

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR  
SAURIOL, Bruno

MISE EN ŒUVRE EN TEMPS RÉEL D'UN  
RÉCEPTEUR HYBRIDE GPS-GALILEO

MONTREAL, LE 13 FÉVRIER 2008

© Bruno Sauriol, 2008

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. René Jr. Landry, directeur de mémoire  
Département de génie électrique à l'École de technologie supérieure

M. Ammar B. Kouki, président du jury  
Département de génie électrique à l'École de technologie supérieure

M. Aurelian Constantinescu, membre du jury  
CAE

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 30 JANVIER 2008

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## REMERCIEMENTS

Je remercie premièrement mon directeur, Prof. René Jr. Landry, pour m'avoir accueilli dans son laboratoire et pour m'avoir donné l'opportunité de découvrir la navigation par satellite. Je remercie également les membres de mon jury, Dr. Ammar B. Kouki et Dr. Aurelian Constantinescu, pour avoir bien voulu prendre le temps d'évaluer ce document.

Je tiens ensuite à remercier mes collègues et amis, sans qui la réalisation de ce projet n'aurait sans doute pas été possible. Tout d'abord merci à Dr. Yacine Adane, mon mentor radiofréquence (RF), pour sa précieuse aide dans l'élaboration de la tête RF. Merci aussi à André Zalzal pour son généreux support tout au long du projet. Un grand merci à Iurie Ilie pour tous ses conseils et pour m'avoir transmis autant de connaissances en matière de positionnement par satellite. Merci aussi à François Laramée, Cédric Chavagneux, Thomas Delaporte et Dr. Di Li et pour leur aide et pour leur bonne humeur habituelle. Je remercie tout spécialement Jean-Christophe Guay pour son support inconditionnel ainsi que pour toute l'aide qu'il a pu m'apporter en cette fin de projet.

Finalement, je tiens à remercier ma conjointe, Amélie, ainsi que mes parents, Denis et Louise, pour leur support moral et pour leur appui tout au long de mes études.

Ce mémoire a été rendu possible grâce au support financier du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) ainsi qu'au Fond québécois de la recherche sur la nature et les technologies (FQRNT).

# MISE EN ŒUVRE EN TEMPS RÉEL D'UN RÉCEPTEUR HYBRIDE GPS-GALILEO

SAURIOL, Bruno

## RÉSUMÉ

Le début du 21<sup>e</sup> siècle marque un tournant au niveau de la radionavigation par satellite. Parmi les événements importants, mentionnons l'avènement de Galileo, le tout nouveau système de positionnement de l'Union européenne, ainsi que la modernisation du GPS (*Global Positioning System*) des États-Unis. On prévoit que ces systèmes, regroupés sous l'acronyme GNSS (*Global Navigation Satellite System*), mettront à la disposition des utilisateurs civils une très grande quantité de signaux vers les années 2015, ceci permettant un positionnement d'une précision et d'une fiabilité sans précédent. Or, tout ce potentiel ne sera obtenu que par l'intégration des signaux GNSS, une tâche pouvant s'avérer très complexe compte tenu des technologies limitées de recherche et de développement (R&D) actuellement employées par l'industrie.

Ce mémoire présente la mise en œuvre en temps réel d'un récepteur de navigation par satellite GNSS hybride (GPS et Galileo). Ce projet, basé sur un FPGA (*Field-Programmable Gate Array*), constitue une avance technologique face aux procédés actuellement employés par l'industrie, qui sont coûteux et peu flexibles. Ceci permettrait donc d'offrir une solution intéressante aux problèmes d'intégration des signaux GNSS.

Le récepteur de navigation a été développé afin de démoduler les signaux GPS et Galileo en bande de fréquence L1 (1575,42 MHz). L'architecture système demeure toutefois très flexible et permettrait bientôt de démoduler des signaux en bande L2 (1227,6 MHz) et L5 (1176,45 MHz), par exemple. Le récepteur a été mis en œuvre via l'élaboration de trois principales sections, soit la tête radiofréquence (RF), les canaux de démodulation en fréquence intermédiaire (IF) et la solution de navigation en bande de base. L'architecture, principalement numérique, a été judicieusement développée de manière à permettre l'intégration de plus de 50 canaux de démodulation au sein d'un même FPGA. De plus, l'utilisation d'un ordinateur personnel (PC) afin de calculer la solution de navigation a permis d'obtenir des performances encore jamais égalées, permettant une fréquence de mise à jour de la position de 1 kHz en temps réel.

Le récepteur développé a été validé de plusieurs façons, que ce soit par simulation en temps réel à l'aide d'un simulateur de constellation GNSS, ou encore directement à l'extérieur grâce à de vrais signaux GPS et Galileo. Les performances du récepteur ont aussi été étudiées, ce qui a permis de déterminer une précision de 2,2 m RMS horizontalement et de 4,5 m RMS verticalement au niveau du positionnement à l'aide du GPS seulement. De plus, un temps de démarrage de moins de 16 s a pu être mesuré ainsi qu'une capacité de poursuivre les signaux jusqu'à un ratio  $C/N_0$  de 30 dB Hz. Or, toutes ces caractéristiques sont similaires aux récepteurs de navigation que l'on retrouve présentement sur le marché, ce qui indique un très bon fonctionnement pour un premier prototype.



# **DEVELOPMENT OF A REAL-TIME HYBRID GPS AND GALILEO RECEIVER**

SAURIOL, Bruno

## **ABSTRACT**

The 21<sup>st</sup> century announces major changes in the area of satellite radio navigation. The advent of Galileo, the brand new satellite positioning system designed by the European Union, along with the United-States' Global Positioning System (GPS) modernisation, are among those changes. Those systems, brought together under the name "Global Navigation Satellite System" (GNSS), should make available a tremendous number of signals for civilian applications by 2015, thus offering better than ever accuracy and reliability. However, all the possibilities offered by those GNSS systems should be only obtained by tight receiver integration, yet a complex task considering the Research and Development (R&D) technologies currently used in the industry.

This master's thesis presents the development of a real-time hybrid GNSS (GPS and Galileo) receiver. This project is based on a Field-Programmable Gate Array (FPGA), which is a significant advance compared to the expensive and lowly flexible technology currently used in the industry. A FPGA-based approach could actually be an interesting solution to the GNSS signals integration problem.

The receiver has been developed in order to demodulate L1-band (1575.42 MHz) GPS and Galileo signals. However, due to the receiver high level of flexibility, L2-band (1227.6 MHz) and L5-band (1176.45 MHz) signals could be soon processed, for example. The development of the receiver has been done in three major steps: the development of the Radio Frequency (RF) front-end, the Intermediate Frequency (IF) demodulation channels and the baseband navigation solution. The system architecture, mostly digital, has been designed to allow the integration of over 50 demodulation channels in a same FPGA. Besides, the computation of the real-time navigation solution using a Personal Computer (PC) offers astounding performances with position update rates of up to 1 kHz.

The developed receiver has been validated through a series of various real-time tests using a GNSS constellation simulator. Real GPS and Galileo signals have been also directly used outside. The receiver performance is another point that has been studied. Tests allowed us to determine a positioning precision of 2,2 m RMS horizontally and 4,5 m RMS vertically. Moreover, times to first fix below 16 s have been monitored, along with minimal C/N<sub>0</sub> track-down ratios of 30 dB-Hz. All these characteristics are quite similar to current on-market receivers, clearly showing the very good performance of our first prototype receiver.

# TABLE DES MATIÈRES

Page

INTRODUCTION .....	1
CHAPITRE 1 MODERNISATION DE LA NAVIGATION PAR SATELLITE.....	3
1.1 Historique de la navigation par satellite.....	3
1.1.1 Les débuts de la radionavigation.....	3
1.1.2 Systèmes de radionavigation terrestre .....	4
1.1.3 Débuts de la radionavigation par satellite : le système Transit.....	4
1.1.4 Présentation du système GPS.....	5
1.1.5 Présentation du système GLONASS .....	7
1.2 Modernisation de la navigation par satellite .....	8
1.2.1 Modernisation du GPS.....	8
1.2.2 Modernisation de GLONASS .....	12
1.2.3 Galileo.....	12
1.2.4 WAAS.....	16
1.2.5 Beidou, QZSS et IRNSS .....	17
1.2.6 Quelques avantages du GNSS .....	18
1.2.7 Modernisation des récepteurs de navigation.....	18
1.3 Résumé.....	22
CHAPITRE 2 PRINCIPES DE LA NAVIGATION PAR SATELLITE .....	24
2.1 Le principe de la trilatération .....	24
2.2 Le positionnement par satellite de façon concrète.....	27
2.2.1 Distances et pseudo-distances.....	27
2.2.2 Impacts de la vitesse de déplacement sur les signaux.....	29
2.2.3 Positionnement en 2D.....	31
2.2.4 Système de coordonnées géodésiques WGS-84 .....	32
2.2.5 Calcul de la position des satellites à partir des éphémérides .....	33
2.2.6 Calcul des corrections d'horloge des satellites .....	39
2.2.7 Conversion des temps GPS, Galileo et UTC .....	41
2.2.8 Passage des coordonnées cartésiennes aux coordonnées géographiques .....	42
2.2.9 Modèle géopotential EGM96.....	44
2.2.10 Effet de la rotation de la Terre sur les coordonnées des satellites .....	46
2.3 Sources d'erreurs et précision du positionnement par satellite.....	49
2.3.1 Définition des erreurs RMS, à $1\sigma$ et à $2\sigma$ .....	49
2.3.2 Erreurs liées aux segments spatial et de contrôle.....	52
2.3.3 Disponibilité sélective (SA).....	52
2.3.4 Délais ionosphériques .....	53
2.3.5 Délais troposphériques.....	54
2.3.6 Multitrajets.....	55
2.3.7 Erreurs liées au segment utilisateur .....	56
2.3.8 Dilution de précision.....	57



2.3.9	Évolution temporelle de la DOP .....	61
2.3.10	Précision des mesures de vitesse.....	63
2.3.11	Résumé des sources d'erreurs et précision du positionnement par satellite ..	63
2.4	Conclusion .....	65
CHAPITRE 3 DÉMODULATION EN TEMPS RÉEL DES SIGNAUX GPS L1 C/A .....		66
3.1	Introduction au CDMA .....	66
3.1.1	Principes de base du CDMA.....	66
3.1.2	Robustesse du CDMA face au bruit et aux brouilleurs.....	71
3.2	Structure des signaux GPS L1 C/A.....	75
3.3	Architecture générale du récepteur GPS L1 C/A développé .....	79
3.4	Développement de la tête RF .....	83
3.4.1	Conception et réalisation de la tête RF .....	83
3.4.2	Test de la tête RF .....	87
3.4.3	Évolution de la tête RF .....	92
3.5	Mise en oeuvre des canaux de démodulation en fréquence intermédiaire.....	93
3.5.1	Vue d'ensemble .....	93
3.5.2	Théorie sur les boucles à verrouillage de phase.....	96
3.5.3	Conception de la boucle à verrouillage de phase GPS L1 C/A .....	100
3.5.4	Conception de la boucle à verrouillage de délai GPS L1 C/A.....	105
3.5.5	Réalisation des oscillateurs contrôlés numériquement (NCO) .....	110
3.5.6	Réalisation du générateur de codes pseudo-aléatoires C/A .....	114
3.5.7	Réalisation des multiplicateurs .....	116
3.5.8	Réalisation des intégrateurs .....	117
3.5.9	Mise en place de corrélateurs à espacement variable .....	119
3.6	Réalisation et test du contrôleur automatique de gain (AGC) .....	121
3.7	Implémentation d'un microcontrôleur embarqué .....	123
3.7.1	Architecture interne du microcontrôleur.....	123
3.8	Estimation du ratio SNR .....	126
3.9	Mise en œuvre d'un mécanisme d'acquisition des signaux.....	130
3.9.1	Principes généraux sur l'acquisition des signaux .....	130
3.9.2	Détection des signaux lors de l'acquisition .....	133
3.9.3	Phase initiale de poursuite des signaux.....	137
3.10	Synchronisation au message de navigation.....	142
3.11	Mesure du temps satellite.....	146
3.12	Mesure de la pseudo-distance, de l'ADR et de la vitesse relative.....	148
3.13	Mesure en temps réel de la fonction de cross-corrélation.....	150
3.14	Implémentation en temps réel au sein d'un FPGA .....	152
3.14.1	Utilisation des ressources FPGA .....	152
3.14.2	Charges de calcul du microcontrôleur embarqué.....	156
3.15	Conclusion et recommandations .....	158
CHAPITRE 4 DÉMODULATION EN TEMPS RÉEL DES SIGNAUX GALILEO .....		160
4.1	Structure des signaux Galileo E1 .....	160
4.2	Prototypage rapide d'un récepteur Galileo E1 .....	164

4.3	Mise en œuvre des canaux de démodulation en fréquence intermédiaire .....	164
4.3.1	Vue d'ensemble .....	164
4.3.2	Conception de corrélateurs BOC universels .....	167
4.3.3	Réalisation des générateurs de codes pseudo-aléatoires Galileo E1 .....	172
4.4	Particularités du mécanisme d'acquisition des signaux Galileo E1 .....	174
4.5	Implémentation en temps réel au sein d'un FPGA .....	176
4.5.1	Utilisation des ressources FPGA .....	176
4.5.2	Charges de calcul du microcontrôleur embarqué.....	178
4.6	Conclusion et recommandations .....	179
CHAPITRE 5 ARCHITECTURE LOGICIELLE À HAUT NIVEAU .....		181
5.1	Vue d'ensemble de la couche logicielle à haut niveau .....	181
5.2	Gestion des canaux GNSS .....	184
5.3	Calcul de la solution de navigation.....	186
5.3.1	Technique de base pour la détermination de la position.....	186
5.3.2	Solution de navigation par la méthode des moindres carrés.....	189
5.3.3	Solution de navigation à l'aide du filtre Kalman étendu (EKF) .....	190
5.3.4	Modifications apportées au filtre de Kalman étendu .....	198
5.4	Charge de calcul liée à l'architecture logicielle à haut niveau.....	201
5.5	Conclusion et recommandations .....	205
CHAPITRE 6 TESTS EN TEMPS RÉEL .....		207
6.1	Sommaire des tests réalisés.....	207
6.2	Étude du mécanisme de mesure du SNR (test #1) .....	208
6.2.1	Méthodologie du test #1 .....	208
6.2.2	Résultats et analyse du test #1 .....	209
6.3	Test statique en environnement simulé (test #2).....	211
6.3.1	Méthodologie du test #2.....	211
6.3.2	Résultats et analyse du test #2 .....	212
6.4	Test dynamique en environnement simulé (test #3) .....	215
6.4.1	Méthodologie du test #3.....	215
6.4.2	Résultats et analyse du test #3 .....	217
6.5	Évaluation du temps de démarrage en environnement réel (test #4) .....	220
6.5.1	Méthodologie du test #4.....	220
6.5.2	Résultats et analyse du test #4 .....	223
6.6	Test statique en environnement réel (test #5) .....	224
6.6.1	Méthodologie du test #5.....	224
6.6.2	Résultats et analyse du test #5 .....	225
6.7	Test dynamique en environnement réel (test #6) .....	229
6.7.1	Méthodologie du test #6.....	229
6.7.2	Résultats et analyse du test #6 .....	232
6.8	Capture des signaux Galileo (test #7) .....	235
6.8.1	Méthodologie du test #7.....	235
6.8.2	Résultats et analyse du test #7 .....	236
6.9	Test en environnement difficile (test #8) .....	238



6.9.1	Méthodologie du test #8.....	238
6.9.2	Résultats et analyse du test #8 .....	239
6.10	Conclusion et améliorations.....	245
CONCLUSION .....		247
RECOMMANDATIONS .....		249
ANNEXE I RÉSUMÉ DES CARACTÉRISTIQUES DU RÉCEPTEUR HYBRIDE .....		251
ANNEXE II SURVOL DE L'INTERFACE GRAPHIQUE UTILISATEUR (GUI) .....		253
ANNEXE III MÉTHODE DE NEWTON.....		269
ANNEXE IV SPLINES CUBIQUES .....		272
ANNEXE V ERREUR RÉSIDUELLE LIÉE À LA CORRECTION DE LA ROTATION DE LA TERRE.....		279
ANNEXE VI CALCUL DE L'INTÉGRALE SOUS LA FORME $\sin^2(AX)/X^2$ .....		285
ANNEXE VII SCHÉMA DÉTAILLÉ DE LA TÊTE RF DÉVELOPPÉE.....		288
ANNEXE VIII SCHÉMAS DÉTAILLÉS DE L'ARCHITECTURE IF .....		289
ANNEXE IX DESCRIPTION DES FICHIERS DE SAUVEGARDE (LOG FILES) .....		297
BIBLIOGRAPHIE .....		302

## LISTE DES TABLEAUX

	Page
Tableau 2.1	Paramètres de l'ellipsoïde et autres constantes du modèle WGS-84.....33
Tableau 2.2	Paramètres propres aux éphémérides GPS et Galileo .....34
Tableau 2.3	Paramètres de correction d'horloge pour les systèmes GPS et Galileo.....39
Tableau 2.4	Mesures de précision les plus fréquentes.....51
Tableau 2.5	Conversion entre CEP, $1\sigma$ , $2\sigma$ , $3\sigma$ , RMS et 2DRMS .....52
Tableau 2.6	Budget d'erreur pour les mesures de pseudo-distances GPS sans SA .....64
Tableau 3.1	Liste des principaux discriminateurs de PLL.....102
Tableau 3.2	Bande de bruit et coefficients de filtres de boucle numériques typiques .....104
Tableau 3.3	Résumé des paramètres de la PLL .....105
Tableau 3.4	Liste des principaux discriminateurs de DLL .....108
Tableau 3.5	Résumé des paramètres de la DLL .....110
Tableau 3.6	Correspondance des niveaux logiques aux niveaux des signaux.....116
Tableau 3.7	Utilisation des ressources FPGA par le récepteur GPS L1 C/A .....154
Tableau 4.1	Paramètres des générateurs de code du satellite GIOVE-A.....173
Tableau 4.2	Utilisation des ressources FPGA par le récepteur GPS et Galileo.....177
Tableau 6.1	Erreurs de positionnement moyennes et déviations standard lors de la simulation en statique .....214
Tableau 6.2	Erreurs de positionnement moyennes et déviations standard lors de la simulation en dynamique .....217
Tableau 6.3	Mesures du temps de démarrage du récepteur .....223
Tableau 6.4	Erreurs de positionnement moyennes et déviations standards lors des tests statiques en environnement réel.....228
Tableau 6.5	Erreurs de positionnement moyennes et déviations standards lors des tests dynamiques en environnement réel .....234



## LISTE DES FIGURES

	Page
Figure 1.1	Modernisation du spectre en fréquence du GPS.....11
Figure 1.2	Spectre en fréquence Galileo. ....15
Figure 1.3	Première génération de récepteurs GPS.....19
Figure 1.4	Trois récepteurs GPS voués à trois applications différentes.....20
Figure 1.5	Architecture typique d'un récepteur de navigation. ....21
Figure 1.6	Spectres en fréquence Galileo et GPS Bloc III. ....23
Figure 2.1	Principe de la trilatération en 2D. ....24
Figure 2.2	Principe de la trilatération en 3D. ....25
Figure 2.3	Effet des incertitudes sur la trilatération. ....26
Figure 2.4	Mesure du délai de transmission des signaux de navigation. ....27
Figure 2.5	Système d'axes selon la norme WGS-84.....32
Figure 2.6	Représentation des éléments orbitaux.....38
Figure 2.7	Représentation du rayon de courbure d'un ellipsoïde. ....42
Figure 2.8	Ellipsoïde, géoïde et ondulation du géoïde. ....44
Figure 2.9	Ondulation du géoïde EGM96. ....46
Figure 2.10	Changement de référentiel dû à la rotation de la Terre.....47
Figure 2.11	Distribution normale et centrée.....50
Figure 2.12	Réception simultanée d'un signal et d'un multitrajet (à gauche) et réception d'un multitrajet seul (à droite). ....56
Figure 2.13	Exemples de distributions spatiales 2D (de gauche à droite) : bonne géométrie et mauvaise géométrie.....57
Figure 2.14	Nombre de satellites en vue et DOP correspondante à Montréal durant les 28 et 29 juin 2007.....62

Figure 3.1	Modulation d'un train de données par un code d'étalement. ....	67
Figure 3.2	Densité spectrale de puissance d'un train de données avant et après étalement spectral (cas général). ....	67
Figure 3.3	Démodulation d'un signal CDMA transmis dans un canal bruité. ....	72
Figure 3.4	Étalement d'un brouilleur lors de la démodulation d'un signal CDMA. ....	75
Figure 3.5	Schéma de modulation et spectre fréquentiel des signaux GPS L1 C/A. ..	76
Figure 3.6	Puissance minimale en ciel dégagé des signaux GPS lors de leur réception au sol. ....	76
Figure 3.7	Relation de temps entre le code C/A et les données GPS. ....	77
Figure 3.8	Segmentation du message de navigation GPS. ....	77
Figure 3.9	Format des mots TLM et HOW. ....	78
Figure 3.10	Structure des sous-frames du message de navigation GPS. ....	79
Figure 3.11	Architecture générale du récepteur GPS L1 C/A développé. ....	80
Figure 3.12	Photographie de la carte de développement. ....	81
Figure 3.13	Diagramme en couches du récepteur de navigation développé. ....	82
Figure 3.14	Photographie du PC de développement. ....	83
Figure 3.15	Schéma de la tête RF développée. ....	84
Figure 3.16	Photographie de la version préliminaire de la tête RF. ....	85
Figure 3.17	Schéma de branchement des horloges et LO. ....	86
Figure 3.18	Lobe GPS central mesuré à la sortie de la tête RF. ....	87
Figure 3.19	Mesure de la bande passante de la tête RF. ....	89
Figure 3.20	Dérive de l'horloge Stanford CG635 durant la nuit du 17 au 18 avril 2007. ....	90
Figure 3.21	Dérive de l'horloge Symmetricom 8040 durant la nuit du 18 au 19 avril 2007. ....	90
Figure 3.22	Photographie de la plateforme de développement finale. ....	92

Figure 3.23	Architecture IF générale. ....	94
Figure 3.24	Architecture d'un canal de démodulation.....	95
Figure 3.25	Schéma bloc d'une PLL classique. ....	97
Figure 3.26	PLL linéarisée autour d'un point d'équilibre.....	97
Figure 3.27	Schéma bloc de la boucle à verrouillage de phase développée. ....	100
Figure 3.28	Courbes de réponse des principaux discriminateurs de PLL. ....	102
Figure 3.29	Schémas bloc de filtres de boucles numériques d'ordre 0 (a), d'ordre 1 (b) et d'ordre 2 (c). ....	103
Figure 3.30	Disposition des corrélateurs en avance, en phase et en retard sur la fonction d'autocorrélation des signaux GPS L1 C/A. ....	106
Figure 3.31	Schéma bloc de la boucle à verrouillage de délai développée. ....	107
Figure 3.32	Courbes de réponse des principaux discriminateurs de DLL. ....	108
Figure 3.33	Schéma bloc des NCO de code (gauche) et de porteuse (droite).....	110
Figure 3.34	Synthèse de l'horloge de DLL.....	113
Figure 3.35	Générateur de codes pseudo-aléatoires GPS C/A.....	115
Figure 3.36	Démonstration d'une commande d'avance de deux bribes.....	116
Figure 3.37	Schéma bloc d'un multiplicateur 1xN bits. ....	117
Figure 3.38	Schéma bloc d'un intégrateur. ....	118
Figure 3.39	Mise en place de corrélateurs à espacement variable. ....	120
Figure 3.40	Schéma bloc du contrôleur de gain automatique. ....	122
Figure 3.41	Schéma de branchements pour le test en temps réel de l'AGC.....	122
Figure 3.42	Réponse temporelle de l'AGC à un stimulus de type échelon. ....	123
Figure 3.43	Architecture générale du microcontrôleur embarqué.....	124
Figure 3.44	Ordinogramme des programmes du microcontrôleur embarqué. ....	125
Figure 3.45	Distribution statistique en sortie des corrélateurs pour divers $C/N_0$ . ....	127



Figure 3.46	Schéma bloc du canal de calibration.....	129
Figure 3.47	Déviation standard du ratio SNR due au bruit dans le canal. ....	130
Figure 3.48	Procédure d'acquisition séquentielle des signaux. ....	131
Figure 3.49	Diagramme d'états du programme d'acquisition et de poursuite. ....	133
Figure 3.50	Algorithme de détection à pas variables des signaux. ....	134
Figure 3.51	Probabilité de détection d'un signal en sortie de l'algorithme de recherche à pas variables .....	135
Figure 3.52	Détermination de la probabilité de fausse alarme par essai simple en sortie des corrélateurs ( $P_{fa}$ ). ....	137
Figure 3.53	Courbe de réponse spectrale normalisée en sortie des intégrateurs. ....	139
Figure 3.54	Mécanisme d'affinage par dérivation de la fréquence DLL .....	140
Figure 3.55	Accélération radiale maximale pouvant être encaissée au cours de la phase initiale de poursuite des signaux. ....	142
Figure 3.56	Exemples de synchronisation aux bits de données par recherche de la transition la plus abrupte. ....	143
Figure 3.57	Probabilité de synchronisation aux bits de données ( $C/N_0 = 30$ dB Hz) .	144
Figure 3.58	Probabilité d'erreur de synchronisation ( $C/N_0 = 30$ dB Hz). ....	145
Figure 3.59	Extraction des mesures pour le calcul de $t_{SV}$ . ....	147
Figure 3.60	Extraction des mesures pour le calcul de l'ADR.....	150
Figure 3.61	Diagramme bloc du corrélateur de mesure. ....	151
Figure 3.62	Principe de fonctionnement du corrélateur de mesure.....	151
Figure 3.63	Mesure de la fonction de cross-corrélations des satellites GPS PRN #1 (gauche) et GIOVE-A (droite) à Montréal le 26 mai 2007.....	152
Figure 3.64	Architecture simplifiée d'une cellule logique de Virtex-4 de Xilinx. ....	153
Figure 3.65	Répartition totale des LUT pour le récepteur GPS L1 C/A. ....	155
Figure 3.66	Répartition des LUT au niveau d'un canal de démodulation GPS L1 C/A.....	155

Figure 3.67	Charge de calcul du Microblaze pour un récepteur à 24 canaux. ....	157
Figure 4.1	Schéma de modulation et spectre fréquentiel des signaux Galileo E1. ...	161
Figure 4.2	Exemple de modulation TMBOC(6,1,2/11). ....	161
Figure 4.3	Segmentation du message de navigation Galileo E1. ....	163
Figure 4.4	Structure des sous-frames du message de navigation Galileo E1. ....	163
Figure 4.5	Architecture IF générale du récepteur GNSS développé. ....	165
Figure 4.6	Architecture d'un canal de démodulation Galileo E1. ....	166
Figure 4.7	Fonctions d'autocorrélation BOC(1,1) et BOC(2,1). ....	167
Figure 4.8	Fonctions de cross-corrélation BOC(1,1) et BOC(2,1) telles que mesurées en sortie des corrélateurs. ....	168
Figure 4.9	Suppression de l'ambiguïté de corrélation par génération de sous-porteuses en phase et en quadrature. ....	168
Figure 4.10	Fonctions de cross-corrélation BOC(1,1) obtenues pour des sous-porteuses en phase et en quadrature. ....	169
Figure 4.11	Multiplexage des sous-porteuses en phase et en quadrature. ....	170
Figure 4.12	Erreur en sortie du discriminateur enveloppe AMRN lorsqu'appliqué aux corrélateurs BOC universels développés. ....	172
Figure 4.13	Générateurs de codes pseudo-aléatoires Galileo E1-B/C. ....	173
Figure 4.14	Diagramme d'état du programme d'acquisition et de poursuite Galileo. .	175
Figure 4.15	Répartition totale des LUT pour le récepteur hybride GPS et Galileo. ...	178
Figure 4.16	Répartition des LUT au niveau d'un canal de démodulation Galileo E1.	178
Figure 5.1	Schéma bloc de la couche logicielle à haut niveau. ....	181
Figure 5.2	Schéma bloc du segment de gestion et de solution de navigation. ....	182
Figure 5.3	Ordinogramme simplifié du mécanisme de gestion des canaux. ....	185
Figure 5.4	Schéma synthèse du filtre de Kalman étendu. ....	191
Figure 5.5	Affichage du gestionnaire de tâche de Windows lors de l'exécution de la solution de navigation à 20 Hz. ....	202

Figure 5.6	Capture d'écran lors de l'exécution de la solution de navigation à 1 kHz.....	202
Figure 5.7	Estimation de la charge CPU exigée par la solution de navigation. ....	204
Figure 6.1	Simulateur GS7700 de Spirent.....	208
Figure 6.2	Montage pour l'étude du mécanisme de mesure du SNR. ....	209
Figure 6.3	Évaluation de la linéarité du mécanisme de mesure de SNR.....	210
Figure 6.4	Satellites visibles lors de la simulation en statique. ....	211
Figure 6.5	Erreur de positionnement horizontale et HDOP. ....	213
Figure 6.6	Erreur de positionnement verticale et VDOP. ....	213
Figure 6.7	Détails de la trajectoire simulée. ....	216
Figure 6.8	Représentation 3D de la trajectoire simulée. ....	216
Figure 6.9	Erreurs de positionnement nord-sud, ouest-est et verticale. ....	218
Figure 6.10	Correspondance entre les erreurs de vitesse et les stress de jerk. ....	218
Figure 6.11	Localisation des antennes GPS sur le toit de l'ÉTS; vues direction sud-est (gauche) et nord-ouest (droite). ....	221
Figure 6.12	Photographie satellite de l'ÉTS. ....	222
Figure 6.13	Amplificateur (gauche) et antenne (droite). ....	222
Figure 6.14	Présence de données aberrantes en sortie de la solution de navigation. ..	225
Figure 6.15	Distribution horizontale des solutions de navigation. ....	227
Figure 6.16	Altitudes relatives en fonction du temps.....	227
Figure 6.17	Photographie du véhicule de test. ....	230
Figure 6.18	Montage employé lors des tests dynamiques en environnement réel. ....	230
Figure 6.19	Trajectoire de test en environnement réel. ....	231
Figure 6.20	Superposition des solutions de navigation horizontales. ....	233
Figure 6.21	Différence entre les solutions de navigation. ....	233



