans ce projet, or ce capteur pourrait être intégré dans le future pour améliorer la mesure d'altitude du mobile.

3.2.2 Accéléromètres

Les accéléromètres de type MEMS permettent de mesurer les forces d'accélération sur leurs 3 axes. Ils mesurent des forces statiques et dynamiques, basées sur les lois de Newton. C'est-à-dire la projection du vecteur de gravité terrestre ($g = 9.81 \text{ m/s}^2$) et les forces résultants d'un mouvement ou d'une vibration. Le fonctionnement de l'accéléromètre pour le projet ibNav est de type masse-ressort, comme le montre la Figure 3.6. Le déplacement de l'électrode mobile permet de mesurer le déplacement de la masse par rapport à son support et d'en déduire l'accélération.

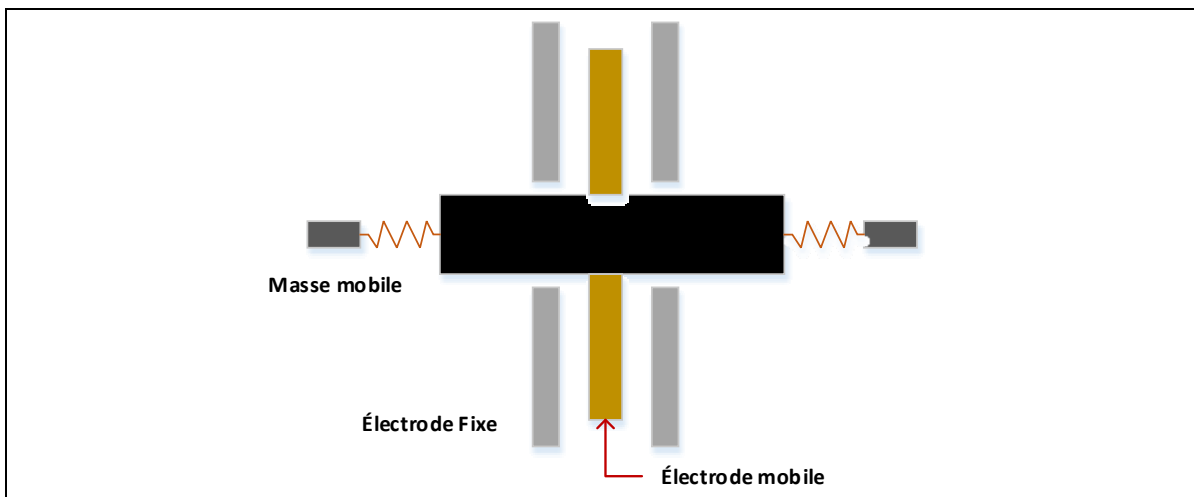


Figure 3.6 Schéma du fonctionnement d'un accéléromètre masse-ressort de type MEMS
Adaptée de Zeljko (2012)

3.2.3 Gyroscopes

Les gyroscopes (ou gyromètres) de type MEMS permettent de mesurer la vitesse angulaire sur leurs 3 axes (roulis, tangage et lacet). Cette mesure est obtenue par l'effet de Coriolis, suivant le principe du Pendule de Foucault. A noter que ce type de gyroscope n'est pas assez précis pour mesurer la vitesse de rotation de la terre.

Le fonctionnement du gyroscope du projet ibNav est aussi de type masse-ressort, il fonctionne à l'aide de 2 masses vibrant en opposition de phase. Afin d'utiliser l'effet de Coriolis, ses masses sont continuellement en vibration à l'aide d'impulsions électrostatiques. Dès qu'une force externe est appliquée, une vibration est créée sur l'axe orthogonal au déplacement continu des masses par la force de Coriolis.

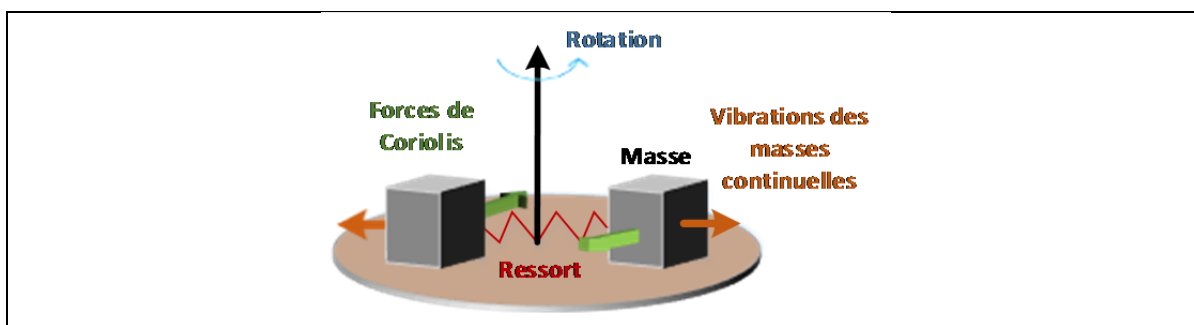


Figure 3.7 Schéma du fonctionnement d'un gyroscope masse-ressort de type MEMS
Adaptée de Zeljko (2012)

3.2.4 Magnétomètres

Les magnétomètres de type MEMS permettent de mesurer le champ magnétique terrestre sur leurs trois axes. Ils permettent essentiellement de connaître l'emplacement du Nord magnétique. Cependant, la performance des magnétomètres est réduite dans un endroit très métallique et / ou de nombreux matériels électroniques sont en fonctionnement. Le champ magnétique se retrouve perturbé et les mesures sont erronées. La technologie utilisée est un capteur de type AMR (Magnétorésistance anisotropique), le magnétomètre détecte le champ magnétique appliqué perpendiculaire à la circulation du courant grâce à la variation de la

valeur de ses résistances. Le capteur est sous la forme d'un pont de Wheatstone et est composé de magnétorésistance ayant la même valeur. Le pont est toujours alimenté afin d'avoir un courant passant à travers les résistances. Lorsqu'un champ magnétique est appliqué, le vecteur de magnétisation de deux des résistances opposées tourne en direction du courant, ce qui provoque une augmentation de leur résistance. A l'inverse, pour les deux autres résistances, le vecteur de magnétisation tourne à l'opposé du sens du courant, ce qui diminue leurs résistances. (Pasolini, 2009).

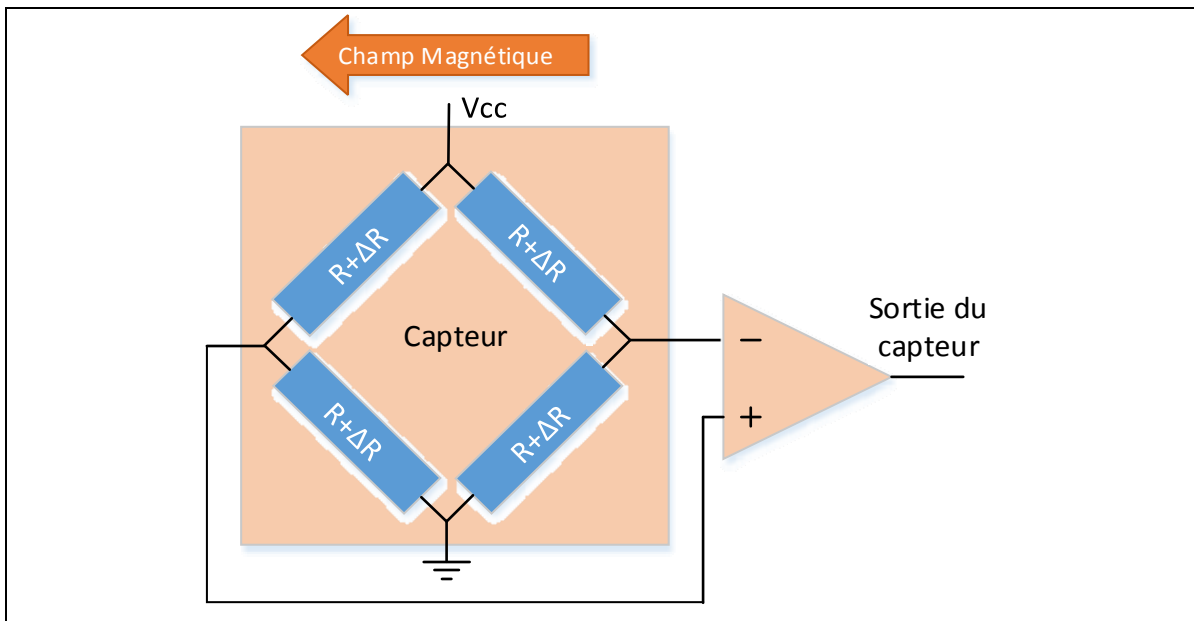


Figure 3.8 Schéma du fonctionnement d'un magnétomètre AMR de type MEMS
Adaptée de Pasolini (2009)

3.2.5 Fusion des capteurs

L'objectif d'utiliser un IMU combinant différents capteurs est de pouvoir fusionner les données afin d'obtenir un système plus puissant, offrant une solution ayant une meilleure précision. L'IMU du projet ibNav utilise les trois capteurs précédents ainsi que la température interne afin de calculer sa solution (attitude, vitesse et position). La fusion des capteurs se fait à l'intérieur d'un filtre de Kalman. Par exemple pour le modèle AHRS, la

vitesse angulaire des gyroscopes permettant d'obtenir l'attitude du mobile est corrigée à partir des données d'accélération, de champ magnétiques et de la détection d'évènements.

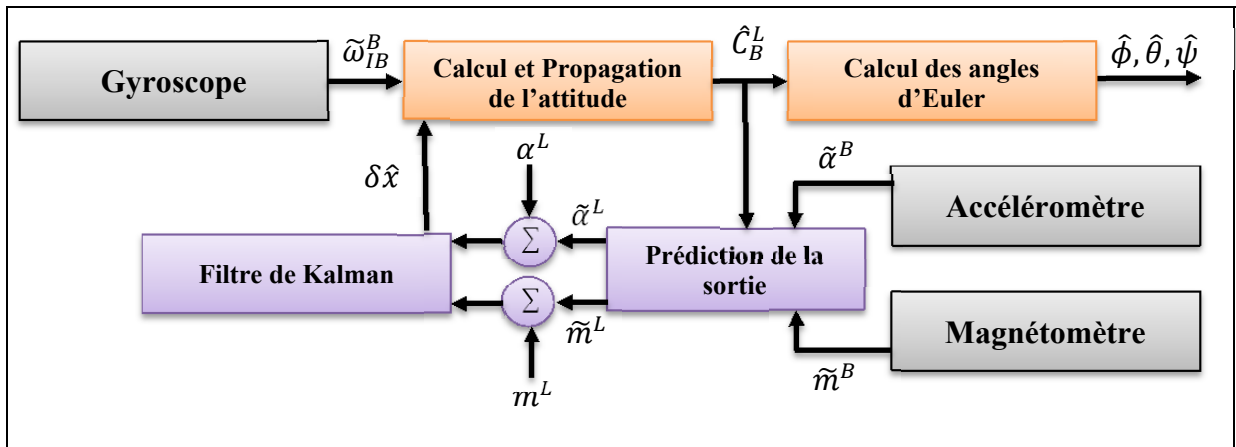


Figure 3.9 Schéma d'une fusion des capteurs pour un algorithme AHRS
Adaptée de Farrell (2008)

La Figure 3.9 présente un modèle de base AHRS utilisant un filtre de Kalman, le modèle développé dans le cadre du projet ibNav est présenté au CHAPITRE 5.

3.3 Sources d'erreurs des capteurs inertiels

Comme vue dans le CHAPITRE 2, il existe deux types d'erreurs (déterministe et stochastique) propres aux capteurs inertiels, entraînant la dérive de la solution. Ces erreurs doivent être caractérisées et modélisées afin de les corriger.

Les mesures provenant de chaque capteur doivent prendre en compte ces différentes erreurs (Park, 2004, Lavoie 2012, Zeljko, 2012). On peut d'ores et déjà écrire le vecteur de mesure général pour chaque capteur inertielle :

- pour les accéléromètres : $\tilde{\underline{a}}^S = \underline{a}^S + \delta \underline{a}^S + \underline{\eta}_a$ (3.38)

- pour les gyroscopes : $\tilde{\underline{\omega}}_{IS}^S = \underline{\omega}_{IS}^S + \delta \underline{\omega}^S + \underline{\eta}_\omega$ (3.39)

- pour les magnétomètres : $\tilde{\underline{m}}^S = \underline{m}^S + \delta \underline{m}^S + \underline{\eta}_m$ (3.40)

où :

- $\underline{\tilde{a}}^S; \underline{\tilde{\omega}}_{IS}^S; \underline{\tilde{m}}^S$: Vecteurs de mesure des accélérations, des vitesses angulaires et du champ magnétique terrestre mesurés (respectivement);
- $\underline{a}^S; \underline{\omega}_{IS}^S; \underline{m}^S$: Vecteurs de mesure des accélérations, des vitesses angulaires et du champ magnétique terrestre réel (respectivement);
- $\delta\underline{a}^S; \delta\underline{\omega}^S; \delta\underline{m}^S$: Vecteurs d'erreurs de type déterministe des accélérations, des vitesses angulaires et du champ magnétique local (respectivement);
- $\underline{\eta}_a; \underline{\eta}_\omega; \underline{\eta}_m$: Vecteurs d'erreurs de type stochastique représentant un processus aléatoire tel que le bruit blanc.

3.3.1 Erreurs de type déterministe

Ces erreurs de type non-aléatoire sont estimées et corrigées à l'aide d'une méthode de calibration qui sera détaillée au CHAPITRE 4. Dans le cadre de ce mémoire, nous nous intéresserons aux erreurs illustrées à la Figure 3.10.

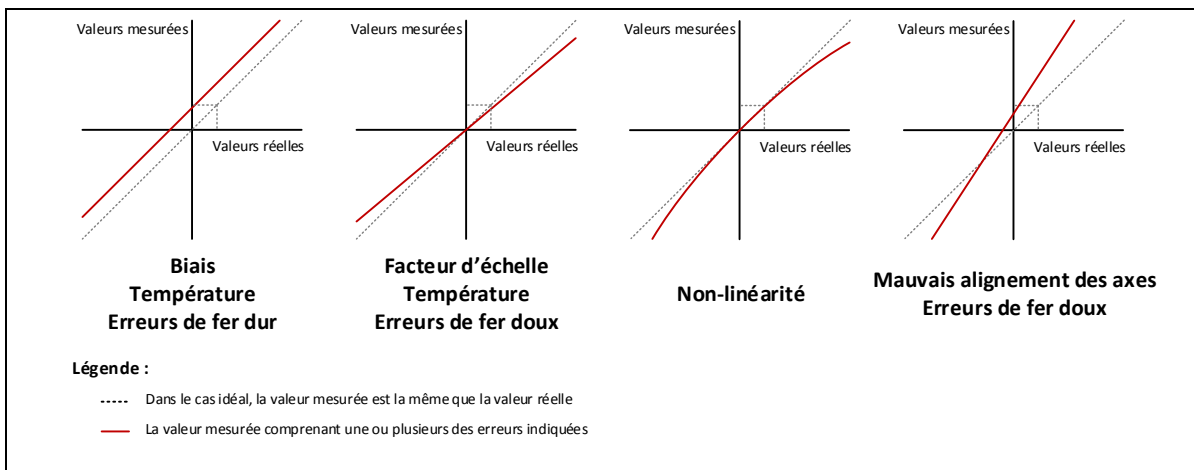


Figure 3.10 Principales erreurs entre les valeurs mesurées et réelles d'un capteur inertielle et d'un magnétomètre
Adaptée de Ladetto (2003)

Pour les accéléromètres et gyroscopes, les erreurs sont les suivantes :

- **le biais** : C'est une valeur constante que l'on obtient en sortie des capteurs inertiels en scénario statique, soit en l'absence d'excitations en entrée. Ces erreurs sont toujours présentes durant le fonctionnement des capteurs. Cette valeur est également influencée par la température de ceux-ci. On retrouve le terme de « Zero-g level » qui signifie l'absence d'accélération en entrée pour les accéléromètres et de « Zero rate level » qui signifie l'absence de vitesse angulaire en entrée pour les gyroscopes;
- **le facteur d'échelle** : Cette erreur correspond à la sensibilité du capteur, soit le gain du capteur qui est un ratio entre la valeur mesurée en sortie et l'unité de mesure que l'on souhaite obtenir. Par exemple, si l'accéléromètre a une sortie numérique, on obtient les mesures en LSB mais on souhaite par la suite obtenir cette valeur en accélération (g). Le ratio correspondant à la sensibilité est donc exprimé dans ce cas en LSB/g, c'est un pourcentage correspondant à la déviation. Ce ratio est également influencé par la température des capteurs inertiels;
- **la non-linéarité** : La réponse en sortie du capteur suit souvent un modèle non-linéaire pour les capteurs inertiels à faible coût. Cependant, cette erreur est souvent assez faible et peut souvent être ignorée;
- **le mauvais alignement des axes** : Comme son nom l'indique, elle correspond à une erreur d'orthogonalité. Cette erreur peut se situer entre l'IMU et le capteur lui-même mais aussi entre les axes du capteur. L'angle entre les axes n'est pas de ce fait à 90° mais il est légèrement modifié provoquant ainsi des erreurs se répercutant sur chacun des axes. Souvent, on considère que l'erreur entre les axes du capteur est négligeable et on modélise uniquement l'erreur entre l'IMU et les capteurs inertiels même si cette erreur reste assez faible également;
- **l'influence de la température** : Celle-ci agit sur le biais et le facteur d'échelle, elle est cependant difficile à modéliser puisqu'elle suit un modèle non-linéaire.

Pour les magnétomètres, les erreurs correspondant à la distorsion du champ magnétique portent un nom différent et sont réparties au sein de deux catégories (Ladetto, 2003 ; Lavoie 2012) :

- la première concerne **les erreurs de fer dur** du champ magnétique. Celles-ci sont causées par l'IMU lui-même. Ce sont des perturbations électromagnétiques causées par les différents composants formant l'IMU. Ces erreurs sont équivalentes à un biais constant et peuvent être caractérisées par une calibration initiale;
- la deuxième concerne **les erreurs de fer doux** du champ magnétique. Celles-ci sont causées par l'environnement où l'IMU est utilisé. Ces erreurs peuvent provenir des appareils électroniques, du câblage électrique du bâtiment, des transmissions radio. On retrouve également une perturbation du champ magnétique causée par le corps humain mais celle-ci est négligeable lorsque l'utilisateur ne porte aucun objet métallique ou électronique. Cette erreur est équivalente aux erreurs de mauvais alignement et de facteur d'échelle et nécessite une calibration en ligne pour être efficace.

La Figure 3.11 représente les différentes observations du champ magnétique suivant les deux types d'erreurs des magnétomètres. La méthode la plus simple pour visualiser ces erreurs est de représenter les mesures sur l'axe X par rapport à celles prises sur l'axe Y. Lorsqu'il n'y a aucune erreur, les mesures devraient former un cercle centré en 0 (absence de distorsion). Cependant, lorsqu'une erreur de fer dur est présente, le cercle n'est plus centré en son origine et lorsqu'il y a une erreur de fer doux, le cercle prend une forme ellipsoïdale.

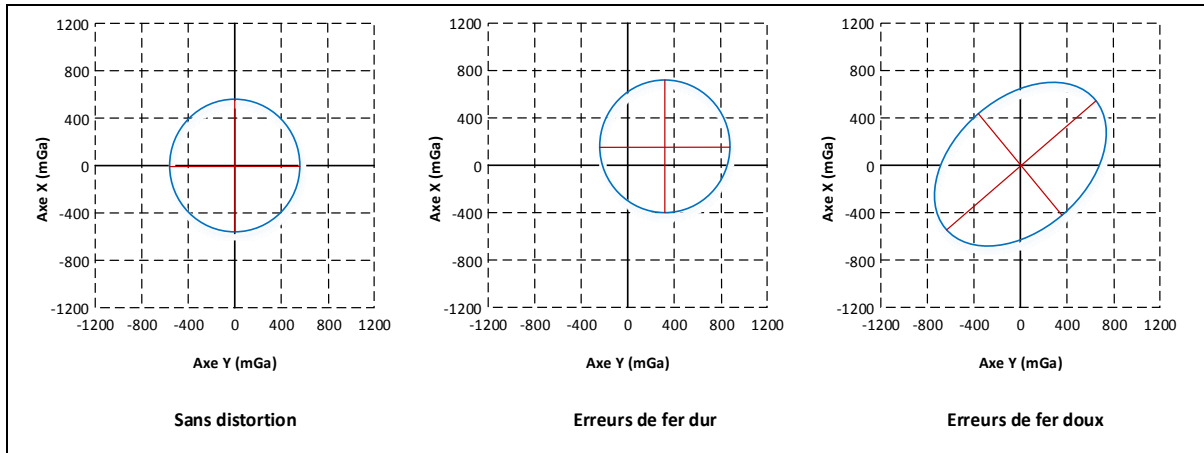


Figure 3.11 Représentation des deux types d'erreurs causant la distorsion du champ magnétique

On peut modéliser les erreurs déterministes, tel que (Fong, Ong et Nee, 2008) :

$$\underline{v}^B = MS(\underline{v}^S - \underline{b}) \quad (3.41)$$

- où :
- \underline{v}^B : Vecteur exprimé dans le repère du mobile
 - \underline{v}^S : Vecteur exprimé dans le repère du capteur (contenant des erreurs)
 - M : Matrice de mauvais alignement
 - S : Matrice de mauvais alignement
 - \underline{b} : Matrice de facteur d'échelle
 - : Vecteur de biais sur les mesures

Les matrices complètes de mauvais alignement M et de facteur d'échelle incluant les erreurs de croisement des axes sont les suivantes :

$$M = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{bmatrix} \quad (3.42)$$

$$S = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \quad (3.43)$$

où : S_{ij} : Valeur du facteur d'échelle suivant les axes i et j
 α_{ij} : Angle de mauvais alignement entre les axes i et j

Afin de simplifier le problème, on utilise directement le produit des deux matrices, on va également la simplifier en considérant que la sensibilité du croisement des axes est négligeable. De plus, on considère que l'axe Y du repère du mobile est situé dans le même plan formé par l'axe X et Y du capteur, de ce fait les angles α_{xz} , α_{xy} , α_{yx} sont négligeables.

On a alors la matrice E suivante :

$$E \approx \begin{bmatrix} S_{xx} & -S_y \alpha_{yz} & S_z \alpha_{zy} \\ 0 & S_y & -S_z \alpha_{zx} \\ 0 & 0 & S_z \end{bmatrix} \quad (3.44)$$

Dans l'exemple ci-dessus, on a considéré que l'alignement est parfait entre les axes des capteurs et ceux du mobile. Cependant, lorsque l'IMU est mis dans un boîtier, il arrive souvent que les deux ne sont pas parfaitement alignés, l'équation (3.41) devient alors :

$$\underline{v}^B = C_S^B [E(\underline{v}^S - \underline{b})] \quad (3.45)$$

où : C_S^B : Matrice de rotation entre le repère du capteur et le repère du mobile

3.3.2 Erreurs de type stochastique

Ces erreurs de type aléatoire sont plus difficiles à modéliser et donc à corriger. La seule qui sera corrigée dans le cadre de ce mémoire est une composante à haute fréquence, le bruit blanc, à l'aide du filtre de Kalman. Il sera modélisé à l'aide de la variance d'Allan sur un

échantillon statique des données des capteurs inertiels. Cette erreur suit une loi normale de moyenne nulle et de variance σ^2 .

3.4 Modèles de propagation de l'erreur

Lorsqu'on calcule l'attitude, la vitesse ou la position du mobile, des erreurs s'accumulent au cours du temps et provoquent la dérive de la solution. Celles-ci doivent premièrement être modélisées afin d'être corrigées à l'aide d'un estimateur tel que le filtre de Kalman.

Erreur de l'attitude :

Les gyroscopes à faible coût présentent des erreurs déterministes et stochastiques non négligeables. De ce fait, il y a une différence entre la vitesse angulaire du repère du capteur ω_{IS}^S et la vraie vitesse angulaire de la plateforme ω_{IP}^P . On a ainsi les équations suivantes :

$$\delta\omega_{CP}^P = \omega_{IP}^P - \omega_{IS}^S = \omega_{IP}^P - \hat{\omega}_{IP}^P \quad (3.46)$$

$$\omega_{IB}^B = C_P^B (\delta\omega_{CP}^P + \omega_{IS}^S) \quad (3.47)$$

- où :
- $\delta\omega_{CP}^P$: L'erreur entre les données mesurées et la vraie vitesse angulaire de la plateforme
 - ω_{IB}^B : Vitesse angulaire du mobile
 - ω_{IP}^P : Vitesse angulaire de la plateforme dite vraie
 - ω_{IS}^S : Vitesse angulaire mesurée par le gyroscope
 - $\hat{\omega}_{IP}^P$: Vitesse angulaire de la plateforme estimée
 - C_P^B : Matrice de rotation afin de passer du repère de la plateforme au repère du mobile

On peut ainsi écrire l'équation de propagation de l'attitude (intégration de l'équation (3.20) avec l'erreur $\delta\omega_{CP}^P$ s'accumulant au cours du temps, cette erreur ayant un impact direct sur l'attitude du système :

$$q_B^L(t) = \int \frac{1}{2} q_B^L(t-1) \otimes C_P^B(\delta\omega_{CP}^P + \omega_{IS}^S)(t) dt \quad (3.48)$$

Erreur de la vitesse et de la position :

Tout comme les gyroscopes, les accéléromètres à faible coût présentent des erreurs non négligeables dans leurs mesures. C'est-à-dire une différence entre l'accélération vraie de la plateforme a^P et celle mesurée par le capteur a^S . Puisque l'accélération est la dérivée de la vitesse par rapport au temps et que la vitesse est la dérivée de la position par rapport au temps, cette erreur va s'accumuler. Les équations sont similaires au gyroscope :

$$\delta a^P = a^P - a^S = a^P - \hat{a}^P \quad (3.49)$$

$$a^B = C_P^B(\delta a^P + a^S) \quad (3.50)$$

L'intégration de l'équation de la vitesse (3.25) avec l'erreur δa^P s'accumulant au cours du temps :

$$V^N(t) = \int C_L^N \left(q_B^L \otimes C_P^B(\delta a^P + a^S) \otimes q_B^{L^{-1}} \right) - g^N dt \quad (3.51)$$

L'intégration de l'équation de la position (3.27) avec l'erreur δa^P s'accumulant au cours du temps :

$$P^N(t) = \iint C_L^N \left(q_B^L \otimes C_P^B(\delta a^P + a^S) \otimes q_B^{L^{-1}} \right) - g^N dt \quad (3.52)$$

3.5 Modèles de référence

Afin de limiter la propagation de l'erreur, il est impératif d'utiliser un estimateur afin de corriger ces erreurs par rapport à d'autres données de référence. Dans le cas contraire, l'attitude dérivera de plusieurs degrés au bout de quelques minutes, il en est de même pour la vitesse et la position.

Puisque les capteurs inertiels que l'on utilise sont à faible coût, on ne peut pas utiliser la rotation de la terre comme référence. Cependant, on peut utiliser le champ magnétique et la gravité terrestre comme référence.

La gravité terrestre :

Deux modèles sont utilisés dans le cadre de ce mémoire, le premier est la définition standard de la gravité, soit :

$$G_E = 9.7803267715 \text{ m/s}^2 \quad (3.53)$$

La deuxième est la formule de la norme WGS84 [Department of defense World Geodetic System 1984], prenant en compte l'emplacement où se trouve le mobile (l'altitude n'est pas prise en compte car son influence est négligeable pour un système fonctionnant au sol), soit :

$$G_\phi = \left(G_E \frac{1 + 0.001931851353 \times \sin^2 \phi}{\sqrt{1 - 0.0066943800229 \times \sin^2 \phi}} \right) \text{ m/s}^2 \quad (3.54)$$

où : G_E : Valeur standard de gravité au niveau de l'ellipsoïde de référence
 ϕ : Latitude du mobile

Le champ magnétique terrestre :

Le modèle utilisé permettant de connaître le champ magnétique à un emplacement précis est basé sur le standard IGRF-11 [IAGA Division V-MOD Geomagnetic Field Modeling]. Les coefficients du modèle sont approximés à partir d'observations, ceux-ci doivent être recalculés constamment car ils changent en fonction du temps. Il est également prévu d'utiliser le modèle CGRF permettant une meilleure précision au Canada. La formule est la suivante (Finlay, 2010) :

$$V(r, \theta, \phi, t) = a \sum_{n=1}^N \sum_{m=0}^n \left(\frac{a}{r} \right)^{n+1} [g_n^m(t) \cos m\phi + h_n^m(t) \sin m\phi] P_n^m(\cos \theta) \quad (3.55)$$

où :	r	: Distance radiale avec le centre de la terre
	θ	: Colatitude
	ϕ	: Longitude Est
	a	: Axe semi-majeur
	g_n^m, h_n^m	: Coefficients de Gauss
	$P_n^m(\cos \theta)$: Schmidt normalisé
	N	: Maximum degré d'expansion

3.6 Filtre de Kalman

Le filtre de Kalman est un estimateur d'état, il permet d'estimer les erreurs d'un système à partir de son état précédent et d'observations aléatoires dans le temps pouvant être bruitées. Dans notre cas, nous devons utiliser le filtre de Kalman étendu puisque notre système est non-linéaire. Cette extension du filtre de Kalman linéarise le système autour de l'estimation de l'état précédent. Afin d'expliquer sommairement le filtre de Kalman et d'être fidèle à notre utilisation dans le domaine inertiel, les contraintes à considérer sont les suivantes :

- le système est à temps discret;
- le système n'est pas invariant;
- le système est autonome, les commandes appliquées sont nulles, $u_K = 0$.

Considérons premièrement le cas linéaire :

On a ainsi le système d'équations suivant :

$$x_{K+1} = \phi_K x_K + G_K \omega_K \quad (3.56)$$

$$z_K = H_K x_K + v_K \quad (3.57)$$

où :	x_{K+1}, x_K	: Vecteur d'états du système
	z_K	: Vecteur d'observation
	ϕ_K	: Matrice de transition du vecteur d'état (discrétisé)

G_K	: Matrice de couplage dynamique du bruit
H_K	: Matrice d'observation
ω_K	: Vecteur de bruit du système
v_K	: Vecteur de bruit des observations

La première équation étant bruitée, elle ne peut être résolue directement, on va de ce fait utiliser les données d'observation z_K afin d'estimer x . Pour un système décrit comme ci-dessus, le filtre de Kalman va se révéler être un outil précieux et très puissant afin d'estimer le système. Puisque c'est un cas purement linéaire, il nous fournit une estimation dite optimale.

La matrice ϕ_K discrétisé peut être calculée à l'aide de l'équation suivante :

$$\phi_K = e^{F \Delta t} \quad (3.58)$$

où : F : Matrice de transition de vecteur d'état (continue)
 Δt : Temps entre l'instant t et $t + 1$

Le vecteur de bruit du système ω_K est caractérisé par une succession de bruit blanc Gaussien de moyenne nulle et de variance Qd_K . Cette dernière est appelée matrice de bruit intégré, elle est construite avec la matrice Q , appelée matrice de covariance des bruits d'état.

La matrice de bruit intégré Qd_K se calcule à l'aide de l'approximation par une série de Taylor d'ordre 2, soit l'équation suivante :

$$Qd_K = \Delta t G Q G^T + \frac{\Delta t^2}{2} (\Phi G Q G^T + G Q G^T \Phi^T) \quad (3.59)$$

Le vecteur de bruit des observations v_m est caractérisé par un bruit blanc Gaussien de moyenne nulle et de variance R . Cette dernière est appelée matrice de covariance des bruits d'observations.

Les valeurs des deux matrices Q et R sont dans la plupart des cas déterminées à l'initialisation du système par une analyse de la variance précédemment effectuée sur un nombre représentatif d'échantillons.

Le filtre de Kalman permet de minimiser la covariance de l'erreur notée P dont les équations sont les suivantes :

$$\delta x_K^- = x_K - \hat{x}_K^- \quad (3.60)$$

$$P_K^- = E[\delta x_K^- \delta x_K^{-T}] \quad (3.61)$$

où :

- δx_K^- : Vecteur d'erreur d'état avant la mise à jour des mesures
- \hat{x}_K^- : Vecteur d'état estimé
- P_K^- : Matrice de covariance de l'erreur avant la mise à jour des mesures

Le filtre de Kalman est décomposé en trois parties : L'initialisation, la prédiction et la mise à jour des mesures. La prédiction et la mise à jour des mesures se répètent pour chaque itération.

L'initialisation :

Le vecteur d'état \hat{x}_0 ainsi que la matrice de covariance P_0 doivent être préalablement estimés.

La valeur de P_0 doit augmenter par rapport à l'incertitude posée sur \hat{x}_0 . Ces valeurs seront notées \hat{x}_{K-1}^+ et P_{K-1}^+ pour $k = 1$.

La prédiction :

L'objectif de cette étape est de faire un estimé du vecteur d'état à partir de l'estimé mis à jour \hat{x}_{K-1}^+ . On le notera \hat{x}_K^- , avec sa matrice de covariance P_K^- . Les équations sont les suivantes :

$$\hat{x}_K^- = \phi_{K-1} \hat{x}_{K-1}^+ \quad (3.62)$$

$$P_K^- = \phi_{K-1} \hat{x}_{K-1}^+ \phi_{K-1}^T + Q d_{K-1} \quad (3.63)$$

La mise à jour des mesures :

Cette étape permet de corriger le vecteur d'état à l'aide du vecteur d'observation z_K . On obtient ainsi \hat{x}_K^+ avec sa matrice de covariance P_K^+ . La première étape consiste à calculer le gain de Kalman K_K qui permet de modifier la confiance accordée à l'observation par rapport à l'estimé du vecteur d'état. L'équation est la suivante :

$$K_K = P_K^- H_K^T (H_K P_K^- H_K^T + R_K)^{-1} \quad (3.64)$$

On calcule ensuite la mesure résiduelle, c'est-à-dire l'erreur entre la donnée observée z_K et la donnée observée estimée \hat{z}_K . Soit :

$$\delta z_K = z_K - \hat{z}_K = z_K - H_K \hat{x}_K^- \quad (3.65)$$

On calcule maintenant l'innovation, qui permettra de corriger le vecteur d'état. On a l'équation suivante:

$$innovation = K_K \delta z_K \quad (3.66)$$

Finalement, on corrige le vecteur d'état et sa matrice de covariance d'erreur associée :

$$\hat{x}_K^+ = \hat{x}_K^- + innovation \quad (3.67)$$

$$P_K^+ = (I - K_K H_K) P_K^- \quad (3.68)$$

Considérons maintenant le cas non-linéaire :

On considère maintenant que le système est non-linéaire (version étendu du filtre de Kalman), ce qui est d'avantage réaliste dans le domaine inertiel. Les équations définies en (3.56 et (3.57) deviennent les suivantes :

$$x_{K+1} = F(x) + G_K \omega_K \quad (3.69)$$

$$z_K = H(x) + v_K \quad (3.70)$$

Où $F(x)$ et $H(x)$ représentent des fonctions non-linéaires. Afin de construire la matrice F et H à partir d'un système non-linéaire, il faut déterminer la Jacobienne. Soit les équations suivantes :

$$F = \left. \frac{\delta f(x)}{\delta x} \right|_{x=\hat{x}} \quad (3.71)$$

$$H = \left. \frac{\delta H(x)}{\delta x} \right|_{x=\hat{x}} \quad (3.72)$$

Par la suite, les étapes sont similaires à celles décrites précédemment.

3.7 Synthèse des principes de la navigation inertielle

Le chapitre reprend les équations et les principes de base pour la navigation inertielle à faible coût permettant le développement des algorithmes décrits dans les CHAPITRE 4 et CHAPITRE 5. Le quatrième chapitre est consacré à la correction des erreurs déterministes (section 4.3) et à la modélisation des erreurs stochastique (4.4) alors que le cinquième chapitre est consacré à la correction de la dérive de la solution. La solution de navigation de chaque IMU, sera donnée à l'aide de deux modèles :

- un filtre de Kalman étendu permettant la correction des valeurs des gyroscopes afin d'obtenir l'attitude du mobile avec plus de précision. On appelle cette étape la fusion des capteurs. L'attitude sera par la suite incorporée sous un format standard pour représenter la gestuelle;

- un filtre de même type pour la correction des valeurs des accéléromètres afin d'obtenir la vitesse et la position du mobile. La position sera également incorporée sous un format standard.

CHAPITRE 4

CALIBRATION DES CAPTEURS INERTIELS

Ce chapitre est consacré à la calibration initiale et continue des capteurs inertiels à faible coût. Ces algorithmes de calibration sont un prérequis indispensable afin d'obtenir une solution de navigation exploitable, ils permettent de réduire les erreurs présentes dans les capteurs afin de se rapprocher le plus fidèlement possible des vecteurs réels d'accélération, de vitesse angulaire et de champ magnétique. Ce chapitre correspond approximativement à 15% du travail total effectué durant la maîtrise.

4.1 Présentation du matériel d'étalonnage

Le seul matériel utilisé nous permettant d'automatiser la procédure de calibration est un bras robotique de type Lynxmotion AL5D. Ce robot est équipé de 5 servos (Base, épaule, coude, poignet et pince), il nous permet ainsi de faire des rotations sur les 3 axes X, Y et Z. L'objectif est de pouvoir créer des scénarios quasi-statique (mouvements se produisant sur un seul axe) et dynamique (mouvement se produisant sur plusieurs axes en même temps).

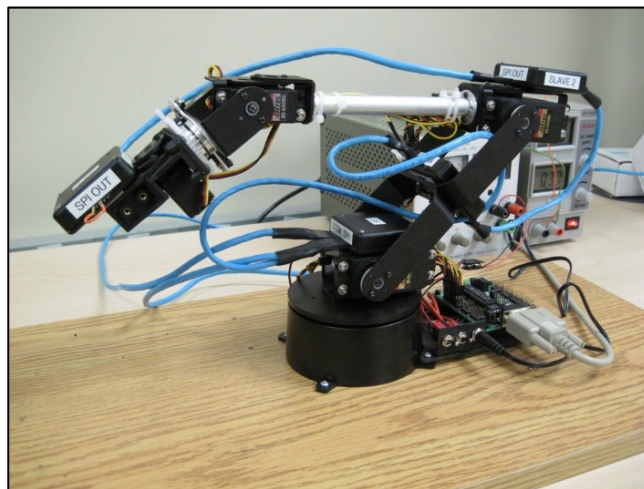


Figure 4.1 Photo du robot de calibration Lynxmotion AL5D équipé avec des IMUs

L'application existante nommée RIOS n'étant pas adéquate pour le projet (peu de précision sur les mouvements, fortes secousses/vibration, manque d'options dans la création de scénarios), une nouvelle application a été développée. Celle-ci a été développée sous Microsoft Visual Studio 2010 en VB.net. La communication avec le microcontrôleur SSC-32 s'effectue à l'aide du protocole RS232. Il n'a pas été utile de modifier le micro logiciel. L'application développée présente trois interfaces principales et une classe pour la communication. Les diagrammes simplifiés des classes et les photographies des interfaces sont les suivantes :

La classe pour la communication RS232 :

Elle contient une fonction permettant de trouver automatiquement sur quel port RS232 est connecté le matériel et dispose des fonctions de base afin d'ouvrir, de fermer le port, de recevoir et d'envoyer des données. Un évènement est créé lorsqu'une donnée est reçue, permettant d'appeler la fonction de réception.

L'interface principale :

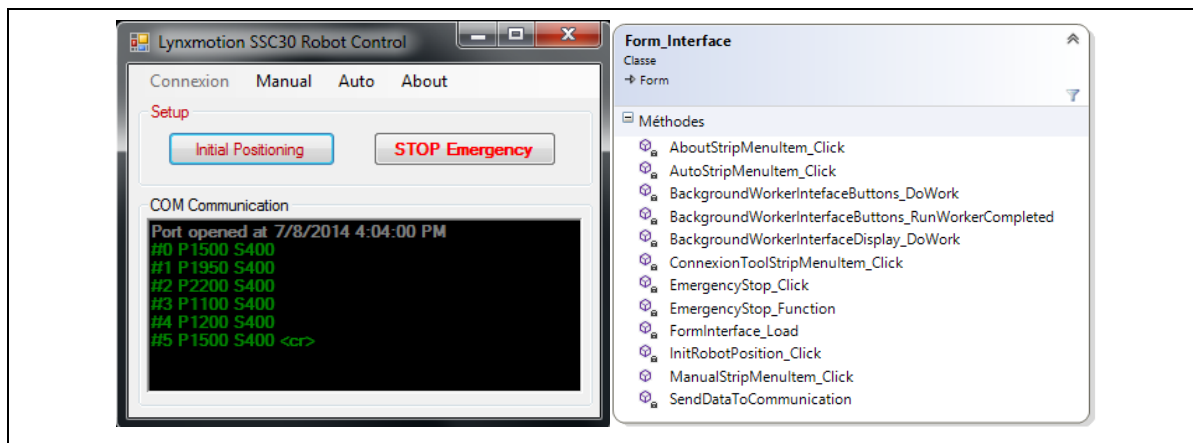


Figure 4.2 Liste des fonctions et imprime écran de l'interface principale

L'interface (Figure 4.2) principale dispose de 4 onglets en haut fonctionnant comme 4 boutons et permettant d'ouvrir des interfaces enfants différentes. Le premier permettra d'ouvrir une interface afin d'établir la connexion avec le robot tandis que le dernier permettra d'ouvrir une interface ayant différentes informations sur le logiciel (version, nom, droit d'auteur).

L'interface dispose également d'une fenêtre dans le bas permettant de voir les événements sur le port, les données envoyées et reçues. Le premier bouton dans la section setup « Initial Positioning » doit être obligatoirement enclenché après une connexion afin de mettre le robot dans sa position initial (Figure 4.1). Le deuxième bouton permet d'arrêter les servos du robot instantanément.

A noter également que l'envoi des données et l'affichage s'effectuent en multitâches afin de ne pas geler l'interface graphique. Deux fonctions gèrent le multitâche, « DoWork » permettant d'effectuer les opérations d'une tâche et « RunWorkerCompleted » permettant de faire les changements sur l'interface graphique.

L'interface pour le contrôle manuelle :

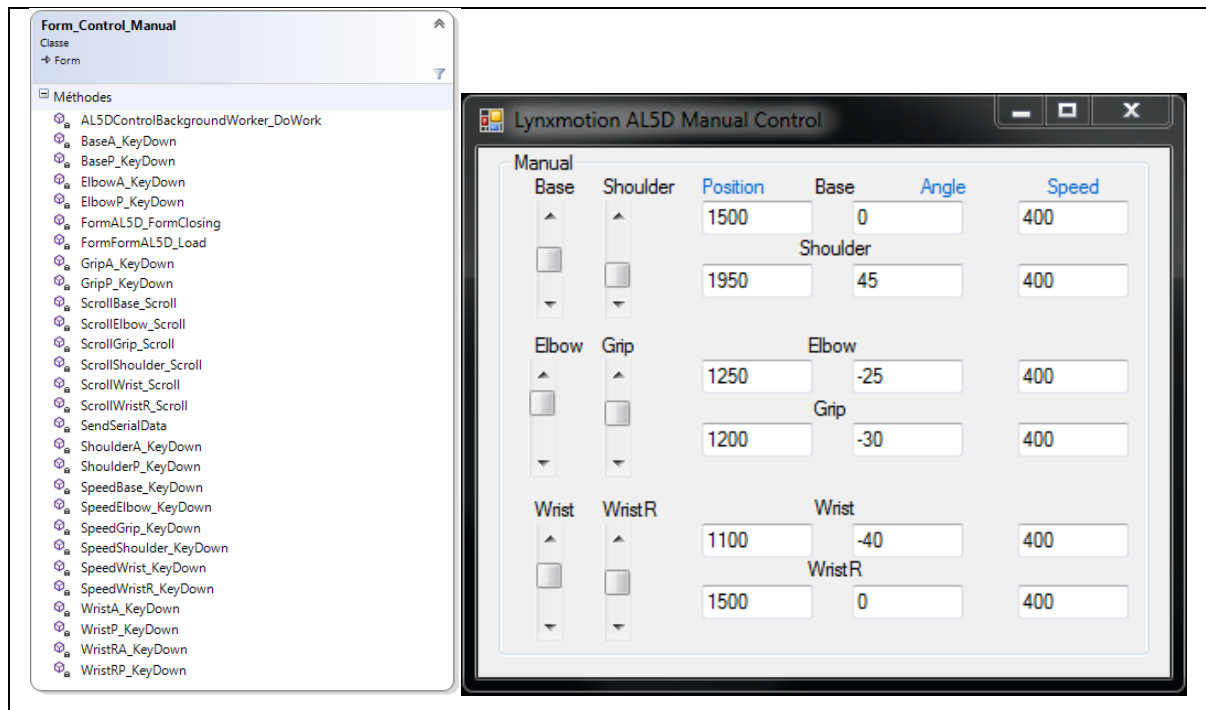


Figure 4.3 Liste des fonctions et imprime écran de l'interface pour le contrôle manuelle

L'interface (Figure 4.3) pour le contrôle manuel permet de contrôler directement les différents servos du robot. On peut choisir la vitesse de déplacement et à quelle position ou angle ils doivent se rendre. On peut également se servir des glissières pour contrôler les servos. Encore une fois, les commandes envoyées passent par une autre tâche afin de ne pas geler l'interface.