Figure 6.22	Trajectoires des satellites GIOVE-A et GPS PRN #2 telles qu'observées à Montréal, le 4 octobre 2007. ....	236
Figure 6.23	Mesures du $C/N_0$ et du décalage Doppler de signaux GPS L1 C/A et Galileo E1 à Montréal, le 4 octobre 2007. ....	237
Figure 6.24	Fonction de corrélation du signal E1 reçu de GIOVE-A. ....	238
Figure 6.25	Trajectoire de test en environnement difficile. ....	239
Figure 6.26	Satellites visibles à Montréal, le 3 octobre 2007, entre 14h30 et 14h50. ....	240
Figure 6.27	Nombre de satellites verrouillés en fonction du temps écoulé. ....	240
Figure 6.28	Trajectoire de test et résultats de positionnement en canyon urbain à Montréal. ....	241
Figure 6.29	Résultats de positionnement vertical en canyon urbain à Montréal. ....	243
Figure 6.30	Photographies en direction nord des rues Jeanne-Mance (gauche) et Crescent (droite) à la hauteur de la rue Ste-Catherine. ....	243
Figure 6.31	Probabilité de verrouillage des satellites en canyon urbain. ....	244

## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

2D	Deux dimensions
3D	Trois dimensions
AGC	Automatic Gain Control
AltBOC	Alternative BOC
AMR	Avance moins retard
BOC	Binary Offset Carrier
BOCcos	Modulation BOC à phase cosinusoidale
BPSK	Binary Phase Shift Keying
CAN	Convertisseur analogique à numérique
CDMA	Code Division Multiple Access
CNA	Convertisseur numérique à analogique
CEP	Circular Error Probable
$C/N_0$	Carrier to Noise Power Density Ratio
C/A	Coarse Acquisition
P	Précision
CRC	Cyclic Redundancy Check
DoD	Department of Defense
DLL	Delay Locked Loop
DSP	Digital Signal Processor
E5	1191,795 MHz
E5a	voir L5



E5b	1207,14 MHz
E6	1278,75 MHz
AMRN	Avance moins retard normalisée
ESA	European Space Agency
ÉTS	École de technologie supérieure
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GEO	Geostationary Earth Orbit
GLONASS	GLObal'naya NAVigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRN	Groupe de recherche en navigation
GUI	Graphical User Interface
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
ITU	International Telecommunication Union
L1	1575,42 MHz
L2	1227,60 MHz
L5	1176,45 MHz
LACIME	Laboratoire de communication et d'intégration de la microélectronique
LEO	Low Earth Orbit
LUT	Look-Up Table

MBOC	Multiplexed BOC Modulation
MEO	Medium Earth Orbit
OS	Open Service
PLL	Phase Locked Loop
PPS	Precise Positioning Service
PRN	Pseudo Random Noise
PVT	Position-vitesse-temps
QPSK	Quadrature Phase Shift Keying
RF	Radiofréquence
RMS	Root Mean Square
SA	Selective Availability
SNR	Signal to Noise Ratio
SoL	Safety of Life
SDR	Software Defined Radio
SPS	Standard Positioning Service
UTC	Temps universel coordonné
URSS	Union des républiques socialistes soviétiques
VLF	Very Low Frequency
VHF	Very High Frequency

## LISTE DES SYMBOLES ET UNITÉS DE MESURE

---

### UNITÉS GÉOMÉTRIQUES

#### Longueur

km	kilomètre
m	mètre
dm	décimètre
cm	centimètre
mm	millimètre
µm	micromètre

#### Angle

rad	radian
°	degré
'	minute d'arc
''	seconde d'arc

---

### UNITÉS MÉCANIQUES

#### Vitesse

m/s	mètre par seconde
km/h	kilomètre par heure
rad/s	radian par seconde

#### Accélération

m/s <sup>2</sup>	mètre par seconde carré
rad/s <sup>2</sup>	radian par seconde carré

#### Jerk

m/s <sup>3</sup>	mètre par seconde cube
rad/s <sup>3</sup>	radian par seconde cube

---

### UNITÉS DE TEMPS

h	heure
min	minute
s	seconde
ms	milliseconde
µs	microseconde
ns	nanoseconde
ps	picoseconde

---

### UNITÉS DE FRÉQUENCE

GHz	gigahertz
MHz	mégahertz

kHz	kilohertz
Hz	hertz

---

### UNITÉS INFORMATIQUES

#### Stockage de données

GB	gigaoctet
MB	mégaoctet
kB	kiloctet
B	octet
Gb	gigabit
Mb	mégabit
kb	kilobit
b	bit

#### Taux de transfert

MB/s	mégaoctet par seconde
kB/s	kiloctet par seconde
Mbs	mégabit par seconde
kbps	kilobit par seconde
bps	bit par seconde
sps	symbole par seconde
Mcps	mégabryte par seconde
cps	caractère par seconde

---

### UNITÉS MÉTÉOROLOGIQUES

#### Température

°C	degré Celsius
K	kelvin

#### Pression

kPa	kilopascal
Pa	pascal

---

### SYMBOLES DE RATIOS

%	pourcent
PPM	partie par million
PPB	partie par milliard
dB	décibel
dB Hz	décibel hertz

---

## INTRODUCTION

Le début du 21<sup>e</sup> siècle marque un tournant dans le domaine du positionnement par satellite. Alors que depuis près de 20 ans le système de positionnement par satellite GPS (*Global Positioning System*) des États-Unis était le seul à opérer de façon fiable et précise, les années 2000 s'annoncent tout autrement avec la venue de plusieurs nouveaux systèmes. Celui le plus attendu, Galileo, le système de positionnement par satellite de l'Union européenne, se veut encore plus fiable et précis que le GPS actuel. Considérant le nombre croissant d'applications liées au GPS et leurs retombées économiques à travers le monde (les ventes de récepteurs GPS auraient à elles seules dépassé les 16 milliards de dollars US en 2003) (Misra et Enge, 2006), on peut donc comprendre l'intérêt que suscite l'avènement d'un nouveau système tel que Galileo.

Le GRN, le Groupe de recherche en navigation du laboratoire LACIME (Laboratoire de communication et d'intégration de la microélectronique) de l'École de technologie supérieure (ÉTS), est conscient des nombreux avantages qu'apporteront les systèmes de positionnement du futur. Afin d'en explorer les avenues, le GRN a donc entrepris un projet appelé *Software Defined Navigation* (SDN) et dont un des buts est de développer un outil de navigation radio logicielle dédié à la recherche. La première étape à ce projet fut la réalisation d'une chaîne de simulation GPS et Galileo complète, c'est-à-dire comprenant le récepteur de positionnement et les divers émetteurs (satellites) de signaux GPS et Galileo (Chavagnieux, 2006). Le travail présenté dans ce document est donc l'étape suivante de ce projet.

Le but premier de ce travail est la mise en œuvre en temps réel d'un récepteur de positionnement hybride GPS et Galileo. De façon plus précise, le travail comprend le développement de 24 canaux de réception GPS C/A (*Coarse Acquisition*) et Galileo en bande de fréquence L1 (1575,42 MHz) au sein d'un FGPA (*Field Programmable Gate Array*). Le tout est complété par l'élaboration d'algorithmes de gestion des canaux et de calcul de la position en temps réel. Bien que cette description soit très sommaire, plus de détails seront fournis tout au long de ce mémoire au sujet de l'architecture du récepteur développé.



Ce mémoire est divisé en 6 chapitres. Le premier chapitre fait un bref suivi historique de la navigation par satellite avant de tracer un portrait global de la situation actuelle. Le second chapitre introduit quant à lui quelques concepts de bases servant à la navigation par satellite, tels que la trilatération, les mesures de pseudo-distances et le calcul de la position des satellites à partir des éphémérides.

Par la suite, le chapitre 3 entre dans les détails de la conception des canaux de réception GPS. On y décrit la chaîne de démodulation des signaux, en commençant par la tête radio fréquence (RF) et en poursuivant avec l'étage de traitement des signaux numériques en bande de fréquence intermédiaire (*Intermediate Frequency* - IF). Le traitement des signaux en bande de base y est aussi abordé. Le chapitre 4 aborde ensuite la conception détaillée des canaux de réception Galileo. Les sections RF, IF et en bande de base y sont donc revues en mettant l'emphasis sur les différences entre les canaux GPS et Galileo.

Au chapitre 5, on effectue un survol de la gestion à haut niveau des canaux. On y présente aussi les algorithmes de navigation les plus fréquemment utilisés, tels que la méthode des moindres carrés et le filtre de Kalman. Finalement, le chapitre 6 expose une série de tests visant à mesurer les performances du récepteur développé dans divers environnements et conditions. Les résultats y sont ensuite présentés et discutés. Une section supplémentaire clôt enfin le mémoire avec quelques conclusions et recommandations générales.

# CHAPITRE 1

## MODERNISATION DE LA NAVIGATION PAR SATELLITE

### 1.1 Historique de la navigation par satellite

#### 1.1.1 Les débuts de la radionavigation

Connaître sa position avec précision n'est pas un désir qui date d'hier. En effet, l'homme tâche depuis plus de 2000 ans de mettre au point des techniques afin de tracer des cartes et d'y déterminer sa position de la façon la plus juste possible (Williams, 1992). Au tout début, la navigation se faisait principalement à l'aide des astres célestes, que ce soit les étoiles, le soleil ou la lune, ou encore à l'aide de repères le long des côtes. Cependant, de telles méthodes s'avérèrent plutôt difficiles en haute mer vu l'absence de points de repère et puisqu'aucun dispositif n'avait encore été développé pour mesurer le temps avec précision. Il fallut donc attendre le 18<sup>e</sup> siècle pour voir apparaître des horloges suffisamment précises pour permettre ce type de navigation. Entre temps, les marins, tels que Christophe Colomb, ont dû avoir recours à des méthodes de navigation à l'estime (mieux connu sous le nom anglais de *dead reckoning*), qui consistent à mesurer la vitesse et la direction d'un navire afin d'estimer la distance parcourue. Pour y arriver, l'utilisation d'un loch (une corde nouée attachée à un flotteur à une de ses extrémités), d'un sablier ainsi que d'une boussole était nécessaire. La précision d'une telle méthode laissait bien sûr place à l'amélioration.

Vu les progrès considérables apportés aux horloges et au sextant durant les 18<sup>e</sup> et 19<sup>e</sup> siècles, la navigation à l'aide des astres célestes devint sans contredit la méthode de prédilection pour la marine. Une telle méthode exigeait toutefois un travail mathématique de longue haleine, ce qui ne dérangeait pas vu la faible vitesse des bateaux à cette époque (Emmott, 2006). Le problème devint cependant tout autre avec l'avènement de l'avion au 20<sup>e</sup> siècle, et des simplifications durent être apportées afin d'accélérer les calculs. On assista alors à l'introduction d'un nouveau métier, celui de navigateur, dont l'unique tâche était de faire les calculs de navigation essentiels au pilote.

Avec l'invention de la radio vers le début des années 1900, on ne tarda pas à remplacer la navigation à l'aide des astres par la radionavigation, plus précise et plus fiable. La Seconde Guerre mondiale accéléra de beaucoup ce processus avec l'apparition de plusieurs systèmes de guidage d'avion, tels que le VOR (*VHF Omnidirectional radio Range*), l'ILS (*Instrument Landing System*) et le MLS (*Microwave Landing System*). Ces systèmes opèrent sur le principe de radiobalises à faible portée et permettent encore aujourd'hui aux avions militaires et civils de se déplacer et d'atterrir en toute sécurité.

### 1.1.2 Systèmes de radionavigation terrestre

Le premier véritable système de radionavigation à longue portée à voir le jour, le Loran (*Long-Range Navigation System*), fit son apparition durant la Seconde Guerre mondiale. Ce système, aussi appelé Loran-A, fut par la suite graduellement remplacé par le Loran-C afin d'améliorer la portée et la précision du système. Le Loran-C, développé vers la fin des années 1950, est encore utilisé à ce jour et offre une précision de positionnement en deux dimensions (2D) de l'ordre de 250 m RMS (*Root Mean Square*) (Misra et Enge, 2006). Néanmoins, le problème du Loran est qu'il repose sur un système de radiobalises multiples, rendant ainsi complexe et coûteux le déploiement du système à l'échelle mondiale. Un autre système développé au même moment que le Loran-C, appelé Omega, releva par contre cet exploit en étant le premier système de positionnement à offrir un service continu à travers le monde entier. Omega reposait sur un système de radiobalises à très longue portée dans la bande VLF (*Very Low Frequency*) et offrait une précision 2D RMS de l'ordre de 1 à 2 km. Le système fut cependant mis hors service en 1997 puisque jugé désuet face au système GPS.

### 1.1.3 Début de la radionavigation par satellite : le système Transit

Vint enfin l'ère spatiale, qui commença en 1957 avec le lancement du premier satellite, Sputnik I, par l'Union soviétique (URSS). Ceci engagea une course à l'espace entre les États-Unis et l'Union soviétique et accéléra considérablement le développement des technologies spatiales. Par conséquent, le premier système de positionnement par satellite, appelé Transit, fut développé et déployé en moins de six ans par les États-Unis, pour être opérationnel vers



1964 (Misra et Enge, 2006). Le système reposait sur une constellation de quatre à sept satellites placés en orbite basse (LEO – *Low Earth Orbit*) à une altitude de 1100 km, et par conséquent, un seul satellite pouvait être vu à la fois avec des passages d’une durée de 10 à 20 minutes, chacun espacé de 100 minutes. La technique consistait essentiellement à déterminer la position d’un utilisateur à partir des mesures de Doppler effectuées sur les signaux reçus. Pour un utilisateur stationnaire ou se déplaçant lentement (le but principal étant alors de guider une nouvelle classe de sous-marins), ceci permettait d’obtenir une précision 2D RMS d’environ 25 m (Misra et Enge, 2006). Le système n’était cependant pas conçu pour répondre aux dynamiques élevées et n’aurait pas fonctionné à bord de voitures ou d’avions par exemple. Le système Transit fut officiellement mis hors service en 1996.

#### 1.1.4 Présentation du système GPS

Le GPS fait son apparition aux États-Unis au courant des années 1970 suite à la fusion de deux projets de la U.S. Navy et de l’Air Force. Le projet porte alors le nom officiel de NAVSTAR GPS. Bien que le premier satellite fût lancé en 1978, on ne déclara le projet officiellement opérationnel qu’en 1995. Contrairement aux systèmes de navigation précédents, le GPS permet d’obtenir une position en trois dimensions (3D – i.e. que l’altitude d’un utilisateur peut être aussi déterminée) à partir de plusieurs satellites en simultanée. Le GPS peut aussi fonctionner pour des dynamiques élevées, telles qu’à bord des autos, des avions et des engins spatiaux. De plus, tous les satellites transmettent leurs signaux de façon simultanée sur les mêmes bandes de fréquence via une technique de modulation appelée CDMA (*Code Division Multiple Access*) (chapitre 3).

La précision horizontale RMS (ou 2D) du GPS s’estime à 5 m et la précision verticale à 7,5 m (Misra et Enge, 2006). Le GPS étant tout d’abord à vocation militaire, le Département de la Défense des États-Unis (DoD – *Department of Defense*) se réserve donc le droit de dégrader la précision des signaux à tout moment pour des utilisateurs non autorisés, tels que les civils par exemple. Le mécanisme employé pour dégrader les signaux GPS, la disponibilité sélective (SA - *Selective Availability*), est abordé plus en détail au chapitre 2. De



plus, les civils n'ont généralement accès qu'au service de positionnement standard (SPS – *Standard Positioning Service*) grâce au code d'étalement C/A, alors que le service de précision (PPS – *Precise Positioning Service*) via le code P(Y) est réservé aux militaires et à d'autres utilisateurs autorisés par le DoD (les codes d'étalement sont abordés au chapitre 3).

Le système GPS est séparé en trois segments principaux : le segment spatial, le segment de contrôle et le segment utilisateur. Ce dernier segment représente essentiellement les récepteurs de navigation employés pour obtenir une position à partir des satellites GPS. Le segment spatial consiste quant à lui en une constellation de 24 satellites GPS opérationnels répartis sur six orbites circulaires différentes et inclinées à  $55^\circ$  par rapport au plan équatorial. Les satellites sont situés sur des orbites moyennes (MEO – *Medium Earth Orbit*) à une altitude de 20180 km et parcourues en 11 h 58 min, faisant en sorte que chaque satellite retrouve la même position dans le ciel au bout d'un jour sidéral ( $\approx 23$  h 56 min). Bien que la constellation complète soit théoriquement constituée de quatre satellites opérationnels et d'un satellite de secours par orbite, pour un maximum de 30 satellites, on compte en 2007 un total de 31 satellites, dont 30 sont actifs (USNO, 2007). Ce surplus de satellites s'explique par la très grande longévité des satellites GPS, qui ont pour la plupart dépassé le double de leur durée de vie estimée.

Vient enfin le segment de contrôle, qui est constitué d'une station maîtresse, contrôlée par l'U.S. Air Force, et d'une série de stations monitrices réparties autour du globe. Le rôle principal du segment de contrôle est de surveiller et de maintenir opérationnel le segment spatial. Ceci implique entre autres de surveiller les orbites et les horloges des satellites GPS afin de prévoir les éphémérides ainsi que les paramètres d'horloges qui seront transmis par les satellites (chapitre 2). Alors que le nombre total de stations de contrôle s'élevait initialement à six, 11 nouvelles stations de la National Geospatial-Intelligence Agency (NGA) ont été ajoutées au cours des années 2005 et 2006 (Manning, 2005). Ceci porte donc le nombre total de stations à 17 mondialement pour une surveillance accrue des satellites GPS.

### 1.1.5 Présentation du système GLONASS

L'équivalent soviétique du GPS, GLONASS (*GLO*bal'*naya N*avigatsionnaya *S*putnikovaya *Sistema*, un acronyme russe se traduisant par « système de navigation global par satellite »), fait son entrée pratiquement au même moment que le GPS. GLONASS fut précédé de deux systèmes identiques à Transit : *Parus*, pour la marine militaire soviétique, et *Tsikada*, pour la marine marchande (Misra et Enge, 2006). Contrairement au GPS, le service de précision standard de GLONASS (SPS) fut offert à la communauté civile dès 1988. Ceci rendit GLONASS très attrayant aux utilisateurs civils vers le milieu des années 1990 alors que la constellation était presque complétée et que la SA était employée sur les signaux GPS.

La précision de GLONASS est similaire à celle du GPS (sans SA), bien que sa fiabilité ait considérablement diminué au cours des dernières années dû à plusieurs pannes et autres problèmes techniques (Cook, 1997). Le segment spatial de GLONASS est constitué de 24 satellites MEO (21 actifs et trois de secours) répartis sur trois orbites circulaires inclinées à  $64,8^\circ$  par rapport au plan équatorial. Les satellites ont une altitude de 19 100 km et accomplissent une révolution complète en 11 h 15 min. Par conséquent, la position des satellites dans le ciel se répète au bout d'une période de huit jours. Quant au segment de contrôle, celui-ci est composé, à l'instar du GPS, d'une station maîtresse et d'une série de stations monitrices. Les stations monitrices sont toutefois aujourd'hui restreintes à l'intérieur du territoire russe.

À l'inverse du GPS, les satellites GLONASS ont une durée de vie très courte et ils doivent être remplacés très fréquemment. Un manque de ressources financières vers la fin des années 1990 a donc rapidement entraîné une réduction de la constellation de satellites. La Russie a toutefois entrepris de rétablir la situation et prévoit que GLONASS sera entièrement opérationnel pour 2010-2011 avec une constellation de 24 satellites (Misra et Enge, 2006). Pour l'instant, plusieurs satellites sont encore manquants avec seulement 11 en activité en date du mois d'août 2007 (CSIC, 2007).

## 1.2 Modernisation de la navigation par satellite

### 1.2.1 Modernisation du GPS

Tel que mentionné précédemment, la durée de vie des satellites GPS a de loin dépassé les attentes que s'était fixé le DoD. Ceci l'a donc obligé à repousser à plus tard la modernisation de la constellation GPS via le lancement des blocs IIF et III. Voici un résumé des blocs lancés et de ceux qui sont prévus prochainement (Misra et Enge, 2006; USNO, 2007) :

- Bloc II : Le premier bloc de satellites GPS opérationnels a été lancé entre 1989 et 1990 (le bloc I était constitué de 12 satellites de démonstration lancés entre 1978 et 1985). La durée de vie des satellites était initialement estimée à 7,3 ans. Le dernier satellite du bloc II a cessé d'être utilisable en novembre 2007, soit plus de 16 ans après son lancement. Le bloc II contenait neuf satellites au total.
- Bloc IIA : Le second bloc de satellites a été lancé entre 1990 et 1997 afin de compléter la constellation. La durée de vie des satellites était aussi estimée à 7,3 ans. Des 18 satellites lancés, encore 15 sont en orbite, dont 14 actifs. Ils sont aujourd'hui âgés de 9 et 16 ans.
- Bloc IIR : Le principal objectif de ce bloc était de regarnir la constellation GPS au fur et à mesure qu'elle vieillissait. Vu l'impressionnante longévité des blocs II/IIA, le DoD a pu toutefois retrancher huit satellites du bloc IIR afin de les moderniser. Le bloc IIR ne compte donc que 13 satellites lancés entre 1997 et 2004. Les satellites sont tous actifs aujourd'hui, à l'exception d'un dont le lancement a été raté en 1997. Les satellites du bloc IIR sont dotés de systèmes de mesure et de communication intersatellitaire, leur permettant ainsi de basculer en mode AUTONAV (*AUTOnomous NAVigation*) advenant une perte de communications avec les stations de contrôle au sol. Le bloc IIR peut donc fonctionner de façon autonome jusqu'à 180



jours sans dégradation de précision. La durée de vie des satellites était initialement estimée à 7,8 ans.

- Bloc IIR-M : Ce bloc est constitué de huit satellites retranchés du bloc IIR pour être modernisés. Quatre satellites ont été lancés entre 2005 et 2007 et le lancement de quatre autres satellites reste encore à être déterminé. Les satellites du bloc IIR-M sont les premiers à transmettre les signaux L2C, de nouveaux signaux GPS civils sur la bande L2 (1227,60 MHz). De nouveaux signaux militaires sont aussi transmis sur les bandes L1 et L2 et font appel à des codes d'étalement plus robustes que l'ancien code militaire P(Y). Ces codes sont appelés M-codes.
- Bloc IIF : Le déploiement de ce bloc est prévu pour 2008 (Leveson, 2006). Bien que pour l'instant rien ne soit définitif, un total de 12 satellites est envisagé. En plus de transmettre les signaux L2C et les M-codes, les satellites du bloc IIF devraient transmettre de nouveaux signaux civils à la fréquence L5 (1176,45 MHz). Les signaux L5, plus robustes et plus fiables que les signaux précédents, pourraient ainsi être employés dans des applications de sauvegarde de la vie (SoL – *Safety of Life*), des applications critiques où la vie de personnes peut être mise en danger.
- Bloc III : Ce bloc est présentement en phase de développement, avec un déploiement prévu pour 2013 (Stansell, 2006). Parmi les objectifs fixés, on compte l'ajout de nouveaux signaux sur la bande L1, appelés L1C, dédiés aux utilisateurs civils. Grâce à un accord signé en 2004 entre les États-Unis et l'Union européenne, on vise à ce que les signaux L1C aient le même spectre fréquentiel que les signaux Galileo L1 (abordés plus loin). Ceci permettrait de rendre davantage interopérables les systèmes GPS et Galileo.

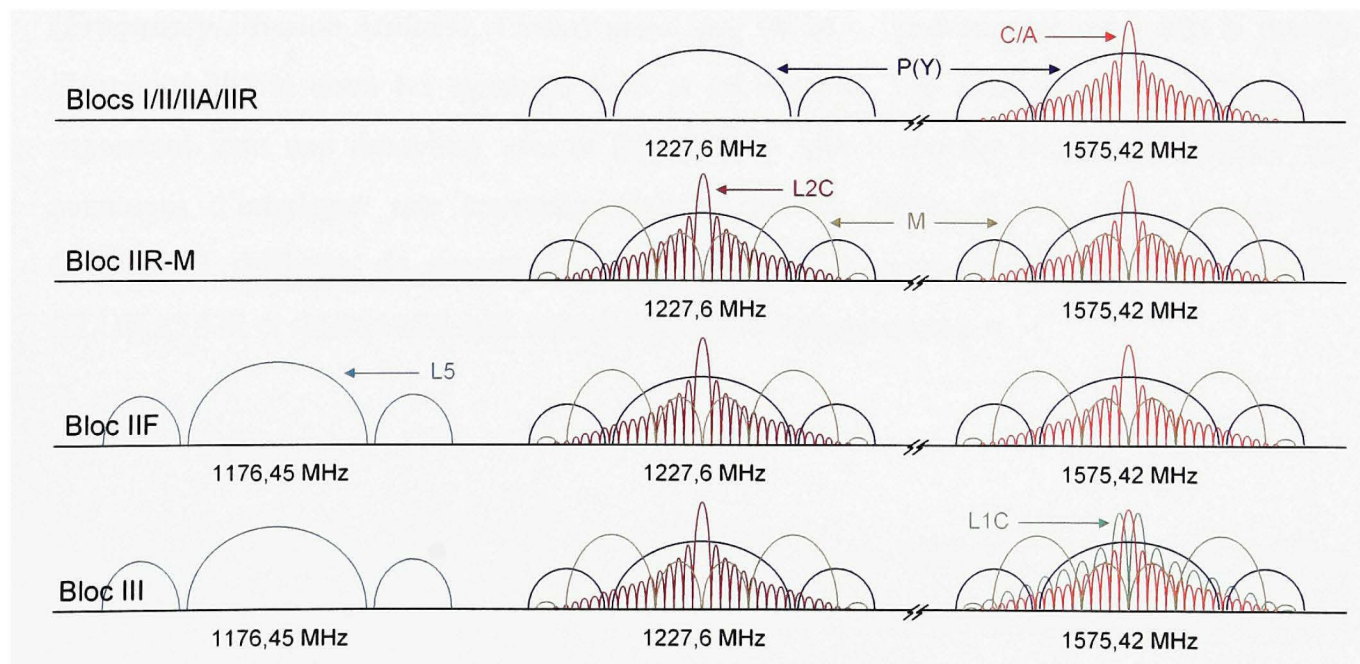
La modernisation du GPS s'est donc entamée en 2005 avec le lancement du premier satellite IIR-M. Ceci a enfin permis aux utilisateurs civils de bénéficier de signaux sur la bande L2 grâce à l'ajout des signaux L2C. Parmi les particularités des signaux L2C, on note

l'utilisation de codes d'étalement plus longs que le code C/A actuel et l'ajout d'un canal pilote en plus du canal de données (soit un canal servant uniquement à l'acquisition et à la poursuite des signaux). Ceci, combiné à une puissance accrue des signaux et d'un mécanisme de correction des erreurs de type FEC (*Forward Error Correction*), permettra d'accélérer l'acquisition des signaux, en plus d'en augmenter la robustesse et la précision (Leveson, 2006). La navigation à l'intérieur des bâtiments s'en retrouvera aussi facilitée, sans oublier qu'une fois combinés aux signaux GPS L1, les signaux L2C permettront à un utilisateur civil de bénéficier des corrections ionosphériques (chapitre 2).

La bande de fréquence L2 n'étant pas sécurisée contre les interférences, il n'est donc pas possible d'employer les signaux L2C pour l'aviation ou pour d'autres applications de type SoL (Misra et Enge, 2006). Il en sera tout autrement des signaux GPS L5 qui seront transmis par le prochain bloc IIF. En effet, les signaux L5 (de même que L1) sont situés à l'intérieur de la bande de fréquence ARNS (*Aeronautical Radio Navigation Services*), une bande mondialement réservée, ce qui leur confère une certaine protection contre les interférences. Les signaux L5 possèdent eux aussi des codes d'étalement plus longs que GPS L1 C/A, une correction d'erreur de type FEC ainsi qu'une puissance accrue. Une modulation QPSK (*Quadrature Phase Shift Keying*) de la porteuse est toutefois employée au lieu du BPSK (*Binary Phase Shift Keying*), ce qui permet de créer un canal pilote sur la voie en quadrature en plus du canal de données sur la voie en phase (ARINC Incorporated, 2001). La fréquence du code d'étalement est aussi choisie dix fois plus rapide que celles des codes L1 C/A et L2C, ce qui résulte en un spectre fréquentiel semblable au code P(Y) militaire sur les fréquences L1 et L2. Le résultat final est l'obtention d'un signal plus robuste et plus précis et dont l'intégrité est assez élevée pour répondre aux exigences d'un service SoL. Enfin, en combinant les signaux L5 aux signaux L1 et L2C, il deviendrait possible de réaliser une technique appelée *tri-laning*, qui permet de virtuellement éliminer les erreurs dues à la géométrie de la constellation et aux multitrajets (Simsy, 2006) (ces deux sujets sont abordés au chapitre 2).

Le futur bloc III réserve encore plus d'améliorations au système GPS. Parmi certains objectifs fixés, on viserait une précision de positionnement inférieure au mètre ainsi qu'une intégrité accrue des signaux. Ces améliorations permettraient ainsi de guider les avions lors d'atterrissages par exemple (Misra et Enge, 2006). L'ajout d'un nouveau signal civil de haute précision sur la bande L1, L1C, fait aussi parti des objectifs. Le type de modulation à être employé serait le BPSK, combiné à un code MBOC (*Multiplexed Binary Offset Code*) selon une entente conclue en juillet 2007 entre les États-Unis et l'Union européenne (les codes MBOC sont abordés au chapitre 3) (EC, 2007). Pour des raisons de compatibilité, Galileo en ferait de même pour ses signaux sur la bande L1.

L'évolution du spectre en fréquence du GPS est illustrée à la Figure 1.1 (Betz et al., 2007). Le spectre original (Blocs I/II/IIA/IIR) y est montré, suivi de chacune des étapes de modernisation. La dernière étape représente le spectre final devant être obtenu lors du déploiement du bloc III, prévu entre 2013 et 2020.



**Figure 1.1 Modernisation du spectre en fréquence du GPS.**



### 1.2.2 Modernisation de GLONASS

La Russie s'affaire présentement à regarnir la constellation GLONASS, dont il manque plusieurs satellites depuis la fin des années 1990. Elle a aussi entrepris de moderniser le système via l'envoi du premier satellite GLONASS-M en décembre 2003. On compte aujourd'hui un total de 16 satellites en orbite, desquels neuf sont de l'ancienne génération et sept sont de la nouvelle (Kosenko, Bartenev et Chebotarev, 2007). On prévoit le lancement de six autres satellites GLONASS-M vers la fin de 2007 (CSIC, 2007).