L'interface pour le contrôle automatique :

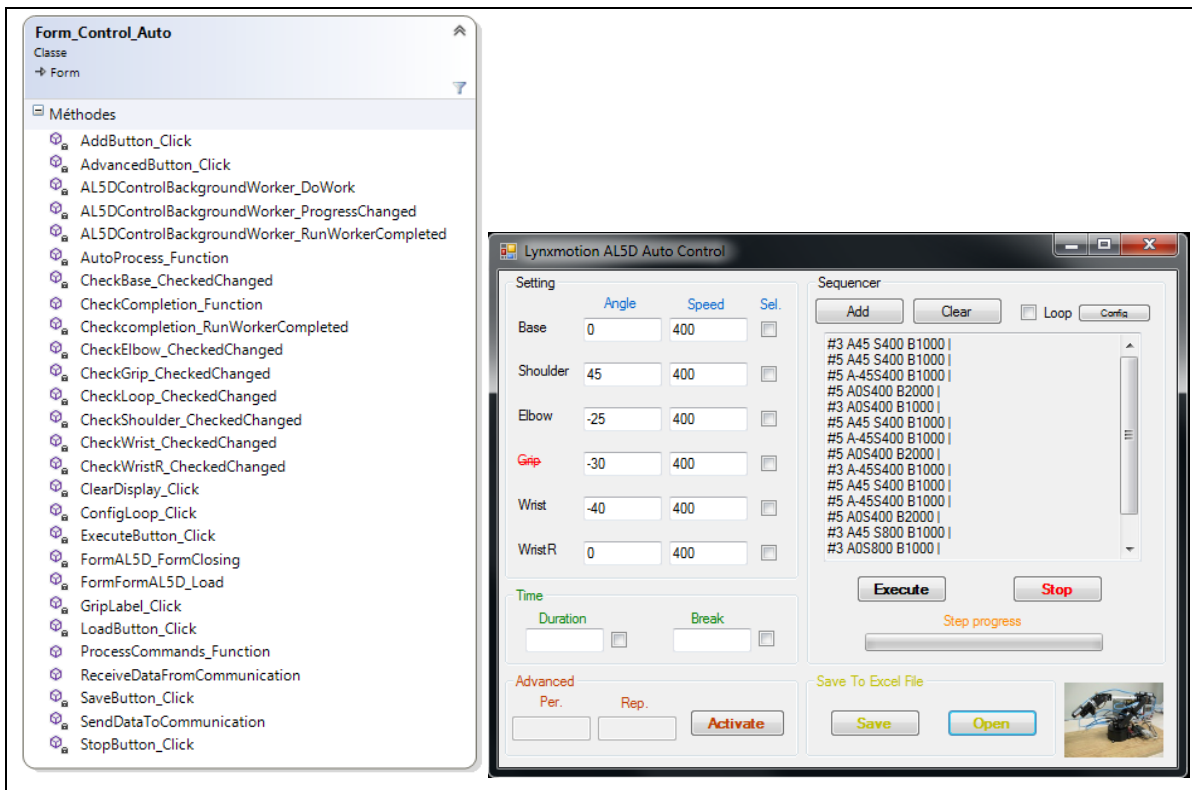


Figure 4.4 Liste des fonctions et imprime écran de l'interface pour le contrôle automatique

L'interface (Figure 4.4) pour le contrôle automatique permet de créer un scénario qui s'effectuera de manière automatique.

- **dans la section « Setting »,** on peut choisir l'angle et la vitesse d'un servo ou de plusieurs servos afin d'effectuer les déplacements en même temps;

- **dans la section « Time »**, il est possible de choisir au lieu de la vitesse, en combien de temps le ou les servos devront atteindre la nouvelle position. On peut également choisir de mettre une pause d'une certaine durée entre chaque action du scénario;
- **dans la section « Sequencer »**, lorsqu'on a complété une action, avec un ou plusieurs servos, on peut l'ajouter en appuyant sur le bouton « Add », celle-ci s'affichera automatiquement dans la fenêtre en dessous. Les différentes actions peuvent être répétées à l'infini en cochant la case « Loop », dans les configurations, il est également possible de choisir le nombre de répétition du scénario;
- **dans la section « Advanced »**, il est possible de lisser les mouvements, c'est-à-dire de mettre une décélération et accélération progressive des servos (Ceci permet d'éviter les secousses);
- **dans la section « Save To Excel File »**, on peut sauvegarder sous le format Excel (xls) le scénario nouvellement créé ou d'en charger un ancien. Il est également possible d'éditer facilement un scénario à l'aide d'Excel pour y faire des modifications.

Les scénarios typiques que l'on utilise placent l'IMU de 12 à 32 positions différentes avec une pause de 5 secondes entre les différentes positions. Ceci nous permet de configurer l'algorithme de calibration initial efficacement et de manière automatique pour chaque IMU du système ibNav.


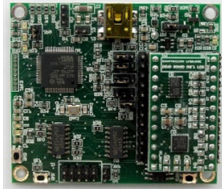


4.2 Présentation des IMUs utilisés pour les tests

Quatre IMUs ont été utilisés afin de tester et de configurer les algorithmes de calibration. Ce sont tous des capteurs à faible coût mais dotés de capteurs inertiels différents. Ces IMUs sont présentés au Tableau 4.1. Les informations techniques de ces différents capteurs inertiels se trouvent en ANNEXE V.

Remarque : La plage de sensibilité a été réglée au minimum sur tous les capteurs pour les algorithmes de calibration exceptée pour le STEVAL-MKI119V1. La correction à l'aide du capteur de température a été effectuée uniquement avec le STEVAL-MKI062V2 et le

STEVAL-MKI119V1. De plus, le STEVAL-MKI119V1, le MPU-9150 MotionFit et le x-IMU ont été utilisés uniquement vers la fin de cette étude afin d'effectuer des tests supplémentaires.

Tableau 4.1 Liste des IMUs utilisés pour tester les algorithmes de calibration

Nom	STEVAL-MKI062V2	STEVAL-MKI119V1	MPU-9150 MotionFit	x-IMU
Constructeur	STMicroelectronics	STMicroelectronics	InvenSense	X-IO Technologies
Photo				
Type de capteurs	LPR430AL LY330ALH LSM303DLH	L3GD20 LSM303DLHC	MPU-9150 AK8975	IMU-3000 LSM303DLH
Sensibilité maximale	Acc : $\pm 2 G$ Gyr : $\pm 300 dps$ Mag : $\pm 1.3 Gs$	Acc $\pm 8 G$ Gyr $\pm 2000 dps$ Mag $\pm 2.5 Gs$	Acc $\pm 2 G$ Gyr $\pm 250 dps$ Mag $\pm 12 Gs$	Acc $\pm 2 G$ Gyr $\pm 250 dps$ Mag $\pm 1.3 Gs$
Fréquence de rafraichissement	50 Hz	200 Hz	80 Hz	256 Hz
Projet du LASSENA	ibNav	Micro-iBB (Chapitre 8)	-	-

4.3 Correction des erreurs de type déterministe

Cette section détaille les algorithmes de calibration créés pour le projet ibNav. Le premier (Figure 4.5) est l'algorithme de calibration initiale utilisant les données de l'accéléromètre, du gyroscope, du magnétomètre et du capteur de température interne au sein d'un filtre de Kalman étendu utilisant le principe de la méthode multi-positions. On obtient ainsi la correction initiale du biais, du facteur d'échelle et du mauvais alignement des axes.

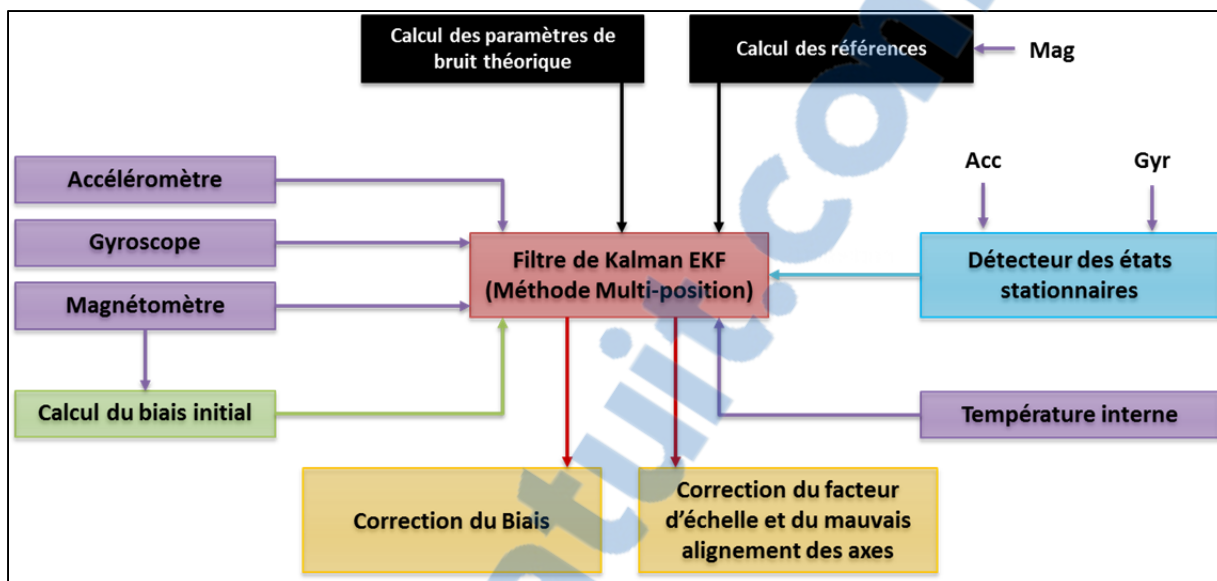


Figure 4.5 Schéma de l'algorithme de calibration initiale du projet ibNav

La deuxième (Figure 4.6) est l'algorithme de calibration continue divisé en deux parties et permettant de corriger le biais et le facteur d'échelle suivant l'influence de la température ainsi que les erreurs de fer doux du magnétomètre à l'aide également d'un filtre de Kalman étendu.

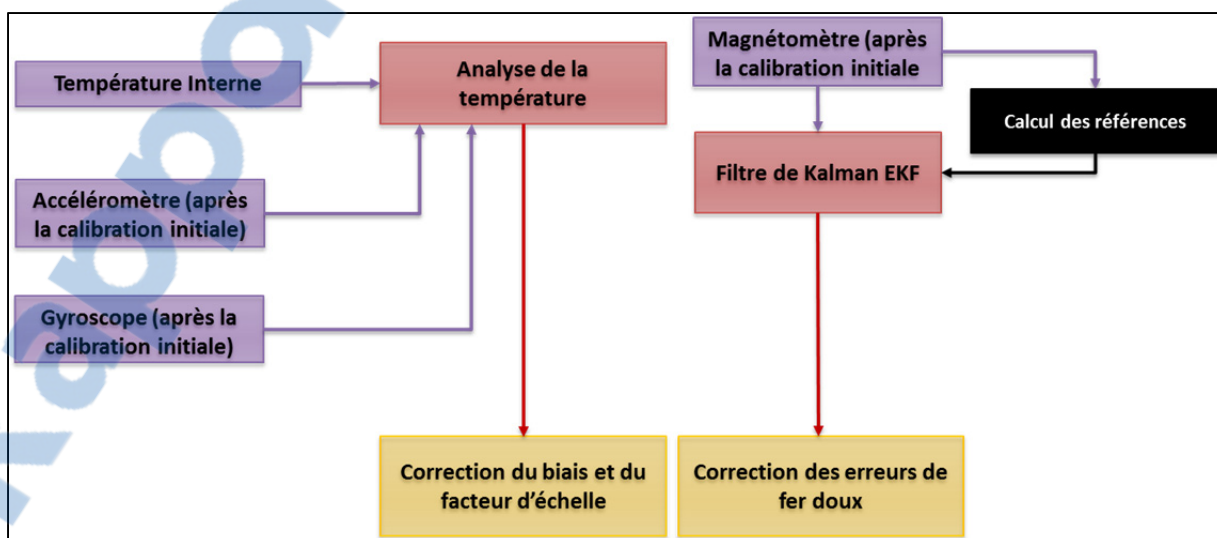


Figure 4.6 Schéma de l'algorithme de calibration continue du projet ibNav

L'objectif de ces méthodes de calibration est de permettre d'exprimer les mesures de chaque capteur inertiel par rapport au repère du mobile. Soit de résoudre l'équation définie en (3.41) :

$$\underline{v}^B = E(\underline{v}^S - \underline{b}) \quad (4.1)$$

Il est donc nécessaire d'estimer la valeur de chaque composante de la matrice E, ainsi que du vecteur de biais, on aura donc 9 composantes à déterminer, soit :

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ 0 & e_{22} & e_{23} \\ 0 & 0 & e_{33} \end{bmatrix} \quad (4.2)$$

$$\underline{b} = [b_x \quad b_y \quad b_z]^T \quad (4.3)$$

4.3.1 Méthode multi-positions

Comme nous l'avons vue dans la revue de la littérature, la méthode dite multi-positions est la plus adéquate afin de modéliser les erreurs déterministes d'une plateforme inertielles sans l'aide de matériel tel qu'une plaque tournante de précision. L'objectif est d'exciter chaque axe de la même façon. Plus on aura de mesures et plus il sera facile d'estimer les inconnues. La démarche permettant d'utiliser cette méthode est indiquée dans le Tableau 4.2 ci-dessous :

Tableau 4.2 Configuration du scénario pour la méthode multi-positions

Type de scénario	Statique
Équipement	Aucun
Nombre de positions	12 à 32
Durée/position (secondes)	5
Durée totale du scénario	70 à 200 secondes
Capteurs requis	Capteurs inertiels (accéléromètre, magnétomètre, gyroscope)
Type de données	Données brutes

Type de capture	Continue (sans interruption)
------------------------	------------------------------

En suivant la démarche de Skog et Händel, 2006 ou de Lavoie, 2012, les différentes inconnues que l'on souhaite estimer peuvent être regroupées à l'intérieur d'un vecteur d'état, tel que :

$$x = [e_{11} \quad e_{12} \quad e_{13} \quad e_{22} \quad e_{23} \quad e_{33} \quad b_x \quad b_y \quad b_z]^T \quad (4.4)$$

Afin de trouver la valeur de ces inconnues, l'objectif est d'étudier la norme des mesures d'accélération, de vitesse angulaire ou de champ magnétique qui devrait être égale à la norme des données de référence. On cherche ainsi à minimiser le problème non-linéaire suivant :

$$L(x) = \sum_{k=0}^{NM-1} \frac{(\|z_k\|^2 - \|f(v_k^S, x)\|^2)^2}{\|z_k\|^2} \quad (4.5)$$

- où :
- $L(x)$: Fonction de coût
 - z_k : Vecteur d'observation
 - N : Nombre d'orientations de la plateforme
 - M : Nombre d'échantillons
 - $f(v_k^S, x)$: Équation du système, égale à l'équation 4.1

Les données de références qui seront utilisées sont des mesures théoriques connues telles que la gravité terrestre, la vitesse de rotation de la terre ou le champ magnétique terrestre. De ce fait il faut que notre système mesure également ces mêmes valeurs afin de pouvoir utiliser la fonction de coût ci-dessus. Pour ce faire, la stationnarité du système devra être imposée pour l'accéléromètre et le gyroscope et puisque l'on souhaite enregistrer en continu afin de rendre la méthode de calibration automatique, il nous faut une méthode de détection des états stationnaires.

4.3.2 Détection des états stationnaires

La détection des états stationnaires est faite à partir des données d'accélération et de vitesse angulaire. L'algorithme comprend quatre conditions qui doivent être toutes respectées afin de considérer que l'on est dans un état stationnaire. Ces quatre calculs sont effectués avec la norme et la variance de ces 2 types de mesures.

Les conditions concernant la norme sont les suivantes (les valeurs sont optimisées pour l'IMU STEVAL-MKI062V2 :

$$C_1 (\text{accéléromètre}) : \quad 0.96 \text{ g} < \|a_K\| < 1.08 \text{ g} \quad (4.6)$$

$$C_2 (\text{gyroscope}) : \quad \|\omega_K\| < 0.3 \text{ rad/s} \quad (4.7)$$

Concernant la variance, on utilise l'algorithme double passe afin de la calculer, les conditions sont les suivantes :

$$C_3 (\text{accéléromètre}) : \quad \left[\frac{1}{10} \sum_{K=1}^{10} (a_K - \bar{a})^2 \right] \leq 0.5 \text{ mg}^2 \quad (4.8)$$

$$C_4 (\text{gyroscope}) : \quad \left[\frac{1}{10} \sum_{K=1}^{10} (\omega_K - \bar{\omega})^2 \right] \leq 0.01 \text{ (rad/s)}^2 \quad (4.9)$$

$$\text{Avec :} \quad \bar{a} = \frac{1}{10} \sum_{k=1}^{10} a_K \text{ et } \bar{\omega} = \frac{1}{10} \sum_{k=1}^{10} \omega_K \quad (4.10)$$

où : \bar{a} et $\bar{\omega}$: Moyenne des accélérations et des vitesses angulaires

Il faut donc que les conditions logiques C_1 et C_2 soient vraies pour 10 échantillons, ainsi que les conditions C_3 et C_4 afin que le système soit considéré en état stationnaire.

Afin d'offrir de meilleurs résultats pour la calibration, il faut que ces conditions soient vraies durant 2 secondes d'enregistrement. On utilise en plus en sortie un filtre médian doté d'une fenêtre de 15 éléments afin de s'assurer que le système soit dans un état stationnaire (voir la Figure 4.7).

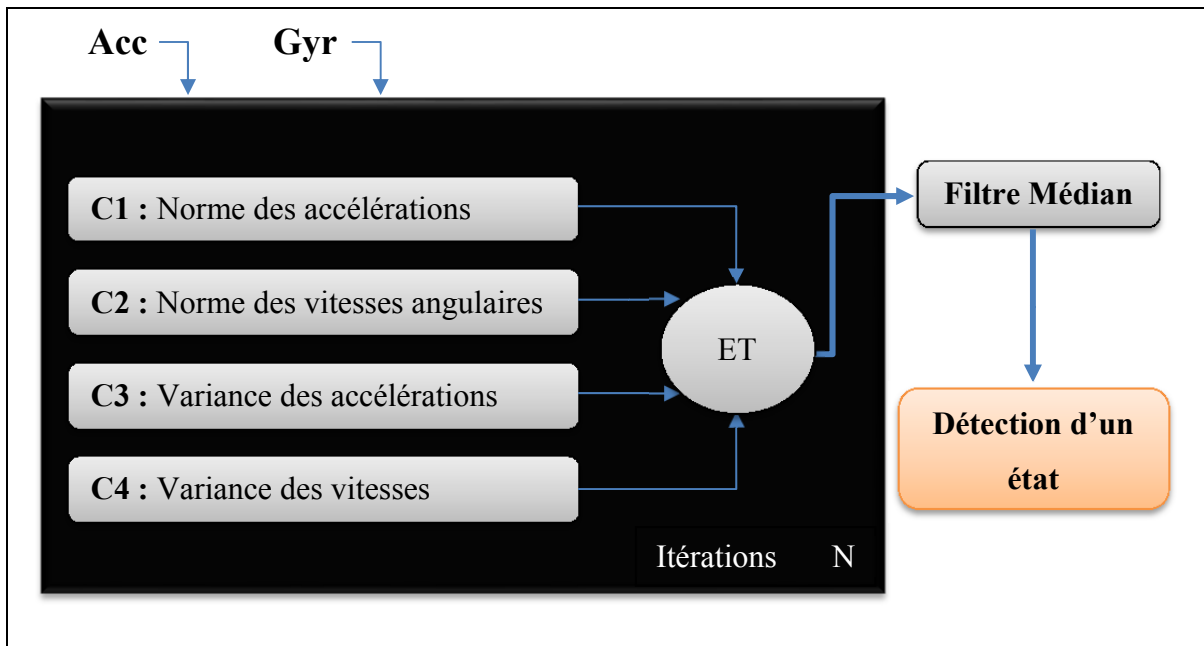


Figure 4.7 Schéma de l'algorithme de détection des états stationnaires

Il est important de noter qu'il faut s'assurer de la stationnarité du système pour que l'algorithme de calibration fonctionne correctement pour l'accéléromètre et le gyroscope. Il est donc préférable de ne pas retenir les états où il y a un doute sur la stationnarité.

La Figure 4.8 représente les résultats obtenus avec l'algorithme ci-dessus développé. Il est important de noter que la référence a été obtenue par rapport au scénario configuré pour le bras robotique. On voit que le résultat que l'on obtient avec l'algorithme est très proche de la référence. L'erreur entre chaque état stationnaire provient essentiellement des secousses du bras robotique dû au poids de l'IMU. Ce point est discuté à la section 4.3.7.

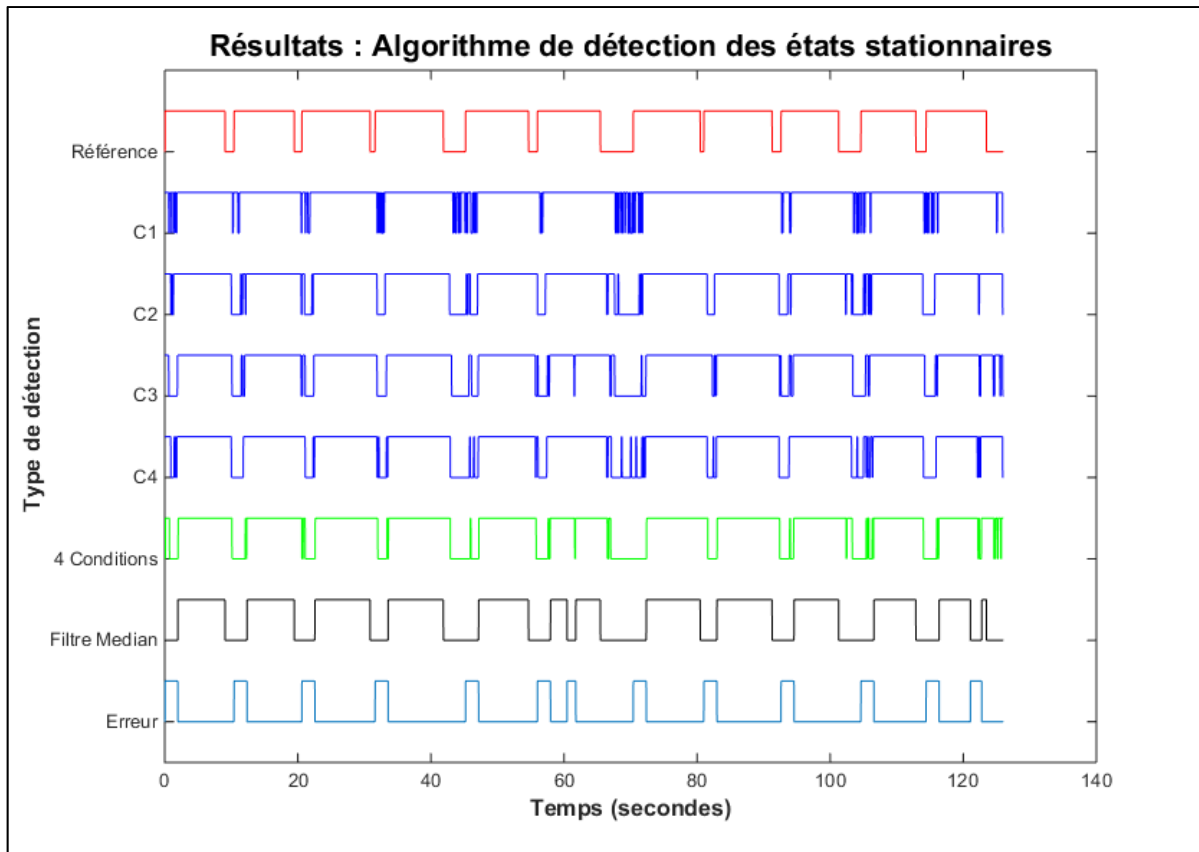


Figure 4.8 Résultats de l'algorithme de détection des états stationnaires

4.3.3 Calibration du biais initial des magnétomètres

Pour le magnétomètre, il est préalablement nécessaire de calculer son biais initial (soit les erreurs de fer dur) puisque celui-ci peut se révéler important. Ces valeurs ainsi évaluées permettront de donner un estimé du biais au vecteur d'état du filtre de Kalman.

La méthode est assez simple, il suffit juste de bouger l'IMU dans toutes les directions et d'enregistrer les valeurs maximales et minimales du champ magnétique sur chacun des axes. On calcule ensuite la valeur moyenne. Soit l'équation suivante :

$$\underline{b}_m = \frac{1}{2}(\underline{m}_{min} + \underline{m}_{max}) \quad (4.11)$$

4.3.4 Calibration initiale des capteurs

La calibration initiale va être effectuée à l'aide d'un filtre de Kalman. On va donc premièrement écrire le système d'équation modélisant ce filtre. Puisque les valeurs que l'on cherche à estimer dans le vecteur d'état (4.4) doivent être constantes dans le temps, la matrice de transition F est invariante et égale à la matrice identité. Quant aux données d'observation, elles sont égales à la norme au carré de la référence théorique (section 3.5). Le système d'équations est donc le suivant :

$$x_{K+1} = I x_K \quad (4.12)$$

$$z_K = h(x) + v_K = \|E(\underline{y}^S(x) - \underline{b})\|^2 + v_K \quad (4.13)$$

Comme nous l'avons dit précédemment, l'équation $h(x)$ étant non-linéaire, il est nécessaire d'utiliser le filtre de Kalman étendu. Afin de rendre le modèle plus puissant, la correction des erreurs déterministes des trois capteurs sera incluse dans un même modèle. De plus puisque le gyroscope utilisé est à faible coût, il n'est pas possible de mesurer la vitesse de rotation de la terre et de ce fait il ne sera pas possible de corriger le facteur d'échelle et le mauvais alignement des axes de ce dernier. Le vecteur d'état (4.4) va donc devenir celui-ci :

$$x = [e_{11a} \ e_{12a} \ e_{13a} \ e_{22a} \ e_{23a} \ e_{33a} \ b_{xa} \ b_{ya} \ b_{za} \\ e_{11m} \ e_{12m} \ e_{13m} \ e_{22m} \ e_{23m} \ e_{33m} \ b_{xm} \ b_{ym} \ b_{zm} \\ b_{x\omega} \ b_{y\omega} \ b_{z\omega}]^T \quad (4.14)$$

Les étapes de réalisation sont les suivantes :

L'initialisation :

Le vecteur d'état \hat{x}_0 doit préalablement être estimé. Malheureusement puisque nous n'avons pas de références et aucun moyen nous permettant d'estimer ces paramètres (excepté pour le biais et le facteur d'échelle du magnétomètre), on considère qu'il n'y a aucune erreur. Soit l'équation suivante :

$$\hat{x}_0 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 2 & b_{x_m} & b_{y_m} & b_{z_m} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (4.15)$$

Puisque cette hypothèse d'erreur nulle est fautive, la matrice de covariance P_0 doit prendre des valeurs de variance correspondant à l'incertitude posée sur \hat{x}_0 . Ces valeurs doivent être assez grandes afin que le système converge et qu'il donne un bon estimé des erreurs. Dans le cas contraire, aucune correction ne sera effectuée. Les valeurs des composantes de P_0 pour la calibration des IMUs du projet ibNav sont inscrites dans le Tableau 4.3. On a l'équation :

$$P_0 = \text{diag}([\sigma_{e_{11a}}^2 \quad \sigma_{e_{12a}}^2 \quad \sigma_{e_{13a}}^2 \quad \sigma_{e_{22a}}^2 \quad \sigma_{e_{23a}}^2 \quad \sigma_{e_{33a}}^2 \quad \sigma_{b_{x_a}}^2 \quad \sigma_{b_{y_a}}^2 \quad \sigma_{b_{z_a}}^2 \\ \sigma_{e_{11m}}^2 \quad \sigma_{e_{12m}}^2 \quad \sigma_{e_{13m}}^2 \quad \sigma_{e_{22m}}^2 \quad \sigma_{e_{23m}}^2 \quad \sigma_{e_{33m}}^2 \quad \sigma_{b_{x_m}}^2 \quad \sigma_{b_{y_m}}^2 \quad \sigma_{b_{z_m}}^2 \\ \sigma_{b_{x_\omega}}^2 \quad \sigma_{b_{y_\omega}}^2 \quad \sigma_{b_{z_\omega}}^2]) \quad (4.16)$$

Tableau 4.3 Valeurs initiales des composantes de la matrice P

P_0	$\sigma_{e_{11}}$	$\sigma_{e_{12}}$	$\sigma_{e_{13}}$	$\sigma_{e_{22}}$	$\sigma_{e_{23}}$	$\sigma_{e_{33}}$	σ_{b_x}	σ_{b_y}	σ_{b_z}
Acc (g^2)	0.1	0.01	0.01	0.1	0.01	0.1	0.5	0.5	0.5
Mag (uT^2)	0.1	0.01	0.01	0.1	0.01	0.1	0.01	0.01	0.01
Gyr (rad^2/s^2)							0.1	0.1	0.1

La dernière étape de l'initialisation consiste à déterminer les valeurs constantes de l'observation z , soit:

$$z_K = [\|a_{ref}\|^2 \quad \|m_{ref}\|^2 \quad \|\omega_{ref}\|^2]^T \quad (4.17)$$

où :

- a_{ref} : Correspond à 1 g (valeur d'accélération conventionnelle de la pesanteur à la surface de la terre)
- m_{ref} : Correspond à l'équation (3.55)
- ω_{ref} : Correspond à 0 puisque le gyroscope ne peut mesurer la vitesse de rotation de la terre égale à $7.292115E^{-5} \text{ rad/s}$

La prédiction :

Puisque les paramètres que l'on cherche à estimer doivent être constants dans le temps, il n'y a pas de prédiction, soit :

$$\hat{x}_K^- = \hat{x}_{K-1}^+ \quad (4.18)$$

$$P_K^- = P_{K-1}^+ \quad (4.19)$$

La mise à jour des mesures :

La première étape consiste à calculer la mesure résiduelle. Pour ce faire, il faut calculer l'équation suivante :

$$\begin{aligned} \delta z_K = [& \|a_{ref}\|^2 - \|E(a^S - b)\|^2 \\ & \|m_{ref}\|^2 - \|E(m^S - b)\|^2 \\ & \|\omega_{ref}\|^2 - \|\omega^S - b\|^2]^T \end{aligned} \quad (4.20)$$

Et puisque le modèle est non linéaire, il faudra par la suite calculer sa Jacobienne H_K , soit :

$$H_K = \left. \frac{\delta h(x)}{\delta x} \right|_{x=\hat{x}} \quad (4.21)$$

A cet instant l'algorithme de détection d'états stationnaires va entrer en jeu. Il va jouer sur les valeurs de la matrice de covariance des bruits d'observations R. Le principe est que le gain de Kalman K_K soit calculé lorsqu'on est en état stationnaire mais prend une valeur tendant

vers l'infiniment petit dans le cas contraire, on dit alors au système que la confiance accordée à l'observation est nulle. La détection d'états stationnaires n'est pas utile pour la calibration des magnétomètres car la norme du champ magnétique doit être identique indépendamment du mouvement et de l'orientation du capteur.

Ensuite, on applique les formules traditionnelles pour le calcul de \hat{x}_K^+ avec sa matrice de covariance P_K^+ . Finalement, les étapes de prédiction et de mise à jours des mesures se répètent pour chaque itération jusqu'à la fin de l'enregistrement des données. Lors de la fin de l'étape de calibration initiale, il faut également que la valeur de température du système soit prise en compte, on l'appellera T_{ref} .

4.3.5 Calibration continue des capteurs

La calibration continue correspond à la prise en compte de la température et à la correction des erreurs de fer doux des magnétomètres (soit la correction des perturbations du champ magnétique dû à l'environnement).

En l'absence de chambre thermique, il n'est pas possible de connaître l'impact réel de la température sur le biais et le facteur d'échelle de l'accéléromètre et des gyroscopes. On va tout de même pouvoir effectuer une correction de ceux-ci grâce aux valeurs théoriques de la documentation technique, soit :

Tableau 4.4 Influence de la température sur les mesures

	Biais	Facteur d'échelle
Accéléromètre	$\pm 0.5 \text{ mg}/^\circ\text{C}$	$\pm 0.01 \text{ } \%/^\circ\text{C}$
Gyroscope Roulis, Tangage	$\pm 0.05 \text{ dps}/^\circ\text{C}$	$\pm 0.07 \text{ } \%/^\circ\text{C}$
Gyroscope Lacet	$\pm 0.02 \text{ dps}/^\circ\text{C}$	$\pm 0.01 \text{ } \%/^\circ\text{C}$

Par exemple pour l'accéléromètre on obtient les équations suivantes :

$$b_K^a = b_{K-1}^a + 0.0005[T_{K-1} - T_K] \quad (4.21)$$

$$E_K^a = \sum_{i=1}^3 E_{K-1}^a(i, i) \times 0.01[T_{K-1} - T_K] \quad (4.22)$$

Concernant le magnétomètre, c'est une version simplifiée du filtre de Kalman vue précédemment permettant de corriger la matrice E durant le fonctionnement du système. Cette dernière se corrigeant lors de la détection d'un état stationnaire. Le vecteur de mesure définie en (4.14) devient donc :

$$x = [e_{11_m} \quad e_{12_m} \quad e_{13_m} \quad e_{22_m} \quad e_{23_m} \quad e_{33_m}]^T \quad (4.23)$$

Et h(x) devient :

$$h(x) = \|E \underline{v}^s(x)\|^2 \quad (4.24)$$

Ce modèle est utilisé juste avant le filtre de Kalman des algorithmes de navigation AHRS et INS (CHAPITRE 5). Une version l'incluant dans la méthode de correction de la dérive est en cours de développement.

4.3.6 Démonstration du besoin de la calibration

Les capteurs inertiels à faible coût présentent des erreurs déterministes importantes, faciles à visualiser. Prenons par exemple le cas de l'accéléromètre où la norme du vecteur de gravité terrestre théorique est égale à 1 g lors de l'état stationnaire de l'IMU. La Figure 4.9 représente des enregistrements effectués pour la calibration initiale, soit des états stationnaires d'environ 5 secondes suivis d'un changement d'orientation rapide. On affiche ici la norme du vecteur de gravité des 4 IMUs à faible coût utilisés pour les tests et on observe que durant un état stationnaire, la valeur n'est pas forcément égale à 1 g. Celle-ci varie suivant l'orientation de

l'IMU et s'explique par la présence d'erreurs de facteurs d'échelle et de mauvais alignements des axes.

Remarque : Le bruit sur la norme du vecteur de gravité est plus élevé pour le Micro-iBB du fait qu'il fonctionne avec une sensibilité maximale de ± 8 g au lieu de ± 2 g. L'algorithme de calibration initiale présent dans le x-IMU était désactivé pour ce test.

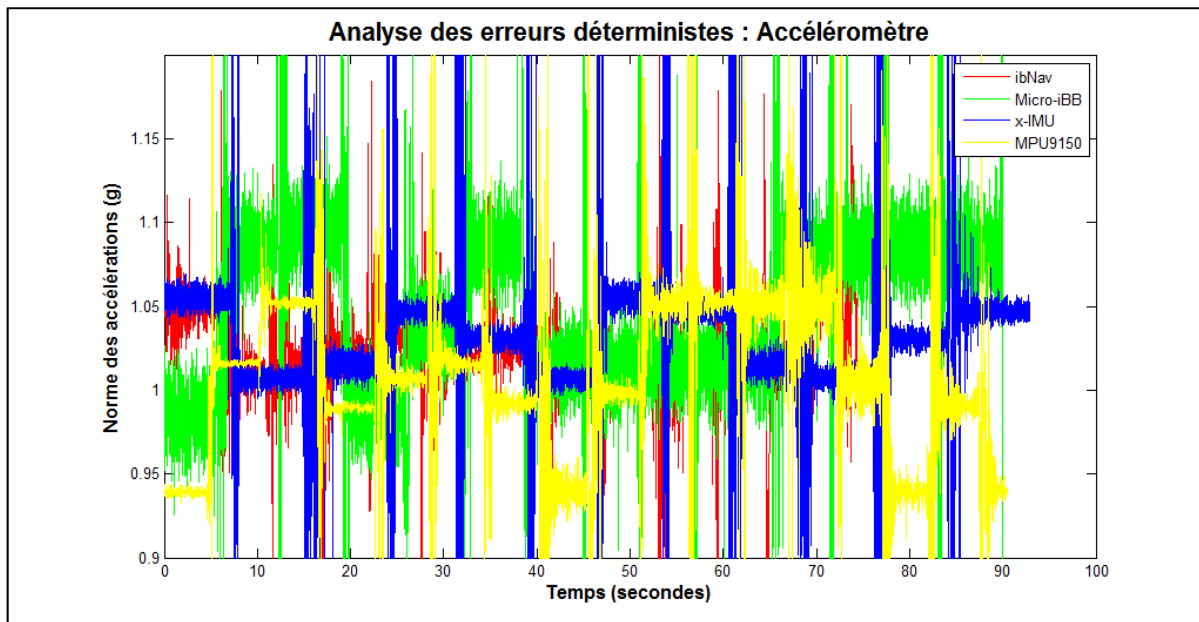


Figure 4.9 Norme des accélérations suivant différentes orientations

Concernant le gyroscope, on constate graphiquement la présence d'un biais puisqu'en l'absence de mouvement la norme de la vitesse de rotation devrait être nulle (Figure 4.10). Pour le magnétomètre, on devrait visualiser la même constante au cours du temps, cependant ces valeurs sont très différentes, on visualise donc des erreurs déterministes importantes (Figure 4.11).

L'algorithme de calibration va donc être indispensable afin d'améliorer la qualité des IMUs à faible coût (ces IMUs étant produits massivement sans qu'une personne est pris soin de les calibrer). Cette amélioration est donc un prérequis indispensable pour les besoins du projet. Dans le cas contraire ces erreurs s'ajouteront continuellement à la solution de navigation et finiront par faire dériver la solution de manière importante.

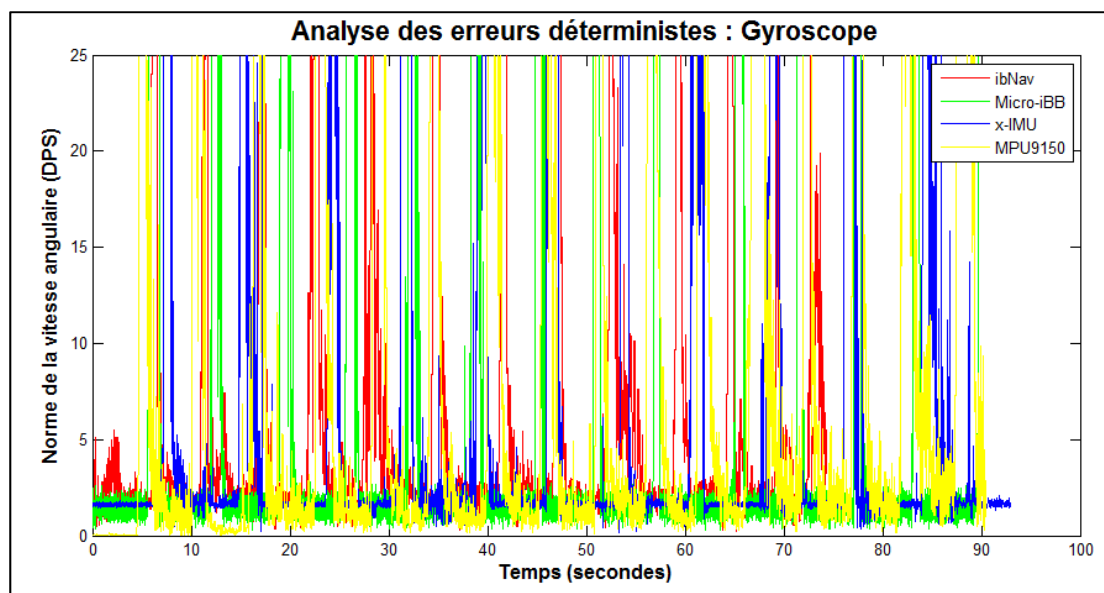


Figure 4.10 Norme de la vitesse angulaire suivant différentes orientations

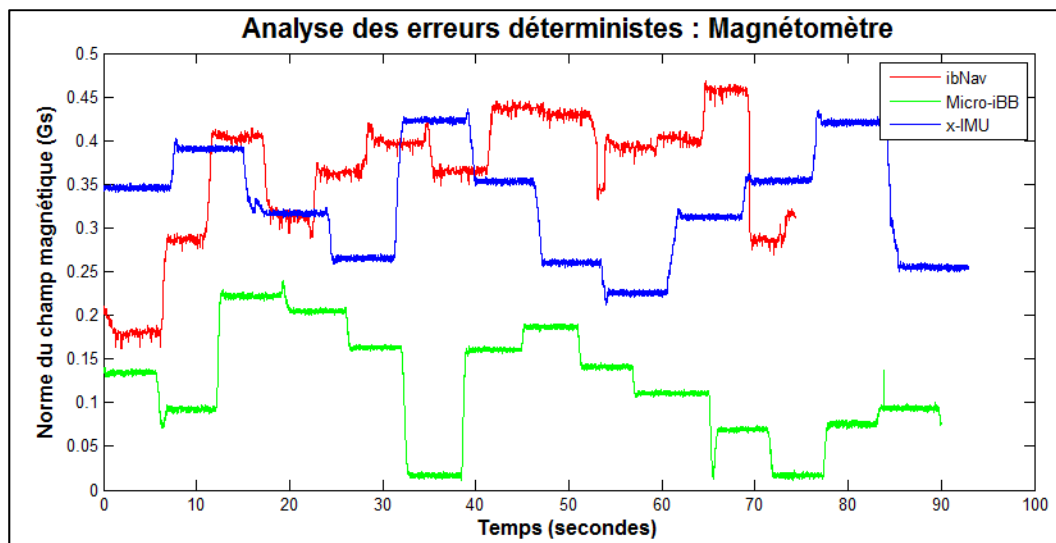


Figure 4.11 Norme du champ magnétique suivant différentes orientations

La Figure 4.12 représente le champ magnétique obtenu avec un des magnétomètres du projet ibNav et sa comparaison avec le champ magnétique terrestre. Le cercle en noir et la sphère en gris représentent le champ magnétique terrestre théorique à l'endroit où l'acquisition a été effectuée, calculé à l'aide de l'équation (3.55). Les formes colorées sont les données

provenant du magnétomètre. On remarque bien la présence d'erreurs déterministes importantes sur les données du magnétomètre, soit :

- de biais (le nuage de points n'est pas centré en 0);
- de facteur d'échelle (le nuage de points n'est pas à la même échelle que la référence);
- de mauvais alignement (Le nuage de points ne forme pas exactement un cercle).

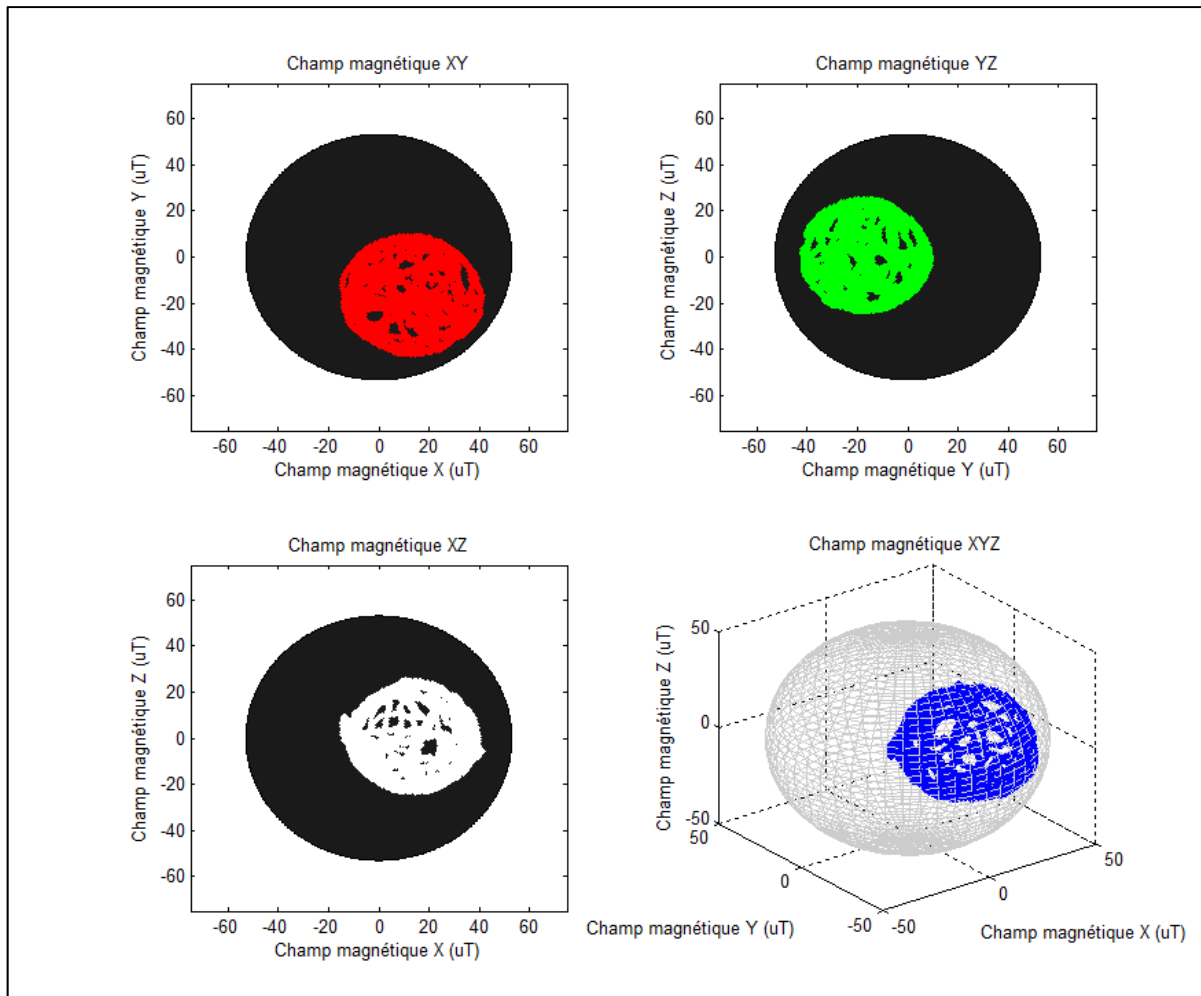


Figure 4.12 Champ magnétique mesuré comparé à la référence théorique

La seconde analyse va démontrer qu'il est indispensable d'effectuer la calibration initiale pour chaque IMU du système ibNav indépendamment. Malgré que chaque IMU du système ibNav est un STEVAL-MKI062V2 et donc utilise les mêmes composants, on remarque au Tableau 4.5 que les erreurs déterministes sont différentes d'un IMU à l'autre. Ce tableau

contient les valeurs trouvées des composantes de la matrice E et du biais pour la calibration de l'accéléromètre de 4 IMUs du système ibNav. Ces différences s'expliquent par le fait que chaque capteur a des erreurs déterministes qui lui sont propres, causées lors de leur conception (Les différentes composantes pour chaque IMU du projet ibNav sont en ANNEXE VI).

Tableau 4.5 Différences d'erreurs entre les IMU STEVAL-MKI062V2 du projet ibNav

	e_{11}	e_{12}	e_{13}	e_{21}	e_{22}	e_{33}	b_x	b_y	b_z
IMU 1 - Torse	0.9811	1.50×10^{-4}	-1.59×10^{-3}	-1.19×10^{-3}	0.9906	0.9709	8.35×10^{-3}	7.84×10^{-3}	2.35×10^{-2}
IMU 2 - Main droite	0.9761	-3.47×10^{-3}	-4.56×10^{-3}	-5.81×10^{-3}	0.9636	0.9902	2.25×10^{-2}	-9.43×10^{-3}	8.43×10^{-3}
IMU 3 - Avant-bras droit	0.9803	5.32×10^{-3}	2.4×10^{-3}	2.18×10^{-3}	0.9811	0.9723	-1.07×10^{-2}	1.65×10^{-3}	-1.35×10^{-2}
IMU 4 - Bras droit	0.9962	-7.39×10^{-4}	-1.24×10^{-3}	-7.51×10^{-4}	0.9949	0.9622	3.15×10^{-2}	3.42×10^{-2}	-3.99×10^{-2}

Finalement la dernière analyse va montrer l'impact de la température sur les données des capteurs inertiels. La Figure 4.13 représente un enregistrement statique de l'IMU du projet ibNav où la température de l'accéléromètre a été diminuée jusqu'à 0°C avant d'être démarré. On remarque la déviation causée sur les données d'accélération qui devraient être stables.

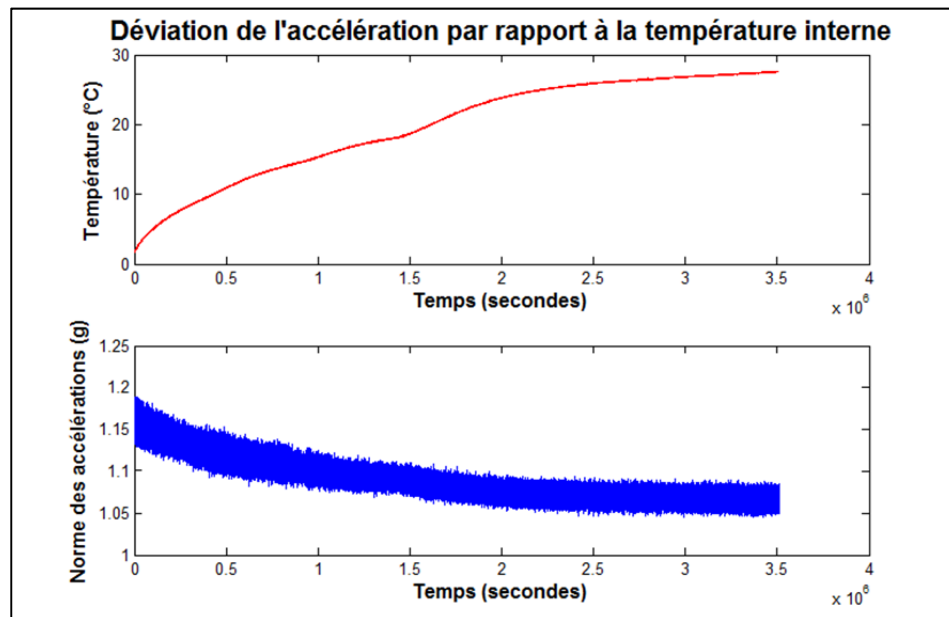


Figure 4.13 Déviation de l'accélération par rapport à la température interne

4.3.7 Analyse des performances et discussion

Le premier résultat qui est présenté à la Figure 4.14 est la norme du vecteur d'accélération avant et après la calibration initiale pour un des capteurs du projet ibNav. La calibration initiale qui est appliquée correspond aux résultats du Tableau 4.5 Différences d'erreurs entre les IMU STEVAL-MKI062V2 du projet ibNav pour l'IMU 1 (Torse). On conclut par l'analyse graphique l'efficacité de l'algorithme dans l'élimination des erreurs déterministes pour l'accéléromètre.

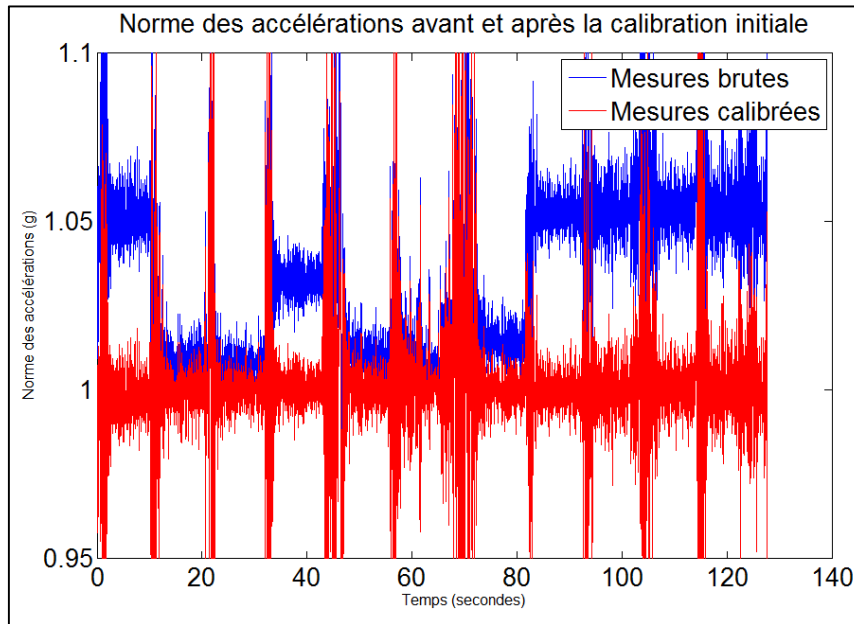


Figure 4.14 Résultat de la calibration pour l'accéléromètre de l'IMU maître du projet ibNav

La Figure 4.15 représente des résultats similaires mais pour le gyroscope. On voit qu'après la calibration initiale, lorsqu'on a un état stationnaire, la valeur tend vers 0 rad/s, ce qui correspond à la valeur de référence choisie. Pour rappel, la méthode multi-position corrige seulement l'erreur de biais pour le gyroscope. Cependant, le facteur d'échelle est légèrement corrigé par l'application de l'équation (4.22) prenant en compte la température interne de l'IMU.

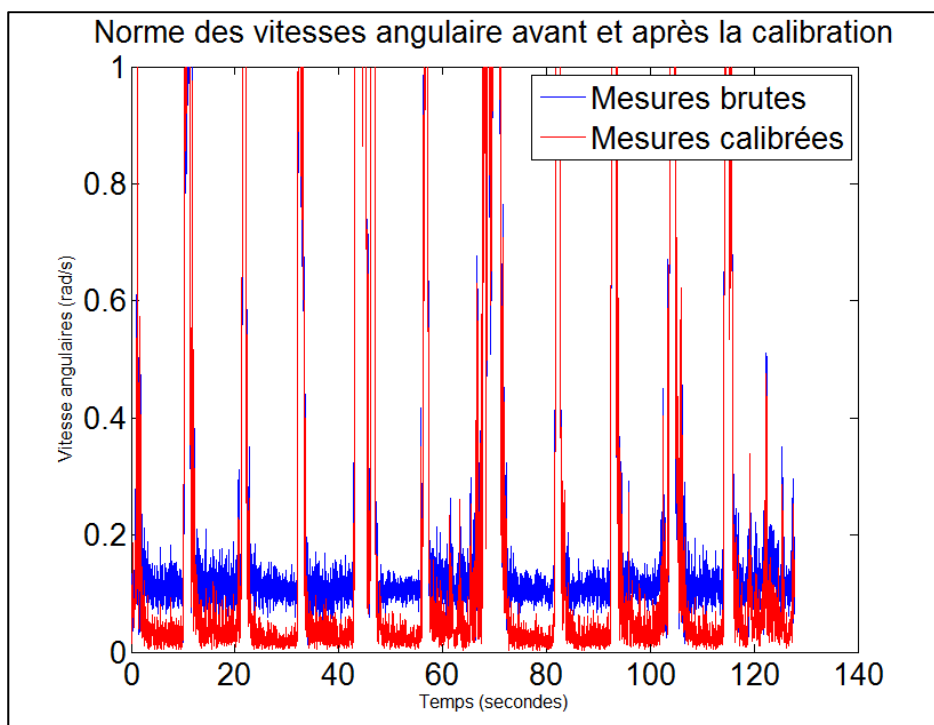


Figure 4.15 Résultat de la calibration pour le gyroscope de l'IMU maître du projet ibNav

Concernant la calibration du magnétomètre, deux étapes vont être nécessaires. Les données utilisées sont représentées à la Figure 4.12. On va premièrement utiliser le filtre de Kalman avec la même méthodologie que pour l'accéléromètre et le gyroscope excepté qu'on n'a pas à prendre en compte la détection des états stationnaires. On obtient ainsi le résultat de la Figure 4.16. On remarque directement la calibration du biais et du facteur d'échelle.

On va deuxièmement effectuer un calcul très simple permettant de placer chaque mesure précédemment calibrée sur la sphère de référence de rayon la norme du champ magnétique terrestre. Ces points sont placés sur le contour de la sphère à l'intersection avec la droite passant par l'origine et par le point calibré. Les résultats de cette deuxième étape sont affichés sur la Figure 4.17.

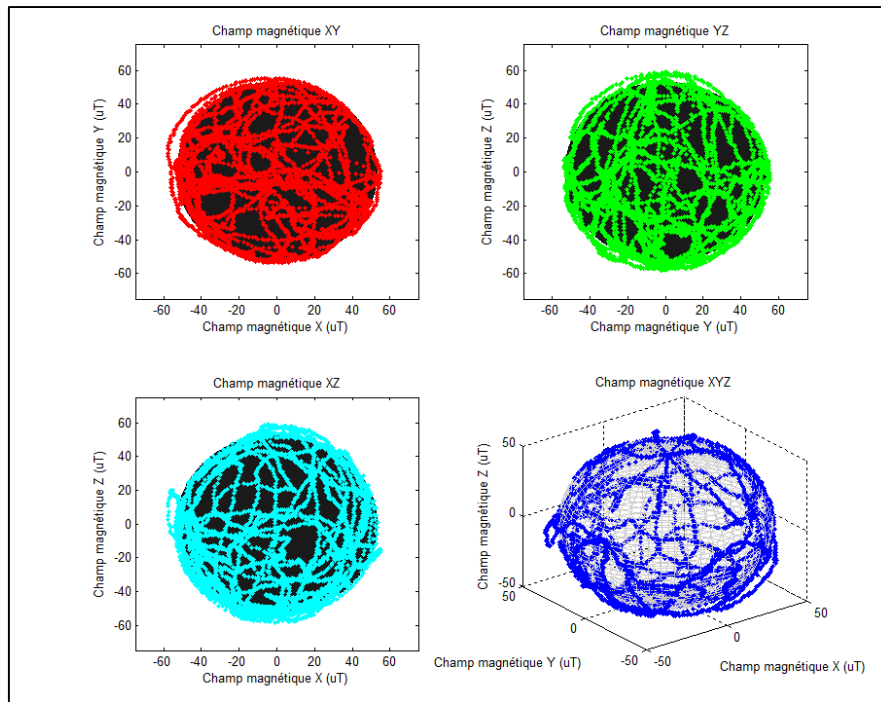


Figure 4.16 Résultat de la calibration EKF pour le magnétomètre du projet ibNav

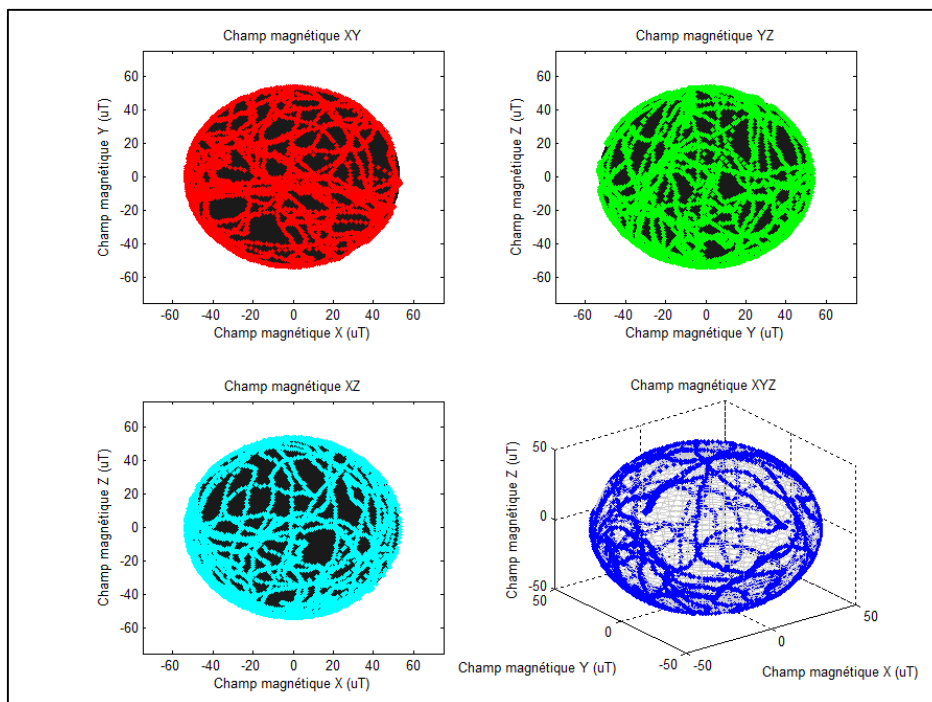


Figure 4.17 Résultat de la calibration EKF + Lissage pour le magnétomètre du projet ibNav

Finalement la Figure 4.18 représente la norme du champ magnétique en fonction du temps, avant et après la calibration initiale, ainsi que l'erreur résiduelle. Suite à l'application du filtre de Kalman sur les données brutes, les erreurs déterministes sont nettement réduites. Les données calibrées lissées ne sont pas visibles sur le graphique car elles correspondent à la droite du champ magnétique terrestre de référence, il n'est donc pas possible d'évaluer l'efficacité du lissage de cette façon. On voit cependant une légère amélioration de la solution de navigation lorsque le lissage est effectué.

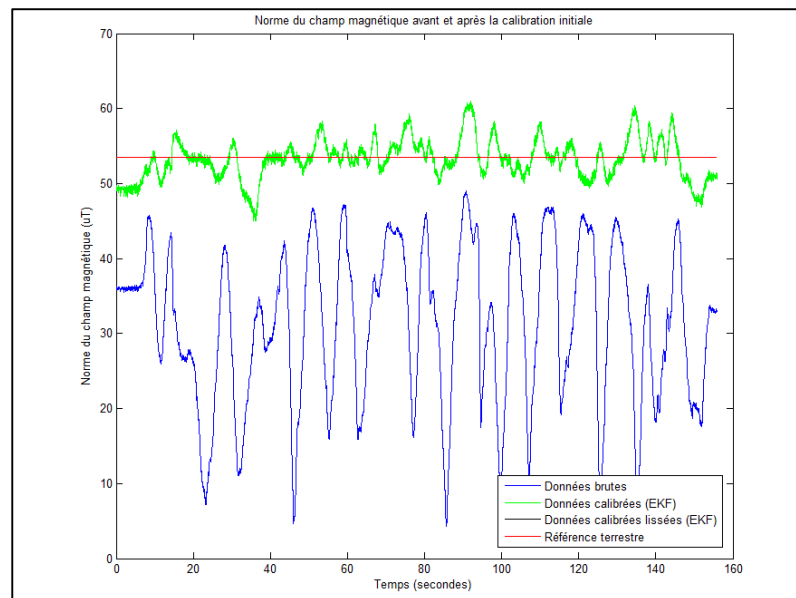


Figure 4.18 Résultat de la calibration pour le magnétomètre de l'IMU maître du projet ibNav

La calibration continue va être une étape absolument nécessaire pour la calibration des erreurs de fer doux (équivalents au mauvais alignement des axes et au facteur d'échelle) puisque le champ magnétique de l'environnement change de manière importante au cours du temps. La Figure 4.19 représente un enregistrement effectué dans un autre environnement où la calibration initiale est appliquée. On remarque que la correction des erreurs de fer dur (le biais) est correcte mais que des erreurs de fer doux sont présentes. On va devoir ainsi effectuer une calibration durant le fonctionnement du système (CC) à l'aide d'une version simplifiée du filtre de Kalman développé pour la calibration initiale. La Figure 4.20 montre le résultat.

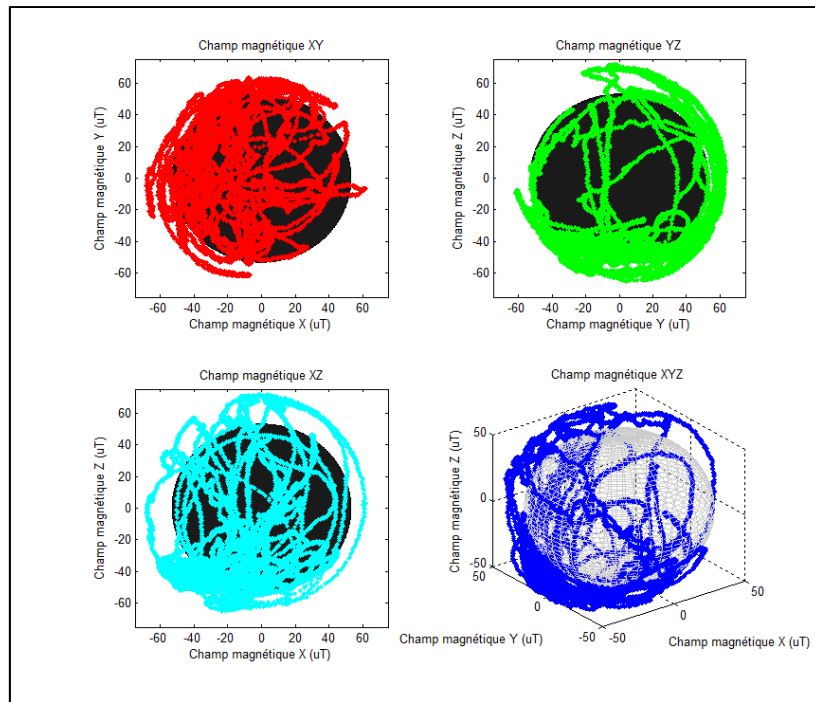


Figure 4.19 Données du magnétomètre durant le fonctionnement du système sans CC

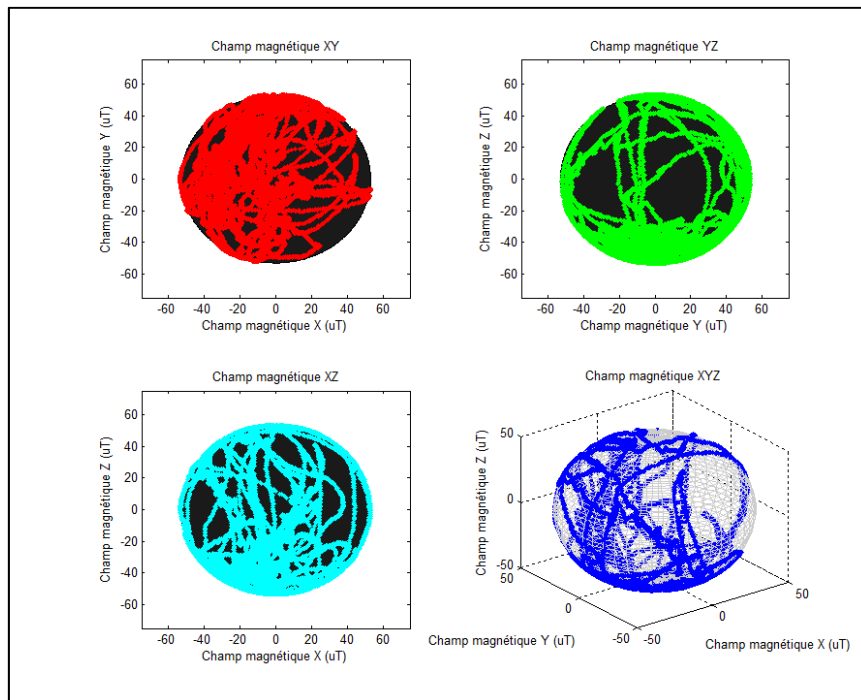


Figure 4.20 Données du magnétomètre durant le fonctionnement du système avec CC

Finalement durant la calibration continue, la température du système va être comparée à celle de référence retenue lors de la fin de la calibration initiale. En appliquant les valeurs théoriques d'influence de la température sur le biais et le facteur d'échelle, on obtient par exemple pour l'accéléromètre la Figure 4.21. Les résultats sont similaires pour le gyroscope.

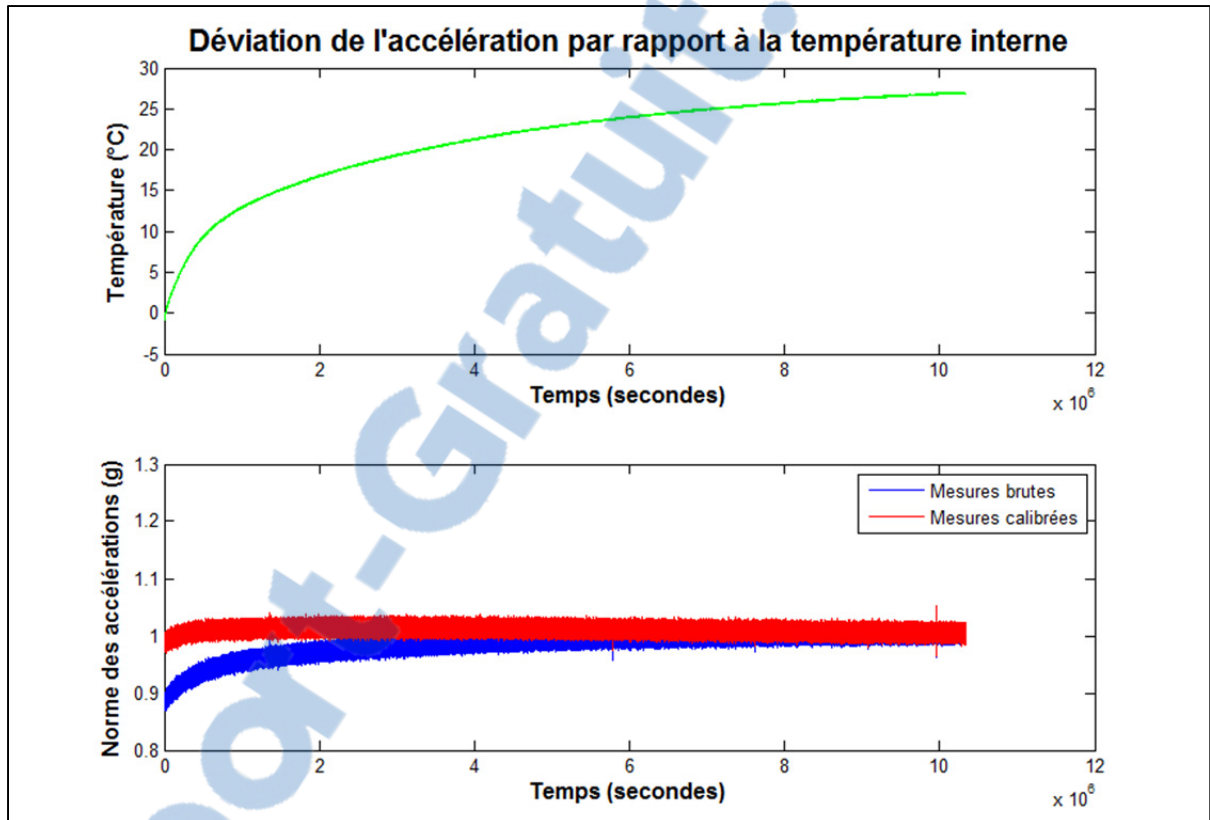


Figure 4.21 Correction des erreurs déterministes de l'accéléromètre à l'aide de la température

En conclusion, ces différentes calibrations des erreurs déterministes permettent d'appliquer des corrections importantes sur les capteurs inertiels à faible coût, permettant un meilleur résultat lors de l'utilisation des algorithmes de navigation.

On peut noter cependant que des améliorations supplémentaires sont possibles, permettant d'avoir un modèle de calibration plus puissant. Certaines améliorations seront effectuées suite à cette maîtrise, notons par exemple :

- l'utilisation d'une chambre thermique permettrait d'avoir un modèle plus précis pour la correction du biais et du facteur d'échelle des capteurs inertiels dû à l'influence de la température;
- l'utilisation d'une plaque tournante permettrait de corriger le facteur d'échelle et le mauvais alignement des axes du gyroscope;
- l'utilisation d'électro-aimants ayant une forte aimantation et aligné de manière précise avec les axes du boîtier, permettrait une meilleure calibration du magnétomètre. La Figure 4.22 représente une plateforme de ce type en cours de développement par l'auteur de ce mémoire. Elle est dotée de 5 électro-aimants, de 2 servomoteurs et d'un microcontrôleur permettant de générer un scénario de calibration de manière automatique;
- l'utilisation d'un bras robotique dont les servomoteurs sont plus puissants serait préférable afin d'éviter les secousses créant un bruit supplémentaire. Idéalement se serait préférable d'utiliser la plateforme en développement, présentée à la Figure 5.2.

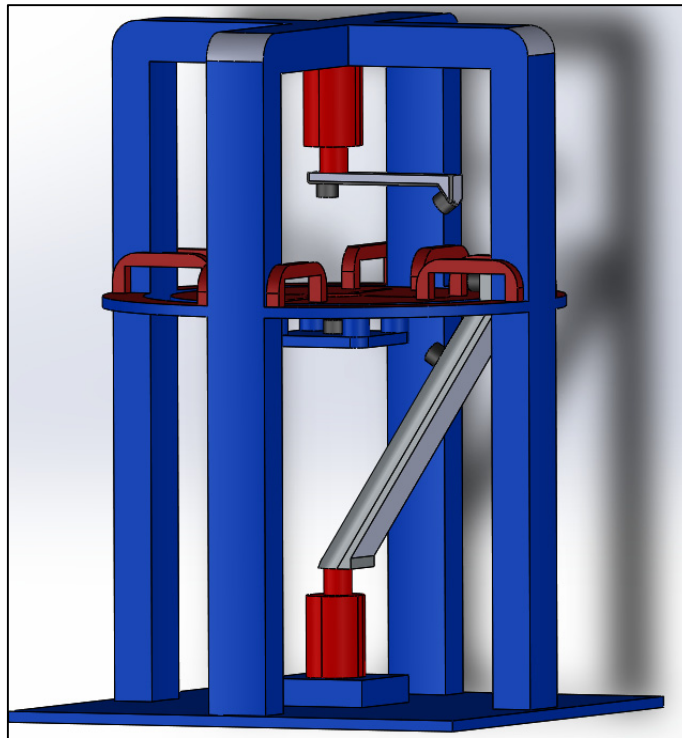


Figure 4.22 Prototype de plateforme de calibration des magnétomètres à faible coût

4.4 Caractérisation des erreurs de type stochastique

Comme vue dans la revue de la littérature, concernant le bruit aléatoire sur les données, nous nous intéresserons uniquement au bruit blanc (une composante à haute fréquence) puisque celle-ci peut facilement être corrigée à l'aide du filtre de Kalman. La correction des autres types d'erreurs aléatoires demanderaient des modèles plus complexes qui ne sont pas étudiés dans le cadre de la maîtrise. Afin de caractériser le bruit blanc, nous utiliserons donc la méthode de la variance d'Allan (AVAR).

4.4.1 Application de la variance d'Allan

La méthode de la variance d'Allan permet de caractériser les erreurs aléatoires sur une trame de données en mesurant leur stabilité sur une période de temps définie. L'analyse s'effectue donc dans le domaine temporel. Afin d'utiliser cette méthode efficacement, il est premièrement nécessaire de faire un enregistrement de N données, avec $N > 1\,000\,000$ échantillons (afin d'avoir un bon estimé des erreurs). L'échantillonnage des données doit se faire à un intervalle de temps constant Δt . Ensuite, on regroupe ces mesures sur un intervalle n , tel que $n < N/2$. On va alors calculer la moyenne des données de chacun de ces groupes, soit :

$$\bar{y}_K(\tau) = \frac{1}{n} \sum_{i=1}^n y_{(K-1)n+i}(\tau) \quad (4.25)$$

où :

- τ : Période de temps du groupe de données K
- n : Nombre d'échantillons du groupe de données K
- $\bar{y}_K(\tau)$: Moyenne des données du groupe K

La variance d'Allan est calculée sur chaque groupe de données grâce à l'équation suivante :

$$\sigma^2(\tau) = \frac{1}{2 \times (N - 2n)} \sum_{i=1}^{N-2n} (y_{i+1}(\tau) - y_i(\tau))^2 \quad (4.26)$$

où : $\sigma^2(\tau)$: Variance d'Allan de taille τ
 N : Nombre total d'échantillons

Finalement, en faisant varier τ , on obtient une courbe similaire à la Figure 2.3 représentant l'écart-type d'Allan en fonction des valeurs de τ . Puisque chaque type de bruit a une pente différente, on peut déterminer facilement les coefficients pour les bruits suivants : Le bruit de quantification, le processus de marche aléatoire, le bruit blanc, l'instabilité du biais et le bruit de rampe. On notera donc que le bruit blanc a une pente de $-0.5 \tau + b$. On peut directement avoir le coefficient du bruit blanc en lisant la valeur de l'écart type pour $\tau = 1$.

4.4.2 Analyse des résultats et discussion

En effectuant l'analyse de la variance d'Allan pour les 3 axes des accéléromètres, gyroscopes et magnétomètres des 4 IMUs à faible coût, on obtient les graphiques suivant :

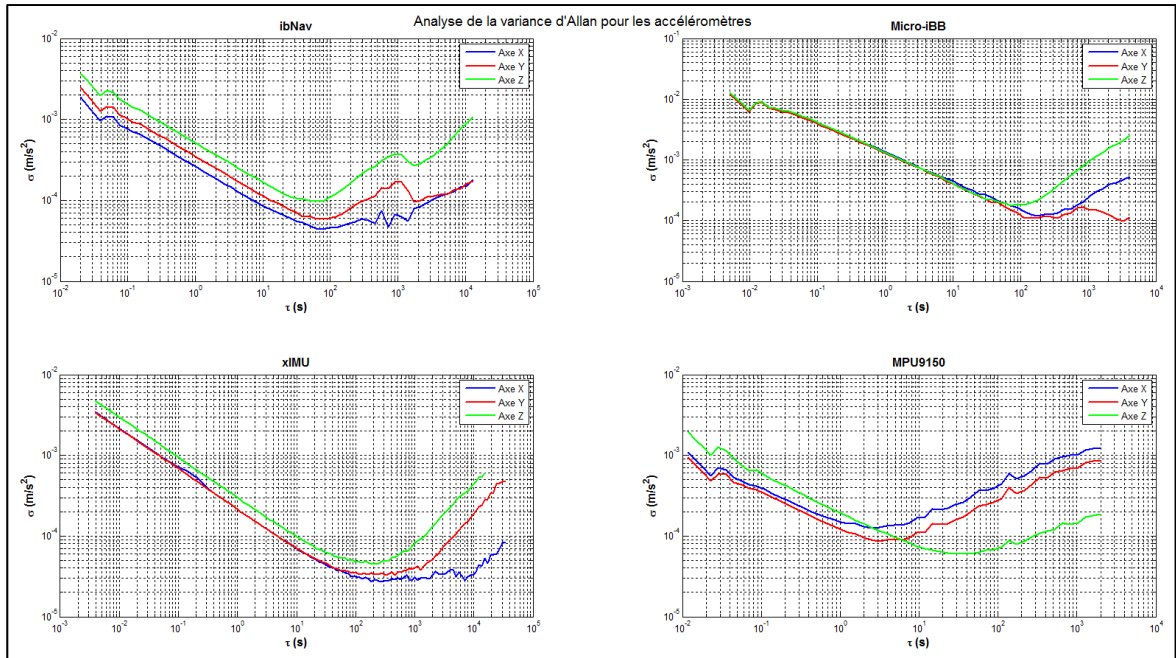


Figure 4.23 Analyse de la variance d'Allan pour les accéléromètres de 4 IMUs à faible coût

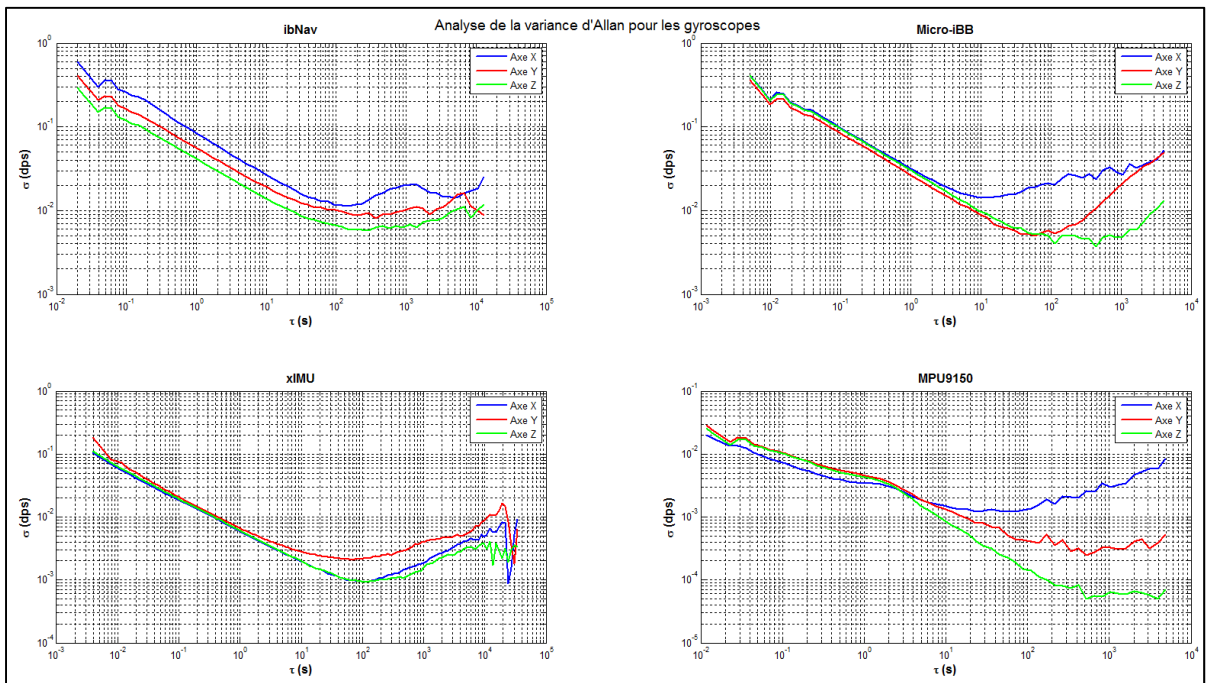


Figure 4.24 Analyse de la variance d'Allan pour les gyroscopes de 4 IMUs à faible coût

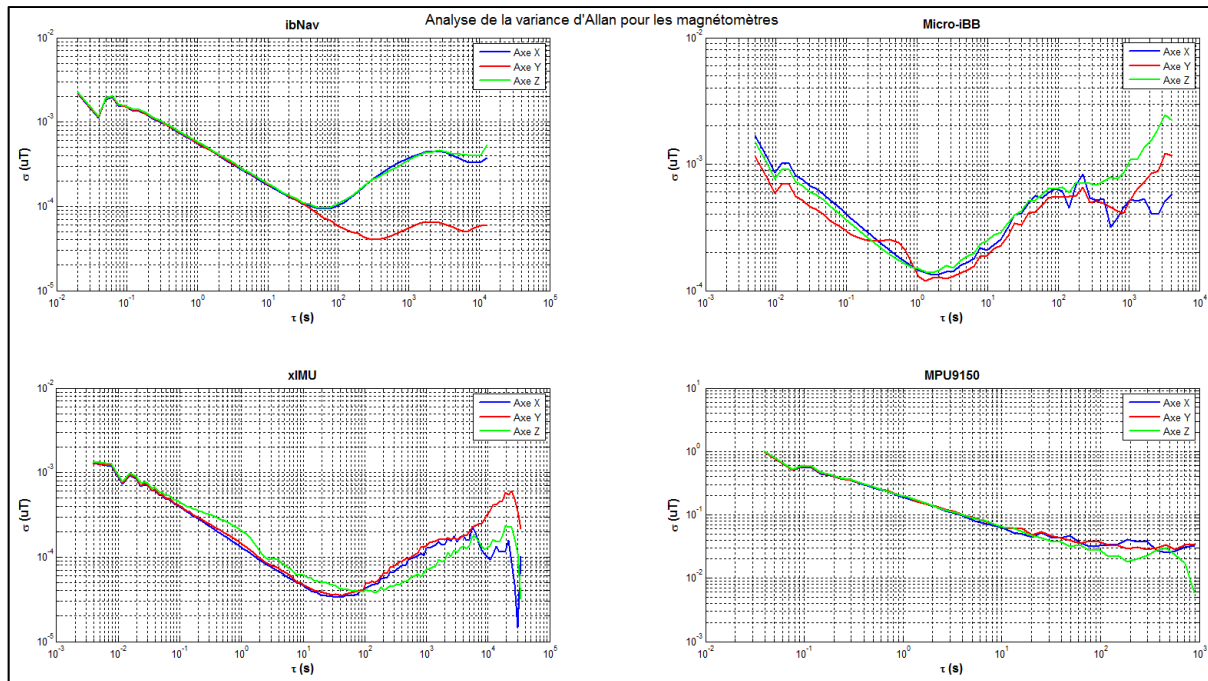


Figure 4.25 Analyse de la variance d'Allan pour les magnétomètres de 4 IMUs à faible coût

Cette analyse a été effectuée sur 6 heures d'enregistrement. Voir le Tableau 4.1 pour connaître la fréquence de rafraîchissement des données pour chaque IMUs.

On obtient alors les coefficients de bruit blanc présentés au Tableau 4.6. Il est également important de prendre en compte la sensibilité maximale de chaque capteur (Tableau 4.1) afin d'analyser les résultats. Les coefficients calculés des 14 IMUs du projet ibNav sont en ANNEXE VII.

Tableau 4.6 Coefficients de bruit blanc calculés suite à l'analyse d'Allan Variance

	Acc X	Acc Y	Acc Z	Gyr X	Gyr Y	Gyr Z	Mag X	Mag Y	Mag Z
	$g \cdot s^{1/2}$			$deg/s^{1/2}$			$\mu T \cdot s^{1/2}$		
ibNav	$2.75E^{-4}$	$3.6E^{-4}$	$5.1E^{-4}$	$8.2E^{-2}$	$4.7E^{-2}$	$4.1E^{-2}$	$4.25E^{-4}$	$4.3E^{-4}$	$4.6E^{-4}$
Micro-iBB	$1.45E^{-3}$	$1.42E^{-3}$	$1.43E^{-3}$	$3.2E^{-2}$	$2.8E^{-2}$	$3.1E^{-2}$	$1.4E^{-4}$	$9.2E^{-5}$	$1.2E^{-4}$
xIMU	$2.12E^{-4}$	$2.13E^{-4}$	$3.07E^{-4}$	$5.8E^{-3}$	$6.7E^{-3}$	$6.05E^{-3}$	$1.4E^{-4}$	$1.6E^{-4}$	$1.9E^{-4}$
MPU-9150	$1.8E^{-4}$	$1.4E^{-4}$	$1.95E^{-4}$	$4.5E^{-3}$	$4.2E^{-3}$	$5.2E^{-3}$	$1.9E^{-1}$	$1.9E^{-1}$	$2.0E^{-1}$

En comparant le bruit blanc présent sur les différents IMUs à faible coût, on remarque que ceux-ci sont du même ordre de grandeur. Le bruit blanc au niveau du Micro-iBB pour l'accéléromètre est cependant plus élevé mais ceci s'explique puisque sa sensibilité maximale est de $\pm 8 g$ au lieu de $\pm 2 g$. Dans le cadre du projet ibNav, une sensibilité maximale de $\pm 2 g$ est suffisante.

On note également que l'accéléromètre des IMUs d'ibNav a un bruit blanc typique d'un capteur à faible coût. Un gyroscope à faible coût a un bruit blanc typique situé entre 1 et 10 $\text{deg} / \text{h}^{1/2}$, alors qu'une version tactique sera de l'ordre 10^{-1} et une version dédiée à la navigation de 10^{-2} . Nous avons dans le cas d'ibNav : $3 \text{ deg} / \text{h}^{1/2}$. La remarque est similaire pour l'accéléromètre.

Ces coefficients seront par la suite utilisés au sein de l'algorithme AHRS et INS afin de corriger cette erreur aléatoire.

CHAPITRE 5

ALGORITHMES D'ATTITUDE ET DE POSITIONNEMENT

Ce chapitre est consacré aux algorithmes de navigation développés pour le projet ibNav. La première partie est l'algorithme de détermination de l'attitude permettant d'obtenir les mouvements du corps humain. La deuxième partie est l'algorithme permettant de détecter les pas de l'utilisateur afin de connaître son positionnement à l'intérieur d'un bâtiment. Cette section représente environ 15% du total de la maîtrise. La deuxième partie est encore au stade du développement et nécessite divers améliorations afin d'être plus efficace.

5.1 Développement de l'algorithme AHRS

Cette section détaille l'algorithme AHRS créé pour le projet ibNav afin d'obtenir l'attitude (roulis, tangage, lacet) de chaque IMU du projet ibNav. La Figure 5.1 représente l'algorithme développé. Il utilise les trois capteurs inertiels dont les données proviennent de la sortie de l'algorithme de calibration continue afin de déterminer l'attitude du système. La dérive de la solution est corrigée à l'aide d'un filtre de Kalman EKF associé à deux modèles de connaissances et à une méthode de soutien.

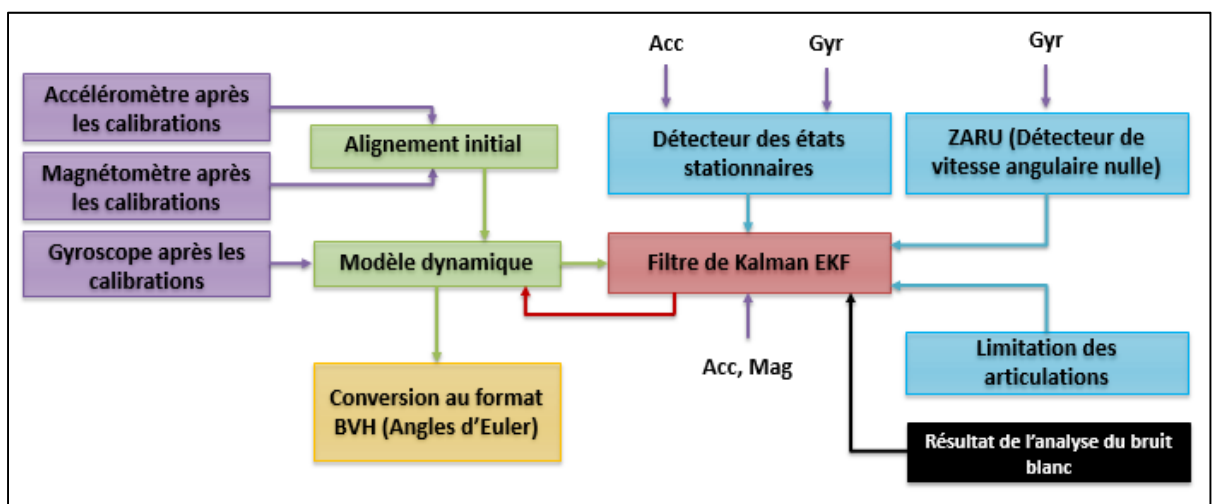


Figure 5.1 Schéma de l'algorithme AHRS du projet ibNav

Les différentes étapes couvrant cet algorithme sont les suivantes :

1. Calcul de l'alignement initial de chaque IMU à l'aide de l'accéléromètre et du magnétomètre. On obtient en sortie le quaternion unitaire initial représentant l'attitude du système au temps $t = 0$. Par la suite, une légère correction de l'alignement du boîtier sera effectuée en multipliant la solution avec une matrice de rotation déterminée grâce au robot de calibration et un scénario quasi-statique;
2. Mise à jour de l'attitude de chaque IMU à l'aide de la vitesse angulaire des gyroscopes, on prédit ainsi l'attitude du système;
3. Mise en place du modèle de connaissance permettant de détecter les états stationnaires et celui permettant la détection de l'état quasi-stationnaire à l'aide de la vitesse angulaire (ZARU). Ces deux modèles donnent des informations au filtre de Kalman EKF afin qu'il sache s'il doit prendre en compte les valeurs d'observation ou s'il est nécessaire de mettre à jour l'attitude du système;
4. Mise en place d'une méthode de soutien prenant en compte l'emplacement de l'IMU sur le corps afin de limiter l'angle minimum et maximum du roulis, tangage et lacet;
5. Correction de la dérive de la solution à l'aide du filtre de Kalman EKF utilisant comme référence les données en provenance de l'accéléromètre et du magnétomètre;
6. Conversion au format BVH, c'est-à-dire le passage du quaternion unitaire aux angles d'Euler (roulis, tangage et lacet) puis à leurs intégrations dans l'architecture BVH (voir la section 3.1.4). L'attitude des IMUs « esclaves » est donnée par rapport à un autre IMU alors que pour l'IMU « maître » son attitude est donné par rapport au repère local.

5.1.1 Alignement initial de l'IMU

L'algorithme d'alignement initial utilisé dans le cadre du projet ibNav est basé sur le livre Farrell, 2008 et permet d'en déduire l'attitude initiale à partir des données de l'accéléromètre et du magnétomètre. Puisque les données à cet instant sont bruitées par des erreurs aléatoires et par les vibrations du corps humain, il sera premièrement nécessaire d'enregistrer l'accumulation de ces données pour chaque IMU lors d'un état stationnaire au démarrage du système. Cet enregistrement idéalement doit contenir une vingtaine de données, soit environ

0.5 secondes dans le cadre du projet ibNav. Il est important de noter que chaque mg d'erreur sur l'accélération correspondra approximativement à 1 mrad d'erreur sur l'estimation de l'attitude initiale.