Le principal avantage des satellites GLONASS-M par rapport à l'ancienne génération de satellites est la diffusion de signaux civils sur la bande de fréquence GLONASS L2 (1242,9375 à 1248,625 MHz) en plus de la bande de fréquence GLONASS L1 (1598,0625 à 1605,375 MHz) (CSIC, 2002). Il faut aussi noter une durée de vie estimée à sept ans pour les satellites GLONASS-M, ce qui est de loin supérieur à la génération précédente. Tous les satellites GLONASS ont cependant la particularité de transmettre des signaux FDMA (*Frequency Division Multiple Access*) plutôt que CDMA, limitant pour ainsi dire le niveau d'interopérabilité entre les systèmes GPS et GLONASS. Les États-Unis et la Russie ont cependant tenu une rencontre vers la fin de 2006 afin d'aborder le sujet et discuter des avantages d'employer une approche similaire (CSIC, 2007). Il y a fort à parier que GLONASS diffusera de nouveaux signaux CDMA à partir de ses prochains satellites GLONASS-K et GLONASS-MK actuellement en développement.

### 1.2.3 Galileo

Galileo voit officiellement le jour en 2001 grâce à une initiative conjointe de l'Union européenne et de l'Agence spatiale européenne (ESA – *European Space Agency*) (Onidi, 2002). Afin d'affirmer sa souveraineté, l'Union européenne veut alors s'affranchir du système GPS des États-Unis, un système de plus en plus utile dans les divers secteurs de son industrie. Un partenariat public-privé (PPP) est donc établi afin de lancer Galileo, le tout premier système de navigation par satellite civil européen. Les États-Unis ne cacheront pas leur mécontentement face à un tel projet, qu'ils considéreront *a priori* comme une menace

pour leur sécurité nationale. Ils finiront cependant par se réconcilier avec l'Union européenne en signant un accord en 2004 visant à rendre les systèmes GPS et Galileo compatibles et interoperables.

Le projet Galileo débute tout d'abord en lion. Les objectifs initiaux sont clairement énoncés et des dates butoirs sont rapidement fixées, tels que vu dans (EC & ESA, 2002) :

- Développement et validation en orbite (2001-2005)
- Déploiement (2006-2007)
- Opérations commerciales (à partir de 2008)

Plusieurs pays étrangers désireux de se joindre au projet, dont le Canada et la Chine, choisissent aussi d'apporter un support financier au projet. Malgré cela, les dates fixées s'avèrent un peu trop optimistes et l'échéancier original n'est pas respecté. Le premier satellite Galileo, nommé GIOVE-A (et le seul à ce jour), est finalement lancé le 28 décembre 2005. La situation commence à devenir critique en 2007, alors que le second satellite, GIOVE-B, éprouve toujours des problèmes techniques et ne peut être lancé. Or, advenant une défaillance de GIOVE-A, Galileo pourrait perdre ses droits de diffuser ses signaux aux fréquences présentement allouées. Ceci est dû à une règle instaurée par l'ITU (*International Telecommunication Unit*), qui stipule que pour maintenir ses droits d'allocation aux fréquences, un système ne doit pas demeurer hors d'onde pendant plus de deux ans (ION Newsletter, 2007).

D'autres problèmes sont aussi éprouvés, cette fois-ci du côté de la structure PPP mise en place. Alors qu'on croyait qu'un PPP accélérerait le déploiement du projet, beaucoup de temps est perdu dans les négociations entre le secteur public et les entreprises privées. Afin de contourner le problème, Galileo pourrait toutefois se transformer sous peu en un programme institutionnel traditionnel, transférant ainsi la charge de financement aux contribuables européens (Inside GNSS, 2007). La mise en service de Galileo pourrait donc être retardée à 2012 dans le meilleur des cas. Les tests continuent toutefois à progresser



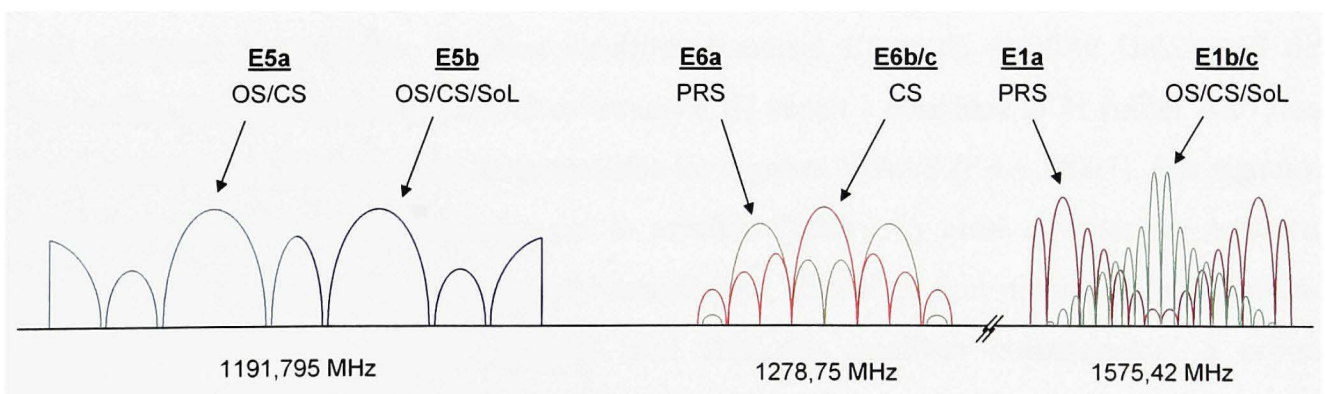
entre-temps à l'aide du satellite GIOVE-A, rendant possible la transmission du premier message de navigation le 4 mai 2007 (ESA, 2007). La construction d'un second centre de contrôle est aussi en cours à Oberpfaffenhofen, dans le sud de l'Allemagne (ION Newsletter, 2007). Ce centre de contrôle permettra de compléter celui déjà existant, situé à Guildford au Royaume-Uni.

Le segment de contrôle de Galileo devrait être composé de deux stations maîtresses, cinq stations monitrices, dix stations à lien montant (*Uplink Station*) et de 29 stations de références (*Sensor Station*) (Misra et Enge, 2006). Le segment spatial est quant à lui composé de 27 satellites MEO et de 3 satellites de secours répartis uniformément sur trois plans orbitaux, chacun incliné à  $56^\circ$  par rapport au plan équatorial. L'altitude des satellites est fixée à 23 222 km, ce qui leur donne une période de révolution de 14 hr 4 min. La position des satellites dans le ciel se répète donc tous les dix jours.

Galileo se propose d'offrir cinq services différents énumérés ci-dessous (Misra et Enge, 2006) :

- OS (*Open Service*) : Un service gratuit et ouvert à tous, similaire au service SPS du GPS mais avec une précision réputée supérieure.
- CS (*Comercial Service*) : Un service commercial payant qui assurera une garantie de service et un certain niveau de performances.
- SoL (*Safety of Life Service*) : Un service payant offrant une haute intégrité pour les applications de sauvegarde de la vie.
- PRS (*Pubic Regulated Service*) : Un service crypté payant, offrant une haute intégrité ainsi qu'une résistance accrue aux interférences, à l'intention des agences gouvernementales (telles que les policiers et les services de secours) et de l'armée.
- SAR (*Search and Rescue Service*) : Un service visant à faciliter la recherche de personnes lors des missions de recherche et de sauvetage; on propose la possibilité de détecter un signal provenant de balises de détresse en moins de 10 minutes à 98% du temps à l'aide des satellites Galileo.

À moins de changements majeurs, les cinq services Galileo devraient être diffusés à trois fréquences différentes, soit L1, E5 (1191,795 MHz) et E6 (1278,75 MHz). Chaque fréquence devrait ensuite être subdivisée en de multiples canaux, chacun supportant des services différents, tels que présentés à la Figure 1.2 (Misra et Enge, 2006). Il faut noter qu'au moins un canal pilote est présent à chacune des fréquences, ce qui rend la poursuite des signaux beaucoup plus efficace et robuste. Une innovation apportée par Galileo est la modulation AltBOC (Alternative Binary Offset Carrier) employée à la fréquence E5. Ce nouveau genre de modulation serait employé pour combiner les signaux E5a (1176,45 MHz, soit la même fréquence que L5) et E5b (1207,14 MHz) sur une seule et même bande de fréquence d'une largeur de 51,15 MHz (ESA, 2006). Ceci permettrait d'offrir aux utilisateurs une précision sans précédent (Sleewaegen, De Wilde et M., 2006). Le traitement de signaux à large bande peut toutefois exiger de se tourner vers des récepteurs plus haut de gamme, ce qui ne correspond pas nécessairement aux budgets ni aux besoins de la plupart des utilisateurs. Pour cette raison, les signaux E5a et E5b peuvent aussi être traités séparément à la manière de signaux QPSK (le lecteur est invité à consulter (ESA, 2006) pour plus de détails sur la modulation AltBOC). Puisque les signaux GPS L5 sont de type QPSK et qu'ils partagent la même plage de fréquence que les signaux E5a, on réalise d'une pierre deux coups en les rendant compatibles. Les signaux Galileo E1B/C et GPS L1C seront eux aussi compatibles puisqu'ils emploieront un type de modulation commun à chacun (soit une modulation BPSK + MBOC, telle qu'abordée précédemment).



**Figure 1.2 Spectre en fréquence Galileo.**

### 1.2.4 WAAS

Le WAAS (*Wide-Area Augmentation System*), est un système d'augmentation par satellite (SBAS – *Satellite Based Augmentation System*) servant à fournir des corrections GPS ainsi que des informations concernant l'intégrité de la constellation. Il fut mis sur pied en 1994 par le département des transports (DOT - *Department of Transportation*) et la Federal Aviation Administration (FAA) des États-Unis. Le principal but du WAAS était de permettre le guidage d'approche d'appareils aériens civils à l'aide du GPS. Il est aujourd'hui utilisé de façon beaucoup plus large, en particulier dans les récepteurs simple fréquence bon marché. Le WAAS est transmis de la même façon que les signaux GPS L1 C/A, et à compter de 2007, ils pourront aussi être transmis sur la bande L5 (Lo et al., 2006). La précision actuellement fournie par le WAAS est de 0,9 m horizontalement et de 1,3 m verticalement à 95% du temps (FAA, 2006).

Le segment au sol du WAAS est composé de près d'une quarantaine de stations de références étendues à la grandeur de l'Amérique du Nord (Wikipedia, 2007f). Chacune des stations mesure en continu les signaux GPS et relaye les informations à une station maîtresse. La station maîtresse calcule et télécharge ensuite les corrections et données d'intégrité à être diffusées par le segment spatial.

Le segment spatial du WAAS a traversé plusieurs changements au cours de l'année 2007. Il était composé au préalable de deux satellites Inmarsat III et du satellite Galaxy 15 de PanAmSat. Puisque le bail des satellites Inmarsat III venait à échéance le 31 juillet 2007, les deux satellites ont depuis cessé de transmettre les signaux WAAS (FAA, 2007). Les signaux continuent toutefois d'être transmis par le satellite Galaxy 15 ainsi que par un nouveau satellite, Anik F1R, loué à la compagnie canadienne Telesat. Il faut noter que les satellites servant à diffuser les signaux WAAS sont tous des satellites commerciaux à orbite géostationnaire (GEO - *Geostationary Earth Orbit*).



Plusieurs autres systèmes SBAS très semblables au WAAS ont vu le jour récemment. On compte parmi eux les systèmes EGNOS (*European Geostationary Navigation Overlay Service*) de l'Union européenne, MSAS (*Multi-functional Satellite Augmentation System*) du Japon et GAGAN (*GPS And Geo Augmented Navigation*) de l'Inde. Un système SBAS canadien, le CDGPS (*Canada-wide DGPS correction service*), a aussi été développé, quoique non compatible avec le système WAAS.

### 1.2.5 Beidou, QZSS et IRNSS

Avec l'engouement suscité au tournant des années 2000 pour la navigation par satellite, d'autres pays ont choisi de développer leur propre système de navigation. Parmi ces pays, on compte la Chine, le Japon et l'Inde dont les systèmes de navigation par satellite respectifs sont : Beidou, QZSS (*Quasi-Zenith Satellite System*) et IRNSS (*Indian Regional Navigational Satellite System*). La Chine est toutefois le seul de ces pays à avoir annoncé le déploiement d'une constellation à couverture mondiale, pour un total de 5 satellites GEO et 30 satellites MEO (le système se nomme aujourd'hui Beidou2 ou Compass) (Inside GNSS, 2006). QZSS et IRNSS sont quant à eux des systèmes régionaux, c'est-à-dire qu'ils ne couvrent qu'une seule partie du globe. Pour y parvenir, les concepteurs de tels systèmes font généralement appel à des satellites géostationnaires ou géosynchrones de manière à assurer une couverture restreinte d'un territoire (Kogure, Sawabe et Kishimoto, 2006). Ceci permet ainsi de réduire le nombre de satellites nécessaires de même que les coûts du système.

Un peu comme Galileo, les systèmes Beidou, QZSS et IRNSS sont conçus pour être compatibles avec le GPS. Ils occupent donc sensiblement les mêmes bandes de fréquences que ce dernier. Le déploiement complet de ces systèmes est prévu aux alentours de 2010-2020.

### 1.2.6 Quelques avantages du GNSS

On appelle « système global de navigation par satellite », ou GNSS (*Global Navigation Satellite System*), l'ensemble de tous les systèmes de navigation par satellite à couverture mondiale. Or, avec la modernisation et le développement de tout nouveaux systèmes GNSS, la possibilité de combiner plusieurs d'entre eux au sein d'un même récepteur de navigation s'avère de plus en plus intéressante. Si on considère le GPS seul par exemple, on parle alors d'au moins 24 satellites en orbite avec la possibilité d'en capter entre 5 et 12 la plupart du temps. En incluant Galileo, on obtiendrait toutefois une constellation GNSS d'au moins 51 satellites et la possibilité d'en capter entre 13 à 20 (Misra et Enge, 2006). De la même manière, l'ajout de GLONASS permettrait d'obtenir un nombre encore plus élevé de satellites visibles (Constantinescu A. et Landry, 2005). Puisqu'il faut au minimum quatre satellites pour obtenir une position en 3D (chapitre 2), ceci augmente donc considérablement les chances d'obtenir une position, et ce peu importe l'environnement dans lequel on se trouve. Un autre intérêt des récepteurs GNSS est qu'en combinant plusieurs satellites, on améliore davantage la précision du positionnement. On note une amélioration d'au moins 20 à 40% de la précision au niveau des récepteurs de navigation lorsque les signaux GPS et Galileo sont combinés comparativement aux mêmes signaux employés séparément (Constantinescu A., Landry et Ilie, 2004). Enfin, puisque les systèmes composant le GNSS sont opérés par des entités de divers pays et continents, on s'assure par l'utilisation d'un récepteur GNSS d'obtenir un service de positionnement fiable, peu importe les crises politico-économiques pouvant survenir dans le monde.

### 1.2.7 Modernisation des récepteurs de navigation

Les récepteurs de navigations sont généralement considérés comme le troisième et dernier segment des systèmes de navigation. Cela ne leur enlève en rien leur importance puisque ce sont eux qui génèrent la presque totalité des retombées économiques de ce secteur par le biais de ventes et de services. Outre l'aspect commercial, le fait le plus impressionnant demeure sans doute le nombre de domaines impliqués lors de la conception de récepteurs de navigation. En voici un bref échantillon : radiofréquences et télécommunications,

microélectronique, informatique, physique, mécanique, astronomie, géomatique et météorologie. C'est la connaissance et le développement de ces sciences qui ont permis de créer les récepteurs de navigation d'aujourd'hui, compacts et abordables.

Les premiers récepteurs de navigation satellites à avoir été développés datent de l'époque de Transit. Ces récepteurs, généralement montés dans des étagères, étaient toutefois trop volumineux pour être transportés par un être humain. Il fallut attendre l'avènement de la microélectronique vers le début des années 1980 pour voir apparaître les premiers récepteurs « portables ». Le récepteur présenté à la Figure 1.3 en est un exemple, pesant plus de 7 kg et coûtant près de 45 000 \$ US (ION, 2004). Depuis, l'évolution de la microélectronique a permis de miniaturiser et de réduire considérablement le coût des récepteurs, à un tel point qu'on arrive aujourd'hui à les intégrer au sein de téléphones cellulaires sans même qu'ils ne passent aperçu (voir Figure 1.4).



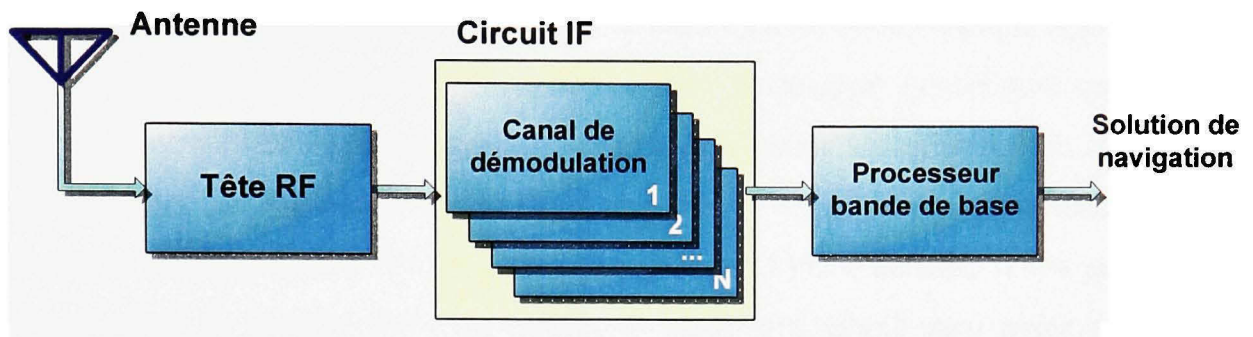
**Figure 1.3** *Première génération de récepteurs GPS.*  
Image tirée de (ION, 2004)





**Figure 1.4** *Trois récepteurs GPS voués à trois applications différentes.*

Les récepteurs de navigation sont habituellement constitués de trois segments : *la tête RF*, dont le but est de capter et de conditionner les signaux satellites, *le circuit IF*, qui sert à démoduler les signaux, et *le processeur en bande de base*, qui s'occupe de traiter les signaux pour en extraire une position (Figure 1.5). De façon abrégée, on parle des segments RF, IF et bande de base. Ces trois segments sont généralement implémentés à l'intérieur d'un ou de plusieurs circuits intégrés, aussi appelés ASIC (*Application Specific Integrated Circuit*). La conception d'un ASIC étant longue, coûteuse et peu flexible, il devient donc difficile pour les manufacturiers de répondre rapidement aux changements qu'entraîne l'arrivée de nouveaux signaux GNSS. Une nouvelle approche beaucoup plus flexible, appelée radio logicielle (SDR – *Software Defined Radio*), permet toutefois de pallier le problème.



**Figure 1.5** *Architecture typique d'un récepteur de navigation.*

La SDR, un concept datant du début des années 1990, repose sur le traitement exclusivement numérique des signaux à l'aide de composants reprogrammables (Sauriol et Landry, 2007). En d'autres termes, on tentera dans un récepteur de navigation SDR de numériser les signaux GNSS le plus près possible de l'antenne (à l'aide d'un convertisseur analogique à numérique – CAN) afin de minimiser le nombre de composants matérielles requises. Puisqu'un maximum de traitement est effectué de façon logicielle, il est alors possible de modifier rapidement l'architecture d'un récepteur SDR, même après sa vente auprès d'un client.

Les composants le plus souvent considérées pour le développement de récepteurs SDR sont les processeurs à traitement de signaux numériques (DSP – *Digital Signal Processor*) et les processeurs numériques à usage général, tels que ceux rencontrés dans les ordinateurs personnels. Ces composants accomplissent généralement bien les tâches associées à la bande de base, qui consistent en une série d'opérations mathématiques séquentielles. Il devient toutefois difficile de traiter efficacement les signaux IF, et ce malgré la puissance des processeurs actuels, puisqu'un grand nombre d'opérations doit être réalisé en parallèle. Les récepteurs de navigation SDR basés uniquement sur des processeurs n'offrent ainsi qu'un nombre restreint de canaux de réception et des performances limitées en terme de fréquence de mise à jour de la position (Won, Pany et Hein, 2006).

Le FPGA est une autre composante à l'occasion utilisée pour le développement de récepteurs SDR. Une raison pouvant expliquer son usage moins fréquent serait la complexité de mise en œuvre, qui s'avère généralement plus grande que pour un simple processeur numérique

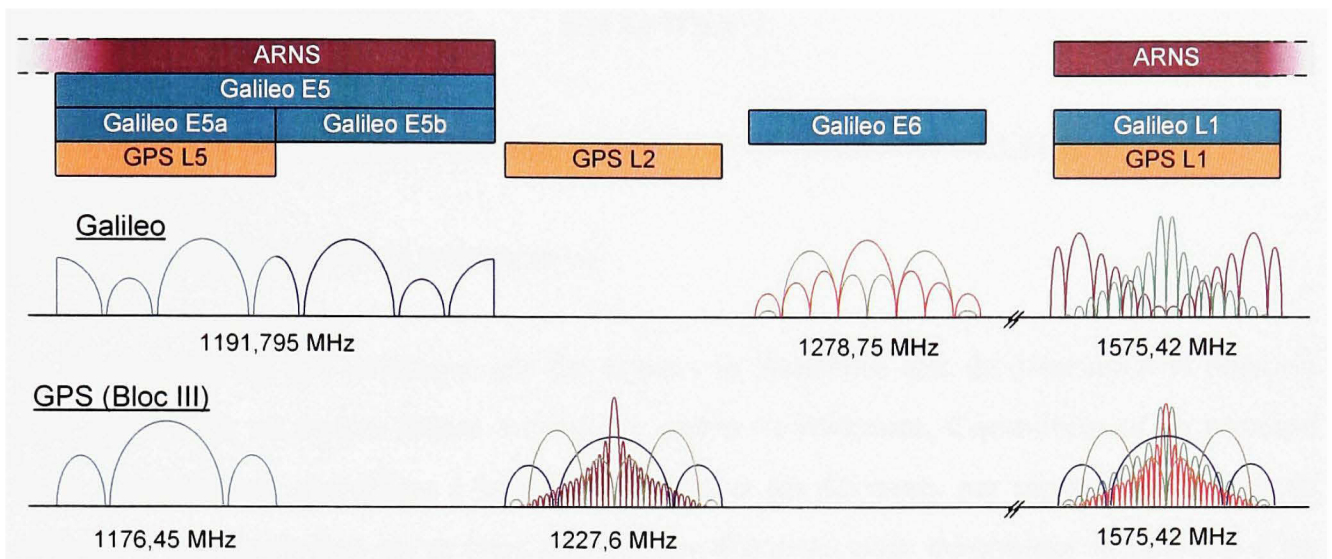
(MacGougan, Normark et Stahlberg, 2005). Le FPGA est un circuit logique reprogrammable, il opère donc de façon parallèle, contrairement aux processeurs numériques qui opèrent de façon séquentielle. Le développement d'architectures parallèles au sein d'un FPGA permet ainsi de traiter les signaux IF à très haute cadence et d'offrir des performances inégalées (Sauriol et Landry, 2007). Vu la grande densité des FPGA actuels, il est aussi possible d'implémenter un nombre élevé de canaux de réceptions GNSS sans avoir à sacrifier les performances du récepteur. Ceci est souhaitable, considérant le grand nombre de satellites et de signaux GNSS qui seront disponibles sous peu. Pour cette raison, le projet présenté dans ce document repose principalement sur l'utilisation d'un FPGA.

### **1.3 Résumé**

La navigation a été introduite dans ce chapitre, débutant il y a plus de 2000 ans à l'aide, entre autres, des astres célestes, et se poursuivant dans les années 1960 grâce à la radionavigation par satellite. Depuis, les systèmes de navigation par satellite n'ont cessé de s'améliorer, offrant un positionnement d'une précision toujours grandissante.

Dans ce chapitre, le début du 21<sup>e</sup> siècle a été dépeint comme un tournant au niveau de la navigation par satellite, alors que plusieurs systèmes sont développés ou modernisés. Inévitablement, ceci crée une certaine compétition entre les systèmes, à savoir qui offrira les services les plus précis et les plus fiables, ou encore qui aura le privilège d'occuper telle ou telle bande de fréquence (voir Figure 1.6). Le réel avantage ne repose néanmoins pas dans la compétition, mais plutôt dans l'intégration de tous ces systèmes en un seul système de navigation par satellite global, appelé GNSS.





**Figure 1.6 Spectres en fréquence Galileo et GPS Bloc III.**

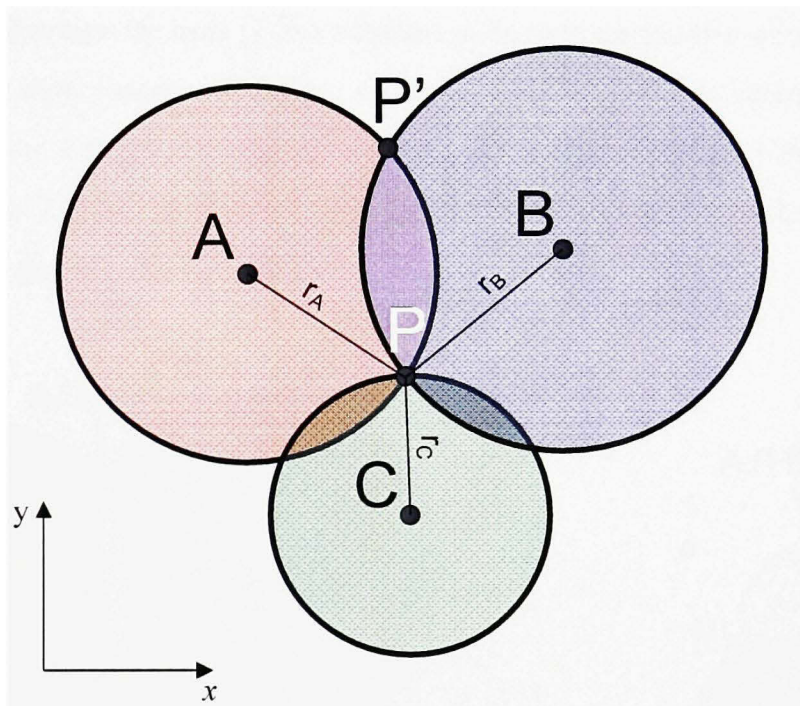
Afin d'exploiter les avantages inhérents au GNSS tout en demeurant compétitive, l'industrie doit se tourner vers de nouvelles technologies de récepteurs de navigation. Parmi les technologies émergentes, on compte les récepteurs SDR, beaucoup plus flexibles et rapides à développer que les récepteurs de navigation actuels. Une catégorie particulièrement intéressante de récepteurs SDR, les récepteurs basés sur les FPGA, sera abordée beaucoup plus en profondeur tout au long de ce document.

## CHAPITRE 2

### PRINCIPES DE LA NAVIGATION PAR SATELLITE

#### 2.1 Le principe de la trilatération

La trilatération est une technique qui fait appel à la géométrie afin de déterminer la position d'un point dans un espace donné à l'aide de points de référence. Contrairement au principe de la triangulation, qui utilise à la fois les angles et les distances par rapport aux points de référence, la trilatération ne se base que sur les distances pour déterminer la position d'un point. La figure ci-dessous illustre bien le principe pour un espace en deux dimensions.



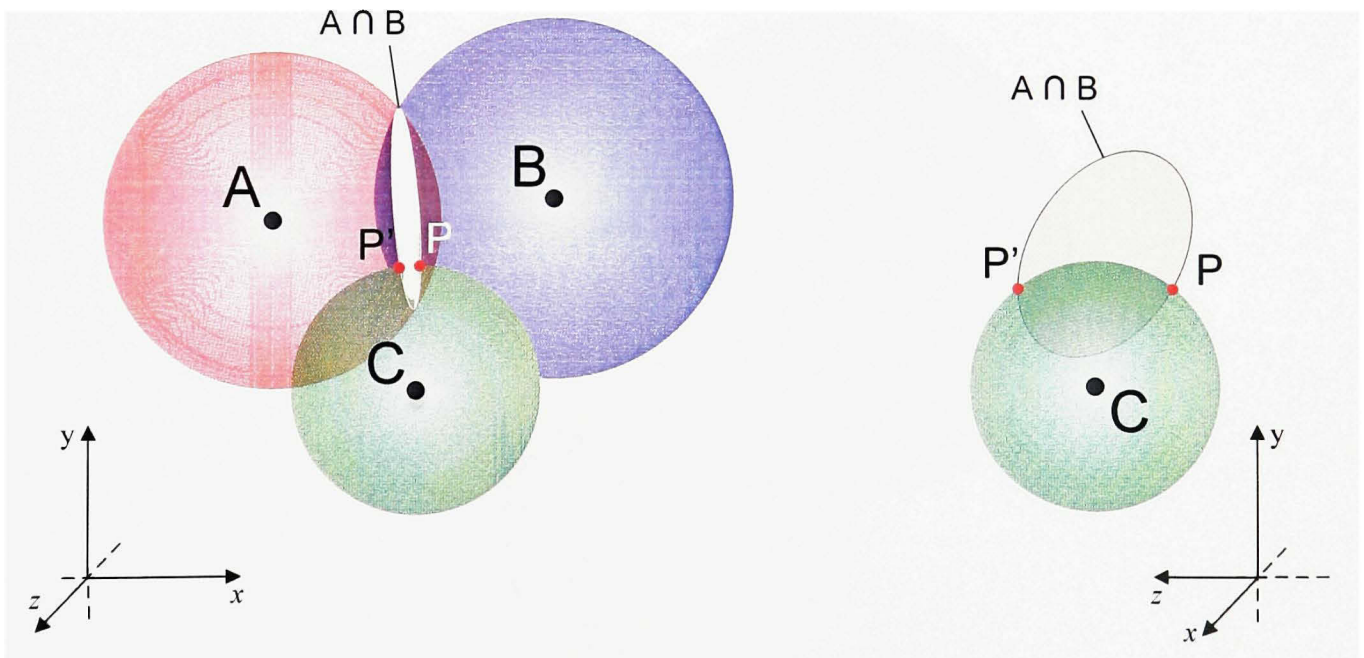
**Figure 2.1 Principe de la trilatération en 2D.**

Si on connaît la position en  $x$  et  $y$  de chacun des points A, B et C ainsi que leur distance respective  $r_A$ ,  $r_B$  et  $r_C$  par rapport à un point P, alors il est possible de déterminer la position exacte de ce point P dans l'espace. Par le théorème de Pythagore, on obtient une série de trois équations non linéaires à résoudre, où  $x_A$ ,  $y_A$ ,  $x_B$ ,  $y_B$ ,  $x_C$  et  $y_C$  décrivent la position des points A, B et C et où  $x_P$  et  $y_P$  sont les inconnues :

$$\begin{aligned}
 r_A &= \sqrt{(x_A - x_P)^2 + (y_A - y_P)^2} \\
 r_B &= \sqrt{(x_B - x_P)^2 + (y_B - y_P)^2} \\
 r_C &= \sqrt{(x_C - x_P)^2 + (y_C - y_P)^2}
 \end{aligned}
 \tag{2.1}$$

Il est à noter qu'une trilatération dans un espace à deux dimensions requiert au minimum deux points de référence pour être exécutée et qu'il en faut au moins trois pour lever l'ambiguïté sur la position, représentée par les points P et P' sur la Figure 2.1.

La trilatération peut aussi être employée pour le positionnement en trois dimensions. C'est la technique que l'on emploie pour le positionnement par satellite, par exemple. Dès lors, l'utilisation d'un minimum de trois points références devient nécessaire afin de déterminer la position d'un point dans l'espace. De plus, au moins quatre points de référence doivent être utilisés afin de lever l'ambiguïté sur la solution de positionnement. Cette ambiguïté est illustrée à la Figure 2.2 par la sphère C qui intercepte à deux endroits distincts le plan de jonction ellipsoïdal des sphères A et B.



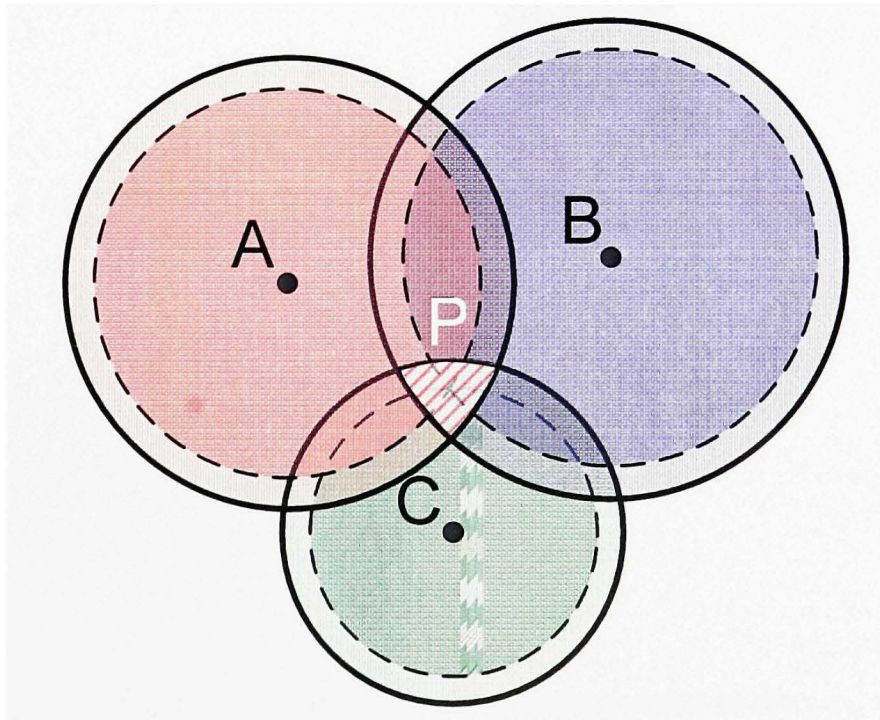
**Figure 2.2 Principe de la trilatération en 3D.**



De la même manière que pour la trilatération en 2D, la trilatération en 3D peut-être résolue par une série de quatre équations non linéaires découlées du théorème de Pythagore. La variable  $z$  est ajoutée afin de décrire la troisième dimension et un quatrième point de référence, appelé D, fait son apparition :

$$\begin{aligned}
 r_A &= \sqrt{(x_A - x_P)^2 + (y_A - y_P)^2 + (z_A - z_P)^2} \\
 r_B &= \sqrt{(x_B - x_P)^2 + (y_B - y_P)^2 + (z_B - z_P)^2} \\
 r_C &= \sqrt{(x_C - x_P)^2 + (y_C - y_P)^2 + (z_C - z_P)^2} \\
 r_D &= \sqrt{(x_D - x_P)^2 + (y_D - y_P)^2 + (z_D - z_P)^2}
 \end{aligned} \tag{2.2}$$

Jusqu'à présent, le principe de la trilatération a été abordé pour un monde idéal et sans erreur. En pratique toutefois, les erreurs sont bien présentes, que ce soit au niveau des mesures de distances effectuées ou de la position des points de référence. Par conséquent, on obtient non plus un point exact dans l'espace, mais plutôt une région de probabilité représentée par la région hachurée à la Figure 2.3.



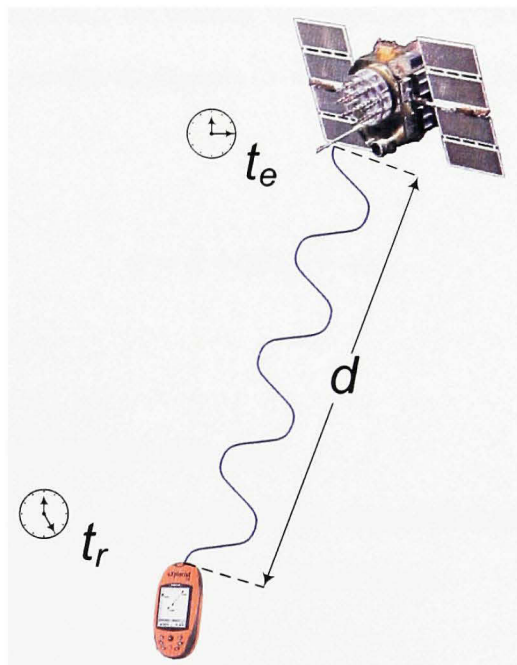
**Figure 2.3** *Effet des incertitudes sur la trilatération.*

## 2.2 Le positionnement par satellite de façon concrète

Tel que mentionné précédemment, c'est le principe de la trilatération 3D qui est le plus couramment employé pour le positionnement par satellite. Selon la théorie, un minimum de quatre satellites est donc nécessaire afin d'obtenir une position sans ambiguïté. Dans les faits toutefois, il est facile de lever cette ambiguïté car une des deux solutions de navigation est généralement située à la surface de la Terre alors que l'autre est localisée loin dans l'espace. On pourrait donc croire *a priori* que trois satellites suffisent à obtenir une position n'importe où sur le globe, or ce n'est pas le cas. Pour y voir plus clair, il faut tout d'abord aborder le concept de pseudo-distances.

### 2.2.1 Distances et pseudo-distances

Les satellites de positionnements, que ce soit GPS, GLONASS ou Galileo, émettent constamment des signaux radio vers le sol. Or, ces signaux radio contiennent de l'information quant à la position des satellites et au temps d'émission,  $t_e$  (voir Figure 2.4).



**Figure 2.4** *Mesure du délai de transmission des signaux de navigation.*

Puisque ces signaux voyagent à la vitesse de la lumière,  $c$ , il s'agit alors de mesurer à l'aide d'un récepteur le temps de réception des signaux  $t_r$  afin de déterminer la distance  $d$  entre un satellite et un utilisateur :

$$d = c(t_r - t_e) \quad (2.3)$$

L'intervalle de temps présenté à l'équation (2.3) représente le délai de transmission des signaux. Vu l'ordre de grandeur de la vitesse de la lumière (soit environ  $3 \times 10^8$  m/s), une incertitude sur le temps de seulement 1 ns se traduit directement par une erreur de distance de 30 cm. Il importe donc de mesurer avec grande précision le délai de transmission des signaux de chacun des satellites afin d'obtenir une position juste. Pour y arriver, on installe généralement des horloges au césium, et plus récemment au rubidium, à bord des satellites, et on les surveille en permanence à partir de postes de contrôle au sol. Pour des fins pratiques, ceci n'est évidemment pas possible au niveau du récepteur. On utilise plutôt une horloge peu coûteuse (une horloge au quartz fait généralement l'affaire) et on suppose qu'il existe un biais inconnu entre le véritable temps de réception des signaux et le celui indiqué par le récepteur. Ceci a pour conséquence de biaiser les mesures de distances effectuées à l'aide du récepteur; on parle donc de pseudo-distances (notées  $\rho$ ) plutôt que de distances à proprement dit :

$$\rho = d + c\Delta t_{bu} + c\Delta t_d \quad (2.4)$$

où :

- $\Delta t_{bu}$  est le biais de l'horloge du récepteur
- $\Delta t_d$  représente toutes les erreurs et délais supplémentaires introduits sur le signal satellite au long de son parcours (abordés plus loin)

Le positionnement par satellite implique donc quatre inconnues principales : la position de l'utilisateur sur les trois axes ainsi que le biais d'horloge du récepteur. Ceci exige d'avoir des



mesures d'un minimum de quatre satellites afin de résoudre un système à quatre équations et quatre inconnues (Tsui, 2000) :

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_u \quad (2.5)$$

où :

- $x_i, y_i$  et  $z_i$  représentent la position du  $i^{\text{ème}}$  satellite
- $x_u, y_u$  et  $z_u$  représentent la position de l'utilisateur
- $b_u$  est le biais d'horloge du récepteur exprimé en terme de distance ( $b_u = c\Delta t_{bu}$ )

Puisque le positionnement par satellite exige de résoudre l'inconnue temps en plus de la position, cela signifie qu'en ayant un récepteur de positionnement, on a virtuellement à sa disposition une horloge atomique. Étant donné que les récepteurs GPS actuels ont un coût très inférieur aux horloges atomiques, ceci a donc permis d'ouvrir la porte à de nombreuses possibilités, que ce soit au niveau de la synchronisation automatique des réseaux sans fil, d'équipements de mesure dans les régions éloignées ou encore de la surveillance minutieuse de réseaux électriques qui s'étendent sur plusieurs milliers de kilomètres. Bien d'autres applications restent toutefois à être découvertes dans ce domaine.

### 2.2.2 Impacts de la vitesse de déplacement sur les signaux

Les satellites de positionnement, tels que les satellites GPS par exemple, se déplacent généralement à haute vitesse le long d'une orbite moyenne (MEO – *Medium Earth Orbit*). Ceci a pour conséquence d'introduire un effet Doppler non négligeable sur les signaux reçus, même pour un récepteur fixe au sol. Puisque le décalage Doppler est directement proportionnel à la vitesse relative entre un satellite et un récepteur (on néglige ici les effets relativistes), il est donc possible de mesurer ce décalage afin de déterminer la vitesse de déplacement d'un utilisateur :

$$\Delta f = -\frac{f \dot{\rho}}{c} \quad (2.6)$$

où :

- $\Delta f$  est le décalage Doppler
- $f$  est la fréquence de la porteuse (i.e. sans effet Doppler)
- $\dot{\rho}$  est la vitesse relative entre un satellite et le récepteur
- $c$  est la vitesse de la lumière

Pour ce qui est des satellites GPS par exemple, ceux-ci se déplacent sur une orbite de 26 600 km de rayon en approximativement 12 heures (Kaplan, 1996), ce qui équivaut à une vitesse moyenne d'environ 4 km/s. Par conséquent, un décalage Doppler de l'ordre de  $\pm 4000$  Hz est introduit sur la porteuse des signaux satellites (Tsui, 2000). Cette valeur est valable pour les signaux dans la bande de fréquence L1 (1574,42 MHz) et un récepteur fixe au sol. Cette valeur peut toutefois atteindre  $\pm 10$  kHz pour un mobile se déplaçant à grande vitesse, tel qu'un avion à réaction par exemple.

Une fois la vitesse relative entre chacun des satellites et le récepteur connue, il est possible de résoudre les équations de navigation afin de déterminer la vitesse d'un utilisateur. Cette vitesse est facilement obtenue en dérivant l'équation (2.5) :

$$\dot{\rho}_i = \frac{d\rho_i}{dt} = \frac{d}{dt} \left( \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_u \right) \quad (2.7)$$

$$\dot{\rho}_i = \frac{(x_i - x_u)(\dot{x}_i - \dot{x}_u) + (y_i - y_u)(\dot{y}_i - \dot{y}_u) + (z_i - z_u)(\dot{z}_i - \dot{z}_u)}{\sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}} + \dot{b}_u \quad (2.8)$$

où :

- $\dot{\rho}_i$  représente la vitesse relative entre le  $i^{\text{ème}}$  satellite et le récepteur
- $\dot{x}_i, \dot{y}_i$  et  $\dot{z}_i$  sont les vitesses de déplacement du  $i^{\text{ème}}$  satellite sur les trois axes

- $\dot{x}_u, \dot{y}_u$  et  $\dot{z}_u$  sont les vitesses de déplacement de l'utilisateur sur les trois axes
- $\dot{b}_u$  est la dérive d'horloge du récepteur

On remarque que quatre inconnues sont encore présentes (soit  $\dot{x}_u, \dot{y}_u, \dot{z}_u$  et  $\dot{b}_u$ ). La résolution de l'équation (2.8) demande donc un minimum de quatre satellites pour être effectuée.

### 2.2.3 Positionnement en 2D

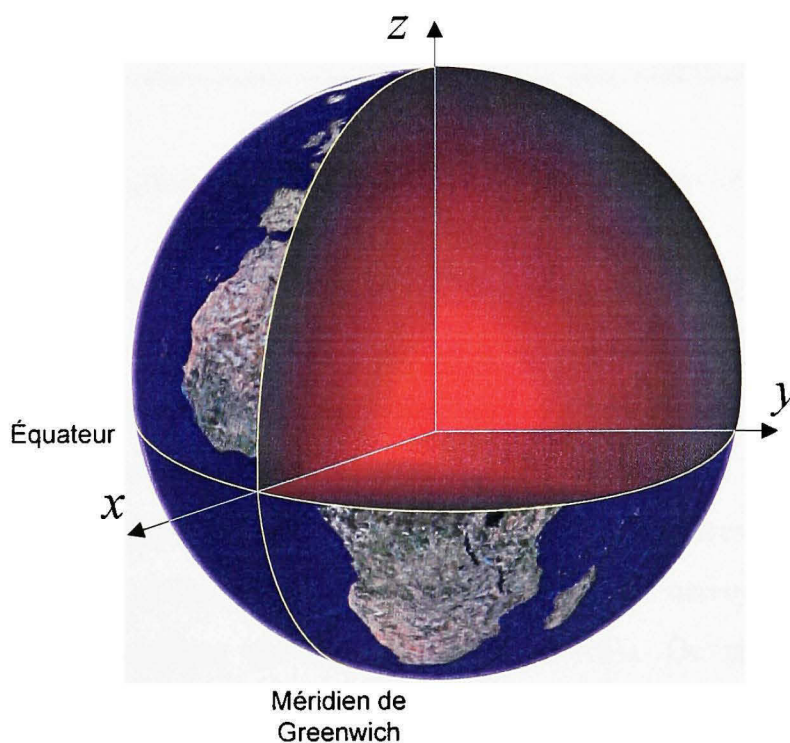
Il peut arriver à l'occasion que moins de quatre satellites de positionnement ne soient visibles par un récepteur. Bien que cela puisse devenir de moins en moins probable avec l'arrivée prochaine de nombreux satellites GNSS, le problème demeure toujours bien réel pour les récepteurs GPS actuels et pour certaines applications à visibilité réduite. Le problème se pose généralement en forêt ou en ville alors que les arbres ou les édifices obscurcissent une bonne partie du ciel. Avec moins de quatre satellites en vue, il devient donc impossible de calculer une position en 3D telle que vue précédemment. Cependant, il est encore possible d'obtenir une position approximative en 2D avec seulement trois satellites. L'approche la plus fréquente consiste à fixer l'altitude par rapport au niveau de la mer du récepteur GPS pour ne calculer que la latitude, la longitude et le biais d'horloge. Évidemment, plus l'estimation de l'altitude est juste, meilleure est la solution de navigation.

Une autre approche pouvant être employée lors de la perte momentanée de satellites consiste à extrapoler le biais et la dérive d'horloge du récepteur à partir de solutions précédemment calculées. En combinant cette approche avec la méthode mentionnée ci-dessus, il devient alors possible d'obtenir une solution temporaire à partir de deux satellites seulement (Bullock et al., 1996). Cette technique n'a toutefois pas été implémentée au sein du récepteur hybride présenté dans ce document. Ceci permettrait néanmoins d'augmenter considérablement la robustesse du récepteur dans des environnements difficiles, tels qu'abordés au chapitre 6.



### 2.2.4 Système de coordonnées géodésiques WGS-84

Le WGS-84 est la révision de 1984 du système géodésique mondial (*World Geodetic System*). Précédé par les systèmes WGS-60, 64 et 72, le WGS-84 est aujourd'hui le système géodésique officiel du GPS. Brièvement, notons qu'il est constitué d'un système de coordonnées cartésiennes dont l'origine est le centre approximatif de la Terre et dont l'axe des  $x$  passe par le méridien de Greenwich au niveau de l'équateur ( $0^\circ$  N  $0^\circ$  E - voir Figure 2.5). L'axe des  $z$  passe quant à lui par l'axe de rotation de la Terre, entrant au pôle Sud et sortant par le pôle Nord. L'axe des  $y$  passe enfin par la mer indienne au niveau de l'équateur ( $0^\circ$  N  $90^\circ$  E). Puisque le référentiel en question est centré et fixé à la Terre, on emploie souvent l'acronyme anglais ECEF pour le décrire (*Earth-Centered, Earth-Fixed*).



**Figure 2.5** *Système d'axes selon la norme WGS-84.*

Dans le système WGS-84, la Terre est modélisée par un ellipsoïde (soit une sphère légèrement aplatie aux pôles) dont le demi-grand axe (ou rayon équatorial) est égal à 6378,137 km et dont le demi-petit axe (ou rayon polaire) est égal à 6356,7523 km. Les divers

paramètres servant à décrire l'ellipsoïde en question sont présentés au Tableau 2.1. Pour plus d'information sur le système de coordonnées WGS-84, le lecteur est invité à consulter la référence suivante : (Misra et Enge, 2006).

Tableau 2.1  
Paramètres de l'ellipsoïde et autres constantes du modèle WGS-84

Paramètres	Symboles	Équations typiques	Valeurs WGS-84
Demi-grand axe	$a$	-	$6\,378\,137 \pm 1\text{ m}$
Aplatissement	$f$	-	$3,352\,810\,664\,747\,480\,5 \times 10^{-3}$
Demi-petit axe	$b$	$b = a(1 - f)$ (2.9)	$6\,356\,752,3\text{ m}$
Excentricité	$e$	$e = \sqrt{\frac{a^2 - b^2}{a^2}}$ (2.10)	$8,181\,919\,084\,262\,149\,2 \times 10^{-2}$
Vitesse de rotation	$\dot{\Omega}_e$	-	$7,2921151467 \times 10^{-5}\text{ rad/s}$
Constante gravitationnelle	$\mu$	-	$3,986005 \times 10^{14}\text{ m}^3/\text{s}^2$
Vitesse de la lumière	$c$	-	$2,99792458 \times 10^8\text{ m/s}$

### 2.2.5 Calcul de la position des satellites à partir des éphémérides

Tel que mentionné précédemment, les satellites de positionnement émettent constamment vers le sol des informations quant à leur position et au temps d'émission des signaux. Les satellites ne transmettent toutefois pas directement leur position, mais plutôt des éphémérides, soit un ensemble de paramètres orbitaux képlériens servant à calculer la position des satellites au temps d'émission des signaux. Ces paramètres, présentés au Tableau 2.2 pour les systèmes GPS et Galileo, sont déterminés à partir de mesures prises au sol, avant d'être téléchargées aux satellites via une station de contrôle. De façon générale, chaque satellite émet ses propres éphémérides et celles-ci sont valides pour des intervalles de l'ordre de quatre heures.

Un ensemble de paramètres similaires aux éphémérides, appelé almanach, est aussi transmis par chaque satellite de positionnement. L'almanach est en fait une version simplifiée des éphémérides, donc moins précise, et dont le but est d'estimer la position de tous les satellites d'une même constellation à un instant donné. Ceci permet notamment d'accélérer

l'acquisition des signaux à partir d'un récepteur (ce sujet est abordé au chapitre 3). Contrairement aux éphémérides, qui ne sont valides que durant quelques heures, l'almanach demeure généralement valide pendant plusieurs semaines, voire quelques mois. De plus, une seule version de l'almanach, contenant l'information sur toute la constellation (ex. : GPS ou Galileo), est transmise par tous les satellites. Vu la quantité d'information transmise, l'almanach est donc plus long à télécharger des satellites que les éphémérides (soit 12,5 et 10 min pour l'almanach comparé à 30 et 50 s pour les éphémérides de GPS et Galileo respectivement) (ICD-GPS-200C, 2000; ESTEC, 2007).

Tableau 2.2  
Paramètres propres aux éphémérides GPS et Galileo

Paramètres	Description	Unités
$M_0$	Anomalie moyenne au temps de référence	rad
$\Delta n$	Différence entre le mouvement moyen et la valeur calculée	rad/s
$e$	Excentricité	-
$A^{1/2}$	Racine carrée du demi-grand axe	m <sup>1/2</sup>
$\Omega_0$	Ascension droite du nœud ascendant	rad
$i_0$	Angle d'inclinaison au temps de référence	rad
$\omega$	Argument du périégée	rad
$\dot{\Omega}$	Vitesse d'ascension droite	rad/s
$\dot{i}$	Taux de variation de l'angle d'inclinaison	rad/s
$c_{uc}$	Amplitude du terme de correction harmonique cosinusoidal de l'argument de latitude	rad
$c_{us}$	Amplitude du terme de correction harmonique sinusoidal de l'argument de latitude	rad
$c_{rc}$	Amplitude du terme de correction harmonique cosinusoidal du rayon orbital	m
$c_{rs}$	Amplitude du terme de correction harmonique sinusoidal du rayon orbital	m
$c_{ic}$	Amplitude du terme de correction harmonique cosinusoidal de l'angle d'inclinaison	rad
$c_{is}$	Amplitude du terme de correction harmonique sinusoidal de l'angle d'inclinaison	rad
$t_{oe}$	Temps de référence éphéméride	s
$\Delta t_{oe}$	Période de validité de l'éphéméride	s

En téléchargeant les éphémérides d'un satellite de positionnement, il est généralement possible de déterminer la position de celui-ci en temps réel avec une précision de l'ordre de 1



à 2 m (IGS, 2007). Certains organismes, tels que l'IGS (*International GNSS Service*), le JPL (*Jet Propulsion Laboratory*) ou RNCAN (Ressources naturelles Canada) fournissent aussi des éphémérides en direct et en différé dont la précision est de l'ordre de quelques centimètres. Cette précision n'est toutefois atteinte qu'à condition de demeurer à l'intérieur de l'intervalle de validité des éphémérides. Ceci est facilement vérifié pour chacun des satellites de positionnement grâce aux équations qui suivent (ICD-GPS-200C, 2000) :

$$t_k = t - t_{oe} \quad (2.11)$$

où :

- $t$  est le temps auquel on désire calculer la position d'un satellite, soit généralement le temps auquel les signaux ont été transmis
- $t_k$  est la différence entre le temps  $t$  et le temps de référence  $t_{oe}$  donné dans l'éphéméride du satellite en question

Pour les systèmes GPS et Galileo, le temps (on parlera de temps GNSS à partir d'ici afin d'alléger le texte) est toujours exprimé en seconde et il est réinitialisé à zéro chaque semaine à minuit durant la nuit de samedi à dimanche. Le temps GNSS ne dépasse donc jamais 604 800 s, soit le nombre total de secondes écoulées dans une semaine. Pour cette raison, il est important de vérifier la valeur calculée pour  $t_k$  dans l'éventualité d'un retournement du temps GNSS (passage de 603 799 s à 0 s) :

$$\text{Si : } t_k > 302400 \Rightarrow t_k \leftarrow t_k - 604800 \quad (2.12)$$

$$\text{Si : } t_k < -302400 \Rightarrow t_k \leftarrow t_k + 604800 \quad (2.13)$$

On peut ainsi considérer  $t$  comme étant à l'intérieur de l'intervalle de validité de l'éphéméride si l'inégalité suivante est respectée :

$$t_k < \Delta t_{oe} / 2 \quad (2.14)$$

Si tel est le cas, on peut alors poursuivre avec le calcul de la position du satellite. Il est toutefois préférable de s'en abstenir dans le cas contraire car les calculs pourraient donner des résultats erronés. Ceci est dû au fait que les paramètres contenus dans les éphémérides sont estimés à l'aide de courbes de régression. Ces courbes sont choisies pour être optimales à l'intérieur de l'intervalle de validité des éphémérides, mais elles peuvent s'éloigner considérablement de la réalité en dehors de ce même intervalle.

Voici donc la procédure synthétisée pour le calcul de la position d'un satellite à partir des éphémérides. Cette procédure débute par le calcul du mouvement moyen  $n_0$  et du mouvement moyen corrigé  $n$  du satellite (ICD-GPS-200C, 2000) :

$$n_0 = \sqrt{\mu / A^3} \quad (2.15)$$

$$n = n_0 + \Delta n \quad (2.16)$$

Puis on poursuit avec le calcul de l'anomalie moyenne  $M_k$  de manière à positionner de façon grossière le satellite le long de son orbite :

$$M_k = M_0 + nt_k \quad (2.17)$$

Il s'agit ensuite de résoudre l'équation de Kepler pour l'anomalie excentrique  $E_k$ . Puisque cette équation est non linéaire, on la résout généralement de façon numérique par la méthode de Newton (se référer à l'annexe 3).

$$M_k = E_k - e \cdot \sin E_k \quad (2.18)$$

À partir de  $E_k$ , on peut ensuite calculer l'anomalie vraie  $v_k$ , puis l'argument de latitude  $\Phi_k$ . À noter ici qu'atan2 représente la fonction d'arctangente à quatre quadrants :

$$v_k = \text{atan} 2 \left( \sqrt{1-e^2} \sin E_k, \cos E_k - e \right) \quad (2.19)$$

$$\Phi_k = v_k + \omega \quad (2.20)$$

Il est ensuite possible de déterminer les termes de correction d'orbite, aussi appelés perturbations harmoniques secondes. Les éléments calculés visent à corriger l'argument de latitude ( $\delta u_k$ ), le rayon ( $\delta r_k$ ) et l'inclinaison d'orbite ( $\delta i_k$ ) :

$$\begin{aligned} \delta u_k &= c_{us} \sin 2\Phi_k + c_{uc} \cos 2\Phi_k \\ \delta r_k &= c_{rs} \sin 2\Phi_k + c_{rc} \cos 2\Phi_k \\ \delta i_k &= c_{is} \sin 2\Phi_k + c_{ic} \cos 2\Phi_k \end{aligned} \quad (2.21)$$

De là, on obtient l'argument de latitude corrigé  $u_k$ , le rayon corrigé  $r_k$  et l'inclinaison corrigée  $i_k$  :

$$u_k = \Phi_k + \delta u_k \quad (2.22)$$

$$r_k = A(1 - e \cos E_k) + \delta r_k \quad (2.23)$$

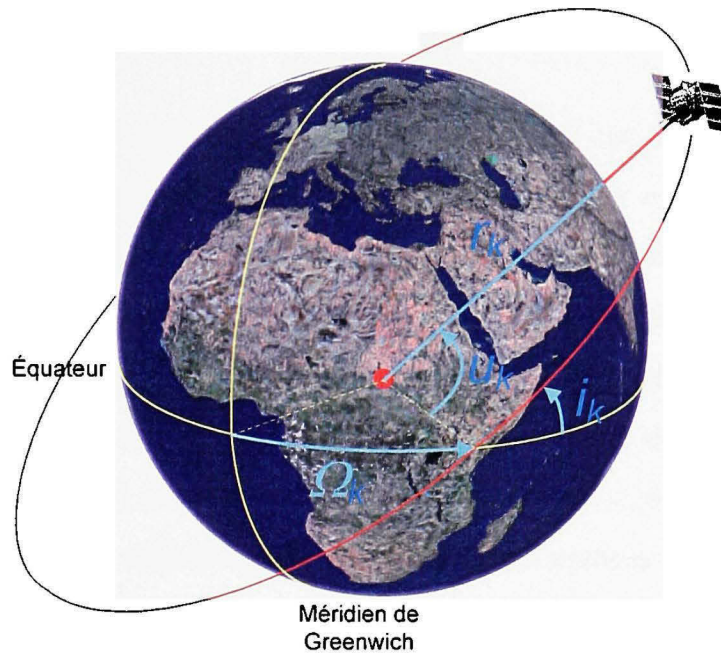
$$i_k = i_0 + \delta i_k + \dot{i} t_k \quad (2.24)$$

Un dernier paramètre reste enfin à être déterminé, soit la longitude du nœud ascendant corrigée ( $\Omega_k$ ) :

$$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e) t_k - \dot{\Omega}_e t_{oe} \quad (2.25)$$

Cette dernière équation tient compte de la vitesse de rotation de la Terre (voir Tableau 2.1) afin de positionner correctement le plan orbital du satellite au temps  $t$  spécifié.





**Figure 2.6 Représentation des éléments orbitaux.**

Les éléments orbitaux  $\Omega_k$ ,  $u_k$ ,  $r_k$  et  $i_k$  ainsi calculés permettent de décrire complètement la position d'un satellite sur son orbite telle qu'illustrée à la Figure 2.6. Les équations qui suivent n'ont donc pour but que de convertir cette position de manière à la représenter sur un référentiel cartésien ECEF, tel que spécifié dans le standard WGS-84 par exemple. La prochaine étape consiste donc à calculer la position du satellite en  $x$  et en  $y$  sur son plan orbital :

$$\begin{aligned} x_k' &= r_k \cos u_k \\ y_k' &= r_k \sin u_k \end{aligned} \quad (2.26)$$

La dernière étape consiste enfin à situer le satellite au sein d'un référentiel ECEF :

$$\begin{aligned} x_k &= x_k' \cos \Omega_k - y_k' \cos i_k \sin \Omega_k \\ y_k &= x_k' \sin \Omega_k + y_k' \cos i_k \cos \Omega_k \\ z_k &= y_k' \sin i_k \end{aligned} \quad (2.27)$$

### 2.2.6 Calcul des corrections d'horloge des satellites

Jusqu'à présent, nous avons supposé que tous les satellites d'une même constellation étaient parfaitement synchronisés entre eux et qu'ils transmettaient leur message de navigation à la même cadence. Dans les faits toutefois, l'horloge de chacun des satellites possède un biais et une dérive qui lui sont propres, faisant en sorte qu'il n'existe pas de synchronisation parfaite entre les satellites. Tel que mentionné précédemment, les stations de contrôle au sol surveillent plutôt les horloges satellites et s'assurent que le biais entre chacune d'entre elles et le temps GPS (ou Galileo) demeure raisonnablement petit (i.e. en dessous de 1 ms). De la même manière que pour les éphémérides, une série de paramètres est ainsi transmise via les satellites afin que les utilisateurs puissent calculer à quelques nanosecondes près les corrections d'erreurs d'horloges. Ces paramètres sont présentés au Tableau 2.3 pour les systèmes GPS et Galileo.