On a ainsi les équations suivantes :

$$\overline{\alpha^S} = -\frac{1}{n} \sum_{i=1}^n \alpha^S(\tau) \text{ et } \overline{m^S} = \frac{1}{n} \sum_{i=1}^n m^S(\tau) \quad (5.1)$$

où : $\overline{\alpha^S}, \overline{m^S}$: Moyenne des accélérations et champ magnétiques (X, Y, Z)
 n : Nombre d'échantillons (20 dans notre cas)

On peut d'ores et déjà calculer l'angle de roulis et de tangage grâce aux équations suivantes :

$$\hat{\varphi} = \text{atan2}(\overline{\alpha_y^S}, \overline{\alpha_z^S}) \quad (5.2)$$

$$\hat{\theta} = \text{atan2}\left(-\overline{\alpha_x^S}, \sqrt{\overline{\alpha_y^S}^2 + \overline{\alpha_z^S}^2}\right) \quad (5.3)$$

Pour la suite des équations, on considère que le repère du mobile est le même que le repère du capteur (suite à la calibration initiale et continue). On sait que la matrice C_L^B peut se décomposer en trois matrices distinctes, on obtient alors, pour le vecteur de champ magnétique terrestre :

$$m^b = C_L^B m^L = C_{R1}^B C_{R2}^{R1} C_L^{R2} m^L \quad (5.4)$$

$$C_{R1}^{R2} C_B^{R1} m^b = C_L^{R2} m^L \quad (5.5)$$

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \sin(\varphi) & \sin(\theta) \cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ -\sin(\theta) & \cos(\theta) \sin(\varphi) & \cos(\theta) \cos(\varphi) \end{bmatrix} m^b = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} m^L \quad (5.6)$$

Puisque l'on connaît l'angle de roulis et l'angle de tangage, on va calculer la partie gauche de l'équation que l'on va appeler m^w , soit le vecteur de champ magnétique terrestre obtenu dans le repère de navigation dérivée :

$$m^w = \begin{bmatrix} \cos(\theta) & \sin(\theta) \sin(\varphi) & \sin(\theta) \cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ -\sin(\theta) & \cos(\theta) \sin(\varphi) & \cos(\theta) \cos(\varphi) \end{bmatrix} m^b \quad (5.7)$$

Afin de résoudre l'équation (5.6) et trouver l'inconnue ψ , on va premièrement calculer les valeurs de m_x^w et de m_y^w grâce à l'équation (5.7) et les valeurs de m_x^L et de m_y^L sont obtenues en utilisant le modèle IGRF-11 présenté à la section 3.5.

On a maintenant le système d'équation suivant à résoudre :

$$\begin{cases} m_x^w = \cos(\psi) * m_x^L - \sin(\psi) * m_y^L \\ m_y^w = \cos(\psi) * m_y^L + \sin(\psi) * m_x^L \end{cases} \quad (5.8)$$

Et on obtient :

$$\begin{cases} \cos(\psi) = \frac{m_x^L \times m_x^w + m_y^L \times m_y^w}{m_x^{L^2} + m_y^{L^2}} \\ \sin(\psi) = \frac{m_x^L \times m_y^w - m_y^L \times m_x^w}{m_x^{L^2} + m_y^{L^2}} \end{cases} \quad (5.9)$$

La valeur de ψ est donc :

$$\hat{\psi} = \text{atan2}(\sin(\psi), \cos(\psi)) \quad (5.10)$$

On peut donc calculer la matrice de rotation C_B^L et le quaternion q_B^L , représentant l'alignement initial de la plateforme, en utilisant les équations présentées à la section 3.1.2.

5.1.2 Correction de la position des IMU avec les boîtiers

Dans la section précédente, on a considéré que le repère du capteur et du mobile étaient les mêmes, suite à la calibration initiale et continue. Cependant, on peut aussi avoir un mauvais alignement des axes provoqué par la mise en place des IMUs dans leurs boîtiers respectifs. Dans le cadre du mémoire, cette correction n'a pas été appliquée mais un prototype de plateforme à faible coût permettant d'effectuer cette calibration a été proposée :

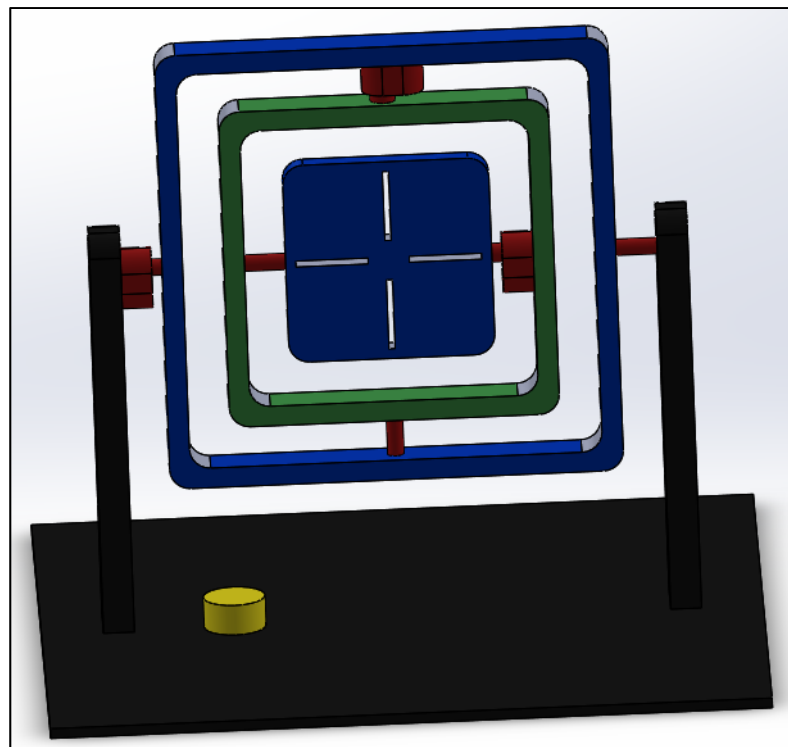


Figure 5.2 Prototype de plateforme de calibration des erreurs de mauvais alignement

Cette plateforme est composée des éléments suivants :

- un inclinomètre de précision (représenté en jaune), permettant de connaître l'angle d'inclinaison de la plateforme par rapport au champ de gravité terrestre;
- trois servomoteurs dotés de freins électromagnétiques contrôlés par un microcontrôleur permettant d'exciter chacun des axes de l'accéléromètre des IMUs suivant la méthode multi-positions;
- un support permettant de fixer et de centrer le boîtier.

Le scénario de calibration se fera de manière automatique et les trames seront transmises en temps réel sur une application PC par le protocole RS232. Grâce aux valeurs recueillis en mode statique pour chaque position, on pourra créer la matrice de rotation C_P^B et ainsi corriger cette erreur de mauvais alignement des axes.

5.1.3 Modèle dynamique de mise à jour de l'attitude

La mise à jour de l'attitude du système (sans correction d'erreurs) est la méthode basée sur l'utilisation des gyroscopes. Puisque ce capteur de type MEMS à faible coût n'est pas assez précis, il est impossible de l'utiliser tel quel, de ce fait, l'attitude sera par la suite corrigée à l'aide d'un filtre de Kalman FEKF (sans inversion de matrice) utilisant comme observation les mesures de l'accéléromètre et du magnétomètre. L'algorithme utilise les équations (3.20) à (3.23). On obtient alors l'équation :

$$q_B^L(t) = q_B^L(t-1) + \Delta t \left(\frac{\dot{q}_B^L(t) + \dot{q}_B^L(t-\Delta t)}{2} \right) \quad (5.11)$$

Il faudra par la suite réorganiser le vecteur $q_B^L(t)$ sous la forme d'une matrice 4x4 pour la multiplication avec le quaternion d'erreur de l'attitude $q^e(t)$, en sortie du filtre de Kalman.

$$q_{B_{corrigé}}^L = \begin{bmatrix} q_{B_w}^L & -q_{B_x}^L & -q_{B_y}^L & -q_{B_z}^L \\ q_{B_x}^L & q_{B_w}^L & -q_{B_z}^L & q_{B_y}^L \\ q_{B_y}^L & q_{B_z}^L & q_{B_w}^L & -q_{B_x}^L \\ q_{B_z}^L & -q_{B_y}^L & q_{B_x}^L & q_{B_w}^L \end{bmatrix} \times \begin{bmatrix} \delta q_w \\ \delta q_x \\ \delta q_y \\ \delta q_z \end{bmatrix} \quad (5.12)$$

5.1.4 Ajout des modèles de connaissances

Deux modèles de connaissances vont permettre un contrôle externe sur le filtre de Kalman :

1. L'algorithme de détection des états stationnaires permettant d'indiquer au filtre de Kalman que les valeurs d'observation peuvent être prises en compte afin de corriger l'attitude du système;

2. L'algorithme de détection de l'état quasi-stationnaire de la vitesse angulaire (ZARU), indiquant au filtre de Kalman que la vitesse angulaire mesurée correspond à l'erreur.

L'algorithme de détection des états stationnaires étant déjà présenté à la section 4.3.2, il ne figurera pas dans cette section. Les constantes pour les conditions C_1, C_2, C_3 et C_4 sont cependant légèrement différentes dûes à la calibration et au bruit créé par le corps humain lors d'un état stationnaire.

Le modèle ZARU est un algorithme très simple, lorsqu'un état stationnaire est détecté, on suppose que la vitesse angulaire que l'on mesure correspond au bruit provoqué par le corps humain, on prend en compte alors cette erreur dans le filtre de Kalman. On a donc, lorsque l'on est en état stationnaire, l'erreur $\delta\omega_{IB}^B = \omega_{IB}^B$ s'ajoutant au vecteur d'observation.

5.1.5 Modélisation du filtre EKF pour corriger la dérive de la solution

Ce que nous voulons corriger est l'erreur sur le quaternion, dans ce cas spécifique, nous avons le modèle de propagation de l'erreur du quaternion, soit :

$$q_B^L = \hat{q}_B^L \otimes \delta q \quad (5.13)$$

où : \hat{q}_B^L : Estimé de la valeur du quaternion
 δq : Erreur du quaternion

Avec finalement, l'équation de propagation de l'erreur du quaternion :

$$\dot{\delta q} = -\omega_{IB}^B \times \delta q - \frac{1}{2}\eta_\omega \quad (5.14)$$

Dans ce cas, nous n'utiliserons pas directement le vecteur d'état mais le vecteur d'erreur d'état afin de ne pas être obligé de linéariser la matrice de transition F et la matrice

d'observation H . A partir de l'équation précédente et en ajoutant l'erreur sur la vitesse angulaire, on obtient donc le vecteur d'erreur d'état suivant :

$$\delta x = [\delta q_{B_x}^L \quad \delta q_{B_y}^L \quad \delta q_{B_z}^L \quad \delta \omega_{IB_x}^B \quad \delta \omega_{IB_y}^B \quad \delta \omega_{IB_z}^B] \quad (5.15)$$

Note : Afin de simplifier l'écriture des équations et éviter les répétitions, la calibration continue (section 4.3.5) n'est pas incluse dans la modélisation de ce filtre EKF. Elle est cependant incluse dans le modèle AHRS final.

La correction de l'erreur de propagation de l'attitude va donc être effectuée en utilisant un filtre de Kalman étendu rapide (FEKF). Puisque nous savons que le vecteur d'erreur d'état change au cours du temps, suivant l'attitude de chaque IMU, la matrice de transition F doit être calculée (contrairement aux méthodes de calibration) et par conséquent, la matrice de couplage dynamique du bruit G (on considère cependant ce bruit comme constant afin de simplifier les équations). Le système d'équations dans le domaine discret est donc le suivant :

$$\delta x_{K+1} = \phi_K \delta x_K + G \omega_K \quad (5.16)$$

$$\delta z_K = H_K \delta x_K + v_K \quad (5.17)$$

Les étapes de réalisation sont les suivantes :

L'initialisation :

Nous estimons qu'à $t = 0$, le vecteur d'erreur d'état $\delta \hat{x}_0$ ne contient aucune erreur, le vecteur prend donc les valeurs suivantes :

$$\delta \hat{x}_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \quad (5.18)$$

Afin que le modèle converge, la matrice de covariance P_0 va prendre des valeurs correspondant à l'incertitude posée sur \hat{x}_0 , soit :

$$P_0 = \text{diag}([\sigma_{\delta q_x}^2 \quad \sigma_{\delta q_y}^2 \quad \sigma_{\delta q_z}^2 \quad \sigma_{q\omega_x}^2 \quad \sigma_{\delta\omega_y}^2 \quad \sigma_{\delta\omega_z}^2]) \quad (5.19)$$

On va également construire les matrices constantes à l'initialisation du système, soit la matrice de covariance des bruits d'états Q et la matrice de covariance des bruits d'observation R (Les données proviennent des coefficients du bruit blanc calculés préalablement à l'aide de la variance d'Allan excepté pour les valeurs de variance pour la méthode ZARU, où on choisira une valeur typique de 0.1 rad/s), soit :

$$Q = \begin{bmatrix} \sigma_{\omega_x}^2 & 0 & 0 \\ 0 & \sigma_{\omega_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega_z}^2 \end{bmatrix} \quad (5.20)$$

$$R = \text{diag}([\sigma_{\alpha_x}^2 \quad \sigma_{\alpha_y}^2 \quad \sigma_{\alpha_z}^2 \quad \sigma_{m_x}^2 \quad \sigma_{m_y}^2 \quad \sigma_{m_z}^2 \quad \sigma_{ZARU_x}^2 \quad \sigma_{ZARU_y}^2 \quad \sigma_{ZARU_z}^2]) \quad (5.21)$$

On peut également construire la matrice de couplage dynamique du bruit G qui est une constante d'après l'équation (5.14), soit :

$$G = \begin{bmatrix} -\frac{1}{2}I \end{bmatrix} \quad (5.22)$$

A partir de cet instant, les variables devront être recalculés à chaque itération. On va premièrement calculer la matrice de transition F et d'après l'équation (5.16), on obtient :

$$F = \begin{bmatrix} -(\omega_{IB}^B \times) & 0 \\ 0 & 0 \end{bmatrix} \quad (5.23)$$

Puisque que l'on travaille dans le domaine discret, il va être nécessaire de discrétiser la matrice de transition F, on va donc obtenir :

$$\phi_K = e^{F\Delta t} \approx I + F\Delta t + \frac{1}{2!}F^2\Delta t^2 \quad (5.24)$$

On va maintenant calculer la matrice de bruit intégré Qd_K à l'aide de la formule (3.59).

Dans cette étape d'initialisation, il nous reste à construire la matrice d'observation H_K , celle-ci va s'écrire :

$$H_K = \begin{bmatrix} -2(\alpha^L \times) & 0 \\ -2(m^L \times) & 0 \\ 0 & I \end{bmatrix} \quad (5.25)$$

La prédiction :

On va faire un estimé du vecteur d'erreur d'état et de la matrice de covariance à partir de l'estimé mis à jour \hat{x}_{K-1}^+ . Voir les équations (3.62) et (3.63).

La mise à jour des mesures :

À toutes les 5 trames et lorsque l'on a détecté un état stationnaire, on va calculer la mesure résiduelle, soit :

$$\delta z_K = \begin{bmatrix} [0 \quad 0 \quad G_E] - C_B^L \alpha^B \\ m_0^L - C_B^L m^B \\ \omega_{ZARU} \end{bmatrix} \quad (5.26)$$

On calcule alors le gain de Kalman, \hat{x}_K^+ et sa matrice de covariance P_K^+ à l'aide des équations (3.64) à (3.68). Finalement, on met à jour l'erreur sur le quaternion δq .

Note : Puisque l'on utilise un filtre FEKF, le calcul du gain de Kalman s'effectue sans inversion de matrice (voir équation (6.1)).

5.1.6 Analyse des performances et discussion

Afin de démontrer le besoin de corriger la dérive de l'attitude au cours du temps, le premier graphique représentera la dérive de l'attitude de l'IMU « maître » du projet ibNav, après la calibration initiale et continue. L'enregistrement est effectué lors d'un scénario statique. A

cet instant le filtre de Kalman n'est pas encore utilisé, de ce fait, le calcul est réalisé uniquement avec les données du gyroscope. Voir la Figure 5.3.

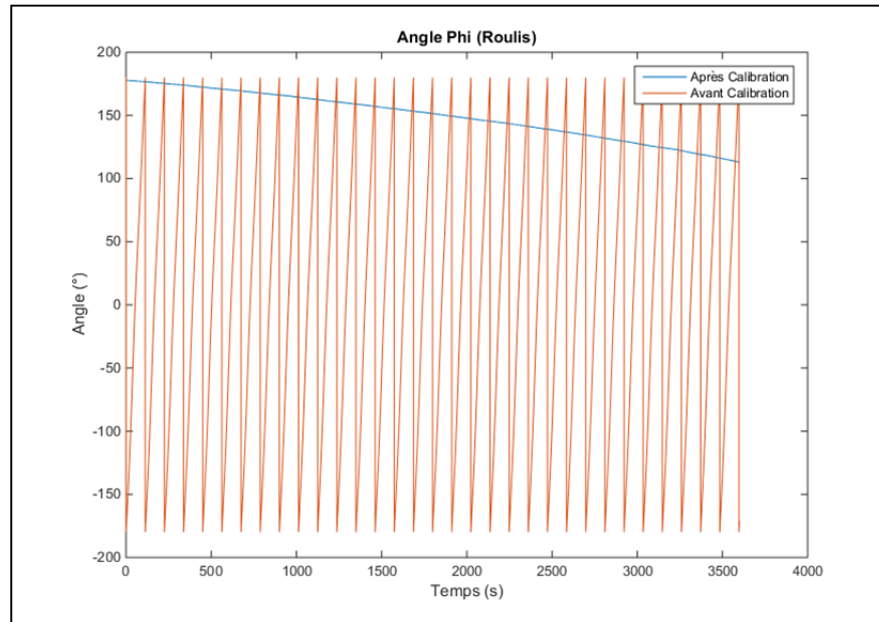


Figure 5.3 Dérive de l'angle Phi (Roulis) avant et après la calibration

Dérive calculée :

- avant la calibration initiale: $11\ 531\ ^\circ/\text{h}$;
- après la calibration initiale (correction du biais) : $65\ ^\circ/\text{h}$.

Pour ce même enregistrement, nous allons ajouter la méthode de correction de la dérive présentée à la section 5.1.5, on obtient alors la Figure 5.4. On remarque que pour un enregistrement statique, lorsque le capteur est en état stationnaire, l'absence de dérive de la solution. Ce comportement s'explique puisqu'en tout temps, la vitesse angulaire est corrigée à l'aide de l'accéléromètre et du magnétomètre.

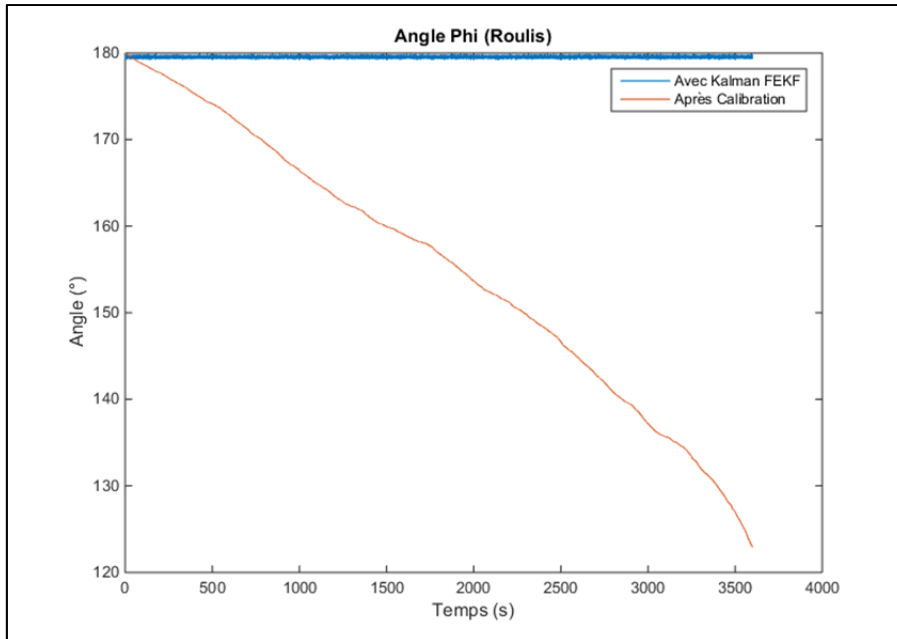


Figure 5.4 Dérive de l'angle Phi (Roulis) avant et après la correction de la dérive

Le prochain enregistrement correspond à un scénario quasi-statique où chacun des axes est excité. On obtient la Figure 5.5 représentant l'estimation des angles d'Euler et la Figure 5.6 représentant l'erreur sur l'estimation. Puisque nous n'avons pas à notre disposition une plateforme inertielle de référence, la référence correspond aux angles du robot de calibration, même si celui-ci n'est pas idéal. Les figures représentent trois enregistrements avec les méthodologies suivantes : FEKF, FEKF + CC (Calibration Continue), FEKF + ZARU + CC pour l'angle d'Euler Theta (tangage). L'angle de roulis et de lacet n'est pas représenté mais ils sont également en mouvement quasi-statique. On remarque également que l'erreur augmente rapidement lorsque l'on est en dynamique, ceci s'explique par l'absence de correction dans cet état.

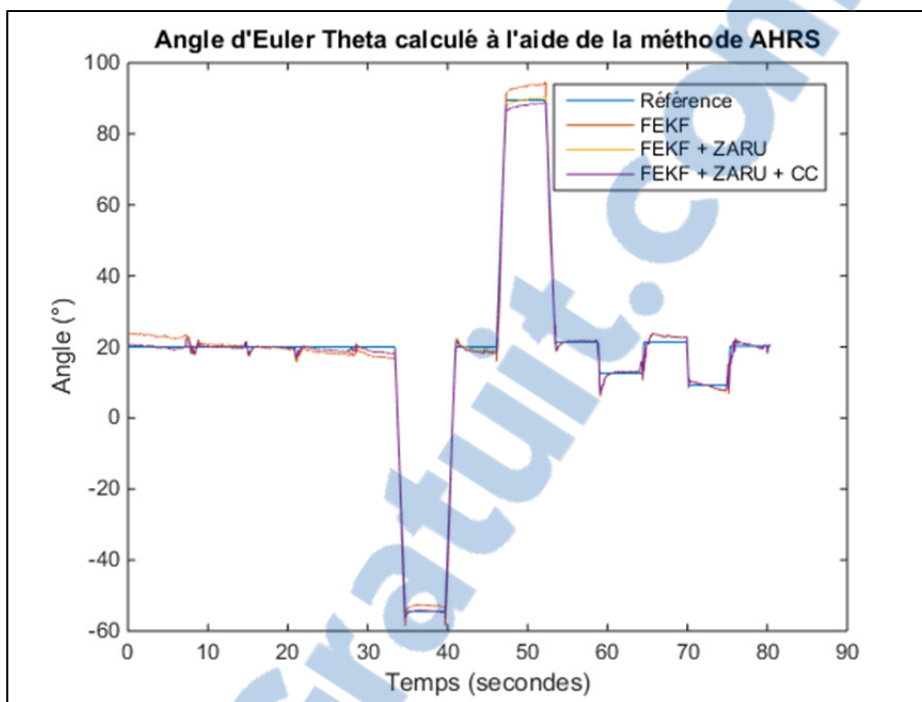


Figure 5.5 Estimation de l'angle d'Euler Theta calculés par la méthode AHRS

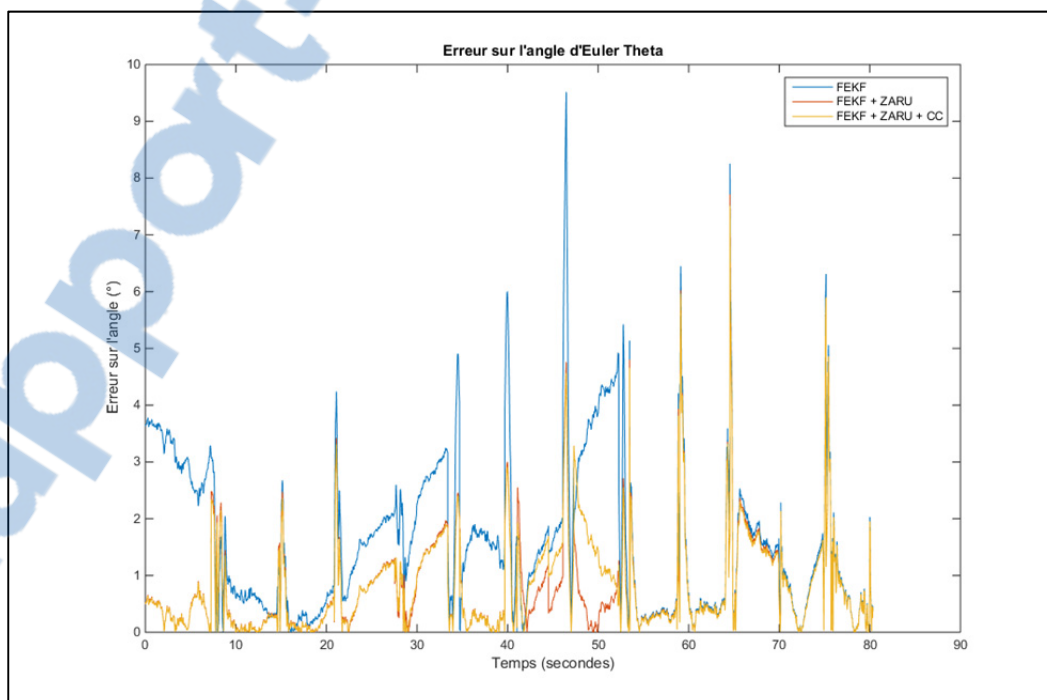


Figure 5.6 Erreurs sur l'angle d'Euler Theta

Note : La référence est celle des angles des servomoteurs, de ce fait; ils ne correspondent pas exactement à l'attitude des IMUs, cette contrainte a été prise en compte afin de calculer les erreurs du Tableau 5.1.

Par analyse visuelle, on remarque que la calibration continue améliore légèrement la solution et que la méthode de correction de la dérive utilisant le modèle de connaissance ZARU permet d'obtenir une erreur plus faible sur chacun des angles. On obtient donc les erreurs suivantes :

Tableau 5.1 Erreur RMS statique et dynamique des angles d'Euler (en degrés) pour un angle de roulis, tangage et lacet de $\pm 90^\circ$

Angles d'Euler ($^\circ$)	FEKF	FEKF + ZARU	FEKF + CC + ZARU
RMS φ statique	0.8456	0.3633	0.3584
RMS φ dynamique	1.3140	0.9847	0.9755
RMS θ statique	1.6786	0.6108	0.5983
RMS θ dynamique	2.8272	1.4132	1.3724
RMS ψ statique	3.4115	1.0739	1.0242
RMS ψ dynamique	3.8776	1.9494	1.9101

L'inconvénient principal est que cette méthode permettant la correction de la dérive perd en performances lorsque l'utilisateur réalise de fortes dynamiques et/ou que les états stationnaires du système sont peu nombreux.

5.2 Développement de l'algorithme INS

Cette section est consacrée à l'algorithme INS du projet ibNav permettant de détecter les pas de l'utilisateur, son positionnement et sa vitesse. Seulement 3 centrales inertielles sont utilisées avec cet algorithme (les deux sur le pied et celui sur le torse, voir section 6.3). La Figure 5.7 représente l'algorithme développé. Cet algorithme est toujours en développement à ce jour à cause de la transition technologique. Il utilise l'attitude calculée précédemment ainsi que les données de l'accéléromètre afin de calculer la position et la vitesse. La dérive de

la solution est corrigée à l'aide d'un filtre de Kalman EKF associé à 3 modèles de connaissances et à une méthode de soutien.

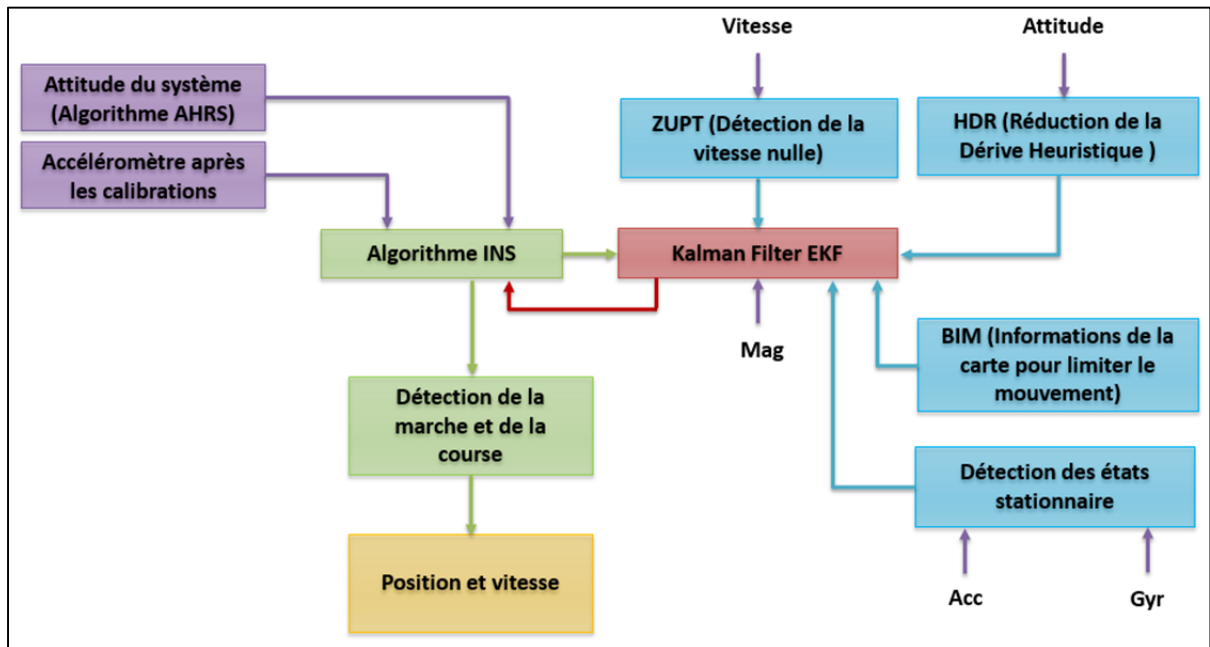


Figure 5.7 Schéma de l'algorithme INS du projet ibNav

Les différentes étapes couvrant cet algorithme sont les suivantes :

1. Calcul et mise à jour de la vitesse et de la position des 3 IMUs. On prédit la vitesse et la position de chaque IMU;
2. Mise en place du modèle de connaissance permettant de détecter les états stationnaires afin d'indiquer au filtre de Kalman que les valeurs d'observations peuvent être prises en compte;
3. Mise en place de deux modèles de connaissance supplémentaire permettant la détection de la vitesse nulle (ZUPT) et la réduction de la dérive heuristique (HDR) permettant de détecter quand une personne marche linéairement. Ces deux modèles permettent donc de corriger d'avantage la vitesse et la position à l'aide du filtre de Kalman EKF;
4. Mise en place d'une méthode de soutien considérant la connaissance de la carte où l'utilisateur se déplace, on a donc la connaissance des emplacements des obstacles;
5. Correction de la dérive de la solution à l'aide du filtre de Kalman EKF;

6. Par rapport aux informations sur la position et la vitesse des 3 IMUs, on pourra entre deux pas déduire si la personne est en train de marcher ou de courir et avoir également la vitesse de l'utilisateur;
7. Les données de positionnement en coordonnées cartésiennes ECEF sont converties en géodésique ECEF pour pouvoir s'intégrer au format KML (voir la section 3.1.5). L'annexe IV indique comment effectuer cette conversion.

5.2.1 Ajout de l'intégration de l'accélération et de la vitesse

Sans correction d'erreurs, l'intégration de l'accélération et de la vitesse se résume à la méthode présentée à la section 3.1.3. L'accéléromètre étant à faible coût, la dérive de la solution avec ce type de méthodes est élevée, on a calculé en moyenne 5.3 mg/s de déviation, soit environ 6 mètres en 1 minute. Il ne sera donc pas possible d'utiliser les formules de base directement, il faudra corriger la dérive à l'aide, dans notre cas, d'un filtre de Kalman FEKF utilisant comme observation les mesures du magnétomètre, les modèles de connaissance et la méthode de soutien. La vitesse est donc calculée à l'aide de la formule suivante :

$$\dot{V}^N = C_L^N \left(q_B^L \otimes a_{Sf}^B \otimes q_B^{L^{-1}} \right) - g^N \quad (5.27)$$

L'attitude q_B^L sera donc calculée à l'aide du modèle AHRS afin d'obtenir le vecteur d'accélération dans le repère de navigation, puis on soustraira la gravité g^N terrestre grâce au modèle présenté à la section 3.5. Il suffira par la suite d'effectuer l'intégration Runge-Kutta du second ordre pour obtenir la vitesse et la position dans le repère de navigation.

5.2.2 Ajout des modèles de connaissances

Le premier modèle de connaissance, l'algorithme de détection des états stationnaires, au même titre que pour l'algorithme AHRS, permet d'indiquer au filtre de Kalman que les valeurs d'observations, les autres modèles de connaissance et la méthode de soutien peuvent être prisent en compte afin de corriger la solution.

L'algorithme de détection de la vitesse nulle (ZUPT), indique au filtre de Kalman que la vitesse mesurée correspond à l'erreur. On suppose donc ici que le bruit de l'accéléromètre combiné à son intégration correspond à l'erreur de la vitesse. Donc, lorsque l'on est en état stationnaire, l'erreur $\delta V_N = V_N$ vient s'ajouter au vecteur d'observation.

L'algorithme de réduction de la dérive heuristique permet de détecter que la personne marche linéairement et de corriger ainsi la dérive de l'attitude. Nous utiliserons la formule de l'article Jiménez, 2010 analysant l'angle de lacet (pour les 2 IMUs sur le pied) et l'angle de roulis pour l'IMU sur le torse afin de détecter les changements d'orientations. Si l'angle en question entre deux pas est faible, c'est à dire inférieure à 10° , on considère que la personne n'a pas changée de direction et que l'angle trouvé correspond à l'erreur (équation (5.28)).

$$\delta\psi_K = \psi_K - \frac{\psi_{K-1} + \psi_{K-2}}{2} \text{ lorsque } \|\psi_K\| < 10^\circ \quad (5.28)$$

Dans cette méthode, le magnétomètre aura le même rôle que l'algorithme HDR, c'est-à-dire corriger l'angle de lacet.

5.2.3 Modélisation du filtre EKF pour corriger la dérive de la solution

Dans ce cas, l'observation provient des données du magnétomètre, des modèles de connaissance et de la méthode de soutien afin de corriger la vitesse et la position du système. Nous allons corriger l'erreur sur la vitesse et la position, les modèles de propagation de l'erreur sont les suivantes :

$$\begin{cases} V^N(t) = V^N(t - \Delta t) + \Delta t \left(\frac{\dot{V}^N(t) + \dot{V}^N(t - \Delta t)}{2} \right) + \delta V^N(t) \\ P^N(t) = P^N(t - \Delta t) + \Delta t \left(\frac{\dot{P}^N(t) + \dot{P}^N(t - \Delta t)}{2} \right) + \delta P^N(t) \end{cases} \quad (5.29)$$

Pour notre modèle INS, nous allons utiliser le vecteur d'erreur d'état suivant (où chaque composante sont des vecteurs) :

$$\delta x = [\delta \underline{P}^N \quad \delta \underline{V}^N \quad \delta \underline{\alpha}^b \quad \delta \underline{q}_B^L] \quad (5.30)$$

Tout comme l'algorithme AHRS, nous allons utiliser le filtre de Kalman étendu rapide (FEKF), le système d'équation dans le domaine discret va être le suivant :

$$\delta x_{K+1} = \phi_K \delta x_K + G \omega_K \quad (5.31)$$

$$\delta z_K = H_K \delta x_K + v_K \quad (5.32)$$

Les étapes de réalisation sont les suivantes :

L'initialisation :

Nous estimons qu'à $t = 0$, le vecteur d'erreur d'état $\delta \hat{x}_0$ ne contient aucune erreur, le vecteur prend donc les valeurs suivantes :

$$\delta \hat{x}_0 = [\underline{0} \quad \underline{0} \quad \underline{0} \quad \underline{0}] \quad (5.33)$$

Afin que le modèle converge, la matrice de covariance P_0 va prendre des valeurs correspondant à l'incertitude posée sur \hat{x}_0 , soit :

$$P_0 = \text{diag} \left(\left[\sigma_{\delta \underline{P}^N}^2 \quad \sigma_{\delta \underline{V}^N}^2 \quad \sigma_{\delta \underline{\alpha}^b}^2 \quad \sigma_{\delta \underline{q}_B^L}^2 \right] \right) \quad (5.34)$$

On va construire les matrices constantes à l'initialisation du système, soit la matrice de covariance des bruits d'états Q et la matrice de covariance des bruits d'observation R , soit :

$$Q = \begin{bmatrix} \sigma_{\alpha_x}^2 & 0 & 0 \\ 0 & \sigma_{\alpha_y}^2 & 0 \\ 0 & 0 & \sigma_{\alpha_z}^2 \end{bmatrix} \quad R = \text{diag} \left(\left[\sigma_{\underline{Z}_{UPT}}^2 \quad \sigma_{H_{DR}}^2 \quad \sigma_{C_P}^2 \right] \right) \quad (5.35)$$

On va maintenant déterminer la matrice de couplage dynamique du bruit G et la matrice d'observation H , soit :

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.36)$$

Il nous reste maintenant à calculer la matrice de transition F , soit :

$$F = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & C_B^N & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.37)$$

La prédiction :

On va faire un estimé du vecteur d'erreur d'états et de la matrice de covariance à partir de l'estimé mis à jour. Voir les équations 3.62 et 3.63.

La mise à jour des mesures :

À toutes les 5 trames et lorsque l'on a détecté un état stationnaire, on va calculer la mesure résiduelle, soit :

$$\delta Z_K = \begin{bmatrix} v_{ZUPT} \\ \psi_{HDR,CP} \end{bmatrix} \quad (5.38)$$

Note : Cette méthode de correction de la dérive peut venir s'intégrer au filtre EKF modélisé à la section 5.1.5.

5.2.4 Discussion

L'algorithme INS pour le projet ibNav présenté dans cette section est une ébauche développé par rapport à la revue de la littérature (CHAPITRE 3). Il n'est cependant pas assez performant pour être utilisé avec des IMUs bas de gamme (<150 \$), il reste plutôt adapté aux IMUs de moyenne gamme (1000-3000 \$). La dérive dans notre cas est trop importante dû aux accéléromètres et à cause de l'absence de sources externes permettant la correction. Les méthodes de filtrage ne sont pas adaptées pour ce problème, une méthode d'apprentissage machine utilisant la méthodologie PDR devra être envisagée pour la suite du projet, tout en considérant qu'il s'agit d'un corps humain en déplacement.

5.3 Développement de méthodes de soutien

Les méthodes de soutien permettent de prendre en considération l'environnement où les IMUs sont utilisés afin de limiter les angles, les positions ainsi que les vitesses possibles. On va ainsi limiter les erreurs. L'usage de ces méthodes nécessite l'ajout des vecteurs associés à l'intérieur des filtres de Kalman de l'algorithme AHRS et INS.

Limitation des angles des articulations du corps humain : Au niveau du rendu et du fichier BVH, il y a une hiérarchie entre les différentes articulations du corps humain. L'attitude pour l'IMU « maître » correspond à la matrice de rotation DCM $C_{B_1}^L$ mais l'attitude des autres IMUs est calculée en fonction de l'attitude de l'IMU précédent (voir Tableau 5.2).

Tableau 5.2 Calcul de l'attitude - Hiérarchie entre les IMUs

IMU du projet ibNav	Attitude de l'IMU
IMU 1 - Torse	$C_{B_1}^L$
IMU 4 - Bras droit	$C_{B_1}^{B_2} = C_L^{B_2} \cdot C_{B_1}^L$
IMU 3 - Avant-bras droit	$C_{B_2}^{B_3} = C_L^{B_3} \cdot C_{B_2}^L$
IMU 2 - Main droite	$C_{B_3}^{B_4} = C_L^{B_4} \cdot C_{B_3}^L$
IMU 8 - Tête	$C_{B_1}^{B_8} = C_L^{B_8} \cdot C_{B_1}^L$

Une liste des limitations des angles des articulations du corps humain est présente en ANNEXE VIII. On remarque qu'en plus de limiter, certains angles d'Euler ne sont pas utilisés.

Cette méthode de soutien a besoin de connaître la matrice C_B^L de chaque IMU afin de calculer les matrices C_B^B , de vérifier si l'angle est autorisé et dans le cas contraire de renvoyer l'erreur δC_B^L au filtre de Kalman de l'algorithme AHRS.

Limitation des déplacements possible sur la carte (BIM) : Cette méthode est en cours de développement, elle est directement utilisée au niveau de l'iPad qui contient les cartes 2D des laboratoires. L'idée est la suivante :

La première étape consiste à convertir les pixels de la carte en coordonnées. On va par la suite créer des trajectoires possibles en considérant les obstacles sur la carte.

Lorsque l'utilisateur souhaitera utiliser la navigation à l'intérieur des bâtiments, il devra entrer sa position initiale. Ensuite, l'iPad recevra le positionnement calculé des IMUs et enverra l'erreur de dérive avec les trajectoires existantes au filtre de Kalman de l'algorithme INS.



Figure 5.8 Interface du logiciel ibNav sur iPad représentant une carte 2D du laboratoire

CHAPITRE 6

RÉALISATION DU PROTOTYPE IBNAV

Ce chapitre présente les développements réalisés au niveau du micro-logiciel des IMUs, de l'architecture matérielle et du système de communication. Il correspond approximativement à 35% du temps de travail total effectué durant la maîtrise (incluant le temps d'autoformation, de réalisation et de test). Ce chapitre relate uniquement les réalisations effectuées durant la phase 3 du projet (section 1.4.3). L'intégration des différentes fonctionnalités requises pour respecter les spécifications du projet ibNav s'est montré une tâche complexe demandant de nombreuses connaissances en programmation embarqué et électronique. La Figure 6.1 présente l'architecture matérielle dans sa globalité.

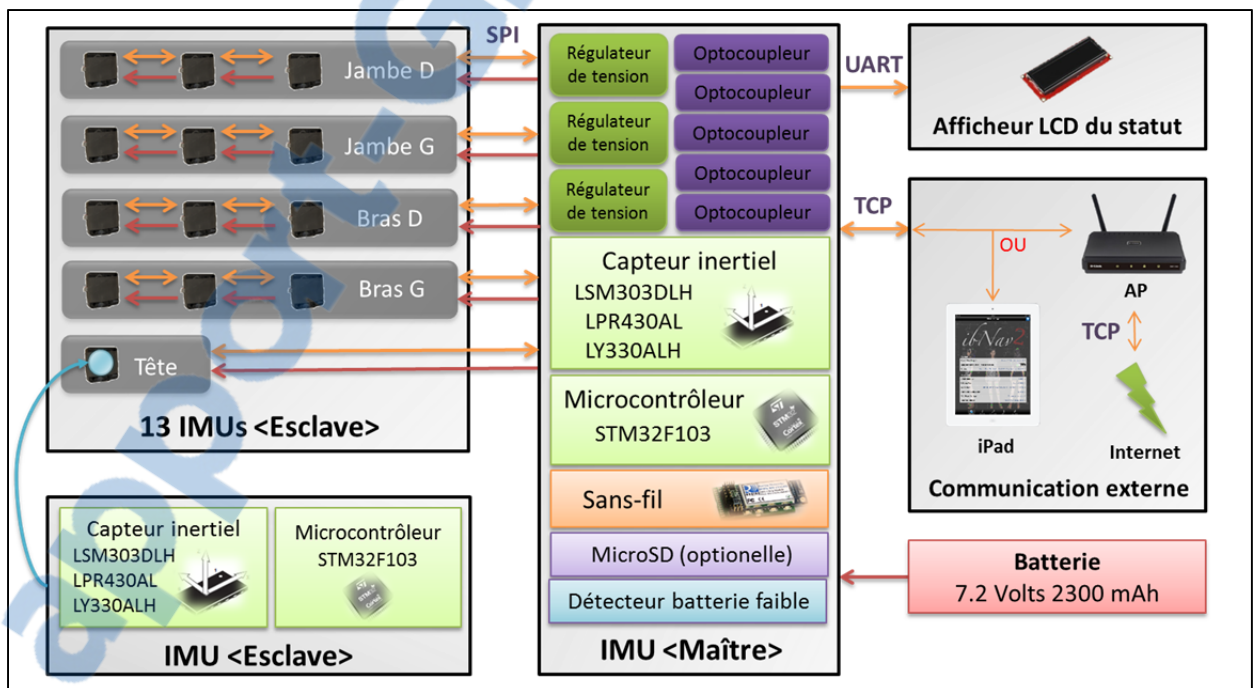


Figure 6.1 Architecture globale du système ibNav

6.1 Conception du micro-logiciel des IMUs

Pour rappel, le microcontrôleur intégré des IMUs du projet ibNav (STEVAl-MKI062V2) est de type ARM 32 bits Cortex M3 sous le numéro de série : STM32F103RE. Ce sont des circuits de prototypage. De base, on a un code source développé sous IAR Embedded Workbench utilisant FreeRTOS et permettant de communiquer avec le logiciel de STMicroelectronics. Lorsqu'on analyse des I/O disponibles sur la plateforme, on remarque que deux protocoles de communications sont disponibles afin d'ajouter des périphériques externes, soit : l'USART2 et le SPI1. L'USART3 peut être utilisé à condition de déconnecter le capteur de température externe et le capteur de pression. L'UART4 peut être utilisé à condition de ne pas utiliser le SDIO (pour la carte SD).

6.1.1 Objectifs de la conception

De manière générale, l'objectif est de récupérer les données brutes des capteurs ainsi que les données calculées de chaque IMU (mouvements et déplacement du corps humain) et de les envoyer à l'iPad en temps réel. Pour ce faire, on a un IMU dit maître, s'occupant de récupérer les données de chaque IMU via le protocole SPI. Par la suite, l'IMU maître envoie toutes les données recueillies au module sans-fil qui établit une communication TCP vers l'iPad. Plus en détail, il faut répondre aux besoins suivants :

- l'objectif est premièrement d'avoir un micro-logiciel permettant de récupérer les données brutes des capteurs inertiels. En effet, le projet étant destiné à la recherche et au développement, il est de ce fait très utile de pouvoir récupérer les données brutes en provenance des capteurs afin d'effectuer du post-traitement sur ces données. Ainsi, il sera possible de tester plusieurs algorithmes inertiels, de les comparer, de les analyser et de les améliorer afin d'obtenir des résultats tendant à répondre à nos attentes. C'est également en récupérant ces données que le système deviendra un moyen d'étude des capteurs à faible coût;

- deuxièmement, il faut intégrer les algorithmes inertiels à l'intérieur de chaque IMU afin d'obtenir les mouvements du corps humain et son déplacement en temps réel. Il est également utile de pouvoir tester plusieurs configurations. A ce stade, il faut bien prendre en compte les limites matérielles lors de la conception du micro-logiciel. En effet, il ne sera pas possible d'envoyer les données brutes et les données calculées en même temps, la communication UART reliée au module sans-fil étant trop lente avec un maximum de 460800 bits/s. Il n'est également pas possible d'implémenter les algorithmes inertiels uniquement dans l'IMU maître, la charge du processeur serait trop importante. De plus, pour la même raison, la récupération des données pourra se faire au maximum à 50 Hz;
- troisièmement, il faut créer une communication SPI afin de communiquer avec chaque IMU indépendamment et de récupérer leurs données. Cette communication pour respecter les spécifications du projet doit fonctionner avec n'importe quel nombre d'IMU connecté (il faut donc arriver à détecter le nombre d'IMU connecté, rapidement et avant que la communication SPI crée sa trame). Elle doit également fonctionner en PnP, c'est-à-dire qu'on pourrait débrancher ou brancher un IMU durant le fonctionnement du système sans avoir besoin de le redémarrer;
- afin de vérifier le bon fonctionnement du système sans avoir besoin de l'iPad, il s'est avéré nécessaire d'ajouter un afficheur LCD, donnant plusieurs informations utiles sur son état de fonctionnement;
- la carte SD reste optionnelle mais elle peut permettre d'utiliser le système sans que l'iPad soit démarré. Ce dernier pourra récupérer les données de la carte SD plus tard. (A noter que cette option étant apparue plus tard dans les spécifications du projet, elle est toujours en développement et n'est pas encore implémentée lors de l'écriture de ce présent document du fait des modifications matérielles à effectuer).