Tableau 2.3  
Paramètres de correction d'horloge pour les systèmes GPS et Galileo

Paramètres	Description	Unités
$a_{f0}$	Biais d'horloge	s
$a_{f1}$	Dérive d'horloge	s/s
$a_{f2}$	Variation de la dérive d'horloge	s/s <sup>2</sup>
$T_{GD}$	Délai de groupe	s
$t_{oc}$	Temps de référence des corrections d'horloge	s
$\Delta t_{oc}$	Période de validité des corrections d'horloge	s

De façon générale, les paramètres  $t_{oc}$  et  $\Delta t_{oc}$  ont la même valeur que les paramètres  $t_{oe}$  et  $\Delta t_{oe}$  des éphémérides. Les équations (2.11) à (2.14) peuvent donc être aussi employées de manière à vérifier l'intervalle de validité des corrections d'horloges.

Le temps GPS (ou Galileo) au moment de l'émission se calcule ensuite directement comme suit (ICD-GPS-200C, 2000) :

$$t = t_{sv} - \Delta t_{sv} \quad (2.28)$$

où :

- $t$  est le temps GPS (ou Galileo) au moment de l'émission des signaux
- $t_{SV}$  est le temps du satellite (SV – *Satellite Vehicule*) à l'émission
- $\Delta t_{SV}$  est la correction d'horloge du satellite

La correction d'horloge se calcule quant-elle ainsi :

$$\Delta t_{SV} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f1}(t - t_{oc})^2 + \Delta t_r - k T_{GD} \quad (2.29)$$

où :

- $\Delta t_r$  est le terme de correction relativiste donné par l'équation (2.39)
- $k$  est un coefficient sans unité qui dépend de la fréquence des signaux reçus. Les valeurs typiques pour le GPS sont de 1 et  $1,64694$  pour les fréquences L1 et L2 respectivement (ICD-GPS-200C, 2000).

Enfin, il s'agit de calculer le terme de correction relativiste tel que :

$$\Delta t_r = \frac{-2\mu^{1/2}}{c^2} e A^{1/2} \sin E_k \quad (2.30)$$

où :

- $\mu$  est la constante gravitationnelle universelle de la Terre et  $c$  est la vitesse de la lumière dans le vide (voir Tableau 2.1)
- $e$  représente l'excentricité de l'orbite du satellite et  $A^{1/2}$  la racine carrée de son demi-grand axe (voir Tableau 2.2)
- $E_k$  est l'anomalie excentrique obtenue après avoir résolu l'équation (2.18)



La théorie de la relativité telle qu'énoncée par Albert Einstein au début des années 1900 stipule qu'en s'approchant de la vitesse de la lumière, un observateur voit sa masse augmenter et le temps se dilater. Or, on tient généralement compte de ce phénomène avant le lancement des satellites de navigation en ajustant la fréquence de leur horloge en fonction de leur vitesse de déplacement moyenne sur leur orbite. Un terme de correction relativiste est tout de même nécessaire car les orbites des satellites ne sont pas circulaires mais plutôt légèrement elliptiques. Par conséquent, des variations de vitesse significatives se produisent lorsque les satellites se déplacent d'un point à l'autre de leur orbite, entraînant du même coup un décalage de leur horloge de l'ordre de quelques dizaines de nanosecondes ( $> 3$  ns d'erreur) (Misra et Enge, 2006).

Une fois les corrections d'horloges calculées, il est aussi nécessaire de les appliquer aux mesures de pseudo-distances. En développant l'équation (2.4) développée précédemment, on obtient alors :

$$\rho = d + b_u + c \left( \Delta t_{iono} + \Delta t_{tropo} + \Delta t_{mt} - \Delta t_{SV} \right) \quad (2.31)$$

Il est important de noter ici le signe négatif appliqué au terme  $\Delta t_{SV}$ , qui est une correction, par opposition aux termes  $\Delta t_{iono}$ ,  $\Delta t_{tropo}$  et  $\Delta t_{mt}$  qui sont des délais supplémentaires liés respectivement à l'ionosphère, à la troposphère et aux multitrajets. Ces sujets sont abordés un peu plus loin dans ce chapitre.

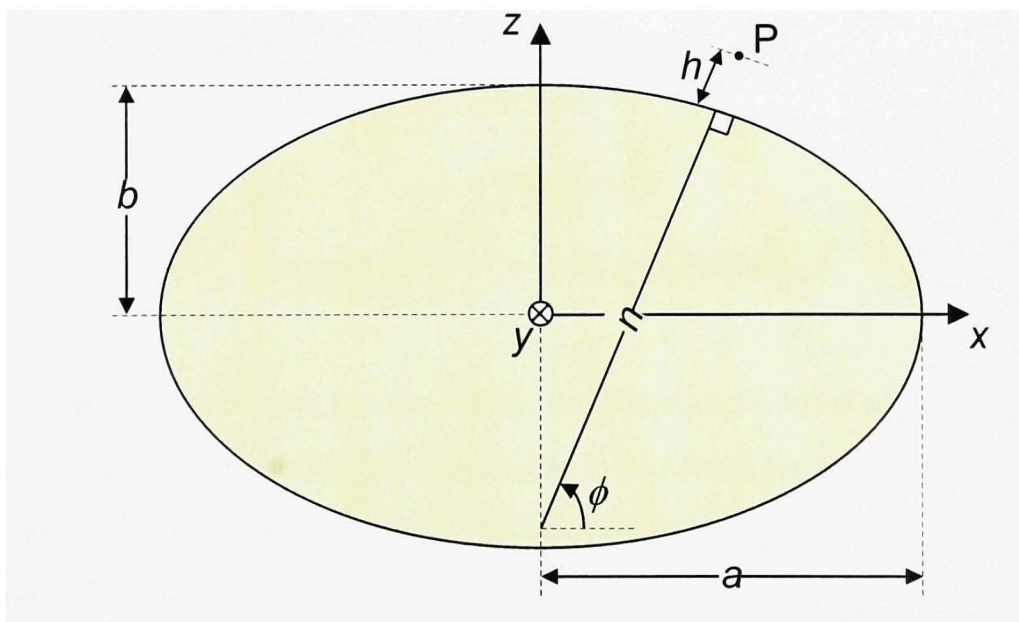
### 2.2.7 Conversion des temps GPS, Galileo et UTC

Puisque le temps GPS diffère légèrement du temps universel, ou UTC (temps universel coordonné), une série de paramètres est aussi transmise via les satellites GPS de manière à calculer le temps UTC avec une précision d'au moins 1  $\mu$ s (ICD-GPS-200C, 2000). Dans les systèmes plus récents, tels que Galileo par exemple, l'information est aussi transmise de manière à convertir le temps Galileo au temps GPS (ESTEC, 2007). En procédant ainsi, il devient possible d'obtenir une solution de navigation avec trois satellites GPS et un seul

satellite Galileo par exemple, chose qui n'est pas possible jusqu'à présent avec le système GLONASS (Cook, 1997; Eastwood, 1990). Dans ce dernier cas, au moins cinq satellites sont nécessaires afin d'estimer le biais entre les temps GPS et GLONASS.

### 2.2.8 Passage des coordonnées cartésiennes aux coordonnées géographiques

Les récepteurs de positionnement opèrent généralement en coordonnées cartésiennes. Il peut toutefois s'avérer plus pratique de convertir ces coordonnées en mesures de latitude ( $\phi$ ), de longitude ( $\lambda$ ) et d'altitude ( $h$ ) afin de se positionner sur une carte géographique, par exemple. Bien que l'opération inverse (i.e. le passage de coordonnées géographiques à cartésiennes) s'effectue directement, il faut cependant procéder par itérations afin de calculer les coordonnées géographiques. Ceci découle du fait que la Terre est considérée comme un ellipsoïde plutôt qu'une sphère parfaite; le rayon de courbure de la Terre ( $n$  ou rayon de courbure normal) à un point P est donc fonction de la latitude tel qu'illustré ci-dessous (se référer au Tableau 2.1 pour les paramètres  $a$ ,  $b$  et  $e$ ).



**Figure 2.7** Représentation du rayon de courbure d'un ellipsoïde.

Le rayon de courbure normal  $n$  se calcule donc (Misra et Enge, 2006) :

$$n = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \quad (2.32)$$

À partir du paramètre  $n$ , il est possible de déterminer directement les coordonnées cartésiennes d'un point à partir de ses coordonnées géographiques, tel que :

$$x = (n + h) \cos \phi \cos \lambda \quad (2.33)$$

$$y = (n + h) \cos \phi \sin \lambda \quad (2.34)$$

$$z = [n(1 - e^2) + h] \sin \phi \quad (2.35)$$

Des équations (2.33) à (2.35), on peut ensuite découler tour à tour les équations de conversion des coordonnées cartésiennes à géographiques (Wikipedia, 2007c) :

$$\phi = \text{atan} \left( \frac{z + e^2 n \sin \phi}{\sqrt{x^2 + y^2}} \right) \quad (2.36)$$

$$\lambda = \text{atan} 2(y, x) \quad (2.37)$$

$$h = \cos \phi \sqrt{x^2 + y^2} + \sin \phi (z + e^2 n \sin \phi) - n \quad (2.38)$$

Puisque le terme  $\phi$  se retrouve des deux côtés de l'équation (2.36) et que le terme  $n$  est aussi fonction de  $\phi$ , il est alors préférable de résoudre le problème par itération. Il s'agira dans un premier temps d'approximer la latitude, puis de répéter les étapes 2 à 4 de l'algorithme ci-dessous jusqu'à ce que l'erreur soit suffisamment petite :

$$1) \phi_0 = \text{atan} \left( \frac{z}{\sqrt{x^2 + y^2}} \right) \quad (2.39)$$



$$2) \ n_i = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi_i}} \quad (2.40)$$

$$3) \ \phi_{i+1} = \text{atan} \left( \frac{z + e^2 n_i \sin \phi_i}{\sqrt{x^2 + y^2}} \right) \quad (2.41)$$

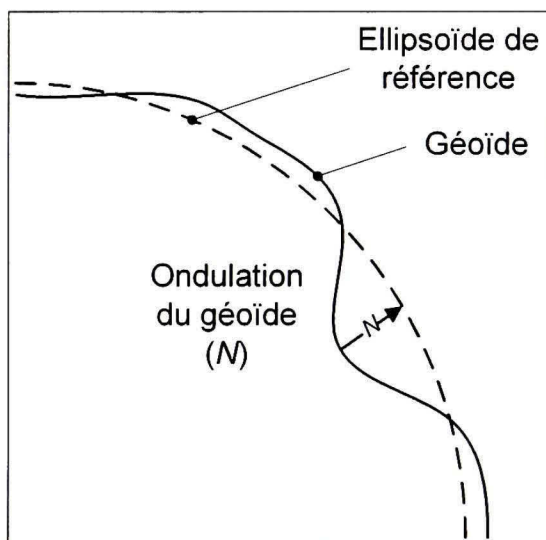
$$4) \ \text{erreur} = |\phi_{i+1} - \phi_i| \quad (2.42)$$

5) Retourner à l'étape 2 si  $\text{erreur} > \text{erreur}_{\min}$

Il faut noter que l'algorithme ci-dessus converge assez rapidement, de telle sorte que 10 itérations sont généralement suffisantes pour obtenir une précision millimétrique au niveau de la latitude.

### 2.2.9 Modèle géopotential EGM96

La NASA, en collaboration avec la *National Imagery and Mapping Agency* (NIMA), a développé en 1996 un nouveau modèle gravitationnel de la Terre (*Earth Gravitational Model 1996* – EGM96). Ce modèle visait entre autres à reproduire le plus fidèlement possible le géoïde, une surface équipotentielle du champ gravitationnel terrestre choisie pour être voisine au niveau moyen des mers, et ainsi pallier les défauts de l'ellipsoïde WGS-84.



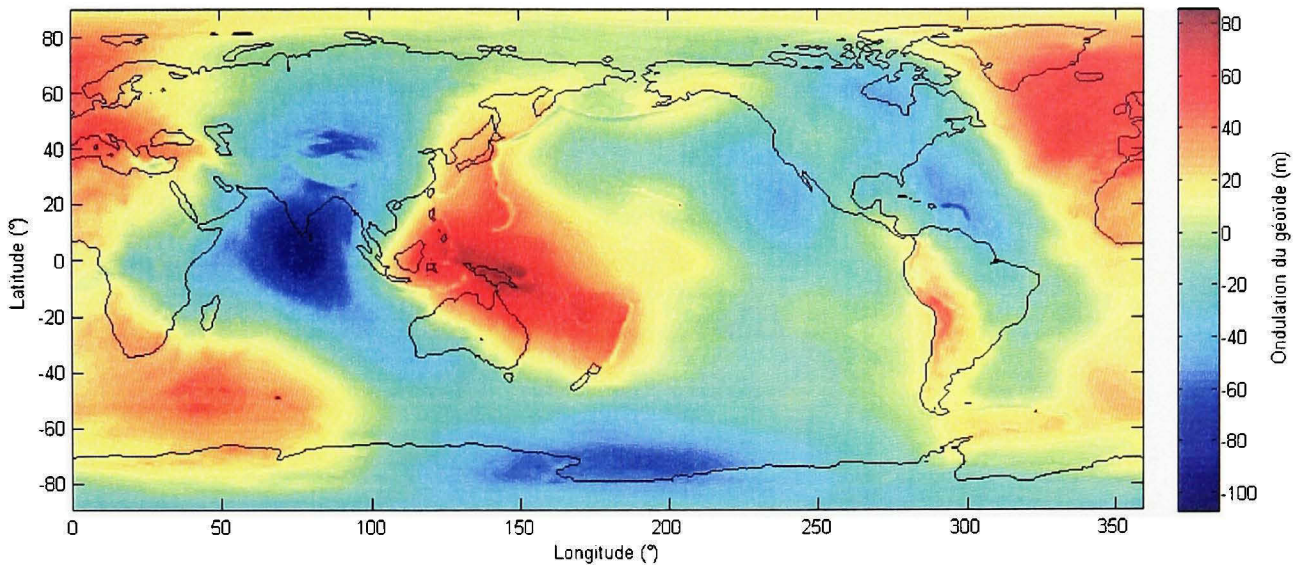
**Figure 2.8** *Ellipsoïde, géoïde et ondulation du géoïde.*

Dans les faits, de légères irrégularités sont décelables au niveau du champ gravitationnel terrestre, ayant pour conséquence de faire varier le niveau moyen des mers d'un endroit à l'autre sur le globe. Ces variations par rapport à l'ellipsoïde de référence WGS-84, appelées ondulations du géoïde ( $N$  - voir Figure 2.8), peuvent ainsi atteindre des valeurs allant de -105 m à +85 m tel qu'illustré à la Figure 2.9. Il est donc impératif de corriger ces ondulations au sein d'un récepteur de positionnement afin de connaître son altitude réelle par rapport au niveau moyen de la mer ( $H$ ). L'équation ci-dessous (EGM96, 2004) est employée afin de calculer cette valeur à partir de l'altitude au-dessus de l'ellipsoïde de référence ( $h$ ) :

$$H = h - N \quad (2.43)$$

Le modèle EGM96 se calcule par un développement en harmoniques sphériques d'ordre 360 et de degré 360. Vu la complexité des équations qui en découlent, le lecteur est toutefois invité à consulter la référence suivante pour plus de détails : (EGM96, 2004). Il faut mentionner que le développement en question est une opération mathématique très lourde impliquant le calcul de plus de 130 000 sinus et cosinus à la fois. Ce genre de calcul ne se prête généralement pas bien aux microprocesseurs embarqués employés dans les récepteurs de positionnement actuels. Par conséquent, des simplifications doivent être apportées au modèle afin qu'il puisse être calculé en temps réel dans les récepteurs, entraînant du coup une perte de précision. Une approche plus simple consiste plutôt à enregistrer une grille précalculée dans la mémoire du récepteur et à effectuer une interpolation entre les points de la grille.

Pour le récepteur hybride présenté dans ce document, un algorithme de type spline cubique est employé sur une grille dont le maillage est de 15' x 15' (où « ' » représente des minutes d'arc - consulter l'annexe 4 pour plus d'information sur la spline cubique). La grille en question, qui peut être trouvée dans (EGM96, 2004), couvre la surface entière de la Terre, pour un total de plus d'un million de points.



**Figure 2.9 Ondulation du géoïde EGM96.**

### 2.2.10 Effet de la rotation de la Terre sur les coordonnées des satellites

L'effet de la rotation de la Terre sur la position des satellites de navigation est un autre phénomène dont on doit tenir compte dans les récepteurs de positionnement. Dans les faits, un délai de transmission non négligeable existe entre les satellites de navigation et les utilisateurs au sol. Or, un utilisateur calcule sa position par rapport à un repère de coordonnées fixé à la Terre, mais le fait est que la Terre tourne durant l'intervalle de transmission des signaux. Par conséquent, la position de l'utilisateur lors de la réception des signaux se retrouve dans un référentiel différent de celui des satellites au temps d'émission (voir Figure 2.10).

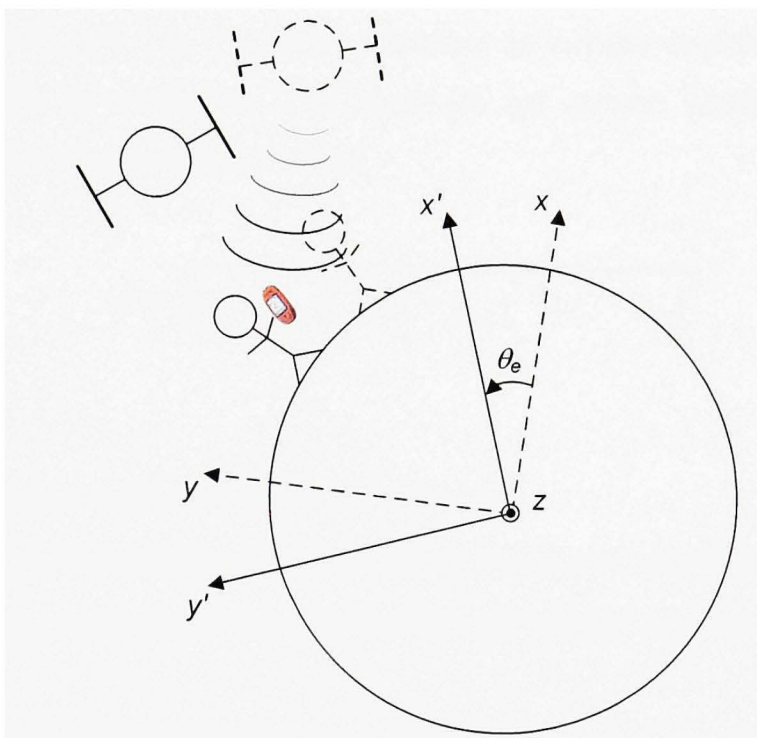
Afin d'avoir un ordre de grandeur du décalage introduit par ce phénomène, prenons l'exemple d'un satellite GPS situé directement au-dessus d'un utilisateur à l'équateur. Le rayon d'orbite d'un satellite GPS étant d'environ 26 600 m, la distance le séparant de l'utilisateur est alors de 20 200 km (on estime le rayon de la Terre à l'équateur à environ 6400 km). À partir de l'équation (2.3), on peut alors calculer le délai de propagation des signaux comme étant :



$$\Delta t_t = t_e - t_r = \frac{d}{c} \approx \frac{20\,200 \times 10^3 \text{ m}}{3 \times 10^8 \text{ m/s}} = 67,3 \text{ ms} \quad (2.44)$$

où :

- $\Delta t_t$  est le délai de transmission des signaux
- $t_e$  et  $t_r$  sont les temps d'émission et de réception
- $d$  est la distance séparant le satellite de l'utilisateur
- $c$  est la vitesse de la lumière dans le vide



**Figure 2.10 Changement de référentiel  
dû à la rotation de la Terre.**

Connaissant la vitesse de rotation de la Terre ( $\dot{\Omega}_e$  - voir Tableau 2.1), il est alors possible de calculer l'angle de rotation effectué par celle-ci ( $\theta_e$ ) durant cet intervalle de 67,3 ms :

$$\theta_e = \dot{\Omega}_e \Delta t_t \approx 67,3 \times 10^{-3} \text{ s} \cdot 7,29 \times 10^{-5} \text{ rad/s} = 4,906 \text{ } \mu\text{rad} \quad (2.45)$$

Puisque le satellite est situé directement au-dessus de l'équateur, le déplacement associé à la rotation du système de coordonnées peut être facilement calculé par trigonométrie :

$$\Delta s = r_{sat} \sin \theta_e = 26600 \times 10^3 \text{ m} \cdot \sin(4,906 \times 10^{-6} \text{ rad}) \approx 130,5 \text{ m} \quad (2.46)$$

où :

- $\Delta s$  est le déplacement du satellite associé à la rotation du référentiel ECEF
- $r_{sat}$  est le rayon d'orbite du satellite

Vu le déplacement de 130,5 m, il est donc important de corriger ce phénomène afin d'obtenir une bonne solution de navigation. Les équations qui suivent permettent de corriger ce problème de rotation d'un référentiel ECEF :

$$d = \sqrt{(x_u - x_{sat})^2 + (y_u - y_{sat})^2 + (z_u - z_{sat})^2} \quad (2.47)$$

où :

- $d$  est la distance entre un satellite et l'utilisateur
- $x_u, y_u$  et  $z_u$  représentent la position ECEF de l'utilisateur au temps de réception
- $x_{sat}, y_{sat}$  et  $z_{sat}$  représentent la position ECEF du satellite au temps d'émission

L'équation ci-dessus découle directement de l'équation (2.2) et permet de réaliser une première approximation de la distance entre un satellite et l'utilisateur. De là, il s'agit de reprendre l'équation (2.44) de manière à calculer le délai de propagation  $\Delta t_t$  des signaux entre le satellite et l'utilisateur :

$$\Delta t_t = \frac{d}{c} \quad (2.48)$$

Il s'agit ensuite de calculer l'angle de rotation  $\theta$  associé au référentiel ECEF :

$$\theta = -\dot{\Omega}_e \Delta t_t \quad (2.49)$$

Enfin, grâce aux équations qui suivent, il est possible de calculer la position du satellite corrigée au sein du nouveau référentiel ECEF (ICD-GPS-200C, 2000) :

$$\begin{aligned} x'_{sat} &= x_{sat} \cos\theta - y_{sat} \sin\theta \\ y'_{sat} &= x_{sat} \sin\theta + y_{sat} \cos\theta \\ z'_{sat} &= z_{sat} \end{aligned} \quad (2.50)$$

Il s'agit finalement de répéter le processus jusqu'à ce que l'erreur résiduelle soit suffisamment petite (Tsui, 2000). Il faut noter toutefois qu'une seule itération est suffisante pour la majorité des applications, l'erreur résiduelle étant alors inférieure à un millimètre (voir démonstration à l'annexe 5).

## 2.3 Sources d'erreurs et précision du positionnement par satellite

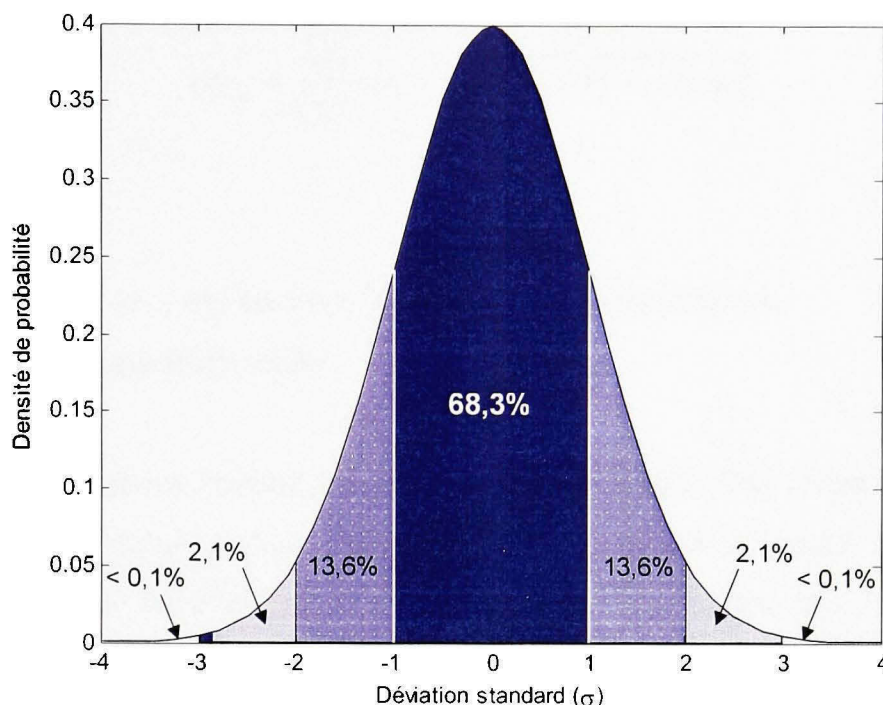
Des mouvements tectoniques aux perturbations cosmiques, les sources d'erreurs rencontrées dans le domaine de la navigation satellite sont nombreuses et variées. Faire l'inventaire exhaustif de tous les phénomènes pouvant influencer de près ou de loin le positionnement par satellite se voudrait sans doute un projet bien ambitieux et dépasserait de loin le cadre de ce document. Pour cette raison, la section qui suit ne résume que les principales sources d'erreurs rencontrées sur le terrain.

### 2.3.1 Définition des erreurs RMS, à $1\sigma$ et à $2\sigma$

Avant d'entrer dans le vif du sujet, il est important de faire le point sur la façon d'exprimer les erreurs de mesure et de calcul. Tout d'abord, les erreurs sont habituellement considérées comme des processus statistiques stationnaires et ergodiques, c'est-à-dire que leur distribution statistique et leur moyenne demeurent invariables dans le temps. On suppose



aussi qu'elles sont distribuées de façon gaussienne et qu'elles ont une moyenne nulle. On parlera donc de distributions de probabilité normales et centrées (voir Figure 2.11).



**Figure 2.11** *Distribution normale et centrée.*

Lorsqu'on se réfère à la précision d'un récepteur de positionnement, on emploie souvent le terme « erreur à 1 écart-type » ( $1\sigma$ ) ou « à 2 écarts-types » ( $2\sigma$ ). Cette notation se rapporte directement à l'aire sous la courbe de la distribution normale présentée à la Figure 2.11 et signifie le niveau de confiance attribué à une mesure. Ainsi,  $1\sigma$  équivaut à 68,3% de probabilité,  $2\sigma$  à 95,5% de probabilité,  $3\sigma$  à 99,7% et ainsi de suite. Par exemple, un récepteur dont la précision 2D est de 1,8 m à  $1\sigma$  signifie que les mesures de latitude et de longitude données par ce récepteur demeurent 68,3% du temps à l'intérieur d'un cercle de rayon de 1,8 m.

Le terme erreur RMS (*Root Mean Square*) est aussi souvent employé pour décrire la précision des mesures. Puisqu'on fait référence à une distribution normale centrée, ce terme a donc exactement la même signification que  $1\sigma$  (l'opération RMS est la définition de l'écart type pour une distribution normale centrée) (Wikipedia, 2007e). On emploie d'ailleurs

l'opérateur RMS lorsqu'il s'agit d'additionner deux ou plusieurs erreurs distribuées de façon normale, tel que :

$$err_{tot} = \sqrt{\sum_{n=1}^N err_n^2} = \sqrt{err_1^2 + err_2^2 + \dots + err_N^2} \quad (2.51)$$

où :

- $err_1, err_2, \dots, err_N$  représentent  $N$  erreurs RMS indépendantes
- $err_{tot}$  est l'erreur RMS totale

Plusieurs autres notations existent afin de décrire la précision d'un récepteur de navigation. Parmi les plus populaires, notons l'erreur circulaire probable (*Circular Error Probable* – CEP), définie comme étant le rayon d'un cercle à l'intérieur duquel on a 50% de probabilité de se situer. Mentionnons aussi l'erreur 2DRMS (deux fois la distance RMS), équivalente à une erreur 2D à  $2\sigma$  (ne pas confondre avec l'erreur RMS 2D, soit une erreur 2D à  $1\sigma$ ). Le Tableau 2.4 résume les notations introduites dans cette section, tandis que le Tableau 2.5 présente les facteurs de conversions servant à passer d'une notation à l'autre (Wikipedia, 2007a).

Tableau 2.4  
Mesures de précision les plus fréquentes

Mesures de précision	Probabilité (%)
$1\sigma$	68,3
$2\sigma$	95,5
$3\sigma$	99,7
RMS	68,3
CEP	50
2DRMS	95,5

Tableau 2.5  
Conversion entre CEP,  $1\sigma$ ,  $2\sigma$ ,  $3\sigma$ , RMS et 2DRMS

Vers De	CEP	$1\sigma$ / RMS	$2\sigma$ / 2DRMS	$3\sigma$
CEP	1,0	1,2	2,4	4,8
$1\sigma$ / RMS	0,83	1,0	2,0	3,0
$2\sigma$ / 2DRMS	0,42	0,5	1,0	1,5
$3\sigma$	0,28	0,33	0,67	1,0

### 2.3.2 Erreurs liées aux segments spatial et de contrôle

Deux sources d'erreurs sont directement imputables aux satellites de navigation et aux centres de contrôle au sol. La première source, qui se retrouve liée aux éphémérides, se traduit par un calcul erroné de la position des satellites. La seconde source est quant à elle liée à l'horloge à bord des satellites et se reflète dans les paramètres de correction d'horloge. Dans les deux cas, ce sont les meilleures estimations qui sont transmises aux satellites pour être ensuite rediffusées aux utilisateurs. Or, une erreur résiduelle subsiste toujours au niveau de ces estimations. Cette erreur tend cependant à diminuer au fur et à mesure que la technologie progresse. À preuve, certaines sources datant des années 1990 (Kaplan, 1996, ; Knight, 1998) traitent d'erreurs à  $1\sigma$  de l'ordre de 5,4 m à ce niveau, alors qu'on parle plutôt de 2,6 m présentement (IGS, 2007).