Finalement pour répondre aux spécifications, nous devons créer notre propre micro-logiciel et ne pas utiliser celui fourni de base par STMicroelectronics. Ce dernier utilisant FreeRTOS et donc prenant une place mémoire supplémentaire, une gestion des ressources supplémentaires ainsi qu'une hausse de la consommation du système. De plus, par sa complexité, le micro-logiciel présente plusieurs erreurs connues qui pourraient venir

perturber le bon fonctionnement de notre système (dans le cas d'une exception interrompant le processus).

6.1.2 Architecture du micro-logiciel des IMUs

La Figure 6.2 représente l'architecture du projet ibNav au niveau du micro-logiciel. Cette section va détailler les fonctions développées ainsi que la structure du programme.

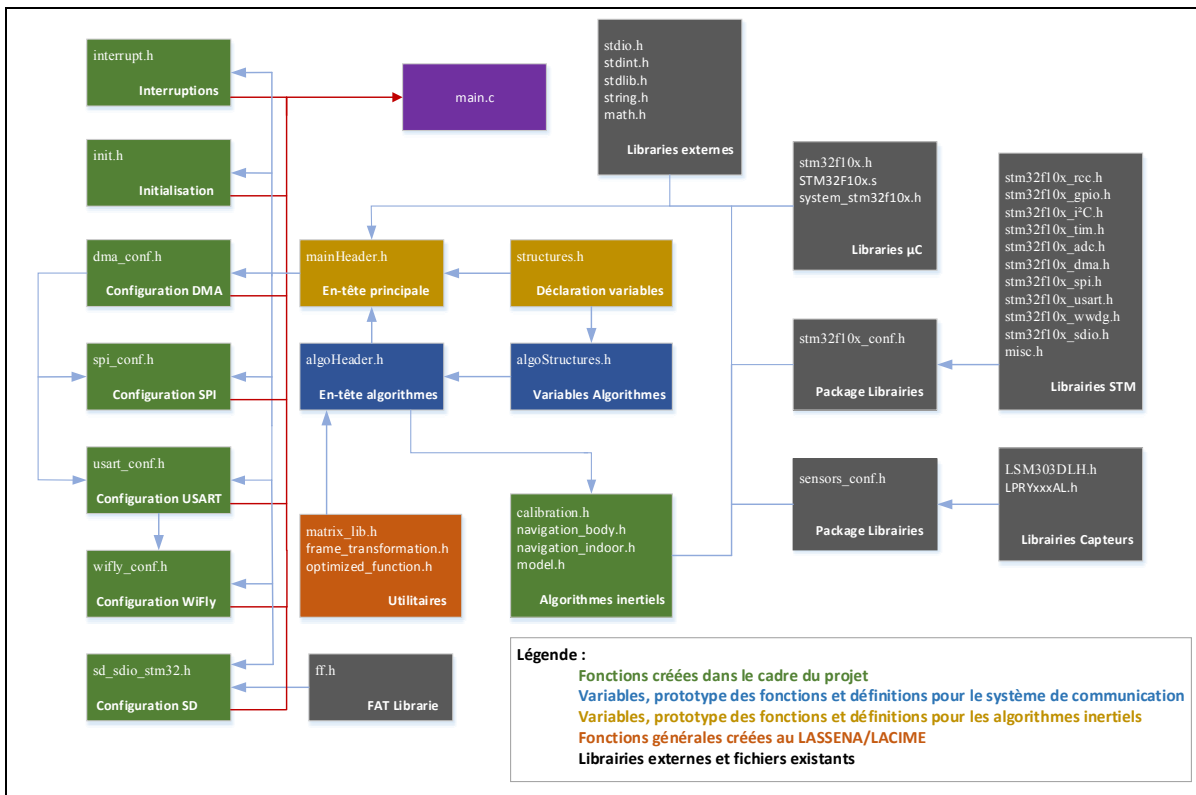


Figure 6.2 Architecture du micro logiciel du projet ibNav version 1.14

Lors de la conception du micro-logiciel, certaines librairies existantes de la société STMicroelectronics ont été ajoutées afin de nous permettre de programmer à un plus haut niveau sans devoir continuellement consulter la documentation technique pour connaître les différentes adresses des registres. Les librairies suivantes sont utilisées :

- « `stm32f10x_rcc.h` » : La librairie RCC permet de configurer et de contrôler les horloges programmables pour chaque périphérique;

- « [stm32f10x_gpio.h](#) » : La librairie GPIO permet la configuration des ports d'entrée / sortie du microcontrôleur;
- « [stm32f10x_i2c.h](#) » : La librairie I2C aussi appelée TWI permet de configurer le bus de données relié aux accéléromètres et magnétomètres;
- « [stm32f10x_tim.h](#) » : La librairie TIM permet de configurer les timers. Dans notre cas, le timer du processus;
- « [stm32f10x_adc.h](#) » : La librairie ADC permet de configurer et d'utiliser le convertisseur analogique / numérique. Celui-ci est utilisé pour les gyroscopes et le capteur de température;
- « [stm32f10x_dma.h](#) » : La librairie DMA permet de faire circuler les données de la mémoire du microcontrôleur vers et depuis les périphériques sans utiliser la puissance de calcul du microcontrôleur;
- « [stm32f10x_spi.h](#) » : La librairie SPI permet de configurer et d'utiliser le protocole de communication SPI. Celui-ci étant utilisé pour l'intercommunication des IMUs;
- « [stm32f10x_usart.h](#) » : La librairie USART permet de configurer et d'utiliser le protocole de communication UART. Celui-ci étant utilisé pour la communication entre l'IMU maître et le module sans-fil ainsi qu'avec le module LCD;
- « [stm32f10x_wwdg.h](#) » : La librairie WWDG permet de configurer et d'utiliser un chien de garde afin de pouvoir redémarrer les IMUs;
- « [stm32f10x_sdio.h](#) » : La librairie SDIO permet de communiquer avec la carte microSD de l'IMU maître;
- « [misc.h](#) » : Cette librairie permet de contrôler les interruptions liées aux périphériques;
- « [LSM303DLH.h](#) » : Cette librairie permet de configurer et d'utiliser l'accéléromètre et le magnétomètre avec le microcontrôleur;
- « [LPRYxxxAL.h](#) » : Cette librairie permet de configurer et d'utiliser les capteurs gyroscopiques avec le microcontrôleur. Note : Elle a été modifiée afin de permettre également la récupération des données du capteur de température interne.

Toutes ces bibliothèques sont incluses dans le fichier « `stm32f10x_conf.h` » et « `sensors_conf.h` ». On a bien entendu inclus les bibliothèques de base en C ainsi que les bibliothèques permettant d'utiliser notre microcontrôleur dans ce projet.

Trois bibliothèques permettant de faire des calculs matricielles « `matrix_lib.h` », la transformation de repères « `frame_transformation.h` » ainsi que des calculs plus rapides « `optimized_function.h` » ont également été créés ou adaptés pour le projet ibNav. Il est intéressant de s'arrêter sur la bibliothèque « `optimized_function.h` » qui contient trois fonctions mathématiques optimisées pour limiter la charge du microcontrôleur :

Gain de Kalman sans inversion de matrice (FEKF) : La méthode utilisée est décrite par Kang, Eyung W., 2008. Il est prouvé que le gain de Kalman peut être calculé sans inversion de matrice et qu'il peut être obtenu à l'aide de l'équation suivante :

$$K_K = \begin{bmatrix} P_{00}/s_0 & P_{02}/s_1 & \cdots & P_{0m}/s_{m-1} \\ P_{10}/s_0 & P_{12}/s_1 & \cdots & P_{1m}/s_{m-1} \\ \vdots & \vdots & \vdots & P_{2m}/s_{m-1} \\ P_{n0}/s_0 & P_{n2}/s_1 & \cdots & P_{nm}/s_{m-1} \end{bmatrix}^T \quad (6.1)$$

où : P_{nm} : Paramètres de la matrice de covariance d'erreurs P_K
 s_m : Paramètres de la diagonale de la matrice $S = H_K P_K^- H_K^T + R_K$

Bien évidemment, cette simplification fait perdre un peu de précision au système mais le gain de performance est important. On remarque un gain allant de 25 à 35% sur la vitesse de traitement de l'algorithme lorsque cette méthode est utilisée avec une influence négligeable sur la précision du système. Cette méthode est donc appelée pour tous les filtres de Kalman excepté celui de la calibration initiale où le temps de traitement n'est pas important. Une remarque importante à mentionner c'est qu'il ne faut jamais effectuer d'inversions de matrice sur un microcontrôleur, la charge du processeur qu'elle demande est beaucoup trop importante.

Racine carré inverse rapide : Cette méthode très utilisée dans le domaine des jeux vidéo est décrite par Lomont, C. 2003. L'algorithme en C est le suivant :

Algorithme 6.1 Racine carré inverse rapide

```
float FastInvSqrt(float x)
{
    float xhalf = 0.5f*x;
    int i = *(int*)&x; // get bits for floating value
    i = 0x5f375a86 - (i >> 1); // gives initial guess y0
    x = *(float*)&i; // convert bits back to float
    x = x*(1.5f - xhalf*x*x); // Newton step, rep increases accuracy
    return x;
}
```

Atan2 rapide : Cette méthode est également très utilisée en C et permet une rapidité de traitement 3 à 5 fois plus rapide. Elle est basée sur l'approximation d'arctangente de Hasting. Soit le code source suivant :

Algorithme 6.2 Arctangente rapide

```
float fastAtan2(float y, float x){
    float atanlocal;
    float z;
    if (x == 0.0f){
        if (y > 0.0f) return 1.5707963;
        if (y == 0.0f) return 0.0f;
        return -1.5707963;
    }
    z = y / x;

    if (fabs(z) < 1.0f){
        atanlocal = z / (1.0f + 0.28f*z*z);
        if (x < 0.0f){
            if (y < 0.0f) return atanlocal - PI;
            return atanlocal + PI;
        }
    }
    else{
        atanlocal = PIBY2_FLOAT - z / (z*z + 0.28f);
        if (y < 0.0f) return atanlocal - PI;
    }
    return atanlocal;
}
```

Au niveau des systèmes de communication avec les périphériques, on a quatre sections :

1. La communication SPI entre les différents IMUs est présente dans la librairie créée « `spi_conf.h` » associée avec « `dma_conf.h` ». Elles vont contenir la configuration SPI avec les DMA associées ainsi que la fonction permettant de construire les trames et la procédure pour la réception et la transmettre des données. De plus, on retrouve la fonction permettant de connaître le nombre d'IMUs connectés au système. Le processus de fonctionnement est indiqué à la Figure 6.5 et Figure 6.6;
2. La communication USART vers et depuis le module sans-fil est présente dans la librairie créée « `usart_conf.h` » associée avec « `dma_conf.h` ». Elle contient également deux fonctions pour la transmission des données par DMA et la réception de commandes par interruption. On retrouve également la fonction permettant de créer et de coder la trame. La Figure 6.5 présente ce processus de communication;
3. La communication USART vers l'afficheur LCD est dans la librairie créée « `usart_conf.h` ». Elle permet de coder et d'envoyer des informations à afficher au lecteur LCD. La procédure ainsi que les informations qui peuvent être affichées sont à la Figure 6.7;
4. La communication SPI avec la carte MicroSD est présente dans la librairie modifiée « `sd_sdio_stm32.h` » associée avec « `ff.h` » permettant de formater le fichier au format FAT. Cette fonction va permettre d'écrire par liaison SDIO des données binaires des capteurs inertiels à l'intérieur d'un fichier. Il sera également possible de les lire à partir de l'iPad.

Le programme a été réalisé en suivant une gestion des variables simple mais efficace. Soit, trois catégories de variables regroupées en quatre structures (Elles sont définies dans le fichier « `structures.h` ») :

- la structure `sRawMeasurement` contient les données brutes en provenance des capteurs inertiels (sur les 3 axes) et du capteur de température;
- la structure `sFlags` contient les différents booléens de type « drapeau »;
- la structure `sCom` contient les variables permettant au système de communication SPI et aux USART de fonctionner;
- la structure `sAlgorithms` contient les variables calculées en provenance des algorithmes inertiels.

Avant d'effectuer la compilation du projet, l'utilisateur peut choisir d'activer ou de désactiver trois définitions différentes, présentes dans le fichier « **definitions.h** » :

- la définition **SPIMASTER** indique au compilateur si on programme l'IMU maître ou les IMUs esclaves;
- la définition **DEBUG** permet d'utiliser une des LED présente sur l'IMU afin de vérifier le bon déroulement du processus;
- la définition **NAME** indique le numéro de série de l'IMU.

Le diagramme global du processus de l'application est le suivant (Figure 6.3) :

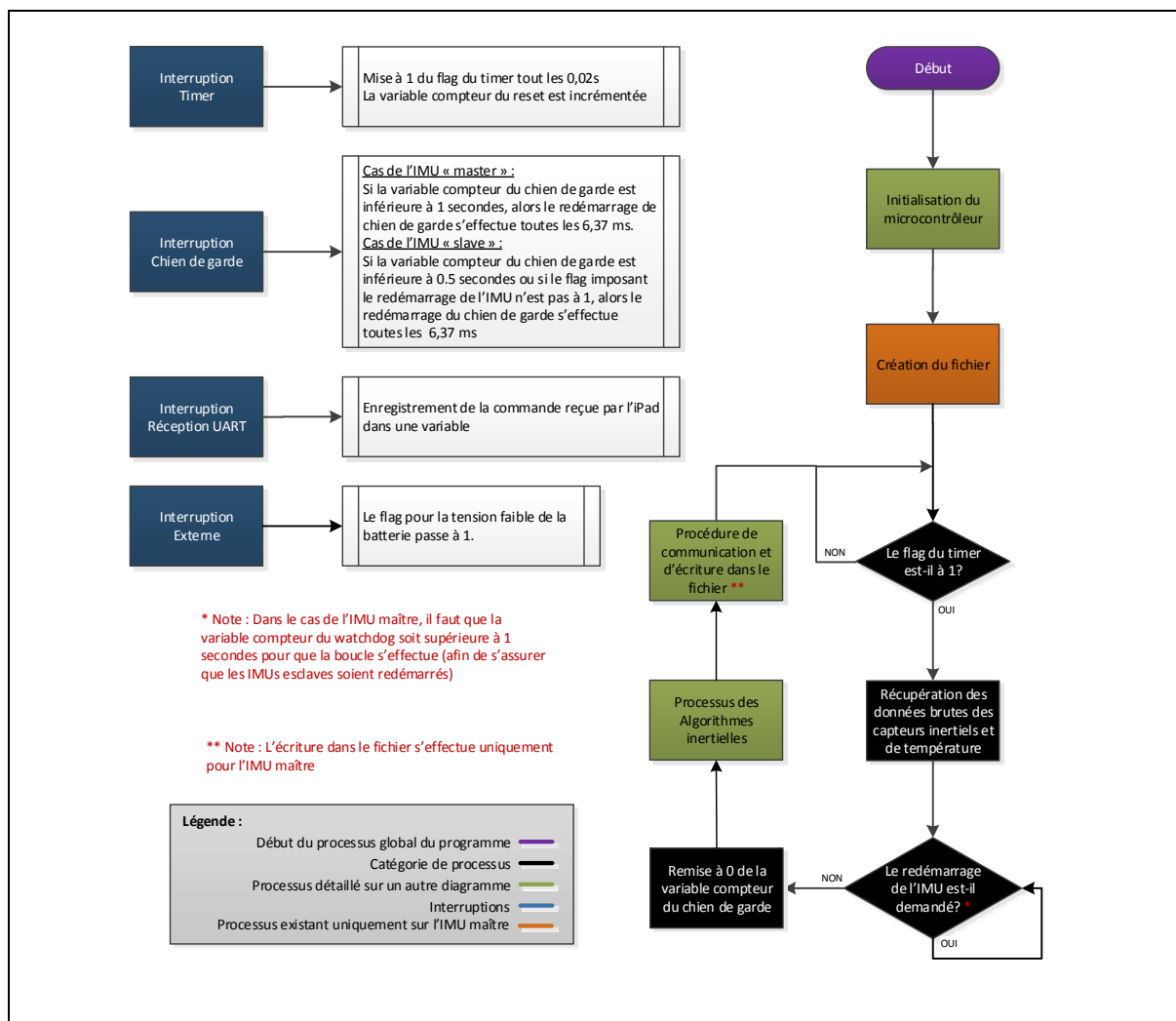


Figure 6.3 Processus global du micro-logiciel des IMU

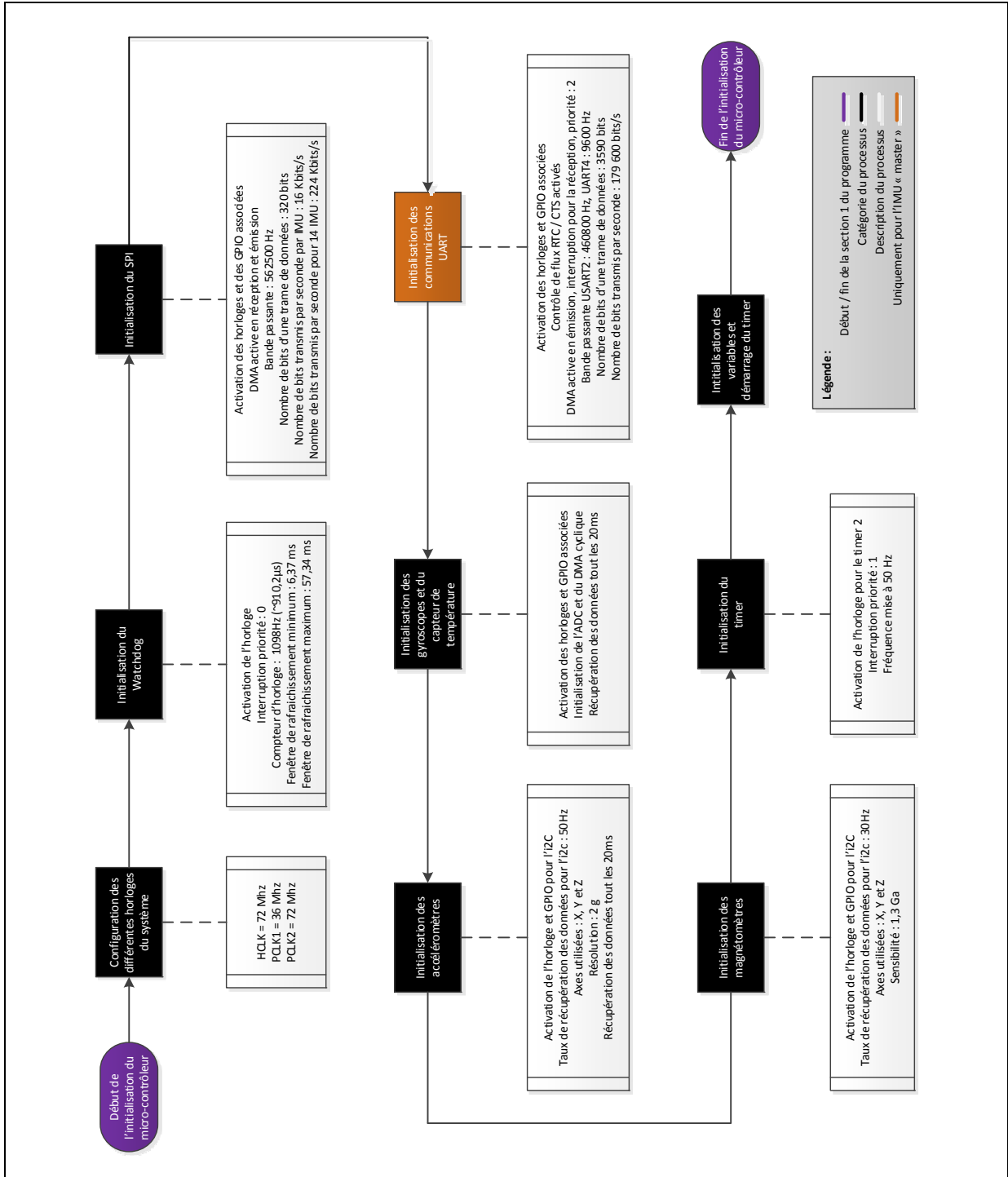


Figure 6.4 Processus concernant l'initialisation du microcontrôleur

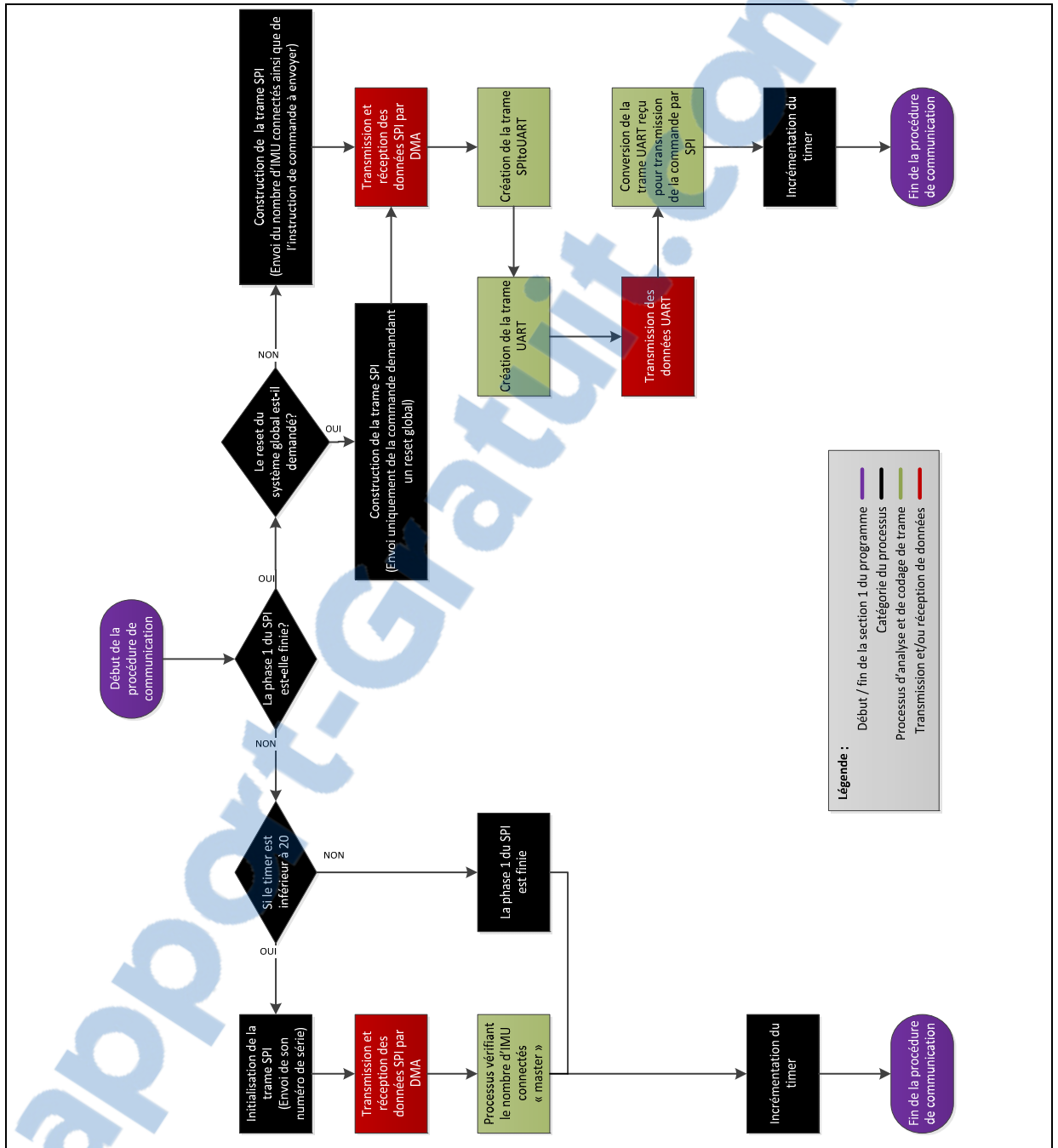


Figure 6.5 Processus concernant la communication pour l'IMU « maître »

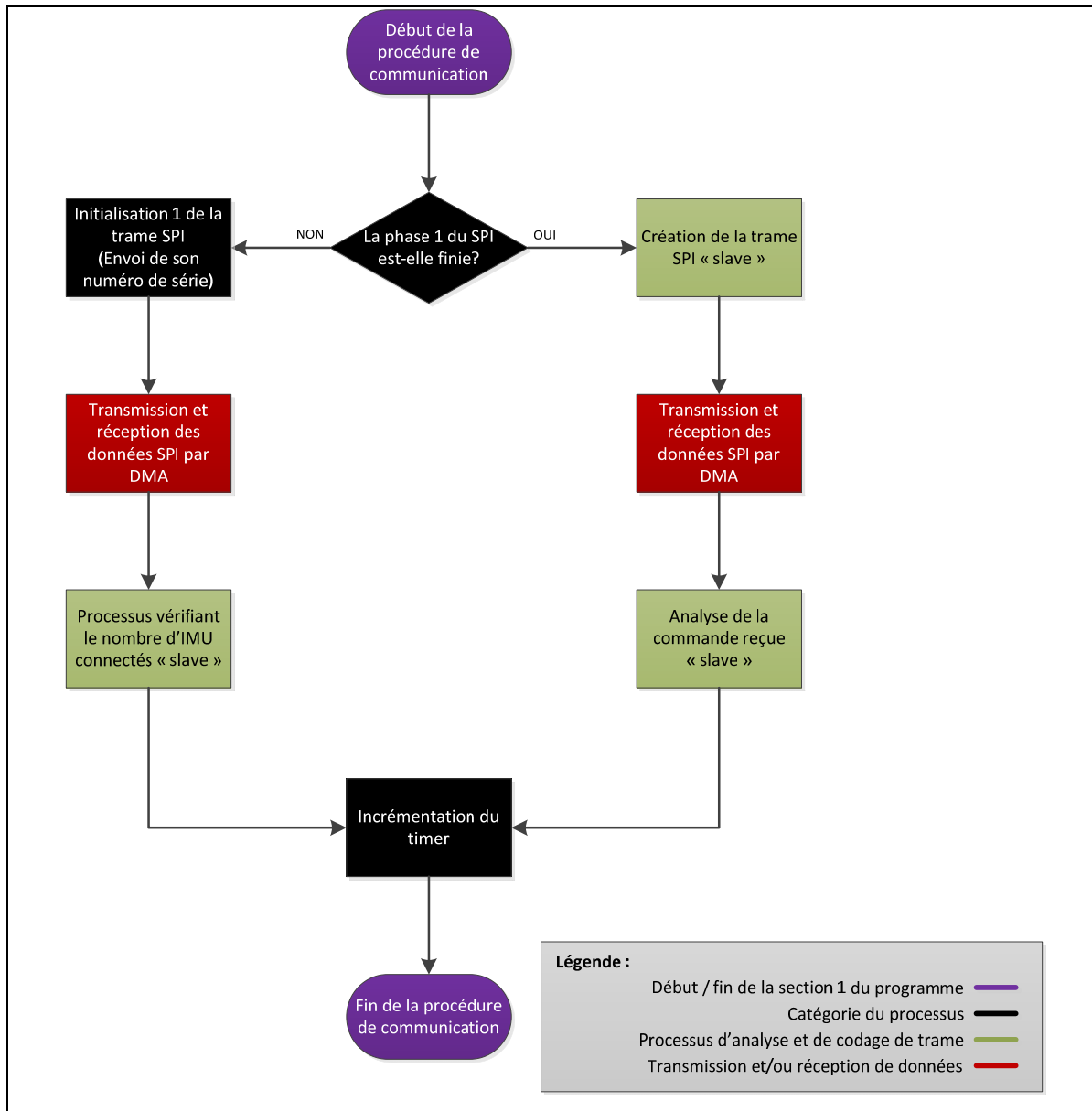


Figure 6.6 Processus concernant la communication pour les IMUs « esclave »

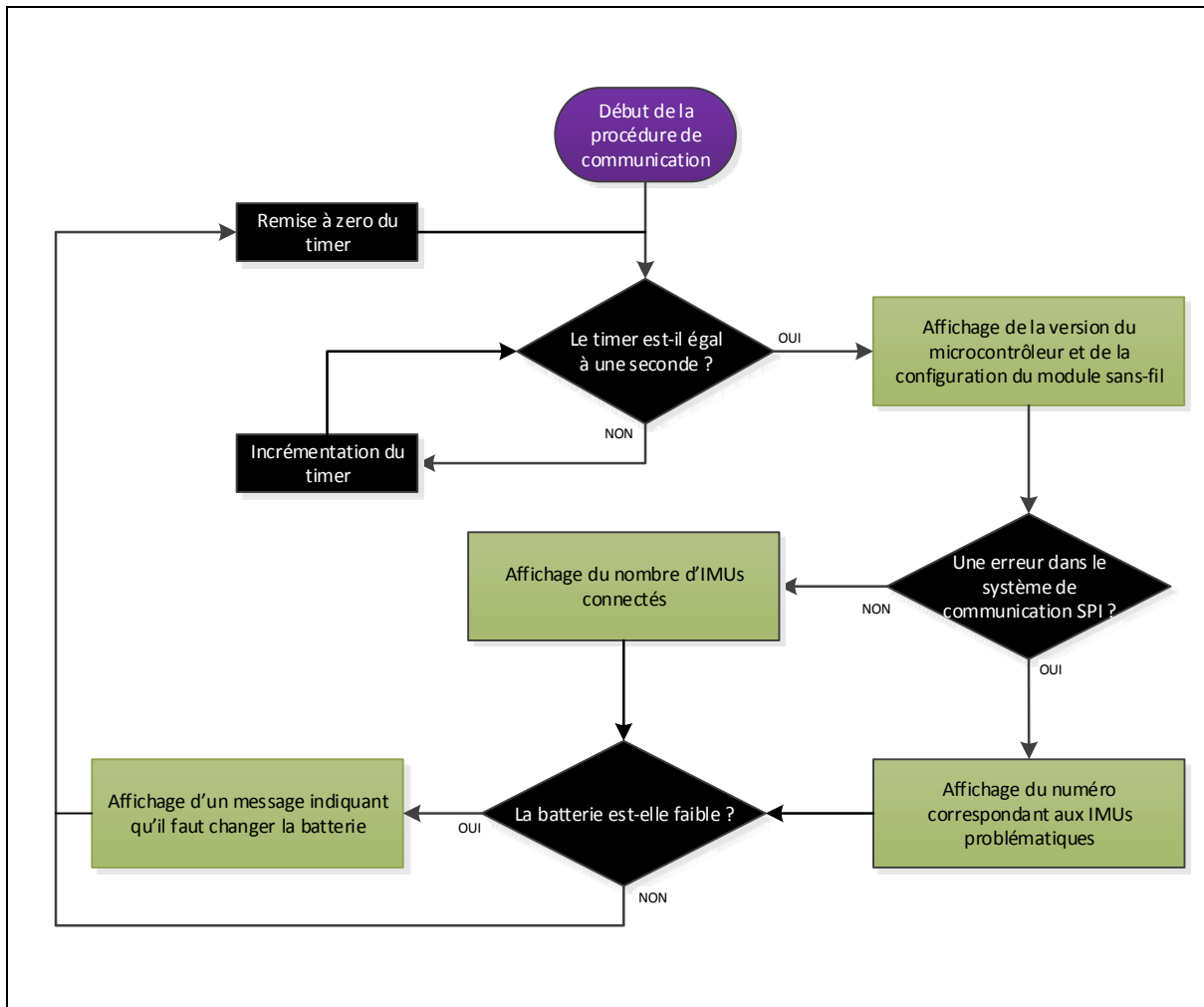


Figure 6.7 Processus concernant la communication avec l’afficheur LCD

Remarque : Les fichiers relatifs aux algorithmes ne sont pas détaillés puisqu’ils sont juste l’implémentation en C des algorithmes présentés au CHAPITRE 4 et CHAPITRE 5.

6.1.3 Fonctionnalités du micro-logiciel des IMUs

Toutes les fonctionnalités nécessaires aux besoins du projet ont été implémentées tout en respectant les performances possibles du système. Ces changements permettent une grande robustesse de celui-ci en intégrant uniquement l’essentiel avec des méthodes d’analyses aussi souples que performantes. Les fonctionnalités sont désormais les suivantes :

La liaison bidirectionnelle UART, WiFly :

Elle est bidirectionnelle, les flux de contrôle sont activés et le débit est à 460800 bits/s. La transmission est effectuée par DMA et la réception par interruption. Le codage du caractère de fin de trame a été simplifié et l'intégrité de celle-ci peut être vérifiée au niveau de l'iPad.

La liaison d'intercommunication SPI :

Dans les précédentes phases, l'initialisation du SPI, correspondant à la détection du nombre d'IMUs connectées au système prenait 12 secondes. Dans cette phase, de nouveaux algorithmes ont été développés et la détection du nombre d'IMU connectées prend désormais 0,4 secondes. De plus lors d'un redémarrage, l'IMU reçoit par le « maître » les informations nécessaires pour recréer son buffer automatiquement et ainsi ne pas perturber le système. De ce fait, dans le cas où une panne survient dans le système, l'IMU peut être retiré et remplacé sans perturber le fonctionnement du système. Il est également possible désormais de connecter ou de retirer des IMU durant le fonctionnement du système, en tenant compte que l'on informe l'IMU « maître » qu'il y a eu un changement. Chaque IMU « esclave » peut recevoir des informations de contrôle. Cette fonctionnalité permet pour l'instant le redémarrage des IMUs, mais une fois les algorithmes fonctionnels, elle permettra l'interaction avec eux. Une vérification robuste de l'intégrité de la trame est mise en place.

Note : Dans les phases précédentes, une fois le système ayant connaissance du nombre d'IMUs connectés, chacun d'entre eux créent leurs trames en utilisant l'allocation dynamique en mémoire. Ceci pouvant être une source d'erreurs potentielle (idéalement, il faut éviter de faire ce type d'allocation mémoire dans un microcontrôleur, pour des raisons de robustesse et de performance), elle est supprimée dans cette nouvelle version. Actuellement, juste la taille de la trame change pour les DMA (réception et émission) mais la place mémoire de la variable reste inchangée.

Le capteur de température interne :

Chaque microcontrôleur dispose d'un capteur de température interne relié à son ADC 1. Ce capteur de température se sert d'une tension de référence qui doit être stable afin d'obtenir

des résultats corrects. Dans la dernière phase développée, le DMA se chargeant de récupérer les données des gyroscopes, récupère en plus les données du capteur de température.

La remise à zéro automatique :

Cette fonction permet le redémarrage des IMUs de façon individuelle si un problème se produit au niveau de la communication ou durant le processus normal d'exécution. Plusieurs cas sont gérés : L'IMU « maître » est doté d'un « chien de garde » permettant son redémarrage après 0,08 seconde. Ce redémarrage se produit uniquement si l'utilisateur souhaite redémarrer tout le système ou dans le cas d'un arrêt du processus normal. Les IMUs « esclave » sont dotés d'un « chien de garde » permettant leur redémarrage après 0,04 seconde ou dans le cas où la variable indiquant le redémarrage voulu par l'utilisateur est mise à 1. Le premier cas est utilisé lors d'un arrêt du processus normal et dans le cas où l'on connecte de nouveaux IMU au système durant le fonctionnement. Le deuxième cas est utilisé lorsque l'utilisateur spécifie un redémarrage, celui-ci pouvant être assignée uniquement à un IMU, lorsqu'on effectue un redémarrage du système global ou dans le cas où l'on connecte de nouveaux IMUs dans le fonctionnement.

Note : Le temps de redémarrage de l'IMU « maître » est plus long afin de permettre aux autres IMUs du système de redémarrer automatiquement si l'un d'entre eux voyait son processus arrêté (dans le cas d'une exception non gérée et non connue à l'heure actuelle).

La réception de commandes :

Lorsque l'iPad envoie une commande, celle-ci est traitée dans l'IMU « maître » qui l'analyse et transfère à son tour une commande par la liaison SPI si nécessaire. Dans le cas présent, les commandes servent uniquement au redémarrage des IMU mais par la suite, ils permettront d'interagir avec les algorithmes inertiels et de choisir les données à transférer. Les commandes reçues peuvent être appliquées de manière individuelle ou d'une manière globale sur l'ensemble du système. Présentement, on peut effectuer un redémarrage global ou individuel des IMU. Ces fonctionnalités permettent entre autre le retrait ou l'ajout d'IMU durant le fonctionnement du système, comme nous l'avons vu précédemment.

Informations importantes :

Il est important de prendre en compte qu'il faut maintenant un système de communication robuste où la tolérance des erreurs de transmission doit être inférieure à 1%. Dans le système développé pour la phase 2, seulement 42,7% des données transmises étaient reçues. Ce qui ne permet pas l'utilisation des données brutes des capteurs inertiels. De plus, les données ne doivent pas être corrompues, sinon l'application et l'analyse de l'efficacité des algorithmes inertiels seront faussées et ne seront pas représentatives. De ce fait, des méthodes de vérification de l'intégrité des trames sont nécessaires pour la liaison SPI et UART.

Problèmes rencontrés :

La difficulté majeure rencontrée lors de l'élaboration du micro-logiciel concerne la combinaison des différentes fonctionnalités, essentiellement au niveau de la procédure de communication.

6.1.4 Résultats et possibilités supplémentaires

Le système de communication a été testé durant 24 heures sans arrêt et aucune panne n'est intervenue. Différents scénarios ont également été testés tel que la déconnexion et la reconnexion d'IMU durant le fonctionnement et la communication a également repris sans problèmes. Après analyse des trames reçues sur Matlab, on obtient plus de 99.5% de trames reçues. Les trames manquantes proviennent de la liaison SPI.

Avec les IMUs que l'on utilise pour le projet ibNav, on reste assez limité au niveau des performances et des I/O disponibles. Une amélioration qui pourrait être faite est de rendre la connexion des IMUs automatique durant le fonctionnement du système. Cette implémentation pouvant également servir à corriger les erreurs dans le système de communication, elle sera peut-être implémentée dans le futur mais il serait plus pertinent de créer nos propres IMUs (voir la section recommandations).

6.2 Développement du système de communication

Le système de communication est composé de quatre protocoles de communication, chacun ayant un rôle crucial pour le fonctionnement du système ibNav. Le système de communication doit être robuste et fiable afin de permettre la transmission des données brutes des capteurs inertiels.

6.2.1 Objectif du développement

Quatre protocoles ont été développés :

- **la communication SPI** : Système utilisé pour la communication entre les différents IMU;
- **la communication UART** : Système de communication utilisé avec l'IMU « maître » afin que celui-ci transmette toutes les données utiles des IMUs vers un périphérique externe;
- **la communication UDP** : Ce protocole de communication est utilisé afin que l'iPad « maître » puisse envoyer les données brutes ou calculés vers d'autres iPads servant uniquement pour l'affichage;
- **la communication TCP** : Ce système de communication a été développé pour la communication entre l'iPad et l'IMU « maître » pour la communication sans-fil.

La communication TCP peut s'effectuer directement avec l'iPad lorsque le WiFly est configuré en point d'accès, on peut également passer par un routeur sans-fil si, par exemple, on aurait besoin d'internet. La communication UDP permet d'utiliser plusieurs iPad en même temps afin de visualiser les données brutes et calculées du système.

6.2.2 Architecture du système de communication

L'architecture du système de communication correspond au schéma à la Figure 6.8.

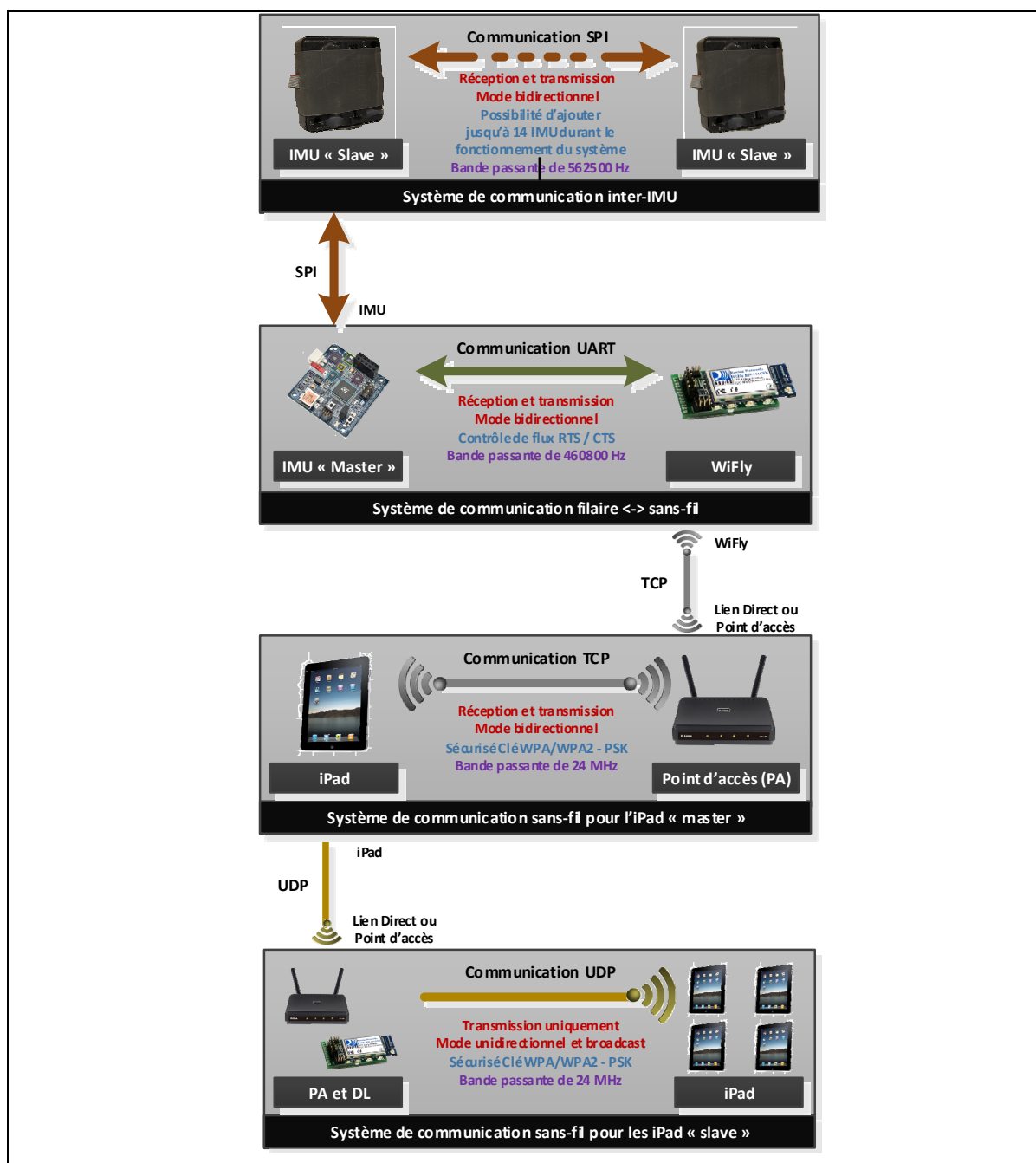


Figure 6.8 Schéma du système de communication mis en place

La communication inter-IMUs :

La communication SPI est bidirectionnelle et fonctionne avec une bande passante de 562 500 Hz. Une trame vaut 304 bits et le débit de données (pour les données brutes) est de 212 800 bits/s. Chaque donnée est codée sur 16 bits. La trame pour l'IMU « maître » et les IMUs « esclave » avant l'initialisation est présentée à la Figure 6.9. La trame pour l'IMU « maître » après l'initialisation est présentée à la Figure 6.10, dans ce cas, NB correspond au nombre d'IMUs détectés et UI à l'information envoyée de l'iPad pour interagir avec les IMUs. La trame pour les IMUs « esclave » après l'initialisation (données brutes) est présentée à la Figure 6.11. Fx correspond à des en-têtes codés avec le numéro de l'IMU, CK1 et CK2 correspondent à un checksum afin que l'IMU « maître » puisse vérifier l'intégrité des trames. Finalement, la trame pour les IMUs « esclave » après l'initialisation (données calculées) est présentée à la Figure 6.12

42	SPI_NAME	166
----	----------	-----

Figure 6.9 Trame pour l'IMU « maître » et les IMUs « esclave » avant l'initialisation

269	NB	524	269	NB	524	...	750	UI	126	750	UI	126	...
-----	----	-----	-----	----	-----	-----	-----	----	-----	-----	----	-----	-----

Figure 6.10 Trame de l'IMU « maître » après l'initialisation

F1	AccX	AccY	AccZ	F2	GyrX	GyrY	GyrZ	F3	MagX	MagY	MagZ	F4	TPS1
TPS2	Tim1	Tim2	CK1	Ck2	F5								

Figure 6.11 Trame des IMUs « esclaves » après l'initialisation (données brutes)

F1	Theta	Phi	Psi	F2	Tim1	Tim2	State	F3	PX	PY	PZ	F4	VX
VY	VZ	F5	CK1	Ck2	F6								

Figure 6.12 Trame des IMUs « esclaves » après l'initialisation (données calculées)

La communication filaire / non filaire (Série / Wifi) :

La communication sans-fil est réalisée à l'aide d'un module nommé WiFly. Ce module développé par la société Roving Networks a pour référence : RN-134. Il permet d'interfacer une liaison filaire UART en liaison sans-fil. Les caractéristiques importantes que l'on dénote sont les suivantes :

- il peut théoriquement fonctionner avec une liaison UART allant jusqu'à 1 Mbits/s mais fonctionne réellement à un débit maximum de 460800 bits/s. Ceci s'explique puisque la liaison entre le microcontrôleur et le module WiFly est réalisée à l'aide de fils d'une longueur d'environ 3 cm. La liaison à 1 Mbits/s serait possible si les deux entités seraient soudées sur un même PCB avec une faible distance les séparant;
- plusieurs modes de fonctionnement : Création d'un réseau Adhoc, point d'accès (PA) (version beta) ainsi que la possibilité de se connecter à un réseau sécurisé WEP, WPA1-PSK, WPA2-PSK;
- facilement configurable en utilisant la liaison UART / RS232;
- fonctionne avec un protocole UDP ou TCP.

La communication est également bidirectionnelle, avec une bande passante de 460 800 Hz. Chaque trame de 3592 bits (on a les données de chaque IMUs) est envoyée à 50 Hz, on a donc un débit de données de 179 600 bits/s. La trame a le format suivant (dans le cas des données brutes) :

10	AccX	AccY	AccZ	GyrX	GyrY	GyrZ	MagX	MagY	MagZ	TPS1	TPS2	
Tim1	Tim2	20	AccX	AccY	AccZ	GyrX	GyrY	GyrZ	MagX	MagY	...	0x0D

Figure 6.13 Trame de transfert de l'IMU « maître » vers l'iPad

6.3 Réalisation de l'architecture matérielle

L'architecture matérielle concerne le réseau filaire permettant l'alimentation et la communication des différents IMUs.

6.3.1 Objectifs de la réalisation

Cette architecture doit être robuste et flexible afin de ne pas nuire aux mouvements de l'utilisateur et de permettre une communication fiable. Chaque IMU est également placé dans un boîtier. Un vêtement a également été conçu afin de pouvoir y inclure le système développé. Voir le résultat à la Figure 6.14.

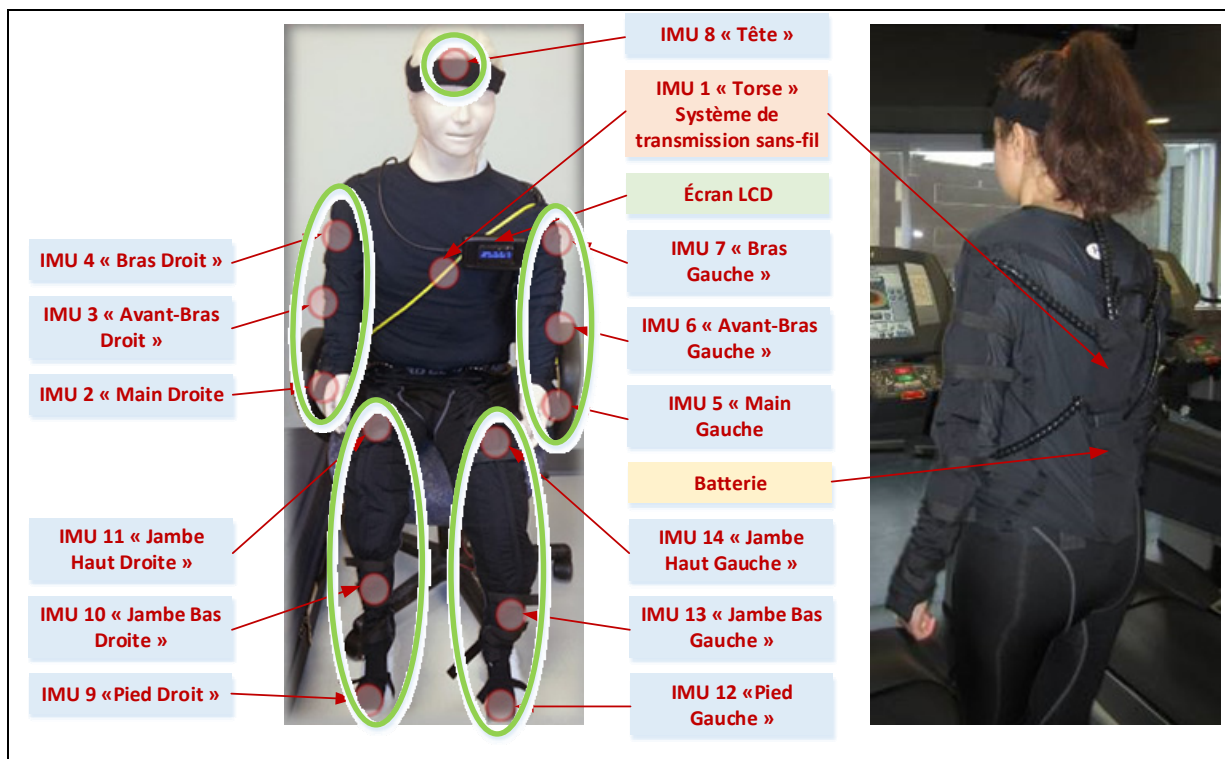


Figure 6.14 Architecture matérielle du système ibNav composé de 14 IMUs

6.3.2 Composition de l'architecture

Le système utilise des câbles spiralés afin d'augmenter la flexibilité et de ne pas nuire aux mouvements de l'utilisateur (Figure 6.15). Ces câbles sont également blindés. Éventuellement, ces câbles pourraient être intégrés à un vêtement spécifique (soldat, pompier ou autres). Des connecteurs sont également ajoutés sur chaque boîtier permettant de ne plus avoir le problème des câbles se cassant à cause de l'usage prolongé du vêtement. De plus, en cas de panne survenant sur un IMU, il est possible de les changer facilement sans devoir modifier tout un sous-système.

Un connecteur est présent sur le boîtier une fois fermé, ce qui permet la reconfiguration du microcontrôleur en cas de mise à jour sans devoir le démonter. Ceci permet surtout de ne pas prendre de risque en déplaçant l'IMU dans le boîtier par la procédure.

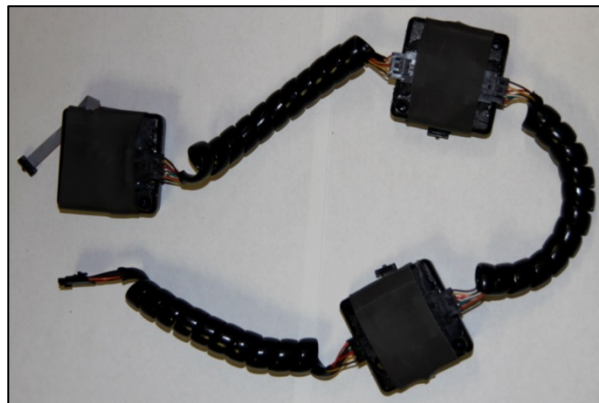


Figure 6.15 Photographie du sous-système pour le bras droit

La Figure 6.16 représente schématiquement le câblage de l'architecture matérielle du projet. On a une alimentation étoilée permettant de réduire l'intensité du courant présent dans chaque sous système à 0.2 A au lieu des 1.3 A circulant dans le système précédent. Puisque le système est décomposé en 5 sous-systèmes, il est également très facile de connecter uniquement certaines parties (par exemple la tête ou juste un bras). A la Figure 6.14, on peut

voir la position de chaque IMU par un cercle rouge et chaque sous-système par un cercle vert.

Finalement, la répartition de l'horloge de façon spiralée permet aussi d'amoindrir l'effet de l'atténuation du signal. Chaque sous-système est doté d'un optocoupleur en entrée (au niveau de l'IMU « maître » afin que la qualité de l'horloge reste bonne. Un régulateur de tension à découplage est également présent pour chaque sous-système (voir Figure 6.16).

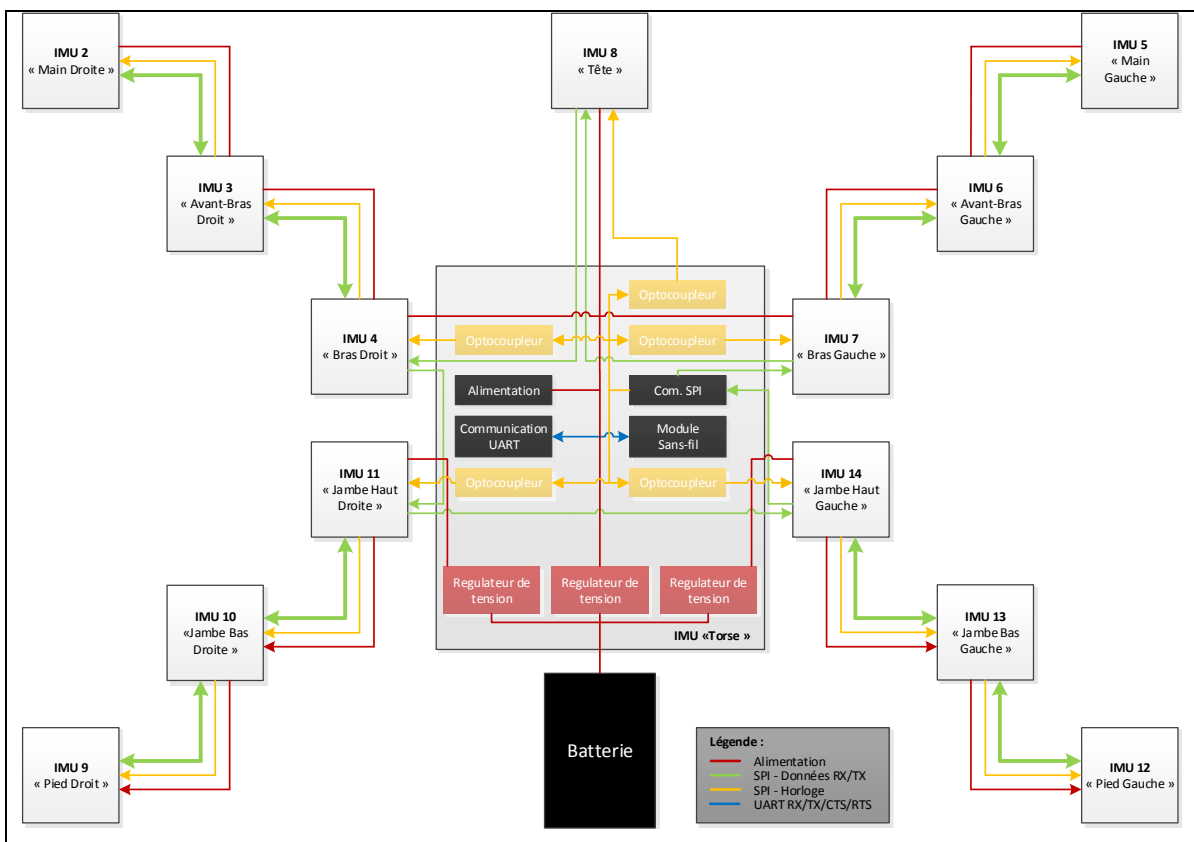


Figure 6.16 Schéma de l'architecture matérielle

Remarque : Seule l'horloge de la communication SPI passe par les optocoupleurs car elle est générée que par l'IMU « maître » et doit être reçue par tous les IMUs « esclave » contrairement aux données qui sont régénérées à chaque IMU.

6.4 Analyse du système ibNav (Coût, poids et consommation)

Le coût de ce prototype est estimé à une valeur de moins de 5000 \$ CAD décomposé comme au Tableau 6.1:

Tableau 6.1 Répartition du coût du premier prototype ibNav

Type	Quantité	Coût approximatif (\$ CAD / Unité)
IMU « Maître »	1	450
IMUs « Esclave »	13	300
Écran LCD	1	50
Batteries (2)	2	25
Câblage	13	5
Vêtement	1	150

Remarque : Il serait cependant possible de réaliser facilement nos propres IMUs au lieu d'utiliser les modules d'évaluation. Dans ce cas le coût du projet serait de 750 \$ approximativement et la robustesse du projet serait nettement améliorée. Ce point est discuté dans les recommandations pour la suite du projet.



Figure 6.17 Photographie des IMUs « esclave » et de l'IMU « maître »

Le poids du prototype incluant le vêtement fait approximativement 1 Kg, il pourrait être d'avantage réduit avec nos propres IMUs.

La consommation du système est de 3.589 Wh (pour le système intégral) avec 244 mWh / IMU. Les batteries que l'on utilise sont dotées de 6 piles AA ayant une capacité de 9.2Wh. Suivant les différents tests effectués, le système peut durer en fonctionnement environ 2 heures 30. Puisqu'il fonctionne avec des piles AA, la batterie peut facilement être changée. On pourrait également utiliser une batterie Lithium-Ion d'une capacité de 45 Wh (d'une valeur de 75\$), théoriquement le système pourrait fonctionner durant 12 heures.

Finalement, cette architecture est bonne et assez robuste pour permettre l'étude des capteurs inertiels. Cependant pour l'utiliser dans des applications commerciales, il faudrait développer nos propres IMUs afin de réduire son coût, d'améliorer sa robustesse et de réduire son encombrement.

CHAPITRE 7

RÉALISATION DE LA PLATEFORME D'ÉTUDES IPAD

Ce chapitre est le descriptif du développement de l'interface graphique utilisateur pour iPad spécialement conçue pour le projet ibNav. Cette partie du développement correspond approximativement à 30% du temps de travail total effectué durant la maîtrise (incluant encore une fois le temps d'autoformation, de réalisation et de test). De nombreuses fonctionnalités ont été implémenté afin d'en faire une interface pour les utilisateurs finaux et pour étudier le comportement des capteurs inertiels. La figure 7.1 présente les fonctionnalités que doit inclure le logiciel.

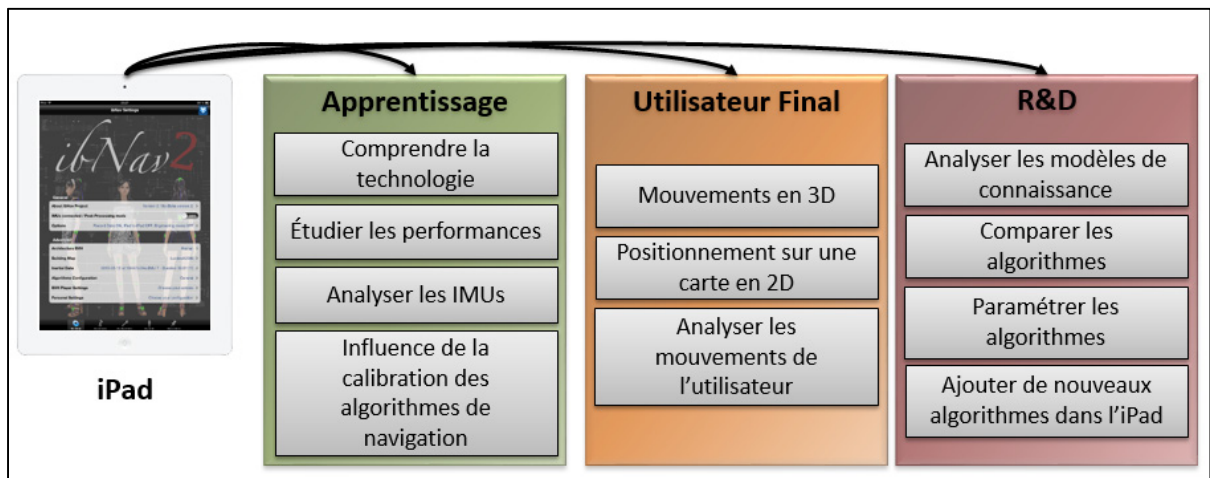


Figure 7.1 Fonctionnalités devant être incluses dans le GUI pour iPad

7.1 Objectifs de la réalisation

Trois sous-objectifs doivent être atteints par la réalisation de ce logiciel (Figure 7.1), associés à trois catégories de personnes différentes :

- les étudiants et les personnes présentes à des événements et exhibitions. C'est le mode apprentissage présentant les technologies inertielles et leur fonctionnement;
- les chercheurs et étudiants des laboratoires de recherche. C'est le mode recherche et développement permettant l'étude des algorithmes inertiels, c'est-à-dire contenir des

fonctionnalités permettant de les modifier, comparer et de tester leurs efficacités et leurs performances;

- les utilisateurs finaux du système. Ce dernier doit permettre de visualiser les mouvements en 3D du corps humain et les déplacements de l'utilisateur sur une carte 2D.

Afin de répondre à ces objectifs, il faudra que l'application inclue les fonctionnalités suivantes :

- le mode temps réel et post-traitement de l'application. Les données brutes et calculées doivent pouvoir être visualisées dans ces deux modes;
- le transfert des données enregistrées. Dans le cas d'ibNav, il peut se faire de trois façon : Via iTunes, via le Dropbox ou par courriels;
- la communication entre plusieurs iPads, afin qu'on puisse visualiser les données sur plusieurs matériels iOS. Il y a donc un mode « maître » et « esclave » à mettre en place et dans ce cas, une communication UDP;
- la communication TCP bidirectionnelle vers le prototype ibNav afin de recevoir les données et de pouvoir le configurer;
- la possibilité qu'on puisse configurer la communication sans-fil afin que le système ibNav fasse lui-même le point d'accès ou qu'il se connecte à un point d'accès existant (afin d'avoir accès à internet et de pouvoir transférer les fichiers);
- le multitâche afin de dissocier l'interface graphique de la communication et des algorithmes;
- la possibilité de lire les fichiers KML et BVH et de pouvoir les éditer;
- la création d'une librairie graphique fonctionnant en temps réel et en post traitement permettant d'afficher des courbes pour les données brutes et calculés;
- la création d'un objet en 3D permettant de visualiser l'attitude du système;
- la création d'une carte 2D pour visualiser le déplacement de l'utilisateur;
- la création d'un squelette 3D pour représenter les mouvements du corps;
- la base de données afin d'optimiser la gestion des enregistrements et les paramètres des algorithmes.

7.2 Implémentation des interfaces graphiques utilisateurs

Le logiciel développé est décomposé en 5 grandes sections reliées par le contrôleur de navigation (Figure 7.2) :

- « **My Setup** » : C'est l'interface principale de l'application qui permet d'accéder aux différentes configurations, au résumé du projet, aux données enregistrées, aux architectures BVH, etc;
- « **My Sensors** » : Cette interface permet de visualiser les données brutes et ceux calculées pour chaque IMU;
- « **My algorithms** » : Cette interface permet de modifier les paramètres des algorithmes et d'étudier leurs performances;
- « **My Body** » : Cette interface permet l'affichage des mouvements du corps en 3D sur un squelette humain;
- « **My Location** » : Cette interface permet de représenter la localisation de la personne sur une carte 2D du bâtiment.

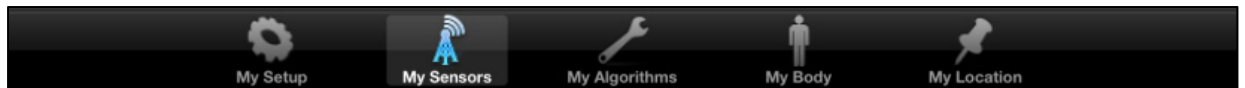


Figure 7.2 Barre de navigation du logiciel iOS

7.2.1 Interface principale

L'interface principale (Figure 7.3) est présentée schématiquement à la Figure 7.4, elle est décomposée en trois grandes parties :

Le premier bouton de section Onglets :

Il permet d'accéder à une interface permettant de configurer l'iPad et le module sans-fil.

- concernant l'iPad, celui-ci peut soit être en mode « maître » ou en mode « esclave » en sachant qu'il ne peut y avoir qu'un iPad « maître ». Les iPad « esclave » reçoivent les données brutes et calculées de l'iPad « maître » afin de les afficher. Ceci est très utile lors

des présentations afin d'avoir par exemple, un iPad affichant les graphiques de données brutes et un autre pour les mouvements du corps;



Figure 7.3 Interface principale et à propos du logiciel ibNav sur iPad

- concernant le module sans-fil, lorsque la connexion est établie avec le système ibNav, on peut reconfigurer le module sans-fil en point d'accès (afin que le système soit autonome sans avoir besoin de routeur) ou qu'il se connecte à un point d'accès (afin d'avoir un accès à internet pour le partage des données);

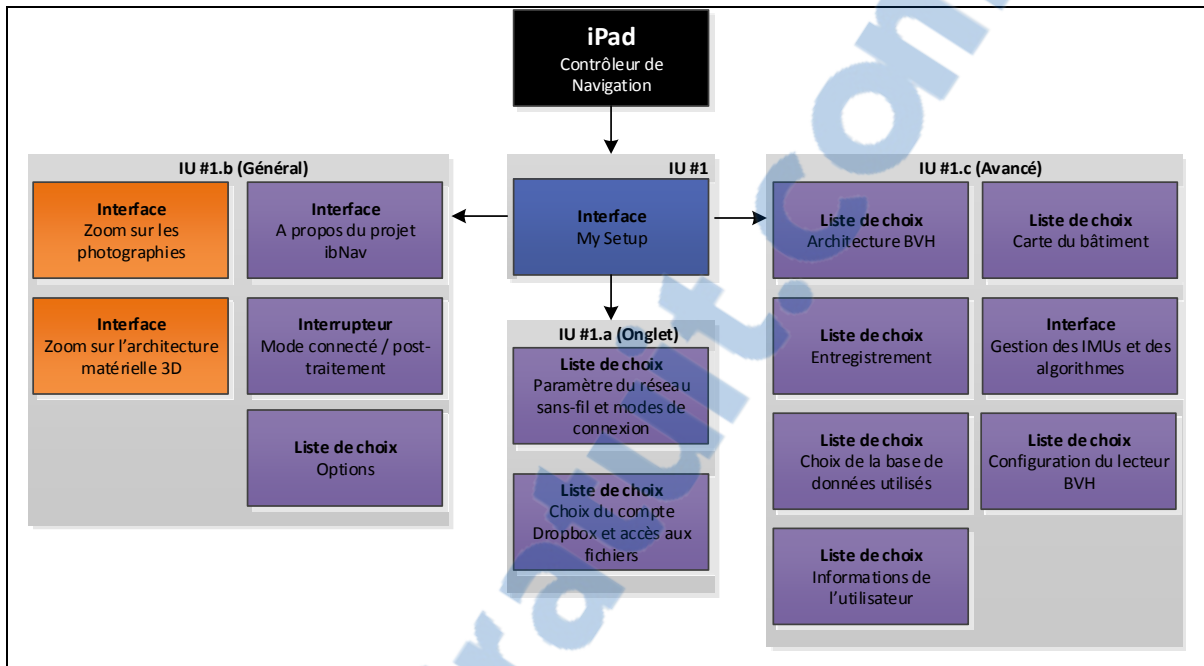


Figure 7.4 Schéma des différentes sections de l'interface principale

Le deuxième bouton de section Onglets :

Il permet d'accéder au Dropbox, on peut choisir sur quel compte Dropbox l'application sera liée; visualiser les fichiers qu'il contient et les télécharger. Le troisième bouton (non présent sur la Figure 7.3) permet de configurer le serveur SMTP afin de pouvoir envoyer les données par courriel.

La section Général :

Elle est décomposée en trois parties :

- la première est un résumé du projet ibNav (Figure 7.3) avec les versions, des photographies du projet et le modèle 3D où sont positionnés les IMUs. Si la personne touche une image, celle-ci se mettra en pleine écran et les fonctions de zoom et rotation seront activés. Si la personne touche le modèle 3D, celui-ci permet d'afficher différents point de vue du modèle en 3D;
- la deuxième est l'interrupteur permettant de se connecter au système ibNav;
- le troisième contient 3 options :
 - activer/désactiver les enregistrements;

- activer/désactiver la communication entre iPad;
- activer/désactiver le mode ingénieur (Ce mode de fonctionnement contient pour le moment des fonctions avancés au niveau de la configuration du module sans-fil, des capteurs et des algorithmes).

La section Avancé :

Elle est décomposée en sept parties :

- le choix de l'architecture BVH, celle-ci représente différentes articulations du modèle utilisé, dans notre cas le corps humain. Ce fichier peut facilement être édité et modifié avec l'iPad; Ceci peut être utile si on souhaite effectuer des modifications mineures, par exemple la distance entre deux articulations;
- le choix de la carte du bâtiment, soit la carte 2D du bâtiment où le système va être utilisé;
- le choix de l'enregistrement à rejouer. Si l'interface n'est pas connectée au système ibNav il est possible de visualiser un scénario précédemment enregistré. Ce fichier contient uniquement les données brutes en binaire provenant du système ibNav. Si ce fichier n'a jamais été rejoué, la première étape consistera à le convertir afin de l'inclure dans la base de données et qu'il puisse être rejoué par la suite plus rapidement. Pour chaque fichier, l'utilisateur a des informations concernant la date, l'heure du début et la durée de l'enregistrement ainsi que le nombre d'IMUs connectés (voir Figure 7.5);

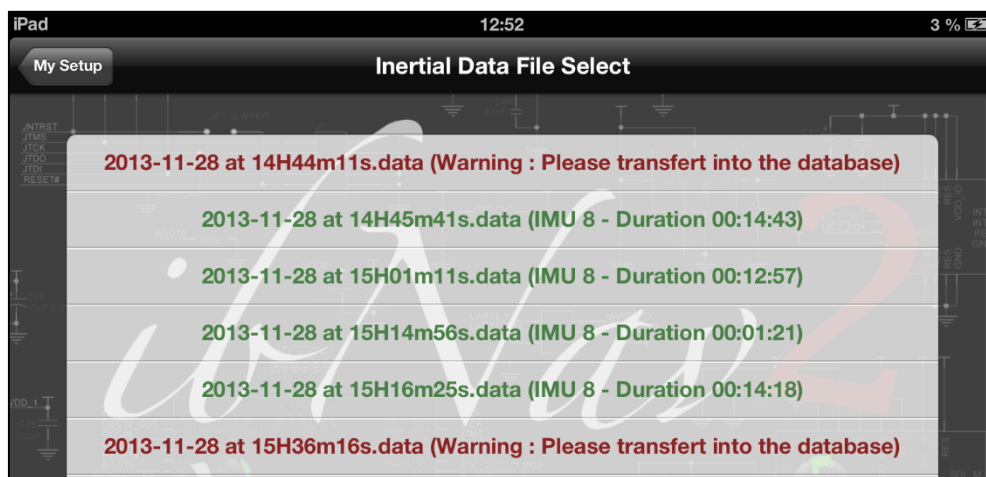


Figure 7.5 Exemple de l'interface permettant de sélectionner un enregistrement

- la gestion des IMUs et des algorithmes. Cette interface est en cours de développement, elle permet, lorsque l'iPad est connecté au système ibNav, de reconfigurer les capteurs inertiels et de choisir/configurer les algorithmes qui seront utilisés au sein de chaque IMU;
- le choix de la base de données à utiliser. Cette interface permet de charger une base de données différence, entre autre si l'utilisateur veut une configuration spécifique préalablement enregistré;
- la configuration du lecteur BVH. Cette interface en cours de développement permettra d'activer ou de désactiver des options pour le lecteur BVH développé sous unity3D;
- les informations sur l'utilisateur utilisant le projet. Cette interface est pour une utilisation futur, elle permettra à l'utilisateur d'entrer des informations le concernant tel que son poids, sa taille, la vrai distance entre chacun des capteurs, etc. Ceci afin d'améliorer l'algorithme de navigation à l'intérieur des bâtiments.

7.2.2 Interface pour les données brutes

L'interface pour les données brutes (Figure 7.6), elle permet de visualiser sous forme de graphique les données des accéléromètres, gyroscopes, magnétomètres, température interne et le timer interne fonctionnant à 1000Hz de chaque IMUs (cadre 1). Cette interface peut tout aussi bien fonctionner en temps-réel qu'avec un enregistrement en post-traitement.

Les différents IMUs sont représenté sur un modèle 3D : Un IMU en rouge veut dire qu'il n'est pas connecté au système alors qu'un IMU en vert indique qu'il est disponible. L'IMU sélectionné par l'utilisateur sera en bleu. La sélection s'effectue de manière tactile en touchant l'IMU en question sur le modèle 3D. Il est également possible d'utiliser les flèches en haut du modèle 3D afin de changer son orientation et ainsi visualiser tous les IMUs. Si l'utilisateur le souhaite il peut également visualiser la table indiquant le nom de chaque IMUs, il peut également les sélectionner à partir de celle-ci (cadre 2).



Figure 7.6 Interface pour les données brutes du logiciel ibNav sur iPad

Sur chaque graphique, lors du mode post-traitement, il est possible de zoomer sur les graphiques en utilisant deux doigts, il est possible de se déplacer également sur le graphique en utilisant un doigt et en le faisant glisser vers la droite ou la gauche. Lors du mode temps-réel, le graphique a une mémoire tampon de 10 secondes, par la suite les courbes se déplaceront par la suite au fur et à mesure que les données arriveront (cadre 3).

En bas de l'interface, l'utilisateur visualise des informations sur l'enregistrement, il peut également ajouter un commentaire et le synchroniser avec l'API Dropbox afin de le partager à d'autres utilisateurs (cadre 4).

Finalement si l'utilisateur sélectionne l'onglet « Parameters », il peut configurer l'unité de chaque graphique pour les différents capteurs. Dans le mode ingénieur il est possible de modifier les constantes (cadre 5).

7.2.3 Interface pour les algorithmes

Cette interface est en cours de développement, elle permet les points suivants :

- visualiser les angles d'Euler, la vitesse et la position sous forme de courbes;
- visualiser l'attitude du système à l'aide d'un cube en 3D;
- dans le mode post-traitement, elle permet d'effectuer certains algorithmes sur les données brutes, par exemple l'algorithme de calibration. L'utilisateur pourra voir les résultats, il pourra changer certains paramètres ou utiliser d'autres algorithmes similaires afin de comparer les résultats.

7.2.4 Interface pour la capture de mouvements et la localisation à l'intérieur des bâtiments

L'interface pour la capture de mouvements a initialement été développée en OpenGL. C'est un modèle 2D en fil de fer d'un corps humain. Ce développement initial ne pouvant pas être porté en 3D pour la suite du projet (c'est également très complexe de directement utiliser de l'OpenGL), nous avons choisi par la suite d'utiliser le moteur 3D d'Unity3D afin de réaliser cette interface (Figure 7.7). Ce dernier modèle n'est pas encore intégré à l'iPad.

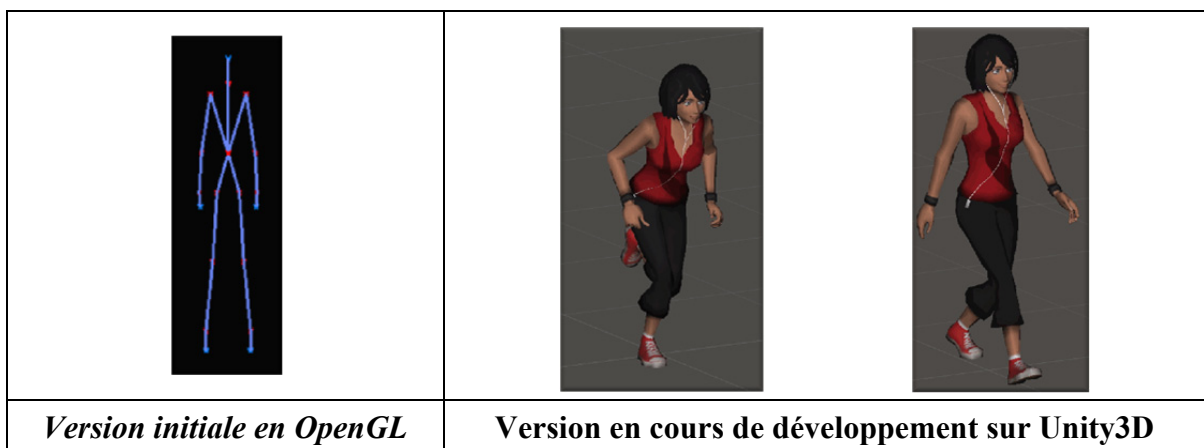


Figure 7.7 Version initiale et en cours du développement du modèle pour la capture de mouvements

L'interface pour la navigation à l'intérieur des bâtiments contient actuellement certaines cartes 2D du laboratoire LACIME et LASSENA mais le développement afin d'afficher la position de l'utilisateur sur la carte n'a pas encore été réalisé à ce jour.

7.3 Outils intégrés

Outre les interfaces graphiques, différents outils ont été intégrés / développés pour le logiciel afin de permettre les différentes fonctionnalités qu'offre le logiciel ibNav pour le système de communication, la gestion de la base de donnée et la visualisation.

7.3.1 Mise en place de la communication UDP / TCP

Les bibliothèques « open source » AsyncSocket et AsyncUDPSocket ont été utilisées afin de gérer la communication UDP et TCP du projet ibNav. Elles permettent facilement d'établir une communication client dans ces deux protocoles. La réception des données a été configurée afin de fonctionner en multitâche, dans une tâche fonctionnant en arrière-plan ne dérangeant ainsi pas l'interface graphique et les autres fonctionnalités de l'application.

La tâche pour la réception des données TCP a la plus haute priorité afin d'assurer une réception à 50 Hz. Ceci est important puisque les données sont uniquement sauvegardées au niveau de l'iPad (même si dans le futur elles seront sauvegardées également dans une carte microSD).

Concernant la communication UDP, à la fin de chaque trame reçue du système ibNav par l'iPad « maître » (communication TCP), ce dernier envoie, si la fonctionnalité est activée, les trames UDP en « broadcast » afin que tous les iPad reliés sur le réseau soient capables de recevoir les données brutes ou calculées. Les iPad « esclaves » ne peuvent cependant pas interagir avec le système ibNav, uniquement l'iPad « maître » permet de modifier les configurations.

7.3.2 Mise en place d'une base de données

Le Framework CoreData a été utilisé pour la base de données, il permet d'utiliser une base SQL directement à partir de commandes en objective-C. Elle contient 4 entités différentes (voir Figure 7.8) et une dizaine d'interaction avec la base de données ont été implémentés.

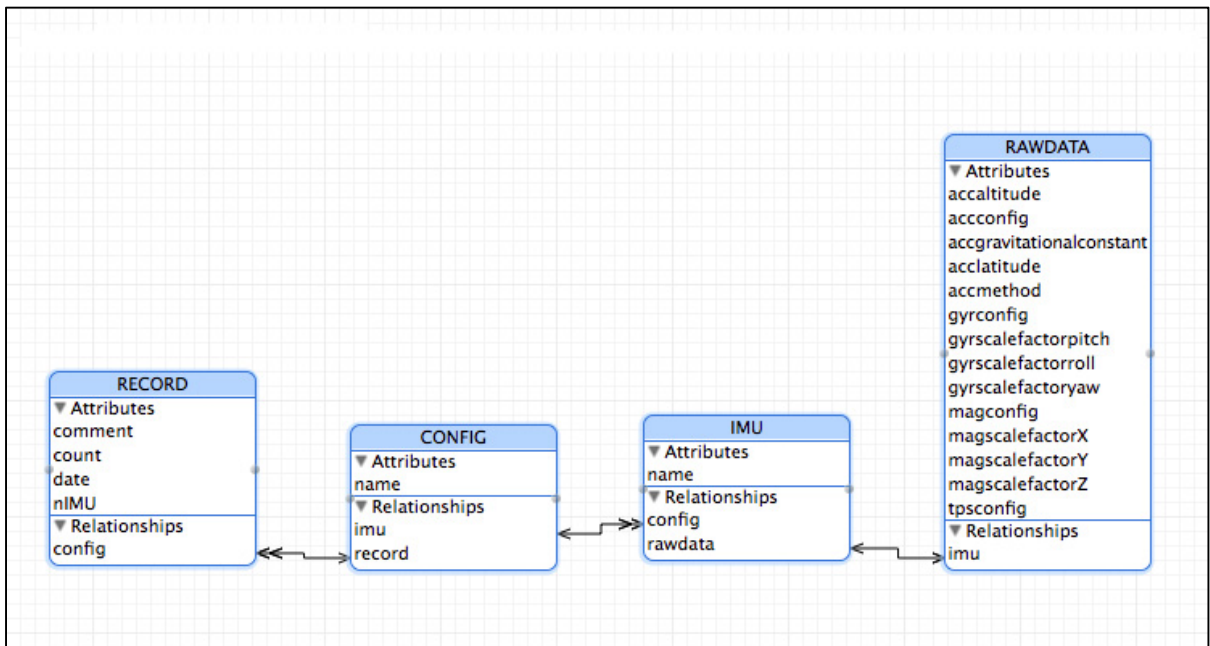


Figure 7.8 Schéma SQL de la base de données SQL du projet ibNav

Les différentes interactions sont les suivantes :

- lors du premier démarrage de l'application, celui-ci crée la base de donnée pour l'entité CONFIG et ses dérivés IMU et RAWDATA. La configuration prendra le nom de « General » et contiendra les configurations de base pour chaque IMU. Cette configuration ne peut être modifiée par la suite, l'utilisateur peut cependant la copier sous un autre nom et la modifier directement avec l'interface de l'iPad. Si celle-ci existe déjà, l'application chargera uniquement la configuration « General »;
- copier, supprimer, rechercher et créer une configuration;
- lorsque l'utilisateur ajoute un enregistrement à la base de données, l'algorithme enregistre la date, le commentaire et calcul le nombre d'IMUs présents dans le fichier. Par la suite il

calculera le nombre de trame pour chaque IMU, cette information est très utile pour la librairie permettant d'afficher les graphiques.

Remarque : La base de données est en constante évolution, elle sera nettement plus importante lorsque l'application sera finalisée. Les différentes interactions ont cependant été implémentées et il est assez simple maintenant de l'utiliser et de poursuivre le développement.

7.3.3 Réalisation d'une interface 2D / 3D utilisant la librairie OpenGL puis Unity3D

Cette partie a déjà été discutée à la section 7.2.4. L'idée serait dans le futur d'implémenter le modèle 3D créé sous 3Ds Max (voir Figure 7.9) au niveau d'unity3D afin de visualiser les capteurs en mouvement.

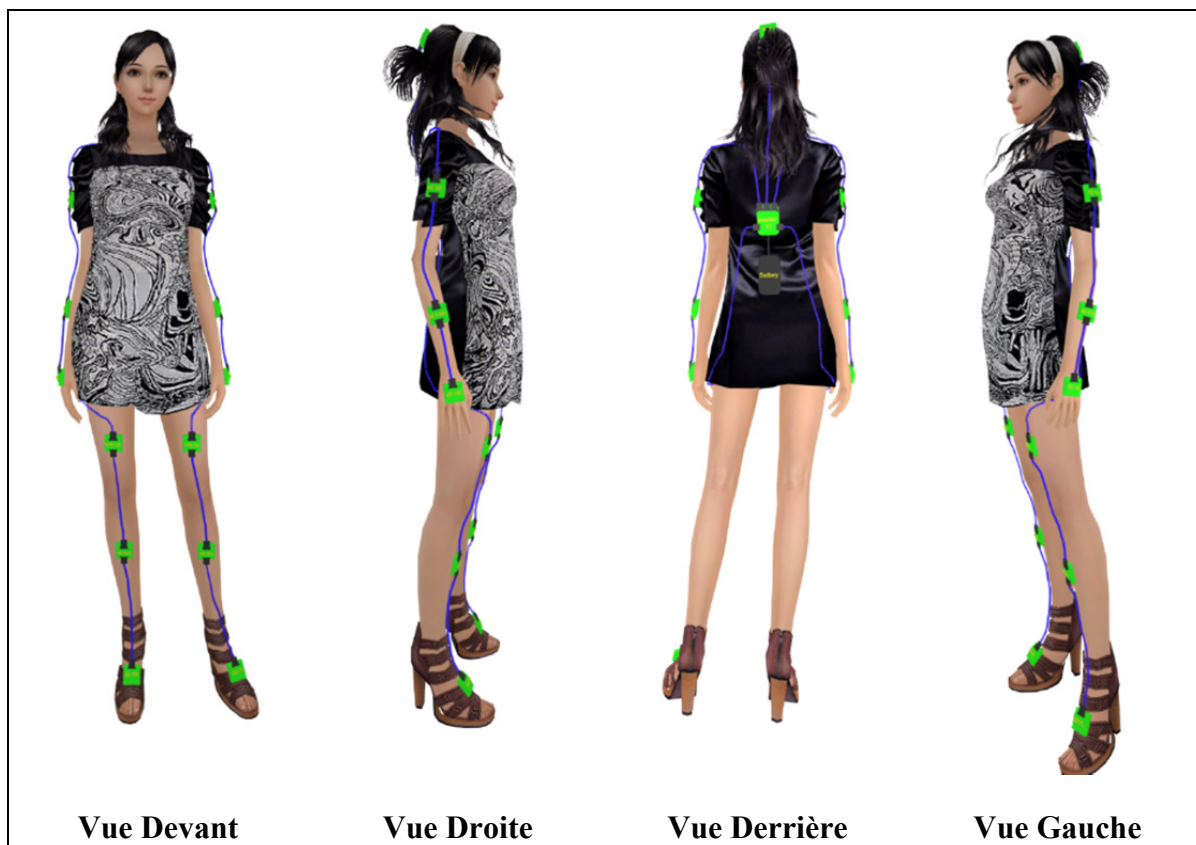


Figure 7.9 Modèle 3D du projet ibNav développé sous 3Ds Max

7.3.4 Réalisation d'une librairie pour la création de graphiques

Pour le projet ibNav, nous avons créé des librairies pour iOS permettant d'afficher des données sous forme de courbes fonctionnant tout aussi bien en temps réel qu'en post-traitement. Ces librairies se nomment « GraphView » et « GraphViewRealTime ».

L'utilisation est très simple. Il faut premièrement initialiser les variables en indiquant la couleur et l'unité de chacun des axes. Par la suite deux fonctions seront appelées lorsque l'on souhaite actualiser le graphique, une indiquant la taille total des données à afficher et l'autre contenant le tableau de données.

Il est donc nécessaire pour l'affichage en temps réel d'avoir une tâche en arrière-plan fonctionnant à une certaine fréquence (idéalement 10 Hz) permettant d'actualiser les données. Pour l'utilisation en post-traitement il faut également spécifier la valeur minimale et maximale de l'axe X.

Au démarrage de la tâche principale de la librairie, celle-ci va vérifier premièrement si les valeurs des axes sont cohérentes et calculer la valeur maximale et minimale pour l'axe Y. Elle va ensuite dessiner le cadrage en X et en Y en prenant en compte la taille du graphique puis les axes en prenant en compte l'unité et les données que l'on veut afficher.

Les courbes vont par la suite être dessinées en respectant un pas dépendant du nombre de données que l'on souhaite afficher. En effet dans le mode post-traitement, on peut avoir de nombreuses données (1 heure d'enregistrement donnerait 180 000 données) et les graphiques font une longueur maximale de 540 pixels. Il n'est donc pas utile de dessiner des données qui ne peuvent être visibles. La librairie prend donc en compte le nombre de pixels pouvant être affichés à l'écran et si par exemple il y a 300 données pour 1 pixel, l'application affichera uniquement le minimum et maximum de ces 300 points.

7.3.5 Mise en place de diverses API

Au niveau du projet ibNav, l'API Dropbox et Mailcore2 ont été implémentées :

- la première permet de partager des fichiers de l'application de manière automatique. Ces fichiers peuvent être sur le Dropbox de l'application ou être partagés sur n'importe quel Dropbox d'un utilisateur;
- la deuxième API permet d'envoyer des courriels, celle-ci a été modifiée afin de permettre d'envoyer des fichiers joints aussi bien que des images intégrées. A ce jour, cette librairie n'a pas entièrement été implémentée sur ibNav mais elle l'a été sur le projet iMetrik Show Case (voir section 8.2.2).

CHAPITRE 8

SYSTÈME GLOBAL ET TRANSFERT TECHNOLOGIQUE

Tous les développements détaillés et les tests effectués dans les chapitres précédents ont permis de créer un prototype fonctionnel avec de premiers résultats obtenus très prometteurs. Cependant, le projet nécessite encore des améliorations afin de devenir commercial. Néanmoins, lors du déroulement de la maîtrise, un besoin évident s'est fait sentir sur d'autres projets du laboratoire LASSENA, essentiellement sur le projet VTADS, ce qui est venu ajouter un nouvel objectif à accomplir, non prévu à l'origine, c'est-à-dire d'effectuer des transferts technologiques du développement d'ibNav. A partir de Novembre 2013, tous les développements effectués sur ibNav ont été portés sur d'autres projets. Ce chapitre montre donc qu'en plus des réalisations et des résultats associés à ibNav, l'architecture étant stable, modulaire et flexible, elle a pu répondre aux spécifications demandées dans d'autres projets.

8.1 Synthèse et rendu du projet ibNav

Cette section synthétise les différents développements réalisés pour le projet ibNav dans le cadre de la maîtrise avec les différents tests que nous avons effectués dans le laboratoire.



Figure 8.1 Projet ibNav en fonctionnement avec les 14 IMUs dans la salle de sport de l'ÉTS

8.1.1 Tableaux synthétique des développements réalisés lors de la Phase 3

Tableau 8.1 Synthèse sur la conception du micro-logiciel des IMU

Synthèse sur la conception du micro-logiciel des IMU		
Objectif : Développer un micro-logiciel permettant de répondre aux spécifications du projet en offrant :		
- Système de communication inter- <i>IMU</i>	- Système de communication <i>IMU / GUI</i>	
- Récupération de données inertielles	- Récupération des données de température	
- Faible consommation énergétique	- Utilisation adéquate des ressources	
- Interaction de l'utilisateur avec les <i>IMU</i>	- Grande robustesse du système	
Spécificités	Phase 2	Phase 3
Liaison SPI		
<i>Mode de transmission</i>	Bidirectionnel avec DMA	Bidirectionnel avec DMA
<i>Bande passante</i>	281 250 Hz	1 125 500 Hz
<i>Temps d'initialisation</i>	12 secondes	0,4 secondes
<i>Connexion à chaud</i>	NON	OUI
<i>Vérification de l'intégrité</i>	NON	OUI
<i>Codage / Repères</i>	NON	En-tête et caractères de contrôle
Liaison UART		
<i>Mode de transmission</i>	Unidirectionnel	Bidirectionnel avec DMA Tx et INT Rx
<i>Bande passante</i>	115 200 Hz	460 800 Hz
<i>Contrôle de flux</i>	Désactivé	Activé RTS/CTS
<i>Vérification de l'intégrité</i>	NON	OUI (iPad)
<i>Codage / Repères</i>	Codage du caractère de fin de trame	En-tête et codage du caractère de fin de trame
Récupération des données		
<i>Type de données</i>	Données calculées uniquement	Données brutes et données calculées
<i>Capteurs</i>	Gyroscopique et géomagnétique	Gyroscopique, géomagnétique et température interne
Consommation énergétique		
<i>Pour un IMU</i>	341 mWh	244 mWh
<i>Pour le système complet</i>	4,905 Wh (14 IMU + WiFly)	3,589 Wh (14 IMU + WiFly)
Le microcontrôleur		
<i>Temps du processus</i>	> 50 Hz	< 50 Hz
<i>Espace mémoire</i>	Espace limité	Espace disponible supplémentaire
<i>Visuel de la calibration</i>	OUI (LED)	OUI (iPad)
<i>Interaction utilisateurs</i>	NON	OUI
<i>Gestion des exceptions</i>	NON	OUI
Synthèse des améliorations		
<ul style="list-style-type: none"> - Au niveau de la communication SPI, on a dorénavant un système dont le temps d'initialisation est faible et permet de ce fait un temps de démarrage du système rapide. On peut également changer d'IMUs à chaud (durant le fonctionnement du système et la majorité des exceptions sont gérées). - La mise en place d'en-tête et de caractères de contrôle permet de connaître les erreurs de communication, ce qui est indispensable lorsque l'on souhaite transmettre les données brutes. - La mise en place d'une bande passante élevée pour la communication UART permet la transmission des données brutes des capteurs inertiels et de la température interne. - La réduction de la consommation aide à prévenir les erreurs de communication causées par l'horloge de la communication SPI. - L'espace mémoire supplémentaire permet de prévenir les erreurs au niveau des algorithmes et d'en utiliser des plus performants. - Le temps du processus global rectifié permet de prévenir les erreurs au niveau du système de communication et d'améliorer l'efficacité des algorithmes inertiels. - La mise en place de communication bidirectionnelle permet l'interaction utilisateur au niveau des IMU, ce qui permet d'interagir sur le type de données à envoyer et de redémarrer manuellement les microcontrôleurs en cas d'exceptions non gérées du système de communication. 		