### 2.3.3 Disponibilité sélective (SA)

Tel que mentionné au chapitre précédent, le GPS demeure avant tout un système militaire. Afin de limiter la précision aux utilisateurs non autorisés (les civils, par exemple), le DoD a donc mis en place en mars 1990 un mécanisme visant à dégrader de façon artificielle la qualité des signaux GPS. Ce mécanisme, appelé disponibilité sélective (SA - *Selective Availability*) est mis en oeuvre d'au moins deux façons connues (Grewal, Weill et Andrews, 2001). L'une consiste à altérer les éphémérides alors que l'autre repose sur le principe de tramage des horloges satellites (*dithering*). Une fois la SA activée, une erreur de 30 m ( $1\sigma$ ) s'ajoute au niveau des mesures de pseudo-distances GPS, limitant ainsi la précision de la



solution de navigation à 100 m RMS horizontalement et 156 m RMS verticalement (Grewal, Weill et Andrews, 2001).

La SA a cependant été interrompue le 1<sup>er</sup> mai 2000 suite à l'annonce du président Clinton. Dès lors, les utilisateurs civils ont accès à une précision s'approchant de celle des militaires (Misra et Enge, 2006). Les civils ne bénéficient toutefois pas de toute la robustesse dont disposent les militaires, sans compter que le DoD peut encore revenir sur sa décision et réactiver la SA s'il en juge nécessaire. Ceci a justifié notamment le développement d'autres systèmes de navigation civils, tel que Galileo.

### **2.3.4 Délais ionosphériques**

L'ionosphère est une partie de l'atmosphère qui s'étend d'environ 70 km à 1000 km au-dessus de la surface de la Terre. Sous l'effet des radiations solaires et cosmiques, les gaz présents dans cette couche atmosphérique sont ionisés et des nuages d'électrons libres sont formés. Ces nuages d'électrons agissent de façons variées sur la propagation des ondes électromagnétiques, les réfléchissant dans certains cas (propriété fort exploitée au niveau de la radio amateur), les retardant dans d'autres. En ce qui concerne les signaux provenant des satellites de navigation, ceux-ci sont principalement retardés par l'ionosphère. Certaines irrégularités rencontrées au passage de l'ionosphère, appelées scintillations, peuvent aussi perturber la phase et l'amplitude des signaux, causant à l'occasion des pertes de verrouillages au niveau des boucles de phase des récepteurs.

Puisque les délais introduits par l'ionosphère dépendent du temps de la journée, de la saison, de la position géographique et des activités solaire et géomagnétique, il est généralement difficile de réaliser une prévision exacte. Ces délais, mesurés au zénith, s'échelonnent généralement entre 1 m la nuit et 5 à 15 m tard en après-midi (Grewal, Weill et Andrews, 2001). Étant donné que les signaux près de l'horizon doivent traverser une couche ionosphérique plus épaisse qu'au zénith, des délais aussi grands que 50 m peuvent parfois être atteints pour de faibles angles d'élévations. De façon générale toutefois, on fixe un angle

d'élévation minimal de 5° ou 10°, appelé masque d'élévation, de manière à rejeter tous les satellites se trouvant près de l'horizon. Ceci permet, la plupart du temps, de minimiser les erreurs sur les mesures de pseudo-distances à 10 m ( $1\sigma$ ) (Grewal, Weill et Andrews, 2001).

L'utilisation d'un modèle diffusé par les satellites de navigation permet aux utilisateurs qui ne captent les signaux que sur une seule fréquence de diminuer de 50% les erreurs introduites par l'ionosphère (ICD-GPS-200C, 2000), les limitant ainsi à 5 m ( $1\sigma$ ). Ceci est le cas du récepteur hybride présenté dans ce document. Le lecteur est invité à consulter (ICD-GPS-200C, 2000) pour plus de détails sur ce modèle. Les utilisateurs ayant la chance de bénéficier d'au moins deux fréquences peuvent toutefois limiter cette erreur à 2,3 m ( $1\sigma$ ) (Knight, 1998). Cette dernière approche repose sur le fait que les délais ionosphériques sont en fonction de la fréquence des signaux. Ainsi, les satellites GPS émettent simultanément à deux fréquences distinctes, soit L1 et L2, de manière à permettre aux utilisateurs de mesurer les délais ionosphériques. Il faut toutefois noter que les récepteurs GPS bifréquences coûtent présentement plus chers que les récepteurs à simple fréquence et que le signal GPS L2 militaire n'est pas toujours accessible aux utilisateurs civils (il peut être crypté à l'occasion). Les signaux L2C et L5 permettront toutefois de régler ce dernier problème avec la modernisation du GPS, en plus des signaux E5a et E5b du système Galileo.

### **2.3.5 Délais troposphériques**

La troposphère est la partie la plus basse de l'atmosphère. Elle s'élève jusqu'à 7 km aux pôles alors qu'elle peut atteindre jusqu'à 20 km d'altitude à l'équateur. La troposphère est surtout constituée de gaz secs et de vapeurs d'eau, des éléments qui participent ensemble à retarder les signaux électromagnétiques par le phénomène de réfraction. Les délais introduits sur les signaux provenant des satellites de navigation varient donc en fonction de l'indice de réfraction de l'air rencontré au passage. Pour cette raison, les délais de propagation troposphériques dépendent non seulement de la localisation géographique d'un utilisateur, mais aussi de son altitude, de la saison ainsi que de la météo.



De façon générale, les délais de propagation associés à la troposphère sont de l'ordre de 2,4 m ( $1\sigma$ ) au zénith et de 25 m ( $1\sigma$ ) à 5° d'élévation pour un utilisateur situé au niveau de la mer (Kaplan, 1996). Plusieurs modèles ont cependant été développés au fil des années afin de minimiser l'erreur troposphérique résiduelle autour de 2 m ( $1\sigma$ ) (Knight, 1998). Plus récemment, des études portant sur le modèle troposphérique d'EGNOS ont permis de démontrer la possibilité d'obtenir des erreurs inférieures à 1 m ( $1\sigma$ ) (Penna, 2001). C'est ce modèle qui a été retenu pour le développement du récepteur hybride présenté dans ce document. Le lecteur est donc invité à consulter (Penna, 2001) pour plus de détails.

### 2.3.6 Multitrajets

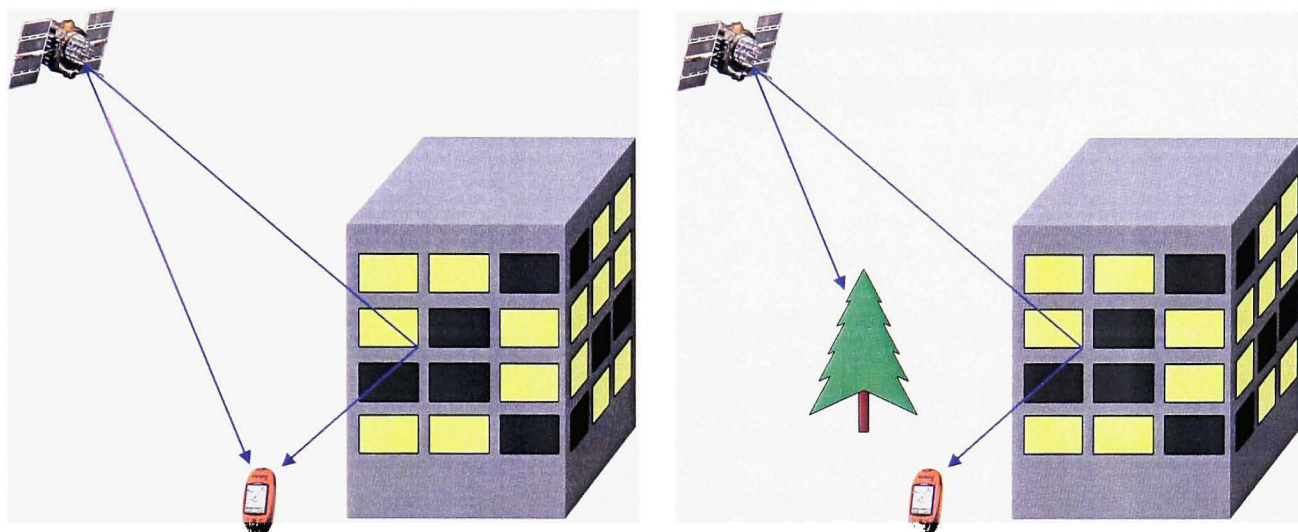
De tous les phénomènes affectant le positionnement par satellite, les multitrajets sont de loin la pire des sources d'erreurs. Bien que la majorité des d'erreurs puissent être minimisées, sinon complètement annulées, par des techniques telles que le positionnement différentiel par exemple (Knight, 1998), ceci n'est pas le cas des multitrajets qui demeurent encore aujourd'hui un problème difficile à contourner.

Le mot « multitrajets » est un terme technique se référant aux réflexions d'un signal électromagnétique. Puisque ces réflexions empruntent des trajets différents du signal principal (donc plus longs), elles peuvent alors induire un récepteur de navigation en erreur en lui fournissant des mesures de pseudo-distances plus grandes qu'en réalité. Deux situations couramment rencontrées sur le terrain sont illustrées à la Figure 2.12.

La première situation, illustrée à la gauche de la Figure 2.12, est la plus fréquente et consiste en la recombinaison d'un ou plusieurs multitrajets avec le signal principal. Pour ce genre de situation, plusieurs méthodes ont été développées ou sont présentement en cours de développement afin de diminuer l'impact des multitrajets sur les mesures de pseudo-distances. La plus célèbre d'entre elles demeure sans doute le *Narrow Correlator* qui permet de limiter aux alentours de 10 m les erreurs causées par les multitrajets au niveau du signal GPS C/A (Grewal, Weill et Andrews, 2001). La modernisation du GPS et l'arrivée prochaine



de Galileo promettent toutefois de minimiser davantage le problème en ayant recours à des codes d'étalement plus robustes face aux multitrajets (Julien et al., 2007).



**Figure 2.12** *Réception simultanée d'un signal et d'un multitrait (à gauche) et réception d'un multitrait seul (à droite).*

La seconde situation, illustrée à la droite de la Figure 2.12, est généralement rencontrée dans des environnements difficiles, tels qu'en ville par exemple, où les édifices et autres objets avoisinants font ombrage aux signaux satellites. Dans une telle situation, où seuls les multitrajets atteignent le récepteur, il est pratiquement impossible de différencier les multitrajets des véritables signaux émis par les satellites. Il est alors nécessaire de se tourner vers d'autres méthodes, généralement plus coûteuses, telles que les antennes réseau à commande de phase (*Phase Array Antenna*) par exemple (Brown et Mathews, 2005), ou les antennes à anneaux d'étranglement (*Choke Ring Antenna*), de manière à éliminer les multitrajets à la source.

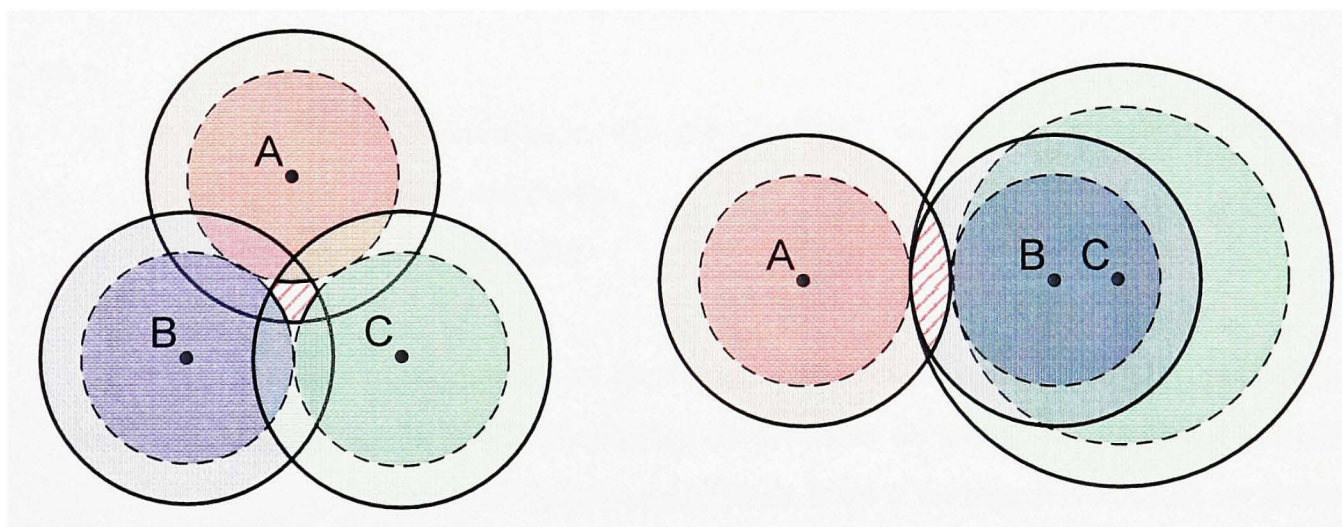
### 2.3.7 Erreurs liées au segment utilisateur

La principale source d'erreurs liée aux récepteurs de navigation est le bruit thermique. Ceci est fort compréhensible puisque les signaux de navigation sont situés sous le niveau du bruit thermique (abordé au chapitre 3). Bien que le rapport signal sur bruit (SNR – *Signal-to-Noise*

*Ratio*) soit de beaucoup amélioré au moment de la démodulation des signaux, une certaine quantité de bruit, estimée à environ 1,5 m RMS, subsiste toujours au niveau des mesures de pseudo-distances. Parmi les autres sources d'erreurs imputables aux récepteurs de navigation, on compte aussi les biais intercanaux (ou bruit d'intercorrelation) qui contribuent à accroître de 0,5 m RMS l'erreur sur les mesures de pseudo-distances (Kaplan, 1996).

### 2.3.8 Dilution de précision

La précision d'une solution de navigation est directement influencée par la disposition spatiale, ou géométrie, des satellites de positionnement. Ainsi, pour une solution de navigation classique avec quatre satellites, on sélectionnera parmi ceux visibles les satellites offrant la distribution spatiale la plus uniforme possible au profit d'une solution de navigation plus précise. Ce principe est illustré à la Figure 2.13.



**Figure 2.13** Exemples de distributions spatiales 2D (de gauche à droite) : bonne géométrie et mauvaise géométrie.

À la Figure 2.13, trois satellites (A, B et C) sont employés afin de déterminer une position en 2D. Chacun des satellites est entouré d'une zone d'incertitude, qui représente l'erreur équivalente totale sur les mesures de pseudo-distances (UERE – *User Equivalent Range Error*). L'UERE est en fait la somme quadratique de toutes les erreurs présentées précédemment et est calculé à l'aide de l'équation (2.51).

Ainsi, pour une bonne géométrie des satellites (i.e. une distribution spatiale uniforme), on obtiendra une région d'incertitude relativement petite au niveau de la solution de navigation (voir région hachurée à gauche de la Figure 2.13). Cette région d'incertitude tendra toutefois à s'agrandir au fur et à mesure que la géométrie des satellites se détériorera (voir région hachurée à droite de la Figure 2.13).

De façon plus rigoureuse, la géométrie des satellites se caractérise par une série d'indices, appelée dilution de précision (DOP – *Dilution of Precision*). L'indice le plus général, soit la dilution de précision géométrique (GDOP), permet d'estimer d'un point de vue global les erreurs de positionnement 3D et de biais d'horloge d'une solution de navigation, telles que (Misra et Enge, 2006) :

$$erreur_{GRMS} = \sigma \cdot GDOP \quad (2.52)$$

où :

- $erreur_{GRMS}$  est une estimation des erreurs RMS de positionnement et de biais d'horloge du récepteur combinées
- $\sigma$  est la valeur RMS de l'UERE

La DOP peut être aussi décomposée en divers autres indices, tels que la PDOP (dilution de précision de la position) ou la TDOP (dilution de précision du temps) de manière à estimer les erreurs de positionnement 3D ( $erreur_{3DRMS}$ ) et de biais d'horloge ( $erreur_{TRMS}$ ) de façon séparée :

$$erreur_{3DRMS} = \sigma \cdot PDOP \quad (2.53)$$

$$erreur_{TRMS} = \sigma \cdot TDOP \quad (2.54)$$

Afin de déterminer ces valeurs de DOP, il s'agit tout d'abord de calculer la ligne de visée entre les satellites et un utilisateur :



$$\mathbf{L}_k = \frac{\mathbf{x}_k - \mathbf{x}_u}{\|\mathbf{x}_k - \mathbf{x}_u\|} = \frac{\begin{bmatrix} x_k - x_u & y_k - y_u & z_k - z_u \end{bmatrix}}{\sqrt{(x_k - x_u)^2 + (y_k - y_u)^2 + (z_k - z_u)^2}} \quad (2.55)$$

où :

- $\mathbf{L}_k$  est le vecteur de ligne de visée pour le  $k^{\text{ième}}$  satellite
- $\mathbf{x}_k$  est le vecteur de position du  $k^{\text{ième}}$  satellite défini par  $x_k$ ,  $y_k$  et  $z_k$
- $\mathbf{x}_u$  est le vecteur de position de l'utilisateur défini par  $x_u$ ,  $y_u$  et  $z_u$

À partir de là, on forme la matrice de géométrie  $\mathbf{G}$  pour les  $K$  satellites employés dans la solution de navigation, telle que :

$$\mathbf{G} = \begin{bmatrix} \mathbf{L}_1 & 1 \\ \mathbf{L}_2 & 1 \\ \vdots & \vdots \\ \mathbf{L}_K & 1 \end{bmatrix} \quad (2.56)$$

On obtient donc une matrice de dimension  $K \times 4$ , où  $K \geq 4$  pour les besoins de l'exercice. Partant du fait que  $\hat{\mathbf{x}}$  et  $\hat{b}$  sont les coordonnées et biais d'horloge estimés à partir d'une solution de navigation, on peut alors démontrer que (Misra et Enge, 2006) :

$$\text{Cov} \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{b} \end{bmatrix} = \sigma^2 (\mathbf{G}^T \mathbf{G})^{-1} \quad (2.57)$$

On définit ainsi la matrice  $\mathbf{H}$  :

$$\mathbf{H} = (\mathbf{G}^T \mathbf{G})^{-1} \quad (2.58)$$

Il est alors possible de réécrire l'équation (2.57) en terme des éléments de la diagonale de la matrice  $\mathbf{H}$ , telle que :

$$\sigma_x^2 = \sigma^2 H_{11}; \quad \sigma_y^2 = \sigma^2 H_{22}; \quad \sigma_z^2 = \sigma^2 H_{33}; \quad \sigma_b^2 = \sigma^2 H_{44} \quad (2.59)$$

où :

- $H_{ii}$  représente le  $i^{\text{ème}}$  élément de la diagonale de  $\mathbf{H}$
- $\sigma_x$ ,  $\sigma_y$  et  $\sigma_z$  sont les erreurs de positionnement RMS en  $x$ ,  $y$  et  $z$
- $\sigma_b$  est l'erreur RMS du biais d'horloge à la sortie de la solution de navigation

En faisant un retour sur les équations (2.52) à (2.54), il est alors possible de définir les indices GDOP, PDOP et TDOP tels que :

$$\text{GDOP} = \sqrt{H_{11} + H_{22} + H_{33} + H_{44}} \quad (2.60)$$

$$\text{PDOP} = \sqrt{H_{11} + H_{22} + H_{33}} \quad (2.61)$$

$$\text{TDOP} = \sqrt{H_{44}} \quad (2.62)$$

De manière à caractériser plus en détail les erreurs de positionnement, il peut s'avérer utile de décomposer la PDOP de façon horizontale (HDOP) et verticale (VDOP). Plusieurs méthodes permettent d'y parvenir, mais la plus simple consiste à modifier la matrice  $\mathbf{G}$  telle que (Misra et Enge, 2006) :

$$\tilde{\mathbf{L}}_k = [\cos el_k \sin az_k \quad \cos el_k \cos az_k \quad \sin el_k] \quad (2.63)$$

où :

- $\tilde{\mathbf{L}}_k$  est le vecteur de ligne de visée pour le  $k^{\text{ième}}$  satellite
- $el_k$  et  $az_k$  sont l'élévation et l'azimut du  $k^{\text{ième}}$  satellite tels que mesurés aux coordonnées de l'utilisateur

À partir des équations (2.56), (2.58) et (2.63), on obtient alors les matrices modifiées  $\tilde{\mathbf{G}}$  et  $\tilde{\mathbf{H}}$ , desquelles on dérive la EDOP (dilution de précision horizontale en est) et la NDOP (dilution de précision horizontale en nord), en plus de la VDOP et de la TDOP :

$$\text{EDOP} = \sqrt{\tilde{H}_{11}}; \quad \text{NDOP} = \sqrt{\tilde{H}_{22}}; \quad \text{VDOP} = \sqrt{\tilde{H}_{33}}; \quad \text{TDOP} = \sqrt{\tilde{H}_{44}} \quad (2.64)$$

On calcule ensuite la HDOP via une moyenne quadratique des EDOP et NDOP :

$$\text{HDOP} = \sqrt{\text{EDOP}^2 + \text{NDOP}^2} \quad (2.65)$$

On peut finalement estimer les erreurs de positionnement horizontale ( $\text{erreur}_{HRMS}$ ) et verticale ( $\text{erreur}_{VRMS}$ ) d'une solution de navigation telles que :

$$\text{erreur}_{HRMS} = \sigma \cdot \text{HDOP} \quad (2.66)$$

$$\text{erreur}_{VRMS} = \sigma \cdot \text{VDOP} \quad (2.67)$$

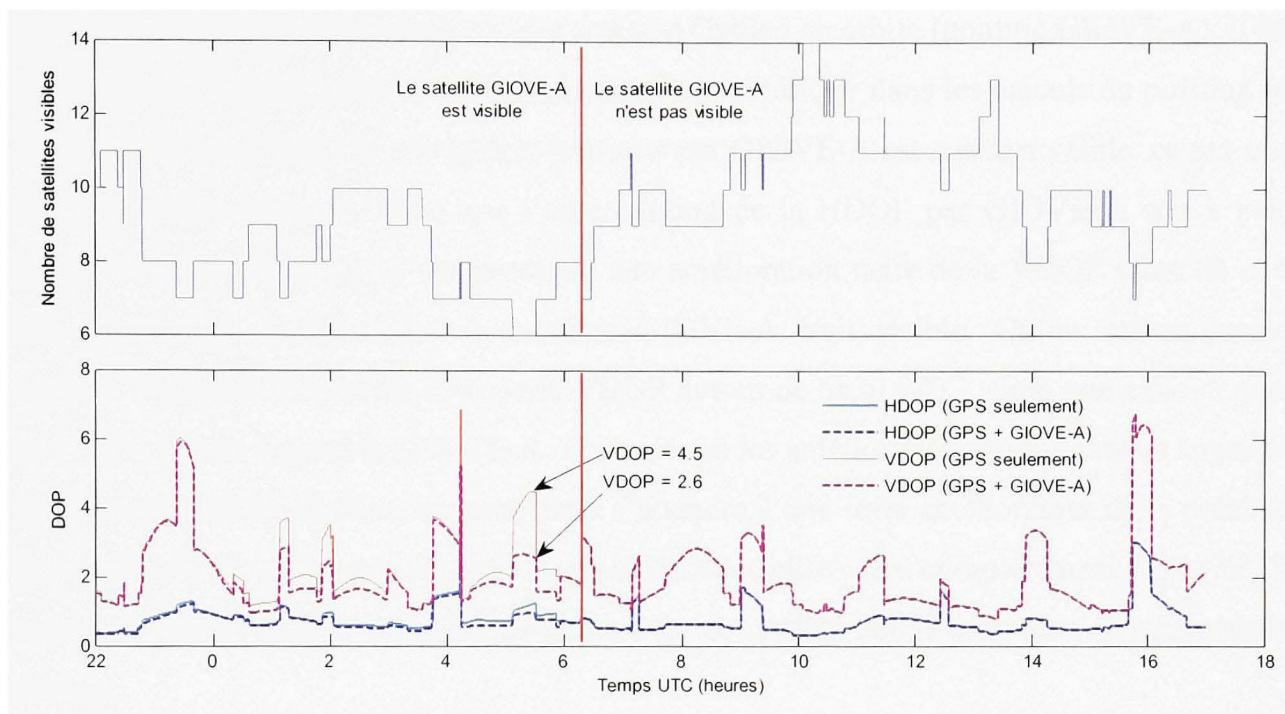
La section suivante reprend le concept de dilution de précision et l'applique de façon pratique dans un contexte réel. Ceci permet de comprendre plus facilement l'utilité des indices tels que la HDOP et la VDOP.

### 2.3.9 Évolution temporelle de la DOP

La DOP est intimement liée au nombre de satellites visibles à partir d'un emplacement ainsi qu'à leur position dans le ciel. Puisque les satellites sont en perpétuel mouvement, la DOP varie donc d'un instant à l'autre, se traduisant par des fluctuations de la précision de la solution de navigation dans le temps. La Figure 2.14 présente le nombre de satellites GPS et Galileo visibles au-dessus de l'ÉTS, à Montréal, ainsi que la DOP correspondante durant les journées du 28 au 29 juin 2007. La HDOP et la VDOP y sont présentées pour la constellation



GPS seule ainsi qu'en incluant le satellite Galileo GIOVE-A. Un masque d'élévation de  $5^\circ$  a été employé.



**Figure 2.14** *Nombre de satellites en vue et DOP correspondante à Montréal durant les 28 et 29 juin 2007.*

La Figure 2.14 montre que le nombre de satellites GPS et Galileo visibles à Montréal peut varier entre 6 et 14 au cours d'une même journée, ce qui est supérieur aux années 1990 alors qu'on observait rarement plus de 10 satellites au même moment (Misra et Enge, 2006). Ceci est en majeure partie dû au lancement récent de plusieurs nouveaux satellites GPS, ce qui pousse leur nombre à 31 en 2007 (USNO, 2007) comparativement à 25 pour 1997 (Misra et Enge, 2006). On note aussi que la HDOP demeure généralement autour de 1 alors que la VDOP se situe plutôt autour de 2. Cette différence entre les deux indices est normale et est due au fait que les satellites situés sous la ligne d'horizon ne sont pas visibles. Ceci a donc pour effet de réduire de moitié la géométrie verticale des satellites.

De façon générale, la VDOP est deux fois plus grande que la HDOP, mais ceci peut être encore pire aux latitudes élevées. Ceci s'explique par l'angle d'inclinaison de  $55^\circ$  à  $56^\circ$  des

orbites GPS et Galileo, résultant en une absence de ces satellites au zénith pour des latitudes supérieures à  $56^\circ$ .

Puisqu'il n'y a présentement qu'un seul satellite Galileo en orbite (nommé GIOVE-A), il faut généralement attendre plusieurs heures avant d'en bénéficier dans les calculs de position (on suppose que le message de navigation transmis par GIOVE-A est *a priori* valide, ce qui n'est pas le cas actuellement). Bien que l'amélioration de la HDOP par GIOVE-A soit à peine perceptible à la Figure 2.14, on remarque une amélioration nette de la VDOP entre 0h et 6h UTC, intervalle durant lequel le satellite GIOVE-A était visible. On va même jusqu'à mesurer une amélioration de 73% de la VDOP autour de 5h30 UTC, alors que celle-ci passe de 4.5 à 2.6 avec l'ajout de GIOVE-A. Étant donné les améliorations significatives apportées par un seul satellite Galileo, on peut donc s'attendre à une forte amélioration de la précision de la solution de navigation lorsque la constellation Galileo sera complètement déployée. Le lecteur est invité à consulter (Constantinescu A. et Landry, 2005) pour plus d'information à ce sujet.

### **2.3.10 Précision des mesures de vitesse**

Tel que vu précédemment, la vitesse d'un utilisateur est généralement déterminée à partir du décalage Doppler de la porteuse des signaux satellites. Or, le bruit thermique et les biais intercanaux sont habituellement les seules sources d'erreurs significatives au niveau de la porteuse, les autres sources, telles que les effets relativistes et les erreurs ionosphériques, étant négligeables. Par conséquent, une précision de 0,2 m/s est généralement atteignable à 95% du temps sur chacun des trois axes ( $x$ ,  $y$  et  $z$ ) en ce qui concerne la vitesse de déplacement de l'utilisateur (DoD, 1996).

### **2.3.11 Résumé des sources d'erreurs et précision du positionnement par satellite**

Un bilan de toutes les sources d'erreurs discutées précédemment (à l'exception de la SA) est présenté au Tableau 2.6. L'UERE, dont la valeur a été calculée grâce à l'équation (2.51), y

figure aussi, en plus d'une estimation des erreurs de positionnement pour les services standard simple fréquence et de précision (SPS/PPS) du GPS.

Tableau 2.6  
Budget d'erreur pour les mesures de  
pseudo-distances GPS sans SA

	PPS Code P	SPS Code C/A
Sources d'erreur	Erreur à $1\sigma$ (m)	
Segments spatial et de contrôle	2,6	2,6
Délais ionosphériques	2,3	5
Délais troposphériques	1	1
Multitrajets*	1,2	2,5
Segment utilisateur	1,6	1,6
UERE	4,1	6,5
Erreur horizontale (HDOP = 1)	4,1	6,5
Erreur verticale (VDOP = 2)	8,2	13,0

\* Valeur tirée de (Knight, 1998)

D'après le Tableau 2.6, l'UERE pour le service standard GPS (sans SA) est de 6,5 m ( $1\sigma$ ). Bien que cette valeur concorde avec ce que l'on retrouve généralement dans la littérature (DoD, 1996), il ne faut pas perdre de vue qu'il s'agit d'un estimé pessimiste basé sur des données théoriques. Des tests menés de façon intensive au-dessus des États-Unis révèlent un UERE beaucoup plus petit avec des valeurs à  $1\sigma$  de l'ordre de 1,6 m ainsi que des erreurs de positionnement à 95% du temps inférieures à 2,4 m horizontalement et 4,8 m verticalement (FAA, 2007). Des valeurs semblables se retrouvent dans les fiches techniques de certains fabricants de récepteurs GPS (Novatel, 2007b)

Toutes les données présentées dans cette section se réfèrent au système GPS actuellement en place, ceci pour la raison qu'il s'agit du système de navigation le mieux documenté et le plus fiable de l'heure. Des tests menés avec le système GLONASS révèlent toutefois des résultats similaires au système GPS (Cook, 1997). Pour ce qui est de Galileo, il est encore trop tôt pour pouvoir déterminer la précision du système. Vu les améliorations qui ont été apportées au GPS au fil des années, on est cependant en mesure de croire que Galileo sera encore plus précis que le GPS actuel.