Tableau 8.2 Synthèse sur le développement du système de communication

Synthèse sur le développement du système de communication		
Objectif : Développer un système de communication robuste, fiable permettant la transmission des données brutes des capteurs inertiels et du capteur de température interne. Il est important de prendre en compte que les erreurs dans la transmission sont néfastes par la suite puisque ces données sont utilisées au sein d'algorithmes inertiels. La fréquence de transmission doit être également constante pour que les algorithmes fonctionnent efficacement. Les implémentations du système de communication sont réparties au niveau du microcontrôleur des IMU, du GUI (l'iPad), de l'architecture matérielle et finalement du module sans-fil WiFly.		
Spécificités	Phase 2	Phase 3
<i>Liaison SPI « IMU depuis/vers IMU »</i>		
<i>Mode de transmission</i>	Bidirectionnel	Bidirectionnel
<i>Taille de trame</i>	96 bits	304 bits
<i>Taux de transmission</i>	67 200 bits/s	212 800 bits/s
<i>Connexion à chaud</i>	NON	OUI
<i>Vérification de l'intégrité</i>	NON	OUI
<i>Codage / Repères</i>	NON	En-tête et caractères de contrôle
<i>Liaison UART « IMU depuis/vers WiFly »</i>		
<i>Mode de transmission</i>	Unidirectionnel	Bidirectionnel
<i>Bande passante</i>	115 200 Hz	460 800 Hz *
<i>Taille de la trame</i>	1568 bits	3592 bits
<i>Taux de transmission</i>	78 400 bits/s	179 600 bits/s **
<i>Contrôle de flux</i>	Désactivé	Activé RTS/CTS ***
<i>Vérification de l'intégrité</i>	NON	OUI (iPad)
<i>Codage / Repères</i>	Codage du caractère de fin de trame	En-tête et codage du caractère de fin de trame
<i>Liaison sans-fil « WiFly depuis/vers iPad »</i>		
<i>Protocole</i>	UDP	TCP
<i>Mode de transmission</i>	Unidirectionnel (Multidiffusion)	Bidirectionnel
<i>Périphérique intermédiaire</i>	NON (Ad hoc)	Point d'accès (AP) ****
<i>Sécurité</i>	Ouvert	WPA / WPA2 – PSK
<i>Liaison sans-fil « iPad vers iPads »</i>		
<i>Protocole</i>	-	UDP
<i>Mode de transmission</i>	-	Unidirectionnel (Multidiffusion)
<i>Périphérique intermédiaire</i>	-	Point d'accès (AP)
<i>Informations supplémentaires</i>		
<i>Architecture matérielle – SPI</i>	Non blindé, série (horloge et données)	Blindé, série (données), étoile (horloge)
<i>Tâche supplémentaire Rx – iPad</i>	NON	OUI
<i>Liaison entre les communications</i>	NON	OUI
Synthèse des améliorations		
<ul style="list-style-type: none"> - La mise en place d'une communication bidirectionnelle au niveau de la liaison UART et de la communication sans-fil permet à l'utilisateur d'interagir avec les IMUs (de façon globale ou individuelle). - L'augmentation de la bande passante de la liaison UART permet la transmission des données brutes des capteurs inertiels et du capteur de température. - Le codage du caractère de fin de trame a été rectifié afin d'être valide et utilisable. - Le passage du protocole UDP à TCP permet de réduire les erreurs de transmission (l'absence de données). - Le réseau sans-fil sécurisé permet de limiter les perturbations causées par l'environnement extérieur. - Le blindage de la liaison SPI permet de lutter contre la perte d'informations causées par les perturbations de l'horloge mais aussi contre les erreurs sur les données. - Le multitâche de l'iPad est nécessaire afin que la communication ne soit pas influencées par l'interaction de l'utilisateur. 		
<p>* Théoriquement le module WiFly pourrait supporter une liaison UART à 1 Mbits/s, cependant en pratique ce n'est pas le cas et il n'est pas possible d'augmenter la bande passante d'avantage.</p> <p>** Malgré le fait que le débit utilisé est plus faible que la bande passante, celui-ci ne pourrait pas augmenter considérablement dû aux limites posées par le module WiFly.</p> <p>*** L'activation du contrôle de flux est nécessaire pour lier les différentes communications et ainsi corriger les erreurs de transmission (pour rappel dans la version précédente, seulement 40% des données parvenaient à l'iPad).</p> <p>**** Le point d'accès était nécessaire puisque le micro-logiciel du WiFly présentant cette fonctionnalité est en version beta et n'est pas réellement fonctionnel et le réseau Ad hoc ne permet pas d'utiliser le protocole TCP.</p>		

Tableau 8.3 Synthèse sur la réalisation de l'architecture matérielle

Synthèse sur la réalisation de l'architecture matérielle		
Objectif : Réaliser une architecture matérielle robuste ne venant pas perturber les mouvements du corps humain et pouvant supporter le système de communication devant être mis en place. Cette architecture matérielle est composée de 14 IMUs, chacun d'entre eux réparti dans un boîtier, il est par la suite câblé et le tout est incorporé dans un vêtement. L'IMU dit « maître » dispose d'un module générant une communication sans-fil. Une batterie est également présente.		
Spécificités	Phase 2	Phase 3
<i>Le câblage</i>		
<i>Flexibilité</i>	NON *	OUI (spiralé)
<i>Blindage</i>	NON **	OUI (2 à 2)
<i>Modularité</i>	NON	OUI
<i>Nombre de fils</i>	4	8
<i>Longueur</i>	> 10 m	5.6 m
<i>Le boîtier</i>		
<i>Connecteurs</i>	NON	OUI
<i>Modularité</i>	NON ***	OUI
<i>Alimentation</i>	GPIO du SPI	Batterie
<i>Debug après fermeture</i>	NON	OUI
<i>L'architecture</i>		
<i>Batterie</i>	OUI	OUI
<i>Répartition de l'alimentation</i>	Série	Étoilée
<i>Répartition de l'horloge du SPI</i>	Série	Étoilée
<i>Transmission des données SPI</i>	Série	Série
Synthèse des améliorations		
<ul style="list-style-type: none"> - Le câblage spiralé a permis d'augmenter la flexibilité et de ne pas nuire aux mouvements de l'utilisateur. Il dispose d'un blindage équivalent à ceux effectué sur les câbles Ethernet et contient plus de fils ce qui permet de réduire considérablement le temps de parcours des données (les câbles entre les 2 bras et 2 jambes sont supprimés). - Les connecteurs sur chaque boîtier permettent de ne plus avoir le problème des câbles se cassant à cause de l'usage prolongé du vêtement. De plus, en cas de panne survenant sur un IMU, il est possible de le changer facilement sans devoir modifier tout un sous-système. - L'alimentation est corrigée et permet ainsi l'utilisation de l'ADC correctement et donc du capteur de température. - Un connecteur est présent sur le boîtier une fois fermé, ce qui permet la reconfiguration du microcontrôleur en cas de mise à jour sans devoir le démonter. Ceci permet surtout de ne pas prendre de risque en déplaçant l'IMU dans le boîtier par la procédure. - L'alimentation étoilée permet de réduire l'intensité du courant présent dans chaque sous système à 0.2 A au lieu des 1.3 A circulant dans le système précédent. - La répartition de l'horloge de façon étoilée permet aussi d'amoindrir l'effet de l'atténuation du signal. 		
<p>* Les câbles dans la précédente version étaient souples mais ils restent adaptés uniquement à l'individu sur qui les mesures ont été effectuées. De plus, certains des IMUs peuvent avoir des positions différentes sur le vêtement et ce type de câble ne permet pas de répondre à cette possibilité.</p>		
<p>** Un blindage métallique était présent sur les câbles utilisés mais celui-ci n'était pas relié à la masse.</p>		
<p>*** Les boîtiers ne disposant pas de connecteurs, plusieurs IMUs étaient reliés ensemble.</p>		

Tableau 8.4 Synthèse sur la conception du logiciel pour iPad

Synthèse sur la conception du logiciel de l'iPad		
Objectif : Cette interface graphique utilisateur doit fonctionner sous trois modes différents permettant ainsi l'étude, l'analyse et la comparaison des différents algorithmes inertiels utilisés tout en permettant la capture de mouvements et la navigation à l'intérieur des bâtiments. De nombreuses interfaces doivent être ainsi développées afin de répondre aux spécifications du projet. Il est important de prendre en compte les limites du matériel lors de la conception.		
Spécificités	Phase 2	Phase 3
Interface utilisateurs <i>A propos du projet ibNav</i> <i>Visualisation / Modification du fichier BVH</i> <i>Visualisation de la carte et interaction utilisateur</i> <i>Visualisation des données sous forme de graphique</i> <i>Modification de la configuration</i> <i>Affichage des mouvements en 3D</i> <i>Localisation à l'intérieur des bâtiments</i> <i>Visualisation des données calculées et l'orientation locale</i> <i>Traitement sur les algorithmes</i>	NON Visualisation NON NON NON OUI NON NON NON	OUI Les deux OUI OUI OUI OUI OUI** OUI** OUI**
Fonctionnalités <i>Types de navigation</i> <i>Modes de traitement – Temps réel avec gestion des algorithmes</i> <i>Modes de traitement – Temps réel sans gestion des algorithmes</i> <i>Modes de traitement – Post traitement</i> <i>Multitâche</i> <i>Intégration d'une base de données</i>	Corporelle NON OUI NON * NON NON	Corporelle et intérieur OUI OUI OUI OUI OUI
Synthèse des améliorations		
<ul style="list-style-type: none"> - Une interface reprenant un résumé du projet a été implémentée, celle-ci sera très utile dans le futur lors de démonstrations. - Une carte du laboratoire de recherche a été incorporée avec les fonctions de déplacement et de zoom tactile. - Les données brutes des capteurs inertiels de chaque IMU ainsi que du capteur de température sont maintenant visualisable sous forme de graphique avec des fonctions de zoom et de moyenne. Cette interface peut également servir à afficher d'autres données sous forme de graphique. - Le GUI est prévu pour fonctionner sous trois modes de traitement différents répondant ainsi aux spécifications du projet. - Le multitâche a été incorporé permettant de faire fonctionner correctement le système de communication et l'application des algorithmes inertiels. - Une base de données sous SQL a été créée afin de gérer plus efficacement les fonctionnalités mises en place dans l'iPad et les fichiers. Le temps de traitement est ainsi fortement réduit. - L'interface pour l'affichage des mouvements du corps en 3D est en train d'être refaite sous Unity3D afin d'y inclure un corps humain sous forme de solides, un sol quadrillé et l'intégration tactile du zoom et du déplacement dans l'environnement 3D. - L'interface permettant l'étude, la modification et la comparaison des algorithmes est en cours de développement. <p>* Le post-traitement n'était pas vraiment implémenté dans la phase 2 du projet puisqu'on pouvait uniquement revoir un enregistrement de données calculées. On était de ce fait dans l'impossibilité de faire du traitement sur les données.</p> <p>** Ces interfaces ont été créées mais elles ne sont pas finies due à la transition technologique. Avant de les finir il faudrait premièrement passer l'application sur la nouvelle version d'iOS et de s'assurer de la fonctionnalité des différents modules.</p>		

Tableau 8.5 Synthèse sur l'implémentation des algorithmes inertiels

Synthèse sur l'implémentation des algorithmes inertiels		
<p>Objectif : L'implémentation des algorithmes inertiels doit permettre dans un premier temps de rectifier les erreurs déterministes et stochastique modélisables à l'aide d'une méthode de calibration et l'utilisation d'un robot de calibration. Dans un deuxième temps, deux méthodes de navigation différentes doivent être utilisées afin d'effectuer la capture de mouvements et la navigation pédestre à l'intérieur des bâtiments. Ces méthodes intégreront différents modèles de connaissance afin de rendre plus performants les algorithmes inertiels utilisés.</p>		
Spécificités	Phase 2	Phase 3
<p><i>Calibration des capteurs inertiels</i></p> <p><i>Correction du biais</i></p> <p><i>Correction du facteur d'échelle</i></p> <p><i>Correction du mauvais alignement des axes</i></p> <p><i>Prise en compte de la température</i></p> <p><i>Calibration continue pour la température et les erreurs de fer doux</i></p> <p><i>Modèle de connaissance – Détection des états stationnaires</i></p> <p><i>Calcul du bruit blanc par la méthode « Allan Variance »</i></p> <p><i>Calcul de la norme du champ magnétique</i></p> <p><i>Calcul du vecteur de gravité théorique</i></p> <p><i>Mise en place du filtre EKF</i></p>	<p>Géomagnétique et gyroscopique</p> <p>NON</p> <p>NON</p> <p>NON</p> <p>NON</p> <p>OUI</p> <p>OUI</p> <p>NON</p> <p>NON</p> <p>OUI</p>	<p>Géomagnétique et gyroscopique</p> <p>Géomagnétique</p> <p>Géomagnétique</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>
<p><i>Algorithmes de navigation – Capture de mouvements*</i></p> <p><i>Modèle AHRS</i></p> <p><i>Conversion au format BVH</i></p> <p><i>Modèle de connaissance ZARU</i></p> <p><i>Modèle de connaissance – Détection des états stationnaires</i></p> <p><i>Mise en place du filtre EKF</i></p> <p><i>Mise en place de la limitation des angles des articulations</i></p>	<p>OUI</p> <p>OUI</p> <p>NON</p> <p>NON</p> <p>OUI</p> <p>NON</p>	<p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI**</p>
<p><i>Algorithmes de navigation – Navigation pédestre intérieur</i></p> <p><i>Modèle INS</i></p> <p><i>Modèle de connaissance – Détection des états stationnaires</i></p> <p><i>Modèle de connaissance – HDR</i></p> <p><i>Modèle de connaissance – ZUPT</i></p> <p><i>Mise en place du filtre EKF</i></p> <p><i>Rectification de la position par BIM</i></p>	<p>NON</p> <p>NON</p> <p>NON</p> <p>NON</p> <p>NON</p> <p>NON</p>	<p>OUI</p> <p>OUI</p> <p>OUI**</p> <p>OUI**</p> <p>OUI**</p> <p>OUI**</p> <p>OUI**</p>
<p>Synthèse des améliorations</p> <p>- Calibration initiale : L'algorithme développé permet de corriger le facteur d'échelle et le mauvais alignement des axes des accéléromètres et des magnétomètres.</p> <p>- Calibration continue : L'algorithme développé utilise un modèle simple afin de corriger la dérive provoquée par la température et permet au magnétomètre de pouvoir être utilisé lorsque celui-ci subit une petite perturbation.</p> <p>- Navigation corporelle : L'algorithme AHRS + ZARU + EKF a été implémenté sur le système ibNav et permet d'avoir une bonne attitude des IMUs, cependant il reste sensible aux fortes dynamiques et lorsque les états stationnaires sont peu nombreux.</p> <p>* La phase 2 contenait un algorithme AHRS avec EKF mais celui-ci présentait des résultats insatisfaisants du fait d'erreurs présents dans l'algorithme.</p> <p>** Ces méthodes ont été développées sur Matlab et/ou Visual Studio mais elles n'ont pas encore été testées sur le système ibNav.</p>		

8.1.2 Faiblesses et forces

Le système ibNav a été testé en laboratoire, dans la salle de sport de l'ÉTS et lors de plusieurs événements (séminaires, portes ouvertes, etc). Lors de ces différents scénarios, nous avons étudié les faiblesses et les forces du prototype au niveau de l'architecture matérielle et du logiciel de l'iPad :

Tableau 8.6 Faiblesses et forces du projet ibNav

	Résultats	Commentaires
Architecture matérielle		
Habillement	Moyen	Les capteurs inertiels sont d'une dimension trop importante ce qui rend le vêtement difficile à mettre.
Démarrage du système	Très satisfaisant	
Utilité de l'afficheur	Excellent	Il permet d'avoir la visibilité s'il y a eu un problème système ou de connectivité.
Mouvements	Très satisfaisant	Aucune gêne des câbles étant placés vers le dos.
Fiabilité du système	Très satisfaisant	Nous n'avons observé aucun bug majeur.
Autonomie	Satisfaisant	Environ 2 heures et 30 minutes d'autonomie par batterie.
Robustesse du réseau sans-fil	Moyen	Lors de certaines présentations où de nombreux système sans-fil étaient présents, le système éprouvait des difficultés à garder la connexion.
Logiciel iPad		
Ergonomie de l'application	Excellent	
Fiabilité de l'application	Très satisfaisant	Quelques bugs restent à régler.
Affichage des données brutes	Excellent	
Affichage de l'attitude de chaque capteur	Satisfaisant	Il faudrait idéalement travailler avec des données brutes à 200 Hz afin d'obtenir de meilleurs résultats.
Affichage des mouvements en 3D	Moyen	Le passage sur Unity3D est en cours
Partage des données	Excellent	Partage par email et/ou dropbox

8.2 Transfert technologique vers VTADS, MEDOC et d'autres projets

Le transfert technologique du projet ibNav a été réalisé sur les projets suivants :

- VTADS, un projet destiné à l'analyse de la conduite et à la détection d'accidents. Pour ce projet, nous avons développé le Micro-iBB reprenant en partie l'architecture d'ibNav, le micro-logiciel, une partie de l'architecture matérielle ainsi que la structure du logiciel pour iPad.

Site internet : <http://lassena.etsmtl.ca/spip.php?rubrique29#34>;

- MEDOC, un projet avec une collaboration internationale afin d'analyser la mobilité des automobilistes. Pour ce projet, nous avons développé une nouvelle version du Micro-iBB.


Site internet : <http://lassena.etsmtl.ca/spip.php?rubrique29#1428>;

- FSAR (L-IMU), un projet dont le système reprend l'IMU du Micro-iBB en y ajoutant un laser pour le positionnement fin en attitude afin de développer des algorithmes avancés en détermination d'attitude.

Site internet : <http://lassena.etsmtl.ca/spip.php?rubrique29#35>;

- TPG, un projet où il fallait étudier les performances d'une centrale inertielle pour un casque de réalité augmentée. À ce titre, les développements faits pour le projet ibNav sur les algorithmes inertiels ont été réutilisés afin de mener cette étude;
- iMetrik Show Case, une application sur iPad servant pour certaines démonstrations de l'un de nos partenaires industriels. Il s'agit d'une simulation d'un véhicule (sous Unity3D), contrôlé par iPad permettant à la fin du scénario de donner un score de réduction d'assurance à l'utilisateur suivant sa conduite et d'autres informations. La structure du logiciel pour iPad d'ibNav a été réutilisée pour cette application.

Tableau 8.7 Logo des différents projets basés sur les développements du projet ibNav

				
VTADS	MEDOC	FSAR	TPG	iMetrik

Les deux prochaines sous-sections illustrent deux de ces exemples de transfert technologique.

8.2.1 Le Micro-iBB, une création à partir d'ibNav

Le Micro-iBB de VTADS a été réalisé de Novembre 2013 à Mai 2014. Il s'agit d'une boîte noire utilisant des capteurs inertiels à faible coût ainsi qu'un récepteur GPS permettant d'obtenir leurs mesures brutes pour une utilisation en post-traitement. Ces mesures sont enregistrées à une fréquence très élevée sur une carte microSD et il est doté d'un système de communication sans-fil. Ce prototype peut tout aussi bien être relié dans un véhicule réel ou sur une voiture radiocommandée professionnelle afin de reproduire des trajectoires de manières automatique (simulation de la conduite et d'accidents). Le Micro-iBB pour MEDOC a été réalisé de Novembre 2014 à Mars 2015, l'architecture est similaire excepté qu'à la place d'inclure un système de communication sans-fil, il est doté d'un lecteur OBD-II afin de lire les informations en provenance de l'ordinateur de bord d'un véhicule automobile.

Afin de développer rapidement ce prototype, nous avons repris un IMU similaire à celui d'ibNav, soit utilisant le même type de Microcontrôleur STM32F103 et des capteurs inertiels équivalents. Le module sans-fil a également été repris (RN-131). L'architecture du micrologiciel est celui développé pour ibNav. La configuration du microcontrôleur, la communication et configuration du module sans-fil ainsi que la communication et configuration de l'accéléromètre/magnétomètre LSM303DLH a été repris (Voir Figure 8.2).

Son interface de contrôle est un logiciel développé sur iPad nommé iVave, ce dernier étant très similaire au logiciel ibNav 2 pour iPad (Voir Figure 8.3). L'architecture du logiciel a été entièrement reprise, elle inclue :

- la mise en place de la base de données SQL;
- la structure des interfaces graphiques;
- la librairie développée permettant d'afficher des courbes en temps réel et en post-traitement;
- le cube en 3D permettant d'afficher l'attitude des IMUs;
- la synchronisation avec l'API Dropbox;
- la gestion entre le temps réel et le post-traitement.

Par la suite, pour répondre aux spécifications du projet, nous avons ajouté un lecteur de carte microSD fonctionnant par SPI, un récepteur GPS (ublox LEA-6T), une batterie Lithium de 2000mAh ainsi qu'une interface USART pour communiquer avec l'autopilote des voitures radiocommandées professionnelles. L'algorithme AHRS pour la détermination de l'attitude a également été modifié à partir d'un autre algorithme ouvert (utilisant la méthodologie du gradient descendant) afin d'offrir de meilleures performances sur un système embarquée. On a également ajouté la possibilité de connecter le système sur un ordinateur par USB et qu'il soit détecté comme un lecteur de masse de données. Concernant l'iPad, nous avons ajouté l'interface permettant de contrôler la voiture manuellement et automatiquement en créant des scénarios.

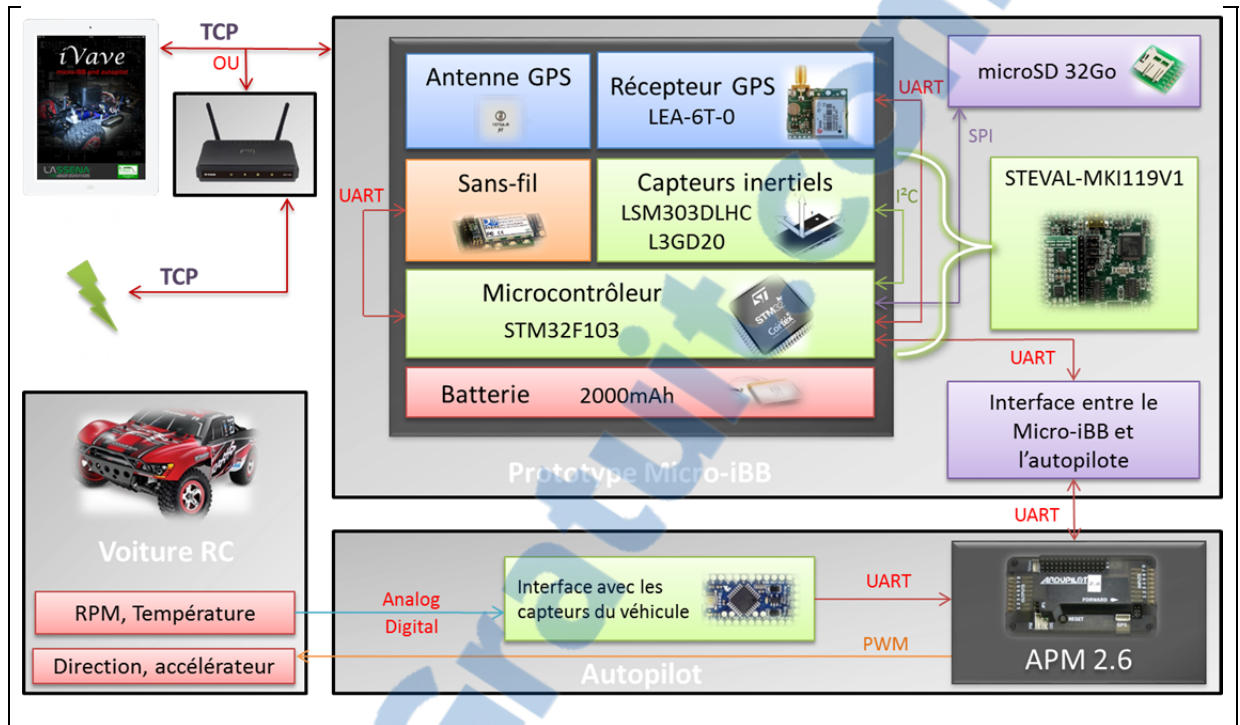


Figure 8.2 Architecture globale du Micro-iBB VTADS

Concernant le projet MEDOC, outre le lecteur OBD-II, nous avons ajouté la possibilité de mettre à jour le micro-logiciel par USB, la mise en veille automatique au bout de 10 minutes en l'absence de détection d'accélération, un bouton d'identification d'incident et la possibilité de rester 6 mois dans un véhicule automobile sans interruption.



Menu de l'interface principale



Visualisation des données brutes et calculés des capteurs inertiels



Attitude en 3D de la plateforme



Réglage de la sensibilité des capteurs inertiels

Figure 8.3 Interfaces sur iPad du logiciel iVave (Partie 1)



Contrôle manuel de la voiture RC

Création de scénarios automatiques

Figure 8.4 Interfaces sur iPad du logiciel iVave (Partie 2)

8.2.2 Le logiciel iOS ibNav 2 vers iMetrik Show Case

Nous avons premièrement investigué le choix d'un moteur 3D fonctionnant sur iOS afin d'améliorer la représentation des mouvements du corps humain. Nous avons choisis Unity3D et une première application a commencé à être développée. Cependant à partir Mars 2014, un de nos partenaires industriels souhaitait que l'on utilise notre expérience d'unity3D afin de leur créer un logiciel pour leurs forums à des évènements. Nous avons donc créé une première interface sur iPad reprenant toujours l'architecture d'ibNav. Cette interface (figure 8.6) permet de poser des questions à l'utilisateur afin qu'il entre son nom, son email, le nombre de kilométrage qu'il effectue par année ainsi que s'il préfère conduire le jour ou la nuit. Une fois ces informations entrées, l'utilisateur peut conduire la voiture (accélération, décélération, guidage et freinage). Cette application est reliée par TCP à une application fonctionnant sur Unity3D sur un PC, l'utilisateur peut alors conduire la voiture à partir de l'iPad (ce dernier étant attaché à un volant). A la fin du scénario de conduite, l'utilisateur

obtient un score qui s'affiche sur l'iPad et celui-ci lui est envoyé directement par courriel. L'iPad retient toutes les informations sur les utilisateurs qui ont utilisés l'application et ces informations peuvent être envoyées par courriel en format CSV à la personne responsable du marketing.



Première interface iPad

Deuxième interface iPad

Interface de conduite de la voiture

Début du scénario de conduite sur unity3D (PC)

Interface avec le résultat de la conduite

Fin du scénario de conduite sur unity3D (PC)

Figure 8.5 Interfaces sur iPad et PC du logiciel iMetrik Show Case



Figure 8.6 Photographie du stand incluant l'iMetrik Show Case (www.imetric.com)

CONCLUSION

Le projet ibNav présenté dans ce mémoire a pris naissance grâce à l'évolution de la technologie MEMS faisant des IMUs à faible coût une solution tout à fait intéressante à étudier pour la capture de mouvements et la navigation à l'intérieur des bâtiments et dont de nombreux défis ont dû être relevés. Non pas uniquement au niveau de la conception des algorithmes mais essentiellement par son défis technologique requérant d'avoir des connaissances dans des domaines variés afin de développer un micro-logiciel, un système de communication, une architecture matérielle, une interface graphique ainsi que des algorithmes inertiels performants. L'atteinte de ces objectifs n'a pas été une chose aisée due au faible coût du projet, notamment dans le choix des technologies utilisées faisant que les différents sous-systèmes manquaient facilement de fiabilité. La conception devait être tout aussi rigoureuse qu'attentive.

Suite à l'état de l'art, des stratégies ont été adoptées au niveau de la calibration des capteurs inertiels et des algorithmes de navigation. Les solutions adoptées devaient respecter le faible coût du système et l'utilisation des IMUs de façon autonome, sans l'utilisation de sources externes. Il fallait également que les méthodes choisies puissent être réalisées dans la durée d'une maîtrise en prenant en compte le développement du système complet. De ce fait, des méthodes traditionnelles utilisant un filtre de Kalman ont été choisies. Concernant le choix des IMUs et le développement de l'application sur iPad, ils ont été imposés au début du projet. Au total, 5 objectifs ont été réalisés au cours de la maîtrise :

Premièrement, un algorithme de calibration initiale a été développé afin de corriger autant que possible le biais, le facteur d'échelle, le mauvais alignement et les erreurs de fer dur des capteurs inertiels. Nous avons vu que cette calibration est nécessaire puisque des erreurs importantes sont présentes sur ce type de capteurs à faible coût. Nous avons donc développé un algorithme utilisant la méthode multi-positions combinée à un filtre de Kalman étendu rapide afin de corriger les erreurs déterministes de ces capteurs. L'avantage de cette méthode est qu'elle ne nécessite pas la réalisation de positions précises et elle peut fonctionner en

temps réel sur des systèmes embarqués. En restant sur le même objectif, un algorithme de calibration continue a également été développé, celui-ci prend en compte la température du système pour corriger la dérive associée et corriger également les erreurs de fer doux des magnétomètres afin que ce dernier offre une meilleure correction au sein de l'algorithme AHRS et INS. Finalement, nous avons utilisé la variance d'Allan afin d'en déduire le bruit blanc sur chacun des axes des capteurs inertiels qui seront utilisées dans les algorithmes de navigation. Par la suite, concernant la réalisation des algorithmes de navigation, à ce titre un algorithme AHRS performant doté d'une méthode d'alignement initial a été développé et permet d'obtenir en étant combiné à la calibration, une erreur d'attitude maximum de 2° et peut de ce fait être utilisé dans notre application. Un algorithme INS pour la navigation pédestre a également été développé mais celui-ci n'a pas été testé sur le système réel et nécessiterait encore des améliorations afin d'être plus performant.

Les trois objectifs suivants concernaient la réalisation du prototype en lui-même, c'est-à-dire l'architecture matérielle et la programmation embarquée. Deux micro-logiciels ont donc été développés pour les STM32F1, un pour l'IMU « maître » et l'autre pour les IMUs « esclaves » afin d'y inclure les fonctionnalités demandés. Les différents systèmes de communication ont été également mis en place, allant de la communication inter-IMUs à la communication sans-fil avec l'iPad ou entre iPads. Finalement, un prototype de vêtement intelligent a été conçu permettant une certaine flexibilité. Le dernier objectif concernait le développement du logiciel pour l'iPad, véritable interface graphique utilisateur permettant la visualisation des données brutes et calculées de chaque IMUs ainsi que de contrôler / paramétrer le système ibNav.

Finalement, au cours de la réalisation de ce projet, une demande plus importante s'est faite ressentir sur d'autres projets du laboratoire LASSENA, de ce fait un transfert technologique suivi d'une réadaptation du système a été réalisé à partir de Novembre 2013. La plupart des développements, que ce soit le micro-logiciel, une partie de l'architecture matérielle (microcontrôleurs et capteurs inertiels), le GUI sur iPad ainsi qu'une partie des algorithmes inertiels ont été réadaptés et réutilisés. On citera essentiellement le projet VTADS avec le Micro-

iBB. Le projet ibNav n'a pas été abandonné pour autant et se poursuivra au cours de l'année 2015.

RECOMMANDATIONS

Le projet ibNav reste à l'heure actuel à l'état d'un prototype, diverses améliorations seront nécessaires afin de le rendre commercial. Nous avons vu dans ce mémoire que les capteurs inertiels à faible coût peuvent être utilisés pour la capture de mouvements mais il semble difficile d'avoir de bonnes performances en navigation intérieure sans l'utilisation de sources externes. Au niveau de l'architecture matérielle, les choix ont été fait essentiellement afin de limiter le coût du projet mais certaines améliorations seraient nécessaires afin d'améliorer la robustesse du projet, la facilité d'utilisation ainsi que les performances. Finalement puisque les besoins du laboratoire LASSENA ont changés au cours de la maîtrise, certaines fonctionnalités doivent encore être implémentées mais je recommande premièrement de suivre la méthodologie suivante avant de poursuivre les implémentations :

Premièrement, au niveau des algorithmes inertiels, je recommande de réaliser les deux plateformes de calibration présentés dans ce mémoire (Figure 4.23 et 5.2) afin d'offrir une meilleure calibration des accéléromètres et des magnétomètres. Ensuite, il serait intéressant maintenant que le laboratoire a en sa possession une chambre thermique et plaque tournante de les utiliser afin d'améliorer la calibration des gyroscopes et la calibration continue présentée dans ce mémoire. Il serait également préférable de continuer le développement de tous les algorithmes sous Matlab afin de pouvoir les étalonner de façon plus efficace (et de les convertir en C par la suite). La calibration continue des magnétomètres pourrait également être améliorée en utilisant une méthode d'apprentissage machine au lieu d'une méthode de filtrage. En effet, la méthode de filtrage à ses limites car elle tend à vouloir corriger les valeurs du magnétomètre suivant la constante de référence mais une méthode d'apprentissage machine pourrait plus facilement avoir connaissance de la correction à effectuer suivant le type d'environnement qu'elle reconnaît. L'inconvénient est qu'il faudrait réaliser de nombreux enregistrements dans des environnements différents et cela sur plusieurs jours. Pour la navigation à l'intérieur des bâtiments, il faudrait réaliser d'avantages de scénarios afin de calibrer les algorithmes et d'en déduire si ceux-ci pourraient être utilisés

tel quel. C'est cependant évident qu'au fil du temps et sans méthode permettant de rectifier la solution que celle-ci dérivera de manière importante. Un baromètre et un capteur ultrasonique pourraient être utiles.

Concernant l'architecture matérielle, je recommande de ne plus utiliser les plateformes de développement STEVAL-MKI062V2 pour les raisons suivantes :

- la plateforme est prévue pour étudier le comportement des capteurs inertiels mais n'a pas été conçu pour l'usage que l'on en fait. On est de ce fait limité par les I/O, l'encombrement et le système d'alimentation n'est pas optimal pour faire fonctionner les CAN. De plus à l'heure actuelle, on peut obtenir des capteurs inertiels de type MEMS à faible coût plus performants et tout intégrés dans une même puce (par exemple le LSM9DS0 de STMicroelectronics);
- il n'est pas possible avec l'architecture actuelle d'avoir un taux de récupération des données des capteurs de plus de 50Hz, il serait cependant fortement recommandé d'avoir au moins 200Hz;
- les données de température ne viennent pas directement des capteurs inertiels mais du capteur de température interne du microcontrôleur;
- la consommation énergétique pourrait être d'avantage réduite.

Outre les inconvénients, il serait préférable de créer nos propres PCB, on pourrait ainsi avoir les IMUs « esclaves » dotés d'un microcontrôleur STM32F4 (comprenant un DSP) avec le capteur inertiel LSM9DS1. La performance du microcontrôleur nous permettrait de faire fonctionner l'algorithme AHRS et INS à 200 Hz et d'ajouter des méthodes d'apprentissage machine. On pourrait également pour chaque branche contenant 3 IMUs ajouter un microcontrôleur STM32W permettant d'effectuer une liaison sans fil vers l'IMU « maître » par le protocole IEEE 802.15.4. Concernant l'IMU « maître », on pourrait lui ajouter un écran tactile au lieu de l'écran LCD permettant de gérer les fonctionnalités de base et ainsi ne pas avoir besoin de l'iPad pour faire fonctionner le système. Cet IMU pourrait être doté d'une carte microSD fonctionnant par SDIO afin de permettre l'enregistrement de toutes les données (brutes et calculées) à une fréquence élevée. Le circuit énergétique pourrait être

refait afin d'offrir un meilleur rendement. On pourrait également ajouter un module sans-fil Ethernet afin d'améliorer la vitesse de communication avec l'iPad.

Afin de donner une idée du coût, ce serait approximativement 20 \$ par IMU « esclave » et de 75 \$ pour l'IMU « maître ». Bien entendu, l'encombrement de chaque IMU serait nettement réduit, environ 15 mm x 15 mm pour chaque IMU « esclave » au lieu du 45 mm x 45 mm actuel.

Concernant l'application développée sur iPad, il faudrait premièrement faire passer le code existant sur la dernière version d'iOS, soit iOS 8.1. Par la suite, je recommande d'utiliser Unity3D pour la représentation des mouvements du corps, en l'intégrant dans l'application existante. Finalement, il faudra ajouter les fonctionnalités restantes.

Je recommanderais une équipe de 4 personnes afin de poursuivre le développement du projet, comprenant :

- une personne en doctorat dont le sujet serait de réaliser un algorithme de navigation à l'intérieur des bâtiments performant;
- un stagiaire provenant d'une école de jeux-vidéo, travaillant sur le développement et l'intégration d'Unity3D dans l'application existante sur iPad;
- un stagiaire en système embarqué afin de réaliser la nouvelle architecture matérielle du projet ibNav;
- une personne en maîtrise continuant le développement des algorithmes de navigation et de calibration du projet ibNav afin de les améliorer et de les rendre plus performants.

Finalement, on s'est rendu compte (pour avoir été contacté à plusieurs reprises) que ce projet intéressait particulièrement des personnes dans le domaine de la santé, essentiellement pour la réhabilitation de personnes ayant subis un traumatisme. Il serait donc tout à fait intéressant de faire un partenariat et d'adapter le projet pour qu'il puisse être utilisé en ce sens.

ANNEXE I

RENDU À LA FIN DE LA PHASE 1 DU PROJET IBNAV

Les différents développements réalisés à la fin de la Phase 1 du projet ibNav, sont les suivants :

Assemblage et calibration de la main robotique Lynxmotion AL5D et mise en places des IMUs, création de l'interface GUI en VB.net et de scénarios de calibration automatique.

- mise en place des protocoles de communication filaire SPI avec la capacité de détecter automatiquement le nombre d'IMUs connectés;
- conception et mise en place de l'architecture matérielle et analyse des bandes passante et de la consommation énergétique;
- création de la représentation 3D pour l'architecture corps humain;
- création du logiciel pour l'utilisation des algorithmes;
- analyse du bruit effectuée sur les IMUs;
- intégration des algorithmes inertiels et mise en place du fonctionnement en temps réel.

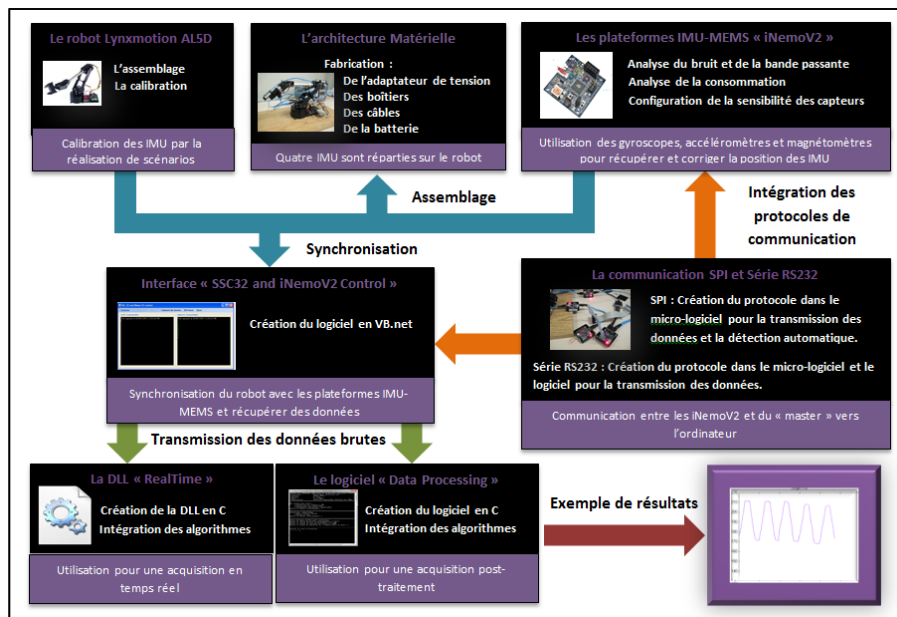


Figure-A-I Rendu à la fin de la phase 1 de développement
du projet ibNav
Tirée de Mixte (2011)

ANNEXE II

RAISONS DE L'IMPLÉMENTATION DE LA PHASE 3

Les problèmes provenaient d'une combinaison de facteurs sur le système complet, suite à des erreurs de développement ou à des choix de conception des étudiants travaillant sur le projet. Beaucoup de corrections ont été appliquées mais le système ne répondait toujours pas à nos attentes. Les étudiants ayant travaillé sur la phase 2 n'étant plus sur le projet et les informations relatant leurs travaux étant manquantes, la recherche des différents problèmes prend trop de temps et ne permet pas au projet d'évoluer. Pour avancer, il n'était plus possible de continuer ainsi, les divers problèmes étant répartis sur le micro-logiciel, le système de communication, l'architecture hardware, le logiciel pour l'iPad et les algorithmes inertiels (calibration et navigation). Pour cette raison, il fallait repartir sur une base d'un système simple, stable et robuste mais présentant tout de même toutes les fonctionnalités nécessaires aux besoins du projet.

Le micro-logiciel :

Lors de la conception du nouveau système, j'ai décidé de créer mon propre micro-logiciel en lui apportant toutes les fonctionnalités nécessaires recensées pour les besoins du projet afin de le finaliser. La raison est que le micro-logiciel existant à de nombreuses fonctionnalités inutiles dans notre projet, telles que l'USB et le code source permettant l'interactivité avec leur interface PC. En enlevant ces fonctionnalités, le système d'exploitation devient un inconvénient et non un avantage. Les raisons sont les suivantes :

- on n'a plus besoin des drivers qu'il inclut;
- il prend une place mémoire supplémentaire;
- une gestion des ressources supplémentaire;
- et par conséquence une hausse de consommation du système.

De plus, par sa complexité, le micro-logiciel présente plusieurs erreurs connues qui pourraient venir perturber le bon fonctionnement de notre système (dans le cas d'une exception interrompant le processus).

Ayant une formation donnée par CMC en collaboration avec ST et l'université de McGill, j'ai eu l'opportunité d'obtenir les informations qui me faisaient défaut afin de me permettre de créer mon propre micro-logiciel avec cet IMU. Cette étape fut très motivante et m'a permis d'acquérir de l'expérience sur la programmation ARM par l'intégration de mon propre système. De plus le code source créé pourra facilement être adapté sur un autre ARM avec d'autres capteurs inertiels, il est donc devenu modulaire.

Le système de communication :

Dans un premier temps, des corrections ont été apportés sur la phase 2, soit :

- modification de la méthode de codage de la liaison UART afin de permettre un nombre quelconque d'IMUs connectés au système;
- modification de la bande passante ainsi que de l'architecture matérielle afin de permettre jusqu'à 14 IMUs connectés au système;
- modification du caractère de fin de trame afin que celui-ci soit fonctionnel;
- modification de la trame SPI afin de recueillir les bonnes données.

Après ces diverses modifications, il était possible de faire fonctionner 14 IMU mais des erreurs dans la trame de communication étaient quand même présentes. De plus la bande passante de la liaison SPI, UART est insuffisante si l'on souhaite récupérer les données brutes ainsi qu'appliquer des méthodes de vérification de l'intégrité des différentes trames. La liaison UDP ne permet pas une réception des données tendant à 100%. Les communications ne sont pas bidirectionnelles et il faudra ses fonctionnalités pour la suite du projet. A ce stade, la robustesse de cette communication étant inadéquate, elle ne pouvait pas être gardé tel quels et des changements au niveau des différents matériels utilisant le système de communication devaient être faite pour continuer le projet.

L'architecture matérielle :

Malgré la souplesse des câbles, ceux-ci ne sont pas flexibles et viennent nuire aux mouvements de l'individu. Certains des IMUs peuvent être également positionnés différemment sur le vêtement et ce type de câble ne permet pas cette possibilité. Un blindage métallique est présent sur les câbles mais celui-ci n'est pas relié à la masse. Plusieurs IMU sont reliés ensemble, il n'y a pas de connecteurs sur les boîtiers, des câbles se cassent au fil du temps que l'on utilise le prototype. L'alimentation de chaque IMU ne s'effectue pas correctement ce qui vient perturber le bon fonctionnement de l'ADC et donc du capteur de température.

Le logiciel pour l'iPad :

Le logiciel n'est pas modulaire puisque toutes les fonctionnalités dont le système de communication sont liées à l'interface graphique 3D pour reproduire les mouvements du corps humains. De plus celui-ci ne dispose pas de multitâche, de ce fait l'interaction de l'utilisateur sur le GUI peut venir perturber le système de communication. L'enregistrement des données s'effectue sur des fichiers ce qui n'est pas non plus optimal si l'on veut les utiliser par la suite (un nombre important de données crée un ralentissement important). Outre les problèmes, afin de respecter les spécifications du projet, il nous faut récupérer les données brutes des capteurs inertiels. Comme nous l'avons déjà vu précédemment, ces données ne doivent pas contenir d'erreurs et elles doivent être reçues à une fréquence précise afin de ne pas agir sur l'efficacité des algorithmes. Il faut donc un autre protocole de communication, soit TCP qui permet plus de robustesse.

L'architecture nécessite également une refonte totale puisqu'il faut incorporer plusieurs modes de traitements et de nouvelles interfaces utilisateurs qui sont décrites ci-après. Il faut également améliorer l'interface existante, essentiellement pour la rendre plus ergonomique pour l'utilisateur. En ce qui concerne l'affichage des mouvements en 3D du corps humain celui-ci doit être modifié afin de permettre à l'utilisateur d'utiliser les fonctions tactiles de zoom et de déplacement. Une grille de repère devra également être ajoutée afin de visualiser plus efficacement les mouvements. Idéalement il faudra utiliser un moteur 3D au lieu des

librairies OpenGL. En ce qui concerne l'affichage du déplacement de l'utilisateur, une carte du laboratoire de recherche devra être incorporée. Cette carte doit également être dotée des fonctions tactiles de zoom et de déplacement utilisateur.