Mentionnons pour conclure que le positionnement par satellite de type simple fréquence (tel que le service SPS du GPS) est loin d'être la seule solution présentement accessible aux utilisateurs civils. Il est en fait possible de se tourner vers une multitude d'autres techniques dans les cas où plus de précision serait nécessaire. L'utilisation de récepteurs bifréquences (pour éliminer les délais ionosphériques), des systèmes d'augmentation locaux ou étendus (tels que le WAAS) et du positionnement différentiel de type RTK (*Real-Time Kinematics*) ne sont que quelques exemples qui permettent de passer d'une précision de l'ordre du mètre à seulement quelques millimètres.

## 2.4 Conclusion

Dans ce chapitre, nous avons passé en revue les principes de base de la radionavigation par satellite. Nous avons tout d'abord vu la technique de la trilatération, la technique principalement employée afin de déterminer une position à partir des satellites. Pour y parvenir, nous avons aussi vu que des mesures de pseudo-distances devaient être effectuées, obligeant du coup à résoudre un système d'équations à quatre inconnues : position  $(x, y, z)$  et temps  $(b)$ . Le système de coordonnées géodésiques WGS-84 a ensuite été abordé, puis nous avons vu comment calculer la position et les corrections d'horloge des satellites à partir des éphémérides. Certaines subtilités ont aussi été abordées, telles que l'ondulation du géoïde et l'effet de la rotation de la Terre sur la solution de navigation. Enfin, nous avons vu les principales sources d'erreurs inhérentes au positionnement par satellite ainsi que quelques solutions afin de les réduire.

Les chapitres qui suivent présentent de façon détaillée le récepteur de navigation que nous avons développé. Tout d'abord, le chapitre 3 met l'accent sur la démodulation des signaux GPS, suivi du chapitre 4 avec la démodulation des signaux Galileo. Le chapitre 5 porte ensuite sur le traitement à haut niveau des signaux, incluant la gestion des canaux de démodulation et la solution de navigation.

## CHAPITRE 3

### DÉMODULATION EN TEMPS RÉEL DES SIGNAUX GPS L1 C/A

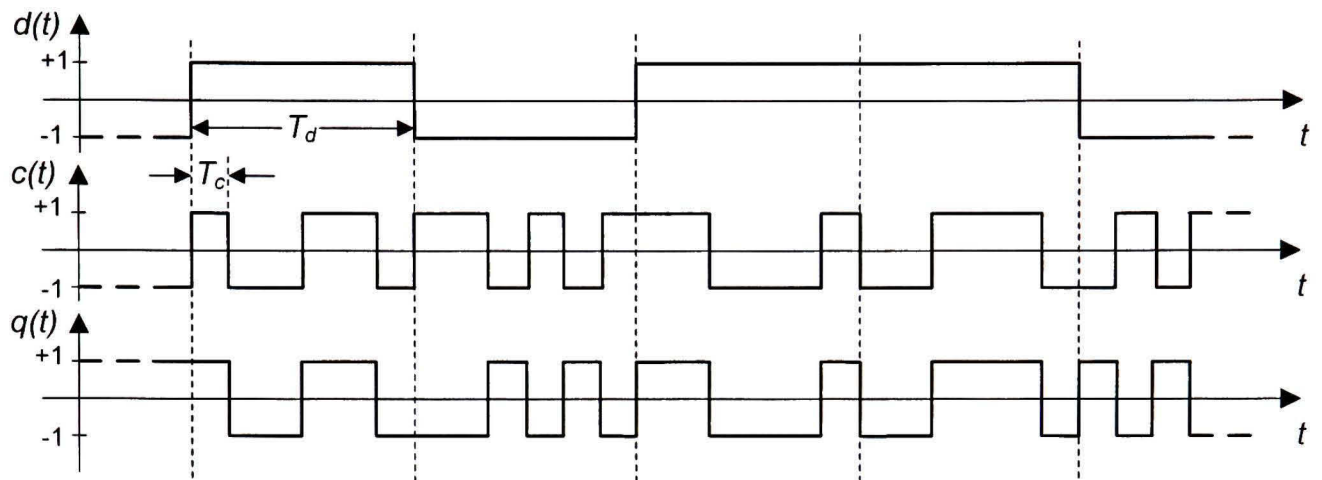
#### 3.1 Introduction au CDMA

##### 3.1.1 Principes de base du CDMA

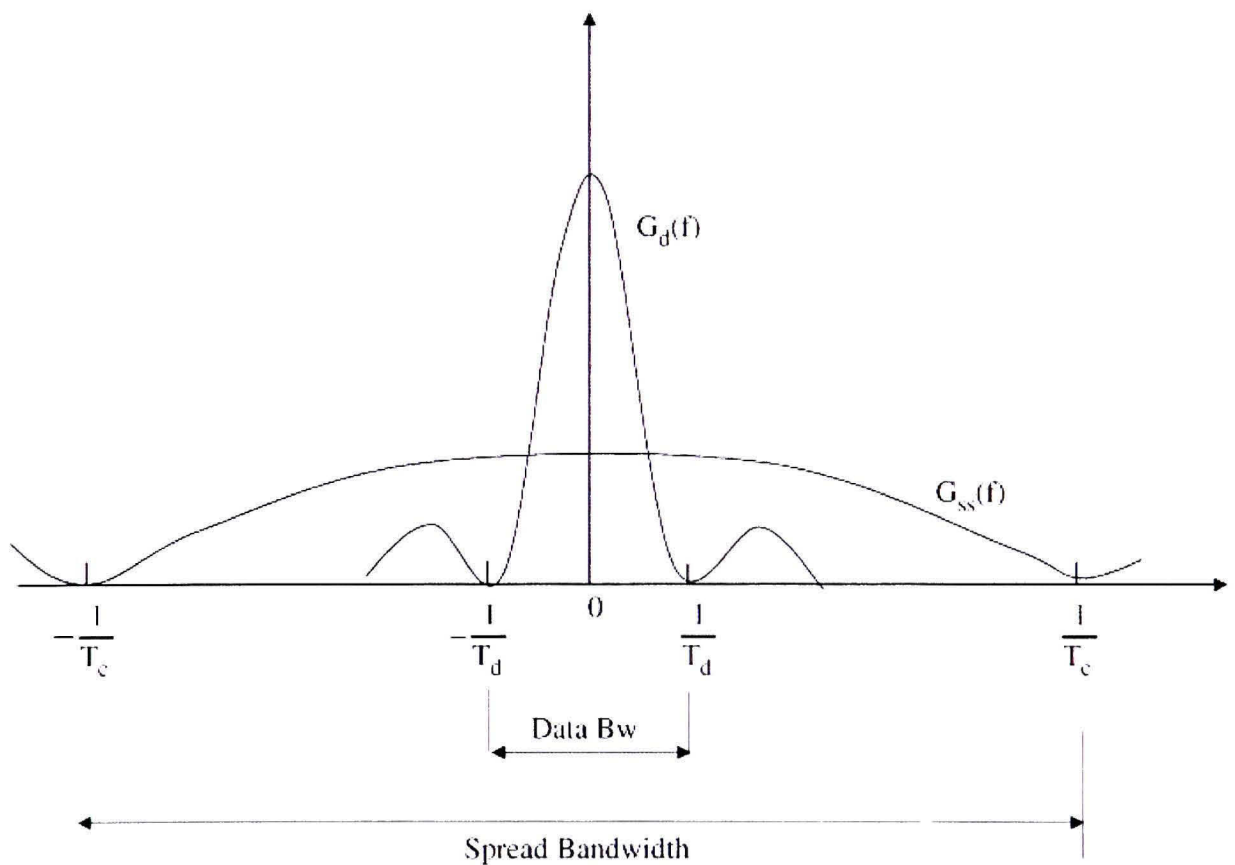
Une panoplie de méthodes existe à ce jour afin de multiplexer plusieurs signaux au sein d'un même canal de communication. La technique d'accès multiple la plus connue est sans doute le FDMA, qui consiste à subdiviser un canal en plusieurs bandes de fréquences distinctes. Il y a aussi le TDMA (*Time Division Multiple Access*), où la stratégie consiste à segmenter de façon temporelle l'accès à un canal de communication. Alors qu'il faille dans les deux cas fractionner le canal de communication, une autre méthode, appelée CDMA, permet à plusieurs émetteurs de communiquer en simultané sur les mêmes bandes de fréquences. Bien qu'il existe plusieurs formes de CDMA, seul le CDMA par étalement du spectre fréquentiel sera abordé dans cette section. Le lecteur est invité à lire (Gerakoulis et Geraniotis, 2001) pour plus d'information sur les autres formes de CDMA.

Deux choses sont requises afin de générer un signal CDMA par étalement spectral (noté seulement CDMA à partir d'ici) : un message contenant des données,  $d(t)$ , et un code d'étalement,  $c(t)$ , chacun variant en fonction du temps ( $t$ ). Les deux signaux sont formés d'une série de pulses carrés pouvant varier de façon aléatoire entre +1 et -1. Ces pulses sont respectivement appelés « bits » et « bribes » (ou *chips* en anglais) pour les signaux  $d(t)$  et  $c(t)$ . Il s'agit alors de multiplier dans le domaine temporel le message au code pour obtenir un signal CDMA par étalement spectral,  $q(t)$  (voir Figure 3.1), à condition cependant que la fréquence du code ( $1/T_c$ ) soit supérieure à celle des données ( $1/T_d$ ) :

$$q(t) = c(t) \cdot d(t) \quad (3.1)$$



**Figure 3.1** *Modulation d'un train de données par un code d'étalement.*



**Figure 3.2** *Densité spectrale de puissance d'un train de données avant et après étalement spectral (cas général).*

Tiré de (Gerakoulis et Geraniotis, 2001)



D'un point de vue fréquentiel, l'enveloppe du signal  $d(t)$  suit la forme d'un sinus cardinal, tel que vu à la Figure 3.2. Sa densité spectrale de puissance en fonction de la fréquence,  $G_d(f)$ , s'exprime ainsi (Gerakoulis et Geraniotis, 2001) :

$$G_d(f) = T_d \left( \frac{\sin(\pi f T_d)}{\pi f T_d} \right)^2 \quad (3.2)$$

où :

- $T_d$  est la période d'un bit de donnée du signal  $d(t)$

Une fois le signal modulé par le code d'étalement  $c(t)$ , on obtient alors la densité spectrale de puissance  $G_{ss}(f)$  suivante (Gerakoulis et Geraniotis, 2001) :

$$G_{ss}(f) = T_c \left( \frac{\sin(\pi f T_c)}{\pi f T_c} \right)^2 \quad (3.3)$$

où :

- $T_c$  est la période d'une bribe du code  $c(t)$

Il faut noter que le cas général est abordé ici. Dans le cas où le code d'étalement se répéterait à intervalles réguliers, ce qui est le cas pour les signaux GPS L1 C/A par exemple, nous aurions plutôt affaire à un spectre de raies tel qu'expliqué dans (Kaplan, 1996, p. 105).

Ainsi, le spectre en fréquence du signal  $d(t)$  se retrouve étalé suite à la modulation, tel que vu à la Figure 3.2, d'où l'appellation « modulation par étalement spectral ». L'opération exactement opposée se produit lors de la démodulation; on parle alors de « désétalement » spectral.

Afin de démoduler un signal CDMA, il s'agit de régénérer le bon code d'étalement au niveau du récepteur,  $c_R(t)$ , puis de le corrélérer avec le signal modulé  $q(t)$  :

$$d(t) = q(t) \star c_R(t) \equiv \frac{1}{T_d} \int_{-T_d/2}^{T_d/2} q(t) c_R(t+\tau) d\tau \quad (3.4)$$

où :

- «  $\star$  » est l'opération de corrélation utilisée lors du désétalement d'un signal CDMA

Si le code d'étalement régénéré est une version identique et parfaitement synchronisée au code de l'émetteur, alors  $c_R(t) = c(t)$  et on obtient :

$$q(t) \star c_R(t) = \frac{1}{T_d} \int_{-T_d/2}^{T_d/2} d(t) c(t) c(t+\tau) d\tau \quad (3.5)$$

Si le signal  $d(t)$  demeure constant tout au long de chaque période d'intégration, l'équation (3.5) se simplifie alors telle que :

$$q(t) \star c_R(t) = d(t) \frac{1}{T_d} \int_{-T_d/2}^{T_d/2} c(t) c(t+\tau) d\tau = d(t) \quad (3.6)$$

Puisqu'ici l'amplitude du code d'étalement est unitaire, son autocorrélation est donc aussi unitaire. On retrouve alors la même équation qu'en (3.4), ce qui prouve que le signal est parfaitement démodulé, toujours à condition que le signal  $d(t)$  demeure constant durant les périodes d'intégration. Pour y parvenir, on s'arrange habituellement pour que les transitions des bits de données aient lieu à la fin des périodes d'intégration. Ceci nécessite donc que le code et le message soient parfaitement synchrones et que la fréquence du code soit un multiple entier de la fréquence du message, telle que :

$$\frac{f_c}{f_d} = \frac{T_d}{T_c} = N_s \quad (3.7)$$

où :

- $f_c$  et  $f_d$  sont les fréquences du code et des données respectivement
- $N_S$  est le facteur d'étalement, soit un entier pouvant être égal à 2, 3, 4, ...

Si les signaux provenant de tous les émetteurs CDMA sont parfaitement synchronisés, il est alors possible d'employer des codes orthogonaux. Ces codes ont la propriété de ne pas interférer entre eux, c'est-à-dire que leur corrélation croisée (ou cross-corrélation) est nulle. Ainsi, plusieurs signaux peuvent être transmis simultanément sans qu'ils n'interfèrent entre eux, tels que :

$$n(t) = q_1(t) + q_2(t) + \dots + q_N(t) \quad (3.8)$$

où :

- $q_i(t)$  représente les signaux CDMA pour les canaux  $i = 1, 2, \dots, N$
- $n(t)$  est la somme de tous les signaux CDMA

Il est alors possible de parfaitement démoduler un signal  $d_k(t)$  parmi tous les autres signaux puisque :

$$n(t) \star c_k(t) = \frac{d_1(t)}{T_d} \int_{-T_d/2}^{T_d/2} c_1(t) c_k(t + \tau) d\tau + \dots + \frac{d_k(t)}{T_d} \int_{-T_d/2}^{T_d/2} c_k(t) c_k(t + \tau) d\tau + \dots + \frac{d_n(t)}{T_d} \int_{-T_d/2}^{T_d/2} c_n(t) c_k(t + \tau) d\tau \quad (3.9)$$

$$n(t) \star c_k(t) = d_1(t) \cdot 0 + \dots + d_k(t) \cdot 1 + \dots + d_n(t) \cdot 0 \quad (3.10)$$

$$\therefore n(t) \star c_k(t) = d_k(t), \text{ ce qu'il fallait démontrer (C.Q.F.D)} \quad (3.11)$$

où :

- $d_i(t)$  représente les données des canaux  $i = 1, 2, \dots, k, \dots, N$
- $c_i(t)$  représente les codes d'étalement des canaux  $i = 1, 2, \dots, k, \dots, N$

Les codes orthogonaux ne sont cependant pas employés pour la navigation par satellite puisque les signaux ne sont jamais parfaitement synchronisés (ne serait-ce qu'à cause des



délais de propagation des signaux qui varient constamment entre les satellites et le récepteur). Pour cette raison, les codes d'étalement sont plutôt basés sur des séquences pseudo-aléatoires (PRN – *Pseudo Random Noise*). Les codes PRN sont constitués d'une série de bribes alternant de façon aléatoire et se répétant après un certain intervalle de temps. On dit de ces codes PRN qu'ils sont pseudo-orthogonaux puisqu'un certain bruit d'intercorrélation subsiste entre les canaux (Gerakoulis et Geraniotis, 2001). On emploie donc des codes d'étalement suffisamment longs pour minimiser cet effet. Pour cette raison, les codes PRN sont souvent produits à l'aide de générateurs polynomiaux, une méthode simple et compacte qui sera abordée plus loin dans ce chapitre. Un très grand nombre de codes différents peut être produit par ces générateurs, desquels on ne retient généralement que ceux qui offrent les plus faibles niveaux d'intercorrélation.

Avant d'être transmis sur les ondes, les signaux CDMA sont généralement modulés par une porteuse sinusoïdale. Les deux types de modulations les plus couramment employées sont alors le BPSK et le FSK (*Frequency Shift Keying*). La modulation BPSK est généralement préférée car plus facile à implémenter. Une simple multiplication du signal CDMA par la porteuse est alors requise, d'où le nom de CDMA à étalement spectral par séquence directe (DS-CDMA – *Direct Sequence CDMA*). La modulation FSK produit quant à elle un signal à étalement spectral par sauts de fréquence (FH-CDMA – *Frequency Hopping CDMA*). C'est le DS-CDMA qui est employé pour GPS et Galileo, il sera donc abordé plus en profondeur dans ce document. Le lecteur est toutefois invité à consulter (Goldsmith, 2005) pour plus de détails sur le FH-CDMA.

### 3.1.2 Robustesse du CDMA face au bruit et aux brouilleurs

De par leur nature, les signaux CDMA offrent une certaine robustesse face aux bruits, interférences ou brouilleurs ambiants. De façon générale, on suppose que le bruit dans un canal de communication est blanc, à moyenne nulle et de distribution gaussienne. On assume aussi que le bruit est non corrélé aux signaux transmis. Par conséquent, le spectre fréquentiel  $G_{ss}(f)$  des signaux peut être « désétalé » indépendamment du bruit lors de la démodulation,

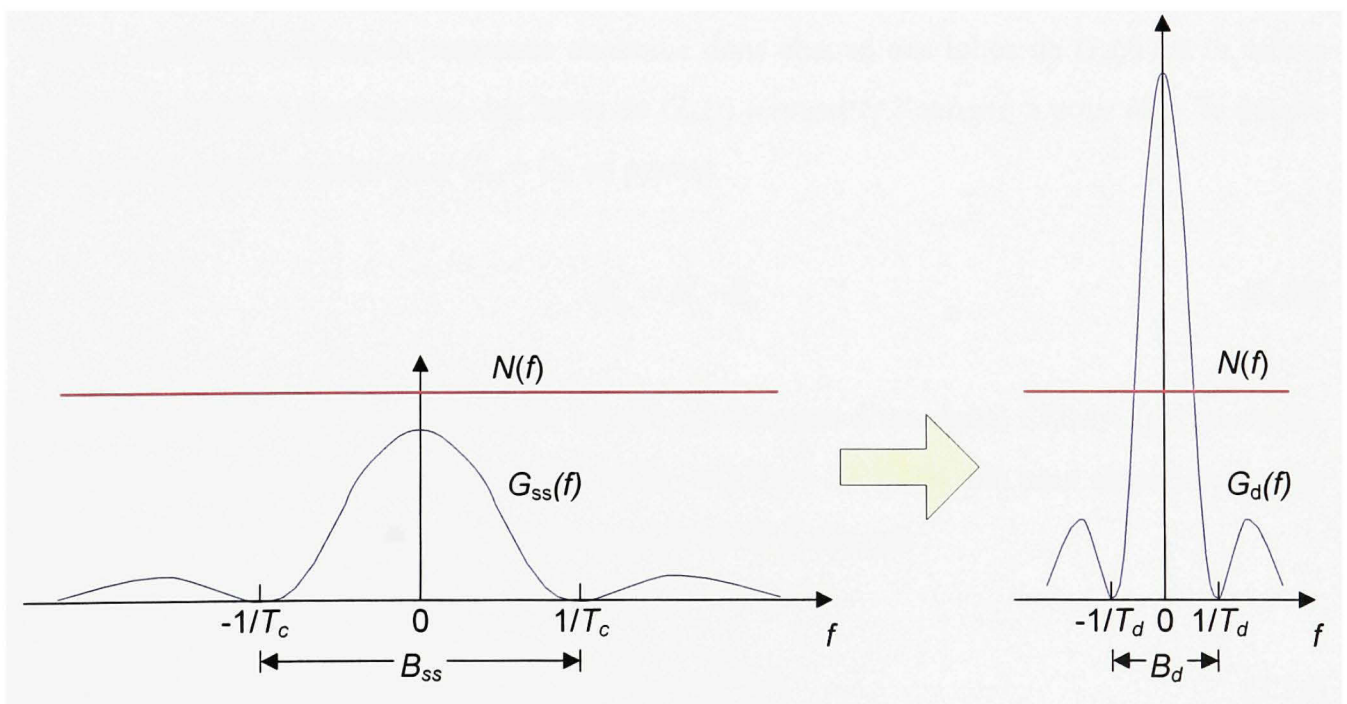
tel qu'illustré à la Figure 3.3. Étant donné que le spectre fréquentiel des signaux,  $G_d(f)$ , se concentre lors de la démodulation, et que celui du bruit,  $N(f)$ , demeure le même, on constate donc un gain au niveau du SNR. Les équations qui suivent montrent comment calculer ce gain appelé « gain d'étalement spectral ».

Afin de déterminer le gain de démodulation, il s'agit tout d'abord de déterminer la puissance du bruit ( $N$ ). Puisque le bruit est blanc, sa densité spectrale demeure donc constante sur toute la bande occupée, ce qui se traduit par :

$$N = N_0 B \quad (3.12)$$

où :

- $B$  est la largeur de bande du bruit (unités : Hz)
- $N_0$  est la densité spectrale du bruit (unités : W/Hz)



**Figure 3.3** *Démodulation d'un signal CDMA transmis dans un canal bruité.*

La puissance d'un signal CDMA ( $C_{ss}$ ) est quant à elle obtenue en calculant l'intégrale de l'équation (3.3) (le cas général est abordé ici) :

$$C_{ss} = \int_{-B_{ss}/2}^{B_{ss}/2} G_{ss}(f) df = T_c \int_{-B_{ss}/2}^{B_{ss}/2} \left( \frac{\sin(\pi f T_c)}{\pi f T_c} \right)^2 df \quad (3.13)$$

Il en va de même pour la puissance d'un signal de données NRZ ( $C_d$ ) avant modulation, dont le spectre fréquentiel a été présenté à l'équation (3.2) :

$$C_d = \int_{-B_d/2}^{B_d/2} G_d(f) df = T_d \int_{-B_d/2}^{B_d/2} \left( \frac{\sin(\pi f T_d)}{\pi f T_d} \right)^2 df \quad (3.14)$$

$B_{ss}$  représente la largeur de bande unilatérale du signal étalé alors que  $B_d$  représente la largeur de bande bilatérale du signal démodulé. Or, de telles équations ne possèdent pas de solution simple. On sait toutefois que la puissance des signaux avant et après modulation demeure la même; en d'autres mots, la puissance contenue dans chacun des lobes de  $G_d(f)$  est la même que celle contenue dans chacun des lobes de  $G_{ss}(f)$  (consulter l'annexe 6 pour plus de détails à ce sujet). Ainsi, on peut fixer  $C_{ss} = C_d$  en posant :

$$B_{ss} = N_s \cdot B_d \quad (3.15)$$

On se rappelle ici que  $N_s$  représente le facteur d'étalement d'un signal CDMA (par exemple, pour un signal GPS L1 C/A,  $N_s = 1,023 \text{ MHz} / 50 \text{ Hz} = 43,1 \text{ dB}$ ). On peut ainsi calculer les SNR des signaux modulés ( $SNR_{ss}$ ) et démodulés ( $SNR_d$ ) tels que :

$$SNR_{ss} = \frac{C_{ss}}{N_0 B_{ss}} \quad (3.16)$$

$$SNR_d = \frac{C_d}{N_0 B_d} = \frac{N_s C_{ss}}{N_0 B_{ss}} \quad (3.17)$$

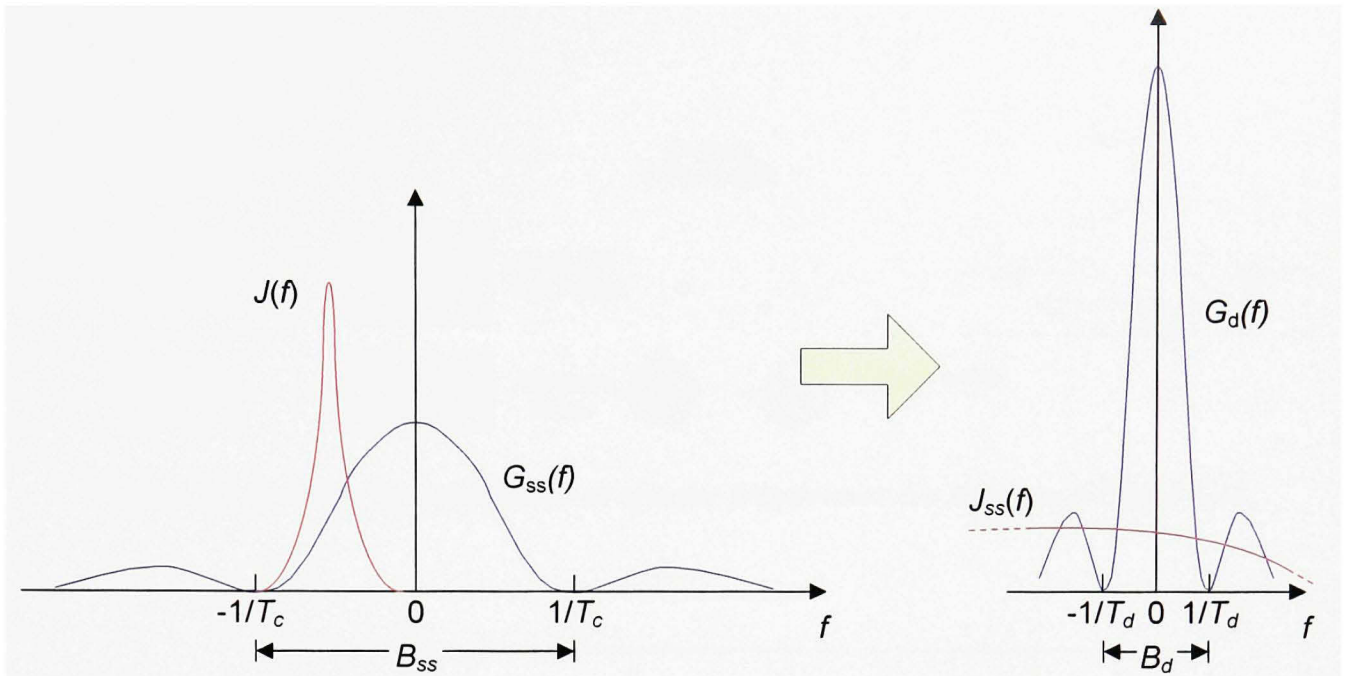


Enfin, le gain en SNR réalisé lors de la démodulation ( $G_{CDMA}$ ) peut être déterminé :

$$G_{CDMA} = \frac{SNR_d}{SNR_{ss}} = \frac{N_s C_{ss}}{N_0 B_{ss}} \bigg/ \frac{C_{ss}}{N_0 B_{ss}} = N_s \quad (3.18)$$

Ainsi, un gain en SNR proportionnel au facteur d'étalement du spectre est réalisé lors de la démodulation des signaux CDMA. Cette propriété est exploitée au niveau du GPS et de Galileo et permet de démoduler les signaux reçus même si ces derniers ont une puissance inférieure à celle du bruit. Cet aspect est abordé à la section suivante.

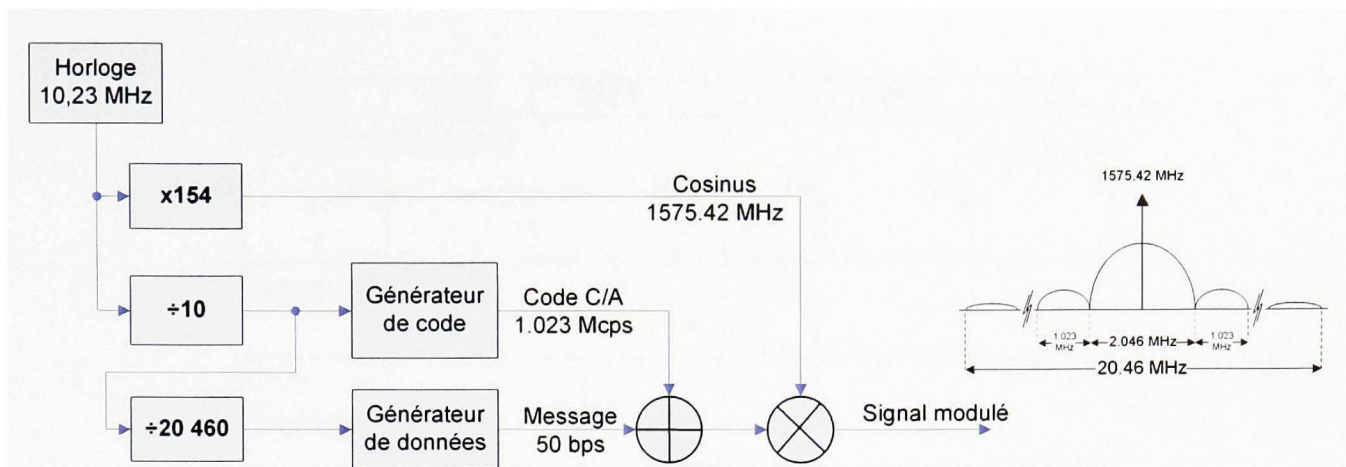
Les signaux CDMA offrent aussi une certaine robustesse contre les interférences, aussi appelées « brouilleurs » par abus de langage (ou *jammer*, en anglais - pour plus de précision, notons que les brouilleurs sont généralement volontaires alors que les interférences sont plutôt non intentionnelles). Puisque les signaux sont à nouveau multipliés par le code d'étalement lors de la démodulation, tout brouilleur introduit dans le canal de communication se retrouve automatiquement étalé en fréquence. Ceci est illustré à la Figure 3.4, où  $J(f)$  représente la courbe de densité spectrale d'un brouilleur et  $J_{ss}(f)$  la même courbe après étalement spectral. Ceci se traduit donc par une amélioration du ratio brouilleur à signal (JSR – *Jammer to Signal Ratio*) telle qu'observée pour le SNR. Le lecteur est invité à consulter (Proakis, 2001) pour plus d'informations sur le sujet.