Il faut ajouter le multitâche pour permettre au système de communication de fonctionner correctement, de ne pas bloquer l'interface utilisateur et de permettre le temps réel avec les algorithmes inertiels. Il faut ajouter une base de données MySQL puisque l'on souhaite utiliser différents modes de traitement et qu'il faut pouvoir étudier et comparer les résultats après l'application des différents algorithmes inertiels. Cette base permet pour un enregistrement (en temps réel) ou préalablement enregistrer d'y incorporer les données brutes de chaque IMU mais aussi les données calculées des différents algorithmes, reprenant les calculs de base et la configuration des algorithmes utilisés.

Les algorithmes inertiels :

Comme nous l'avons déjà vu précédemment, la phase 2 du projet ne nous donnait pas les résultats attendus malgré le fait que de nombreuses corrections ont été effectuées sur les algorithmes eux-mêmes ainsi que sur leurs différentes constantes. Le problème principal est qu'aucune trace des changements sur les algorithmes effectués par les étudiants précédents n'est présente et qu'il se révèle de ce fait très compliquer d'interpréter les modifications. De plus les étudiants ayant travaillé sur la phase 2 ne connaissant pas réellement le fonctionnement des algorithmes inertiels, ces changements ont été faits aléatoirement afin de trouver le problème et ne sont pas réellement valides. De plus, on sait que des problèmes sont présents sur les autres étapes du projet tel que le micro-logiciel des IMU, le logiciel de l'iPad, l'architecture matérielle et le système de communication. Les algorithmes étant implémentés directement sur chaque IMU, il n'est pas possible de les debugger efficacement. En plus de cela, une fois les IMU incorporés dans le vêtement, il n'est plus possible de les debugger. Chaque algorithme devra être repris sur Matlab et testé en simulation avec les données brutes provenant d'un IMU que nous utilisons.

ANNEXE III

PHOTOS DE SYSTÈMES PERMETTANT LA CAPTURE DE MOUVEMENTS

Tableau-A III-1 Photos de systèmes de capture de mouvements



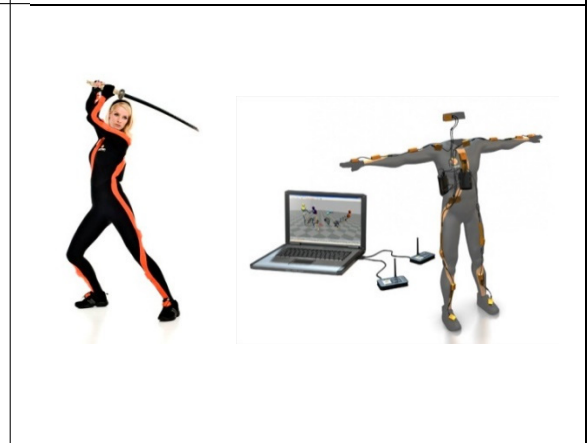
Système optique de la société PhaseScape



Système mécanique Gipsy 7 de la société
Metamotion



Système magnétique MotionStar de la société
Ascension Technologie



Système de capture de mouvements par inertie de
la société Xsens

ANNEXE IV

CONVERTION COORDONNÉES CARTÉSIENNES ET GÉODÉSIIQUES

Le passage de coordonnées géodésiques à cartésiennes dans le repère ECEF est très simple. Dans le système WGS84, on a :

Constantes

Axe semi-majeur : $a = 6378137 \text{ m}$

Excentricité : $e = \sqrt{2f - f^2} = 81.818 \times 10^{-3}$

Calcul

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \text{ et } \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (N + h) \cos \phi \cos \lambda \\ (N + h) \cos \phi \sin \lambda \\ (N(-e^2 + h) \sin \phi) \end{bmatrix}$$

Pour passer des coordonnées cartésiennes à géodésiques, il faut effectuer un calcul itératif, soit :

Étape 1 : Initialisation

$$h = 0 \quad N = 0 \quad \sin \phi = 0 \quad p = \sqrt{x^2 + y^2}$$

Étape 2 : 1^{ère} itération

$$\phi = \tan^{-1} \left(\frac{z + e^2 N \sin \phi}{p} \right)$$

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$$

$$h = \frac{p}{\cos \phi} - N$$

Étape 3 : 2^{ème} itération

$$\sin \phi = \frac{z}{(1 - e^2)N + h}$$






On obtient alors ϕ, N et h

Ensuite, on itère jusqu'à ce que le résultat ne change plus et on en déduit la longitude λ

ANNEXE V

INFORMATIONS TECHNIQUES SUR LES CENTRALES INERTIELLES UTILISÉES

Tableau-A V-1 Informations techniques sur les centrales inertielles utilisées

Nom	STEVAL-MKI062V2	STEVAL-MKI119V1	MPU-9150 MotionFit	x-IMU	ibNav nouvelle génération
Constructeur	STMicroelectronics	STMicroelectronics	InvenSense	X-IO Technologies	STMicroelectronics
Photo					
Type de capteurs	LPR430AL LY330ALH LSM303DLH	L3GD20 LSM303DLHC	MPU-9150 AK8975	IMU-3000 LSM303DLH	LSM9DS1
Dimensions (mm)	Acc/Mag : 5x5x1 Gyr (X,Y) : 4x5x1 Gyr (Z) : 3x5x1	Acc/Mag : 3x5x1 Gyr : 4x4x1	Acc/Gyr : 4x4x1 Mag : 4x4x0.75	Acc/Mag : 5x5x1 Gyr : 4x4x1	Acc/Gyr/Mag : 3.5x3x1
Plage de sensibilité (g, Gs, dps)	Acc : ±2 à 8 Mag : ±1.9 à 8.1 Gyr : ±300 à 1200	Acc : ±2 à 8 Mag : ±1.3 à 8.1 Gyr : ±250 à 2000	Acc : ±2 à 16 Mag : ±12 Gyr : ±250 à 2000	Acc : ±2 à 8 Mag : ±1.9 à 8.1 Gyr : ±250 à 2000	Acc : ±2 à 16 Mag : ±4 à 16 Gyr : ±245 à 2000
Linéarité	Mag : ±1 % de FS Gyr : ±1 % de FS	Mag : ±1 % de FS Gyr : ±0.2 % de FS	Acc : ±0.5 % de FS Gyr : ±0.2 % de FS	Mag : ±1 % de FS Gyr : ±0.2 % de FS	
Densité du bruit	Acc : 218 $\mu\text{g}/\sqrt{\text{Hz}}$	Acc : 220 $\mu\text{g}/\sqrt{\text{Hz}}$ Gyr : 0.03 $\text{dps}/\sqrt{\text{Hz}}$	Acc : 400 $\mu\text{g}/\sqrt{\text{Hz}}$ Gyr : 0.005 $\text{dps}/\sqrt{\text{Hz}}$	Acc : 218 $\mu\text{g}/\sqrt{\text{Hz}}$ Gyr : 0.01 $\text{dps}/\sqrt{\text{Hz}}$	
Stabilité du biais	Acc : ±20 mg Mag : ±20 Gs	Acc : ±60 mg Mag : ±20 Gs Gyr : ±10 à 75 dps	Acc (X,Y) : ±80 mg Acc (Z) : ±150 mg Gyr : ±20 dps	Acc : ±20 mg Gyr : ±20 dps	Acc : ±90 mg Mag : ±1 Gs Gyr : ±30 dps
Sensibilité change avec la température (%/°C)	Acc : ±0.01 Gyr : ±0.01	Acc : ±0.01 Gyr : ±2	Acc : ±0.02 Gyr : ±0.04	Acc : ±0.01 Gyr : ±2	
Biais change avec la température	Acc : ±0.1 mg/°C Gyr : ±0.02 dps/°C	Acc : ±0.5 mg/°C Gyr : ±0.03 à 0.04 dps/°C	Acc : ±0.75 à 1.5 mg/°C Gyr : ±0.16 dps/°C	Acc : ±0.1 mg/°C Gyr : ±0.01 dps/°C	
Taux d'échantillonnage (Hz)	Acc : 50 à 1000 Mag : 0.75 à 75 Gyr : CAN	Acc : 1 à 5376 Mag : 0.75 à 220 Gyr : 95 à 760	Acc : 4 à 1000 Gyr : 4 à 8000	Acc : 50 à 1000 Mag : 0.75 à 75 Gyr : 512	Acc : 14.9 à 952 Mag : 0.625 à 80 Gyr : 14.9 à 952

ANNEXE VI

ERREURS DÉTERMINISTES POUR CHAQUE IMU DU PROJET IBNAV

Tableau-A VI-1 Erreurs déterministes de l'accéléromètre pour chaque IMU du projet ibNav

	e_{11}	e_{12}	e_{13}	e_{21}	e_{22}	e_{33}	b_x	b_y	b_z
IMU 1 - Torse	0.9811	1.50^{e-4}	-1.59^{e-3}	0.9906	-1.19^{e-3}	0.9709	8.35^{e-3}	7.84^{e-3}	2.35^{e-2}
IMU 2 - Main droite	0.9761	-3.47^{e-3}	-4.56^{e-3}	0.9636	-5.81^{e-3}	0.9902	2.25^{e-2}	-9.43^{e-3}	8.43^{e-3}
IMU 3 - Avant-bras droit	0.9803	5.32^{e-3}	2.4^{e-3}	0.9811	2.18^{e-3}	0.9723	-1.07^{e-2}	1.65^{e-3}	-1.35^{e-2}
IMU 4 - Bras droit	0.9962	-7.39^{e-4}	-1.24^{e-3}	0.9949	-7.51^{e-4}	0.9622	3.15^{e-2}	3.42^{e-2}	-3.99^{e-2}
IMU 5 – Main gauche	0.9858	-4.16^{e-3}	-4.79^{e-3}	0.9862	-2.51^{e-3}	0.9710	9.24^{e-3}	2.58^{e-3}	-5.14^{e-2}
IMU 6 – Avant-bras gauche	0.9623	2.83^{e-4}	2.51^{e-3}	0.9939	4.35^{e-3}	0.9872	5.24^{e-2}	-4.23^{e-3}	3.45^{e-2}
IMU 7 - Bras gauche	0.9653	3.91^{e-3}	3.70^{e-3}	0.9973	2.13^{e-4}	0.9862	-2.54^{e-2}	4.25^{e-2}	6.84^{e-3}
IMU 8 - Tête	0.9852	-3.25^{e-3}	4.47^{e-3}	0.9871	-4.51^{e-4}	0.9647	-7.46^{e-3}	-8.21^{e-3}	4.21^{e-2}
IMU 9 – Pied droit	0.9830	2.53^{e-3}	-2.63^{e-3}	0.9030	1.59^{e-3}	0.9834	-3.12^{e-2}	1.78^{e-2}	-3.97^{e-2}
IMU 10 – Jambe B droite	0.9859	-4.76^{e-4}	8.99^{e-4}	0.9897	-7.94^{e-4}	0.9689	3.85^{e-2}	-9.92^{e-3}	6.42^{e-3}
IMU 11 – Jambe H droite	0.9663	4.57^{e-4}	5.62^{e-3}	0.9862	3.96^{e-3}	0.9900	8.47^{e-3}	2.36^{e-2}	8.47^{e-3}
IMU 12 – Pied gauche	0.9828	-6.59^{e-3}	-4.52^{e-3}	0.9668	5.75^{e-4}	0.9956	-1.36^{e-2}	-3.11^{e-2}	2.75^{e-2}
IMU 13 – Jambe B gauche	0.7941	3.36^{e-4}	-9.42^{e-4}	0.9882	-5.34^{e-4}	0.9818	4.33^{e-3}	-2.78^{e-2}	3.85^{e-2}
IMU 14 – Jambe H gauche	0.9768	1.50^{e-4}	3.64^{e-3}	0.9618	4.38^{e-3}	0.9983	-2.29^{e-2}	3.65^{e-2}	4.97^{e-2}

Tableau-A VI-2 Erreurs déterministes des gyroscopes pour chaque IMU du projet ibNav

	b_x	b_y	b_z		b_x	b_y	b_z
IMU 1 - Torse	-5.1^{e-2}	1.83^{e-2}	-1.5^{e-2}	IMU 8 - Tête	-5.2^{e-2}	1.51^{e-2}	2.82^{e-2}
IMU 2 - Main droite	2.56^{e-2}	5.47^{e-2}	-3.3^{e-2}	IMU 9 – Pied droit	2.08^{e-2}	-3.5^{e-2}	1.51^{e-2}
IMU 3 - Avant-bras droit	-1.9^{e-2}	5.25^{e-2}	-4.7^{e-2}	IMU 10 – Jambe B droite	5.44^{e-2}	5.95^{e-2}	-2.6^{e-2}
IMU 4 - Bras droit	-5.2^{e-2}	3.27^{e-2}	-1.6^{e-2}	IMU 11 – Jambe H droite	-4.2^{e-2}	2.88^{e-2}	-1.8^{e-2}
IMU 5 – Main gauche	-1.2^{e-2}	3.62^{e-2}	-5.9^{e-2}	IMU 12 – Pied gauche	-2.9^{e-2}	-3.1^{e-2}	2.1^{e-2}
IMU 6 – Avant-bras gauche	3.53^{e-2}	-2.4^{e-2}	-3.5^{e-2}	IMU 13 – Jambe B gauche	-1.1^{e-2}	-3.4^{e-2}	5.7^{e-2}
IMU 7 - Bras gauche	1.71^{e-2}	-3.0^{e-2}	5.05^{e-2}	IMU 14 – Jambe H gauche	4.9^{e-2}	1.01^{e-2}	-4.5^{e-2}

Tableau-A VI-3 Erreurs déterministes du magnétomètre pour chaque IMU du projet ibNav

	e_{11}	e_{12}	e_{13}	e_{21}	e_{22}	e_{33}	b_x	b_y	b_z
IMU 1 - Torse	2.050	-3.76^{e-2}	2.138	7.28^{e-2}	2.51^{e-2}	2.174	13.83	-16.83	83.46
IMU 2 - Main droite	2.080	4.45^{e-2}	1.993	1.11^{e-2}	-3.73^{e-2}	1.908	-37.68	94.76	14.93
IMU 3 - Avant-bras droit	2.093	5.83^{e-2}	2.119	-5.11^{e-2}	2.22^{e-2}	2.174	99.84	69.44	-35.77
IMU 4 - Bras droit	1.900	1.38^{e-2}	1.910	-8.47^{e-3}	-3.75^{e-2}	2.105	-67.15	77.47	-10.17
IMU 5 – Main gauche	1.992	-1.78^{e-2}	2.015	4.23^{e-2}	9.14^{e-3}	2.134	-10.02	-45.56	85.32
IMU 6 – Avant-bras gauche	1.912	-3.21^{e-2}	2.103	-1.35^{e-2}	3.48^{e-2}	2.102	-61.68	42.18	-95.89
IMU 7 - Bras gauche	2.075	-3.88^{e-2}	2.030	2.33^{e-2}	8.43^{e-3}	1.992	41.80	17.74	-82.27
IMU 8 - Tête	2.004	2.69^{e-2}	1.957	2.35^{e-2}	-1.12^{e-2}	1.952	34.79	-99.39	66.97
IMU 9 – Pied droit	2.165	1.39^{e-2}	2.117	-9.44^{e-3}	2.11^{e-2}	2.015	84.37	-69.60	78.41
IMU 10 – Jambe B droite	1.901	9.74^{e-3}	2.104	-4.23^{e-2}	-1.59^{e-2}	1.932	-70.46	58.81	-48.01
IMU 11 – Jambe H droite	2.031	2.96^{e-2}	1.985	-1.47^{e-2}	9.50^{e-3}	2.147	37.20	10.27	-11.27
IMU 12 – Pied gauche	1.988	-1.87^{e-2}	2.076	3.85^{e-2}	-1.18^{e-2}	2.150	52.80	-29.70	21.12
IMU 13 – Jambe B gauche	2.179	-3.21^{e-2}	2.004	8.22^{e-3}	3.66^{e-2}	1.992	-40.81	87.58	85.18
IMU 14 – Jambe H gauche	2.036	-3.57^{e-2}	2.114	-1.75^{e-2}	2.01^{e-2}	2.0875	11.54	-40.98	80.59

Note : La calibration des magnétomètres est ici à l'état initial, elle dépend également de l'environnement, les erreurs de fer doux étant sensibles à l'environnement.

ANNEXE VII

ERREURS STOCHASTIQUES POUR CHAQUE IMU DU PROJET IBNAV

Tableau-A VII-1 Erreurs stochastiques pour chaque IMU du projet ibNav

	Acc X	Acc Y	Acc Z	Gyr X	Gyr Y	Gyr Z	Mag X	Mag Y	Mag Z
	$g^{3/2}$			$deg/s^{1/2}$			$uT^{3/2}$		
IMU 1	$2.75E^{-4}$	$3.6E^{-4}$	$5.1E^{-4}$	$8.2E^{-2}$	$4.7E^{-2}$	$4.1E^{-2}$	$4.25E^{-4}$	$4.3E^{-4}$	$4.6E^{-4}$
IMU 2	$6.43E^{-4}$	$4.88E^{-4}$	$1.28E^{-4}$	$3.08E^{-2}$	$8.73E^{-2}$	$5.42E^{-2}$	$2.54E^{-4}$	$3.67E^{-4}$	$3.57E^{-4}$
IMU 3	$1.91E^{-4}$	$5.86E^{-4}$	$2.13E^{-4}$	$4.85E^{-2}$	$8.48E^{-2}$	$2.04E^{-2}$	$4.32E^{-4}$	$5.28E^{-4}$	$2.82E^{-4}$
IMU 4	$3.04E^{-4}$	$2.26E^{-4}$	$3.21E^{-4}$	$4.27E^{-2}$	$2.35E^{-2}$	$3.01E^{-2}$	$3.77E^{-4}$	$3.51E^{-4}$	$4.77E^{-4}$
IMU 5	$3.38E^{-4}$	$1.04E^{-4}$	$2.82E^{-4}$	$5.38E^{-2}$	$7.11E^{-2}$	$4.37E^{-2}$	$4.58E^{-4}$	$2.50E^{-4}$	$3.54E^{-4}$
IMU 6	$3.04E^{-4}$	$5.95E^{-4}$	$8.77E^{-4}$	$5.16E^{-2}$	$3.30E^{-2}$	$3.85E^{-2}$	$5.45E^{-4}$	$3.42E^{-4}$	$4.06E^{-4}$
IMU 7	$5.78E^{-4}$	$2.04E^{-4}$	$5.42E^{-4}$	$7.81E^{-2}$	$4.34E^{-2}$	$6.06E^{-2}$	$4.10E^{-4}$	$4.12E^{-4}$	$4.49E^{-4}$
IMU 8	$3.07E^{-4}$	$5.23E^{-4}$	$7.51E^{-4}$	$3.85E^{-2}$	$6.74E^{-2}$	$5.64E^{-2}$	$4.53E^{-4}$	$2.29E^{-4}$	$2.85E^{-4}$
IMU 9	$8.15E^{-4}$	$8.61E^{-4}$	$4.70E^{-4}$	$3.81E^{-2}$	$5.08E^{-2}$	$3.99E^{-2}$	$2.53E^{-4}$	$5.04E^{-4}$	$4.47E^{-4}$
IMU 10	$8.41E^{-4}$	$4.97E^{-4}$	$8.02E^{-4}$	$8.65E^{-2}$	$7.42E^{-2}$	$6.45E^{-2}$	$3.59E^{-4}$	$4.41E^{-4}$	$3.58E^{-4}$
IMU 11	$2.71E^{-4}$	$1.91E^{-4}$	$5.60E^{-4}$	$7.41E^{-2}$	$6.24E^{-2}$	$3.73E^{-2}$	$5.51E^{-4}$	$5.75E^{-4}$	$5.22E^{-4}$
IMU 12	$4.73E^{-4}$	$7.03E^{-4}$	$4.64E^{-4}$	$2.13E^{-2}$	$4.43E^{-2}$	$3.00E^{-2}$	$4.09E^{-4}$	$4.60E^{-4}$	$4.29E^{-4}$
IMU 13	$6.64E^{-4}$	$8.44E^{-4}$	$7.56E^{-4}$	$4.88E^{-2}$	$2.69E^{-2}$	$7.74E^{-2}$	$2.11E^{-4}$	$3.97E^{-4}$	$3.77E^{-4}$
IMU 14	$8.10E^{-4}$	$1.08E^{-4}$	$8.81E^{-4}$	$8.68E^{-2}$	$6.64E^{-2}$	$7.70E^{-2}$	$5.85E^{-4}$	$5.38E^{-4}$	$5.01E^{-4}$


ANNEXE VIII

TABLEAU DES LIMITATIONS DES ARTICULATIONS DU CORPS HUMAIN

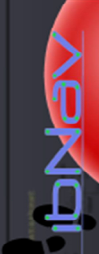
Tableau-A VIII-1 Tableau des limitations des articulations du corps humain


	<i>Euler φ</i>	<i>Euler θ</i>	<i>Euler ψ</i>
IMU 1 - Torse	-45° à 10°	-45° à 45°	-30° à 30°
IMU 2 - Main droite	-45° à 45°	-	25° à -35°
IMU 3 - Avant-bras droit	0° à 120°	-120° à 0°	-
IMU 4 - Bras droit	-120° à 90°	-	-90° à 90°
IMU 5 – Main gauche	-45° à 45°	-	-25° à 35°
IMU 6 – Avant-bras gauche	0° à 120°	0° à 120°	-
IMU 7 - Bras gauche	-120° à 90°	-	90° à -90°
IMU 8 - Tête	-35° à 35°	-45° à 20°	-25° à 25°
IMU 9 – Pied droit	-	0 à 30°	-
IMU 10 – Jambe B droite	-	0° à 90°	-
IMU 11 – Jambe H droite	30 à -30°	-90 à 90°	45° à -90°
IMU 12 – Pied gauche	-	0 à 30°	-
IMU 13 – Jambe B gauche	-	0° à 90°	-
IMU 14 – Jambe H gauche	-30 à 30°	-90 à 90°	-90° à 45°

Note : Les angles ont été adaptés par rapport aux positions des IMUs du projet ibNav.



Le génie pour l'industrie





LABORATOIRE SPÉCIALISÉ EN SYSTÈMES
EMBARQUÉS, NAVIGATION ET AVIONIQUE

Indoor and Body Navigation Prototype using Low-cost Sensors

Adrien Mixte
Adrien.Mixte@lassena.etsmtl.ca

René Jr. Landry
René.Jr.Landry@etsmtl.ca

Department of Electrical Engineering
LASSENA (Embedded Systems, Navigation and Avionics Specialized Laboratory)

Introduction

The ibNav project aims to use only low-cost inertial sensors combined with inertial algorithms to obtain high efficiency in motion capture and indoor pedestrian navigation

- Motion Capture : Body movement's recognition and displaying it in 3D using AHRS algorithms.
- Indoor Pedestrian Navigation : Visualise the movement on a 2D map without external sources using INS algorithms.

The prototype is designed to be composed of a network of 14 IMU using MEMS sensors set in a suit, between each main joint of the body. The IMUs are composed of geomagnetic and gyroscopes sensors working in 9 DOF. System data communication is with SPI, UART, TCP and UDP. Data processing is computed in real time on Apple's device, the iPad. The goal of the GUI is to displaying results and to be a study platform for inertial algorithms (analyse, compare and improve them).

Architecture hardware




Figure 2 : ibNav simplified communication schema

Objectives and Challenges

- Actual trend : Most of motion capture systems are too expensive and need to use external source (eg. Outside-in systems with special room and cameras). Concerning indoor navigation, the trend is to use IPS

Architecture composition :

- 13 cases containing IMUs "slave" platforms, one with the

Calibration

- Gyroscopes : Measurement of the angular rate are made by Coriolis effect using Foucault's Pendulum principle. So are acquired the values for pitch / roll on the axis along the sensor's package and the yaw on the axis perpendicular to it.

Before use these sensors, the first step is to calibrate each sensor axis to reduce the following deterministic errors : Bias (with hard iron of magnetometers), scale factor and axis misalignment (for stochastic errors, white noise is used for each IMU with the Allan Variance Method).

Calibration algorithms scheme

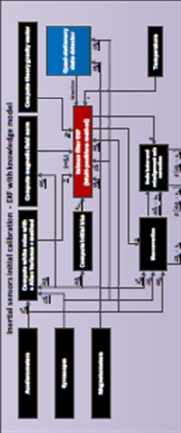


Figure 6 : calibration algorithms scheme

Initial Calibration

The second step of initial calibration uses the Lymnmotion ALSD robot in quasi-stationary scenarios to rectify the position of an IMU within its casing.




Figure 7 : Robot of Calibration

ibNav - Architecture (version 2.6)




Figure 3 : ibNav - Architecture (version 2.6)

Post Processing




Figure 7 : Robot of Calibration

Figure-A IX-1 Poster du projet ibNav (Partie 1)

systems with complex network of external sensors.

- Advantages of ibNav :
The main idea behind the ibNav project is to offer a high efficiency system at a low-cost production, allowing an easy and stealthy deployment anywhere without any external requirements.
- Processing required :
Firstly, to use this low-cost system, it's necessary to find solutions to the many material restrictions. That is, the delicate development of IMU firmware, hardware architecture and GUI with adapted communication. Finally, to use sensors independently without use external source (and hybrid systems), it's necessary to perform a meticulous calibration and implement inertial algorithms applied to raw measurements of geomagnetic and gyroscope sensors to correct sensors errors.

IMU configuration

The network of 14 IMU-MEMS referenced STEVAL-MKI06V2 (developed by ST Microelectronics) has the following features :




Figure 1 : iNemo module and simplified diagram

- The firmware developed in LASSENA give many possibilities and a great reliability to satisfy specifications project.
- Inter-IMU system of communication with SPI protocol bidirectional with DMA, bandwidth of 562 KHz, hot-plug (0.4 seconds time of detection) and check frame integrity.
- IMU / GUI (via a module Wireless*) communication with UART protocol bidirectional with DMA TX and INT RX, bandwidth of 460 KHz and check frame integrity.
- Get raw and computed inertial data in the GUI (in real time).
- Get temperature, accelerations, angular rates and magnetic fields of each IMU (10 DOF) at 50 Hz.
- Low energy consumption (244 mWh / IMU).
- User interaction with each IMU via the GUI.
- Exceptions control by watchdog.
- The RN-134 Wi-Fi module (developed by Roving Networks), interface between the wired (UART protocol) and the wireless (TCP protocol). The TCP protocol is bidirectional and protected by a WPA / WPA2 - PSK key. The inter-GUI communication allowing several iPad to simultaneously show capture motion and indoor pedestrian navigation results, requires the use of a third communication protocol: UDP.

IMU "master" and the module RN-134 and the last with the battery of 25 Wh (Theses are into the suit).

- Coil wiring shielded (4 GND) to flexibility and modularity with connectors of communication and debug on each case.
- Star distribution of energy and SPI Clock to high efficiency.




Figure 4 : ibNav architecture scheme and photography

Graphic User Interface (iPad)

The software in Objective-C allows two different modes, real time and post-processing. The first shows movements for both navigations, raw data and computed data with chart and 3D representation. The last, allows to change calibration and navigation algorithms to compare and analyse results on a previous record in real time. The software has 5 main views :

- My Setup :
Body and indoor view configuration, algorithms configuration, record file choice (post-processing), compute method (real time), map and BVH choices and an about view.
- My Sensors :
Engineer part with raw measurements on charts and IMU user interaction.
- My Algorithms :
Engineer part with computed measurements on charts / 3D and algorithms configuration.
- My Body and My location :
3D motion capture (BVH Reader) and 2D map localization.




Figure 5 : Example of GUI Interfaces

Algorithms navigation design

Motion capture algorithms use an AHRS model with two methods of drift reduction to compute Euler's Angles calculation (ϕ, θ, ψ). During this process, an online calibration is performed to correct the magnetometer's soft iron's influence on the scale factor and axis misalignment and the temperature's on the bias.

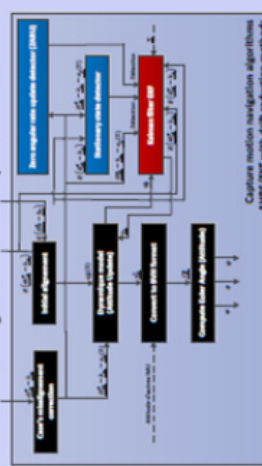


Figure 8 : AHRS algorithms scheme

Indoor Pedestrian Navigation uses an INS model with PDR methodology to obtain the position, velocity and attitude. In addition to an online calibration, several methods of drift reduction are used like ZUPT, ZARU, HDC, Compass and BIM.

Note : All three algorithms use an Extended Kalman filter to calibrate sensors or estimate and correct drift errors.

Currently, the GUI and the indoor pedestrian navigation algorithms are still in development. The others algorithms are in continuous improvement in the LASSENA in order to reach expected results.




Figure 9 : Suit of the ibNav project

Possible uses

ibNav is designed to the research and development for industrial uses and to understand, compare, analyse inertial algorithms. This project covers three distinct sectors :

- Cinema and video games (capture motion mode)
- Military use (both navigation mode) allowing the capture motion of both movements and position of a soldier to improve fighting positions and movements inside buildings.
- Sport (both navigation mode) allowing the analysis of the athlete's movements to improve one's performance.

Sensors and initial calibration

ibNav project uses MEMS sensors (Microelectromechanical system) :

- Accelerometers
Measures the projection of the gravity vector on the 3D axis.
- Magnetometers
Measures the strength of the magnetic field on the 3D axis.

References

- Mico Adhem, "SIGMA 2011 : Développement d'un prototype de navigation corporelle", Ecole de technologie supérieure, Montréal, 2011.
- Mico Adhem, "SIGMA 2011 : Développement d'un prototype de capture de mouvement et de navigation intérieure basé sur l'utilisation de plateformes IMU-MEMS à faible coût", Ecole de technologie supérieure, Montréal, 2012.

Indoor and body navigation prototype using low cost sensors

5 March 2013

Version 1.4

LASSENA, 2013

Figure-A IX-2 Poster du projet ibNav (Partie 2)

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Abbate, N., I. Aleo, A. Basile, C. Brigante et A. Faulisi. 2011. « Design of an Inertial Motion Module ». In *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* (9-13 May 2011), p. 1-4.
- Abbate, N., A. Basile, C. Brigante et A. Faulisi. 2009. « Development of a MEMS based wearable motion capture system ». In *Human System Interactions, 2009. HSI '09. 2nd Conference on.* (21-23 May 2009), p. 255-259.
- Aitenbichler, E., et M. Muhlhauser. 2003. « An IR local positioning system for smart items and devices ». In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on.* (19-22 May 2003), p. 334-339.
- Bachmann, Eric Robert. 2000. « Inertial And Magnetic Tracking Of Limb Segment Orientation For Inserting Humans Into Synthetic Environments ».
- Bakhshi, S., M. H. Mahoor et B. S. Davidson. 2011. « Development of a body joint angle measurement system using IMU sensors ». In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE.* (Aug. 30 2011-Sept. 3 2011), p. 6923-6926.
- Batista, P., C. Silvestre, P. Oliveira et B. Carneira. 2011. « Accelerometer Calibration and Dynamic Bias and Gravity Estimation: Analysis, Design, and Experimental Evaluation ». *Control Systems Technology, IEEE Transactions on*, vol. 19, n° 5, p. 1128-1137.
- Bekkeng, J. K. 2009. « Calibration of a Novel MEMS Inertial Reference Unit ». *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, n° 6, p. 1967-1974.
- Beravs, T., J. Podobnik et M. Munih. 2012. « Three-Axial Accelerometer Calibration Using Kalman Filter Covariance Matrix for Online Estimation of Optimal Sensor Orientation ». *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, n° 9, p. 2501-2511.
- Bird, J., et D. Arden. 2011. « Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [Emerging Opportunities for Localization and Tracking] ». *Wireless Communications, IEEE*, vol. 18, n° 2, p. 28-35.
- Bo, Wang, Li Jie, Meng Yan, Jiang Qiao et Li Gui-Zhen. 2010. « Temperature Compensation for Gyroscope Free Micro Inertial Measurement Unit ». In *Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference*

on.

(17-19 Sept. 2010), p. 1294-1297.

Borenstein, Lauro Ojeda and Johann. 2007. « Non-GPS Navigation with the Personal Dead-Reckoning System ».

Chowaw-Liebman, O., U. Christoph, K. Krempels et C. Terwelp. 2010. « Indoor navigation approach based on approximate positions ». In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on.* (15-17 Sept. 2010), p. 1-7.

Dorveaux, E., D. Vissiere, A. P. Martin et N. Petit. 2009. « Iterative calibration method for inertial and magnetic sensors ». In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* (15-18 Dec. 2009), p. 8296-8303.

Edge, L. 2001. « Centimeter-accuracy indoor navigation using GPS-like pseudolites ».

Guo, P., H. Qiu, Y. Yang et Z. Ren. 2008. « The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system ». In *Position, Location and Navigation Symposium, 2008 IEEE/ION.* p. 1167-1174. IEEE.

El-Sheimy, N., Hou Haiying et Niu Xiaoji. 2008. « Analysis and Modeling of Inertial Sensors Using Allan Variance ». *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, n° 1, p. 140-149.

Farrell, Jay. 2008. Aided navigation : GPS with high rate sensors. Coll. « Electronic engineering ». New York: McGraw-Hill, xxi, 530 p.

Finlay, C., et al. 2010. « International Geomagnetic Reference Field: The eleventh generation » *Geophysical Journal International.* p. 1216-1230.

Fischer, Carl, Kavitha Muthukrishnan, Mike Hazas et Hans Gellersen. 2008. « Ultrasound-aided pedestrian dead reckoning for indoor navigation ». In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments.* (San Francisco, California, USA), p. 31-36. 1410020: ACM.

Fong, W. T., S. K. Ong et A. Y. C. Nee. 2008. « Methods for in-field user calibration of an inertial measurement unit without external equipment ». *Measurement Science & Technology*, vol. 19, n° 8.

- Perales, FJ. 2001. « Human Motion Analysis & Synthesis using Computer Vision and Graphics Techniques. State of Art and Applications ». *Proceedings of the world multiconference on systemics, cybernetics and informatics (SCI2001)*.
- Giroux, Richard. 2004. « Capteurs bas de gamme et systèmes de navigation inertielle: Nouveaux paradigmes d'application ». Ph.D., Ecole de Technologie Supérieure (Canada).
- Grewal, M. S., V. D. Henderson et R. S. Miyasako. 1991. « Application of Kalman filtering to the calibration and alignment of inertial navigation systems ». *Automatic Control, IEEE Transactions on*, vol. 36, n° 1, p. 3-13.
- Guangyi, Shi, He Yang, Ye Feng, Yang Jiancheng, Wang Peng et Jin Yufeng. 2011. « Towards an ubiquitous motion capture system using inertial MEMS sensors and ZigBee network ». In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2011 IEEE International Conference on*. (20-23 March 2011), p. 230-234.
- Hallberg, J., M. Nilsson et K. Synnes. 2003. « Positioning with Bluetooth ». In *Telecommunications, 2003. ICT 2003. 10th International Conference on*. (23 Feb.-1 March 2003) Vol. 2, p. 954-958 vol.2.
- Hui, Liu, H. Darabi, P. Banerjee et Liu Jing. 2007. « Survey of Wireless Indoor Positioning Techniques and Systems ». *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, n° 6, p. 1067-1080.
- Hightower, J., et G. Borriello. 2001. « Location systems for ubiquitous computing ». *Computer*, vol. 34, n° 8, p. 57-66.
- IEEE Recommended Practice for Precision Centrifuge Testing of Linear Accelerometers. 2009. « ». *IEEE Std 836-2009 (Revision of IEEE Std 836-2001)*, p. C1-110.
- Jimenez, A. R., F. Seco, C. Prieto et J. Guevara. 2009. « A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU ». In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*. (26-28 Aug. 2009), p. 37-42.
- Jiménez, A. R., F. Seco, J. C. Prieto et J. Guevara. 2010. « Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU ». In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*. (11-12 March 2010), p. 135-143.
- Kang, Eyung W. 2008. « Radar system analysis, design, and simulation ». In *Artech House*. (Chapter 8.7).

- Kap-Ho, Seo, Oh Seungsub, Park Yongsik, Park Sung Ho et Suh Jin-Ho. 2011. « Human motion reconstruction based on inertial motion sensors ». In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on.* (23-26 Nov. 2011), p. 386-389.
- Kolodziej, K.W., et Johan Hjelm. 2006. *Local Positioning Systems: LBS Applications and Services*. CRC PressINC.
- Kwakkel, S. P. , S. Godha et G. Lachapelle. 2007. « Foot and Ankle Kinematics During Gait Using Foot Mounted Inertial System ». In *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation.* (San Diego, CA, January 22 - 24, 2007), p. 1056 - 1064.
- Ladetto, Q. 2003. *Capteurs et algorithmes pour la localisation autonome en mode pédestre*, 66.
- Lambert, Jean. 2011. « Système de navigation corporelle à l'intérieur des bâtiments sur ipad à base de senseurs inertiels à faible coût ». Ecole de Technologie Supérieure (Canada).
- Lavoie, Philippe. 2012. « Système de navigation hybride GPS/INS à faible coût pour la navigation robuste en environnement urbain ». M. Ing., Ecole de Technologie Supérieure (Canada).
- Le, Manh Hung V. 2009. « Indoor Navigation System for Handheld Devices ». Worcester, Massachusetts, USA.
- Lee, S., B. Kim, H. Kim, R. Ha et H. Cha. 2011. « Inertial sensor-based indoor pedestrian localization with minimum 802.15. 4a configuration ». *Industrial Informatics, IEEE Transactions on*, vol. 7, n° 3, p. 455-466.
- Li, D., R. Landry et P. Lavoie. 2008. « Low-cost MEMS sensor-based attitude determination system by integration of magnetometers and GPS: A real-data test and performance evaluation ». In *Position, Location and Navigation Symposium, 2008 IEEE/ION*. p. 1190-1198. IEEE.
- Link, J.A.B., P. Smith, N. Viol et K. Wehrle. 2011. « Footpath: Accurate map-based indoor navigation using smartphones ». In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on.* p. 1-8. IEEE.
- Lomont, C. 2003. « Fast Inverse Square Root ». *Tech-315 nical report*.
- Madgwick, Sebastian OH, Andrew JL Harrison, and Ravi Vaidyanathan. 2011. « Estimation of IMU and MARG orientation using a gradient descent algorithm ». In

Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on. p. 1-7 IEEE.

- Menache, A. 2010. *Understanding Motion Capture for Computer Animation*. Morgan Kaufmann.
- Mixte, Adrien. 2011. « Stage EE5 : Développement d'un prototype de navigation corporelle ». Telecom Lille1 (France).
- Park, M. 2004. *Error analysis and stochastic modeling of MEMS based inertial sensors for land vehicle navigation applications*. University of Calgary, Department of Geomatics Engineering.
- Park, Minha, et Yang Gao. 2006. « Error Analysis and Stochastic Modeling of Low-cost MEMS Accelerometer ». *J. Intell. Robotics Syst.*, vol. 46, n° 1, p. 27-41.
- Pasolini, Fabio. 2009. « MEMS Accelerometers, Gyroscopes, & Geomagnetic Sensors – Propelling Disruptive Consumer Applications ». *STMicroelectronics*.
- Pomp, A., D.I.J.A.B. Link, I.K. Wehrle et B. Rumpe. 2012. « Smartphone-Based Indoor Mapping ». RWTH Aachen University, Germany.
- Pons-Moll, G., A. Baak, T. Helten, Mu, x, M. Iler, H. P. Seidel et B. Rosenhahn. 2010. « Multisensor-fusion for 3D full-body human motion capture ». In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* (13-18 June 2010), p. 663-670.
- Prayudi, I., et Kim Doik. 2012. « Design and implementation of IMU-based human arm motion capture system ». In *Mechatronics and Automation (ICMA), 2012 International Conference on.* (5-8 Aug. 2012), p. 670-675.
- Prigge, Eric A., et Jonathan P. How. 2004 « Signal architecture for a distributed magnetic local positioning system. » *Sensors Journal, IEEE* 4, no. 6, 864-873.
- Qintuo, Zhang, Tan Zhenfan et Guo Lidong. 2009. « Compensation of Temperature Drift of MEMS Gyroscope Using BP Neural Network ». In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on.* (19-20 Dec. 2009), p. 1-4.
- Renaudin, V., O. Yalak, P. Tomé et B. Merminod. 2007. « Indoor navigation of emergency agents ». *European Journal of Navigation*, vol. 5, n° 3, p. 36-45.
- Renaudin, Valerie, Muhammad Haris Afzal et Gerard Lachapelle. 2010. « New method for magnetometers based orientation estimation ». In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION.* (4-6 May 2010), p. 348-356.

- Renk, E. L., W. Collins, M. Rizzo, F. Lee et D. S. Bernstein. 2005. « Optimization-based calibration of a triaxial accelerometer-magnetometer ». In *American Control Conference, 2005. Proceedings of the 2005.* (8-10 June 2005), p. 1957-1962 vol. 3.
- Retscher, Guenther, et Qing Fu. 2010. « Continuous indoor navigation with RFID and INS ». In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION.* (4-6 May 2010), p. 102-112.
- Roetenberg, Daniel. 2006. « Inertial and Magnetic Sensing of Human Motion ». University of Twente, The Netherlands.
- Sabatini, A. M. 2006. « Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing ». *Biomedical Engineering, IEEE Transactions on*, vol. 53, n° 7, p. 1346-1356.
- Sanchez, A., S. Elvira, A. de Castro, G. Glez-de-Rivera, R. Ribalda et J. Garrido. 2009. « Low cost indoor ultrasonic positioning implemented in FPGA ». In *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE.* (3-5 Nov. 2009), p. 2709-2714.
- Shin, E. H., et N. El-Sheimy. 2002. « A new calibration method for strapdown inertial navigation systems ». *Zeitschrift für Vermessungswesen*, vol. 127, n° 1, p. 41-50.
- Skog, Isaac, et Peter Händel. 2006. « Calibration of a mems inertial measurement unit ». In *XVII IMEKO World Congress.* (Rio de Janeiro, Brazil, September 17-22, 2006), p. 17-22.
- Skog, Isaac. 2010b. « Zero-Velocity Detection—An Algorithm Evaluation ».
- SPASSOV, Ivan. 2007. « Algorithms for mAp-Aided Autonomous indoor pedestrian positioning and navigation ».
- Spinney, J.E. « Mobile Positioning and LBS Applications », *Geography* 88 (2003) 256–265.
- Sturman, David J. 1999. « A Brief History of Motion Capture for Computer Character Animation ».
- Syed, Z. F., P. Aggarwal, C. Goodall, X. Niu et N. El-Sheimy. 2007. « A new multi-position calibration method for MEMS inertial navigation systems ». *Measurement Science & Technology*, vol. 18, n° 7, p. 1897-1907.
- Torrieri, D., M. B. Bendak et G. Ritchie. 2011. « Indoor geolocation by inertial navigation ». In *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011.* (7-10 Nov. 2011), p. 1760-1765.

- Vasconcelos, J. F., G. Elkaim, C. Silvestre, P. Oliveira et B. Cardeira. 2011. « Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame ». *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, n° 2, p. 1293-1306.
- Vissière, D., A.P. Martin et N. Petit. 2007. « Using magnetic disturbances to improve IMU-based position estimation ». In *Proceedings of the European Control Conference 2007*. p. 2853-2858.
- Vlasic, Daniel, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, Jovan Popovi et #263. 2007. « Practical motion capture in everyday surroundings ». *ACM Trans. Graph.*, vol. 26, n° 3, p. 35.
- Vossiek, M., L. Wiebking, P. Gulden, J. Wiegardt, C. Hoffmann et P. Heide. 2003. « Wireless local positioning ». *Microwave Magazine, IEEE*, vol. 4, n° 4, p. 77-86.
- Walder, U., et T. Bernoulli. 2010. « Context-adaptive algorithms to improve indoor positioning with inertial sensors ». In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. (15-17 Sept. 2010), p. 1-6.
- Wang, Hongman, Xiaocheng Zhu et Jiangbo Chang. 2011. « Acceleration and Orientation Multisensor Pedometer Application Design and Implementation on the Android Platform ». In *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*. (21-23 Oct. 2011), p. 249-253.
- Want, Roy, Andy Hopper, Veronica Falc, #227 et Jonathan Gibbons. 1992. « The active badge location system ». *ACM Trans. Inf. Syst.*, vol. 10, n° 1, p. 91-102.
- Ward, Jamie A., Paul Lukowicz, Gerhard Tr\, \#246 et ster. 2005. « Gesture spotting using wrist worn microphone and 3-axis accelerometer ». In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. (Grenoble, France), p. 99-104. 1107578: ACMWerb, J., et C. Lanzl. 1998. « Designing a positioning system for finding things and people indoors ». *Spectrum, IEEE*, vol. 35, n° 9, p. 71-78.
- Wong, Alexander. 2007. « Low-Cost Visual/Inertial Hybrid Motion Capture System for Wireless 3D Controllers ». University of Waterloo.
- Xiaoping, Yun, et E. R. Bachmann. 2006. « Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking ». *Robotics, IEEE Transactions on*, vol. 22, n° 6, p. 1216-1227.
- Y. Wang, X. Jia, H.K. Lee. 2003. « An indoor wireless positioning system based on wireless local area network infrastructure ». In.

- Yan Li, Xiang Luo, Xiang Thomas Ren and Jianguo Jack Wang. 2012. « A Robust Humanoid Robot Navigation Algorithm with ZUPT ».
- Yisheng, Chen, J. Lee, R. Parent et R. Machiraju. 2005. « Markerless monocular motion capture using image features and physical constraints ». In *Computer Graphics International 2005*. (22-24 June 2005), p. 36-43.
- Yokokohji, Y., Y. Kitaoka et T. Yoshikawa. 2002. « Motion capture from demonstrator's viewpoint and its application to robot teaching ». In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*. (2002) Vol. 2, p. 1551-1558 vol.2.
- Zeljko, Zilic. 2012. « Sensor Integration Workshop ». McGill University, Montréal.
- Zhang, R., F. Hoeflinger et L. Reindl. 2012. « Inertial Sensor based Indoor Localization and Monitoring System for Emergency Responders ». *Sensors Journal, IEEE*, vol. PP, n° 99, p. 1-1.
- Zhao, Y., M. Horemuz et L.E. Sjöberg. 2011 « Stochastic modellIng and analysis of IMU sensor errors ».
- Zhiqiang, Zhang, Wu Zheng, Chen Jiang et Wu Jian-Kang. 2009. « Ubiquitous human body motion capture using micro-sensors ». In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. (9-13 March 2009), p. 1-5.