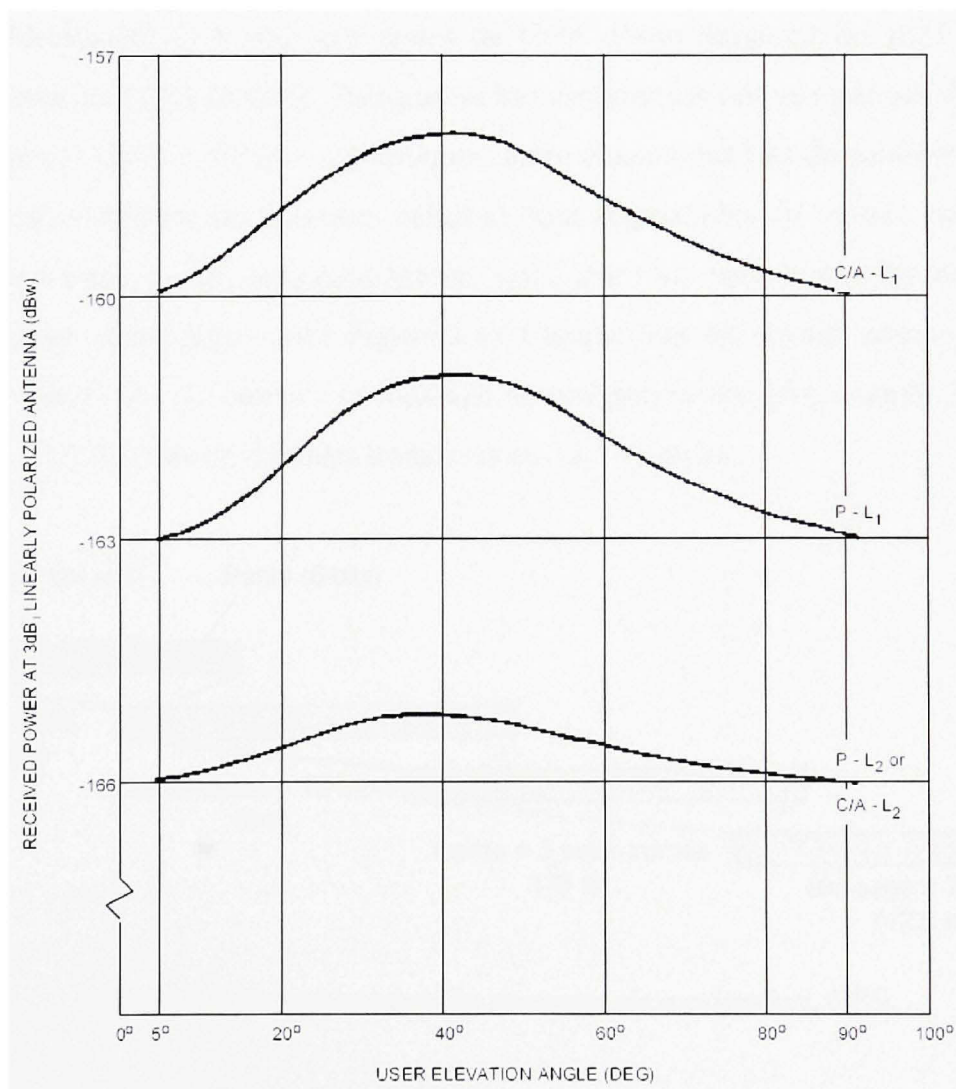
**Figure 3.4** *Étalement d'un brouilleur lors de la démodulation d'un signal CDMA.*

### 3.2 Structure des signaux GPS L1 C/A

Les signaux GPS L1 C/A sont basés sur une modulation de type DS-CDMA, soit une combinaison des techniques CDMA pour l'accès multiple et BPSK pour la modulation de la porteuse. Une horloge dont la fréquence est de 10,23 MHz (soit 10,22999999543 MHz avant l'ajout des effets relativistes) est employée à bord des satellites GPS de manière à cadencer les données et le code d'étalement. Les rythmes respectifs de ces derniers sont de 50 bps et de 1,023 Mcps. Les bits de données sont ainsi combinés au code d'étalement (à l'aide d'une somme modulo-2) avant d'être multipliés par une porteuse dont la fréquence nominale est de 1575,42 MHz, tel que vu à la Figure 3.5. Le signal BPSK ainsi formé est finalement amplifié à une puissance minimale de 33 W (Kaplan, 1996, p. 99) et limité à une largeur de bande de 20,46 MHz avant d'être transmis vers le sol. Vu l'éloignement des satellites par rapport aux utilisateurs, les signaux GPS n'ont plus qu'une puissance d'environ -160 dBW lors de leur réception au sol, tel qu'affiché à la Figure 3.6 (ICD-GPS-200C, 2000). Combiné au bruit thermique ambiant, ceci correspond à un SNR de l'ordre de -30 dB.



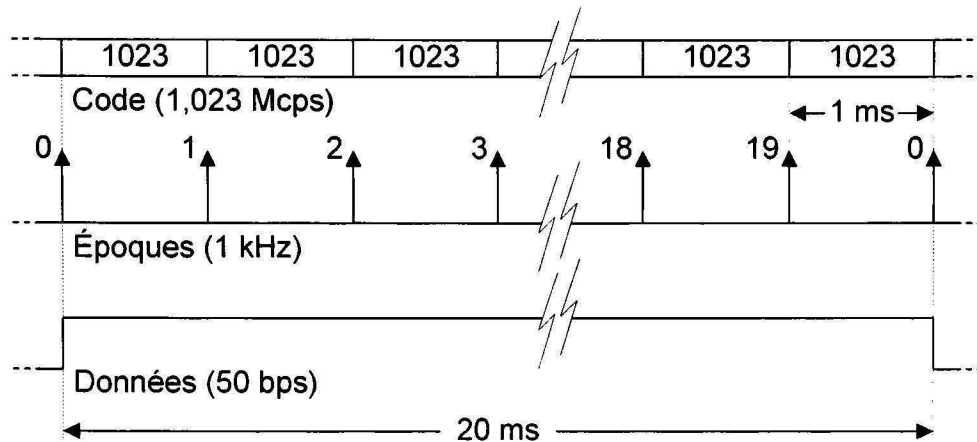
**Figure 3.5 Schéma de modulation et spectre fréquentiel des signaux GPS L1 C/A.**



**Figure 3.6 Puissance minimale en ciel dégagé des signaux GPS lors de leur réception au sol.**

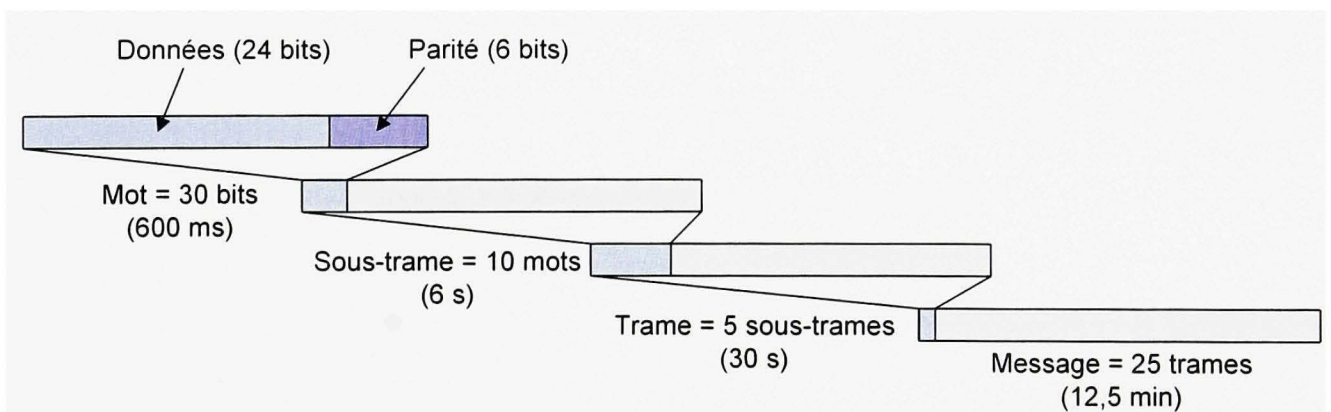
Figure tirée de (ICD-GPS-200C, 2000)





**Figure 3.7** Relation de temps entre le code C/A et les données GPS.

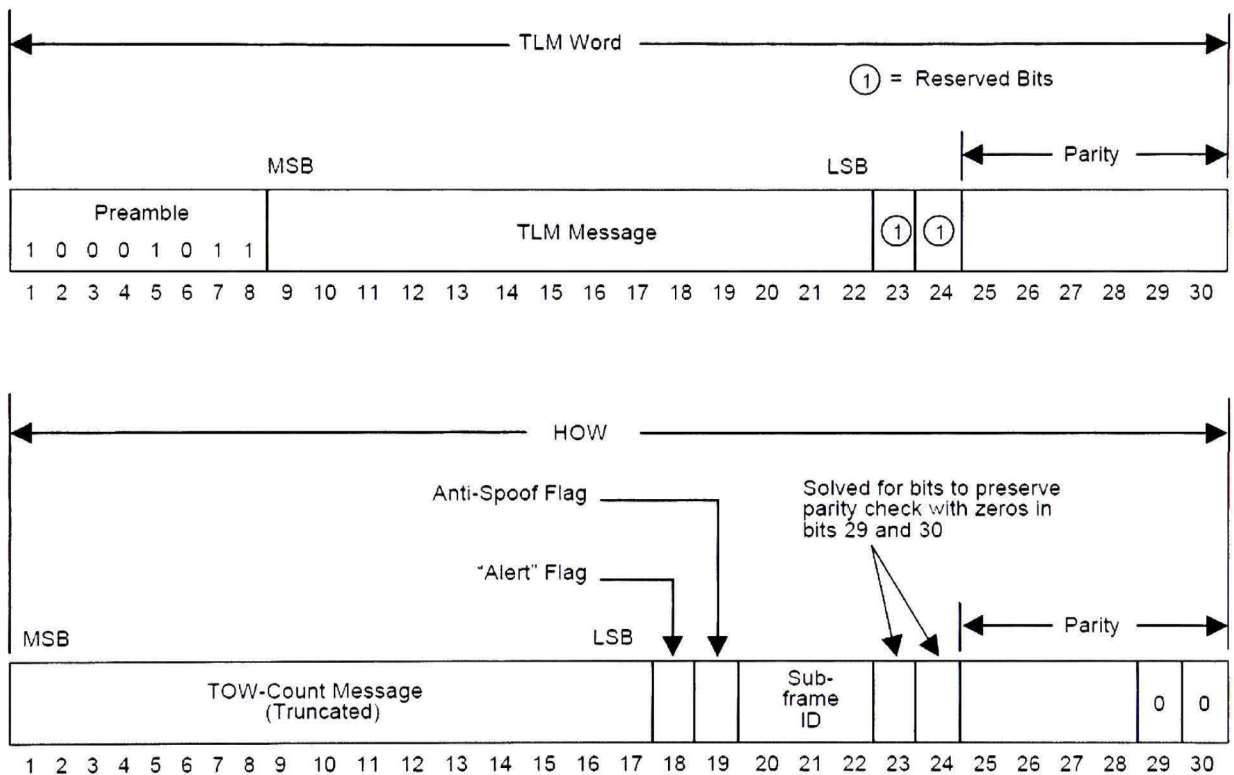
Les codes d'étalement C/A sont des codes de Gold d'une longueur de 1023 bribes et se répètent à toutes les millisecondes. Puisque les bits de données ont une période de 20 ms, il y a donc 20 répétitions du code C/A, ou époques, entre chacun des bits de données (voir Figure 3.7). En ce qui concerne les données, celles-ci sont segmentées en trames, sous-trames et mots : chaque trame contient cinq sous-trames, qui à leur tour contiennent dix mots de 30 bits (soit 300 bits par sous-trame – voir Figure 3.8). Chaque mot est ensuite composé de 24 bits de données et de 6 bits de parité. Un message de navigation complet compte 25 trames, ce qui représente 37 500 bits de données transférés en 12,5 minutes.



**Figure 3.8** Segmentation du message de navigation GPS.

Chacune des sous-trames débute systématiquement par le mot de télémétrie (*TLM Word*), suivi du HOW (*Hand-Over Word*), illustrés à la Figure 3.9. Le mot de télémétrie contient un

préambule, utilisé pour se synchroniser au message de navigation lors de l'acquisition des signaux. Le mot TLM renferme aussi un message spécial réservé aux utilisateurs autorisés seulement, son contenu n'est donc pas accessible aux civils. Le HOW contient quant à lui le numéro de la sous-trame en cours de téléchargement ainsi que le temps GPS au moment de la transmission des signaux. Le temps GPS, essentiel pour calculer la position du satellite et pour déterminer les mesures de pseudo-distances, est fourni via le *TOW-Count* (*Time of Week Count*) transmis dans le HOW.



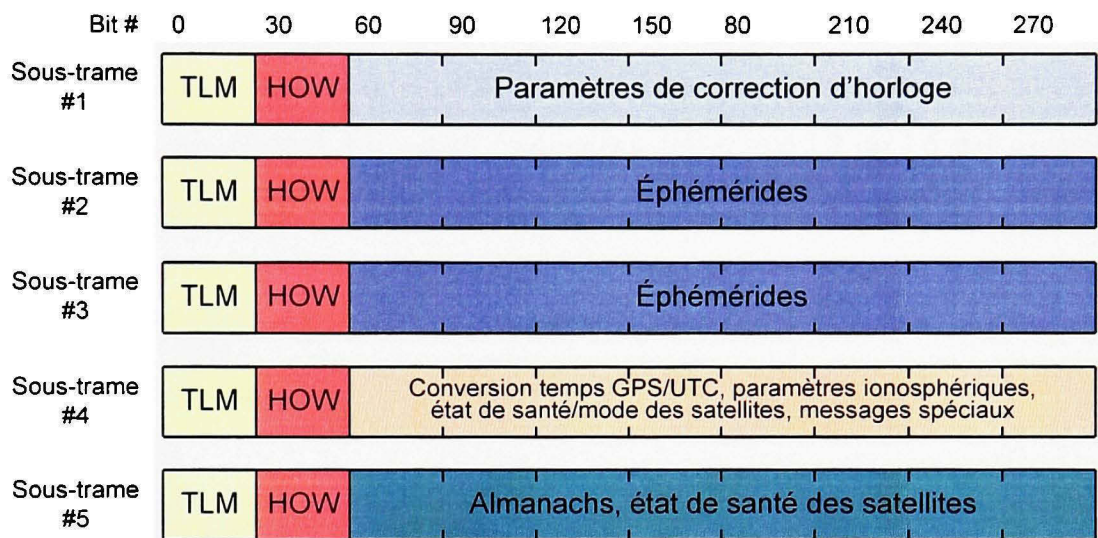
**Figure 3.9 Format des mots TLM et HOW.**

Tiré de (ICD-GPS-200C, 2000)

Deux drapeaux sont aussi transmis dans le HOW : le drapeau AS (*Anti-Spoof*), qui indique si le code Y est employé à la place du code P (i.e. qu'il y a cryptage), et le drapeau d'alerte (*Alert Flag*), qui averti d'une dysfonction du satellite émettant.

Les huit autres mots de chaque sous-trame contiennent à leur tour de l'information relative aux satellites, telle que les éphémérides et les paramètres de correction d'horloge (voir Figure

3.10). Plusieurs autres données y sont aussi transmises, dont l'almanach, l'état de santé des satellites et certains messages dédiés aux utilisateurs autorisés. Le lecteur est invité à se référer à (ICD-GPS-200C, 2000) pour plus de détails sur la structure du message de navigation GPS.



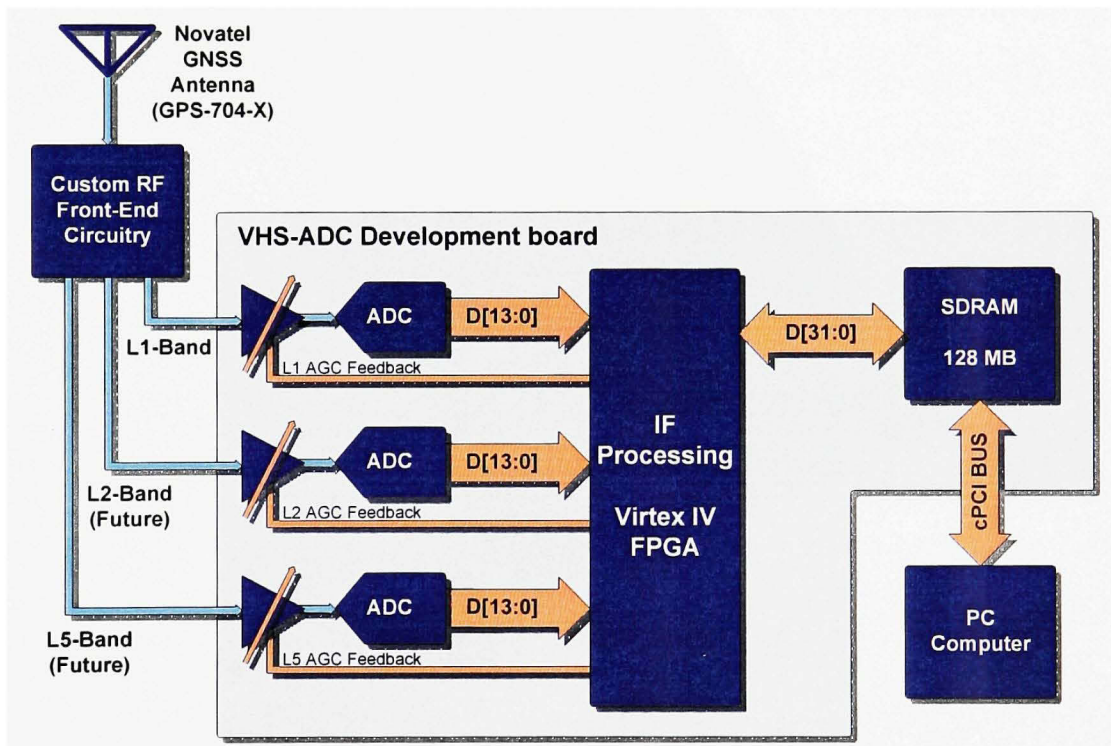
**Figure 3.10** *Structure des sous-trames du message de navigation GPS.*

### 3.3 Architecture générale du récepteur GPS L1 C/A développé

Le développement du récepteur hybride GPS/Galileo sur lequel ce document porte a débuté par la conception d'un récepteur GPS L1 C/A seul. Des canaux de démodulation Galileo ont pu être greffés par la suite, une fois le récepteur GPS réalisé et testé en temps réel. Deux raisons ont motivé une telle approche, la première étant que le document d'interface (ICD, *Interface Control Document*) du tout premier satellite Galileo, GIOVE-A, n'était pas disponible au moment de la conception du récepteur GPS. La seconde raison était de profiter de l'abondance de documentation sur le GPS afin de se forger une expérience en navigation par satellite et ainsi débiter la mise en place d'une plateforme pour le développement de récepteurs GNSS. L'approche s'est avérée somme toute efficace puisque le développement préliminaire du récepteur GPS a nécessité environ 11 mois, soit de janvier à décembre 2006, comparativement à deux semaines en 2008 pour le récepteur Galileo. Le récepteur Galileo est traité plus en détail au chapitre 4.



Le récepteur a été développé en trois segments principaux tels que décrits au chapitre 1, à savoir une tête RF analogique, suivie d'un étage de démodulation IF numérique et d'un processeur en bande de base. La démodulation étant réalisée de façon numérique au sein d'un FPGA, on parle donc d'un récepteur SDR. Les signaux reçus d'une antenne sont tout d'abord conditionnés de façon analogique par la tête RF, puis numérisés à l'aide d'un CAN (ou ADC en anglais – *Analog to Digital Converter*) avant d'être traités par le FPGA. Il faut noter que la tête RF en question a été conçue pour opérer à la fréquence L1 mais qu'elle peut s'adapter facilement afin de fonctionner à d'autres fréquences, telles que L2 ou L5 par exemple (voir Figure 3.11). Plus de détails sont apportés à la section suivante au sujet de la tête RF.



**Figure 3.11** *Architecture générale du récepteur GPS L1 C/A développé.*  
Tiré de (Sauriol et Landry, 2007)

Les CAN et le FPGA sont incorporés sur une même carte de développement, soit la VHS-ADC de Lyrtech (voir Figure 3.12). Notre choix s'est arrêté sur cette carte puisqu'elle était disponible au laboratoire dès le début du projet. La carte comprend entre autres (Lyrtech, 2007) :

- Un FPGA Virtex-4 SX-55 -10 de Xilinx
- 8 CAN de 14 bits de précision (signés) et de fréquence maximale de 105 MHz
- 8 amplificateurs analogiques de 17 dB à gain variable
- 128 MB de mémoire embarquée SDRAM (*Synchronous Dynamic Random Access Memory*)
- Un bus Compact PCI permettant de transférer des données du FPGA à un ordinateur de type PC (*Personal Computer*) à un taux de 20 MB/s



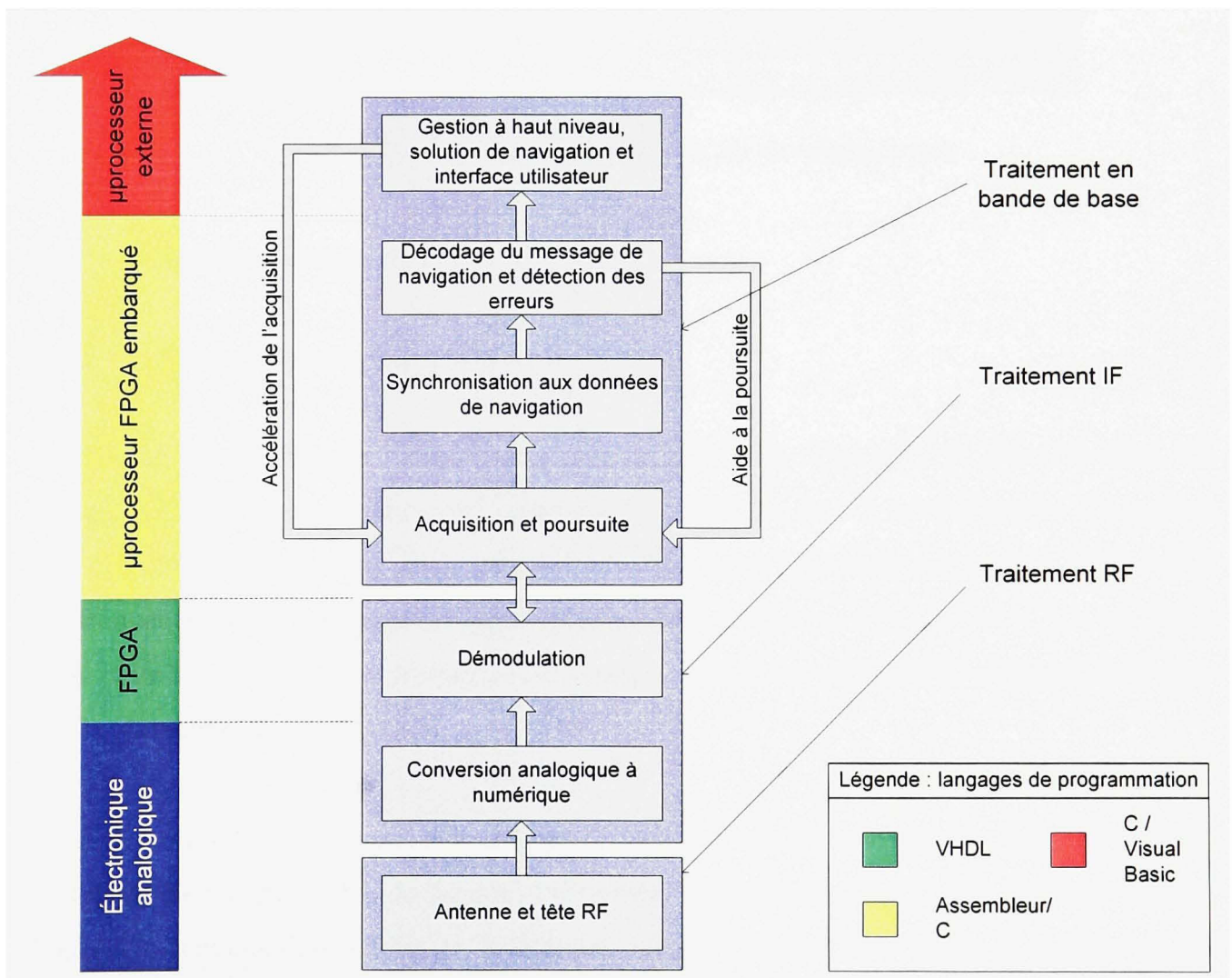
**Figure 3.12 Photographie de la carte de développement.**

Tel que mentionné précédemment, le rôle du FPGA est avant tout de démoduler les signaux GPS numérisés en IF. Ceci est effectué en temps réel à l'aide de 12 canaux en parallèle. Un microcontrôleur a aussi été implémenté au sein du FPGA afin de permettre l'acquisition et la poursuite des signaux en temps réel, facilitant du coup le décodage des messages de navigation, en plus des mesures de pseudo-distances, de fréquences Doppler et d'ADR (*Accumulated Doppler Range*). Le tout est sauvegardé dans la mémoire embarquée (SDRAM), qui peut être simultanément accédée par un ordinateur de type PC (voir Figure



3.14) afin de gérer les canaux de démodulation et calculer la solution de navigation en temps réel (ces deux sujets sont abordés plus en détail au chapitre 5). Le segment IF est ainsi entièrement réalisé au sein du FPGA, alors que le traitement en bande de base est en partie effectué par un microcontrôleur embarqué et par ordinateur PC (voir Figure 3.13). Les langages de programmation pour chacune de ces parties sont :

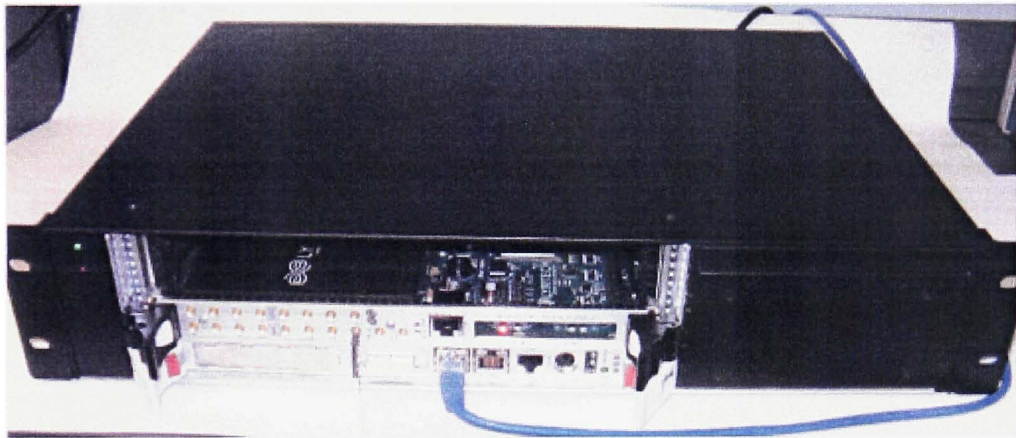
- FPGA : Programmation en langage VHDL
- Microcontrôleur embarqué : Programmation en langage assembleur et C
- Processeur externe : Programmation en langage C et Visual Basic



**Figure 3.13** *Diagramme en couches du récepteur de navigation développé.*



Les sections qui suivent présentent en détail les segments RF et IF ainsi qu'une partie du segment en bande de base. Il faut noter qu'une description synthétisée de l'architecture du récepteur peut aussi être retrouvée dans (Sauriol et Landry, 2007).



**Figure 3.14 Photographie du PC de développement.**

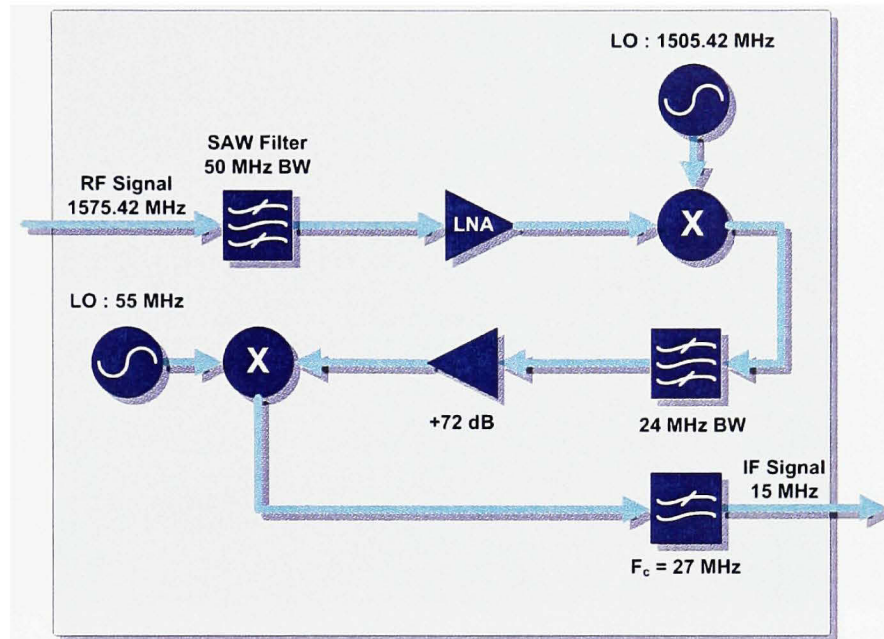
### **3.4 Développement de la tête RF**

#### **3.4.1 Conception et réalisation de la tête RF**

La tête RF est un élément indispensable afin de traiter les signaux GPS. Dans les faits, les signaux GPS ont une puissance de l'ordre de -160 dBW, ce qui est beaucoup trop faible pour être détecté par des CAN à usage général, tels que ceux disponibles sur la carte VHS-ADC. La fréquence élevée des signaux pose aussi problème pour ce type de CAN. La tête RF a donc pour but d'abaisser en fréquence les signaux GPS et de les amplifier suffisamment pour qu'ils puissent être numérisés.

Plusieurs têtes RF GPS sont aujourd'hui disponibles sur le marché. Ces têtes RF offrent toutefois une largeur de bande limitée et n'opèrent qu'à la fréquence L1 (Won, Pany et Hein, 2006). Nous avons donc choisi de développer notre propre tête RF à l'aide de composantes à faible coût facilement disponibles (COTS – *Components-Off-The-Shelf*). Un schéma sommaire de la tête RF est présenté à la Figure 3.15 (Sauriol et Landry, 2007) (voir

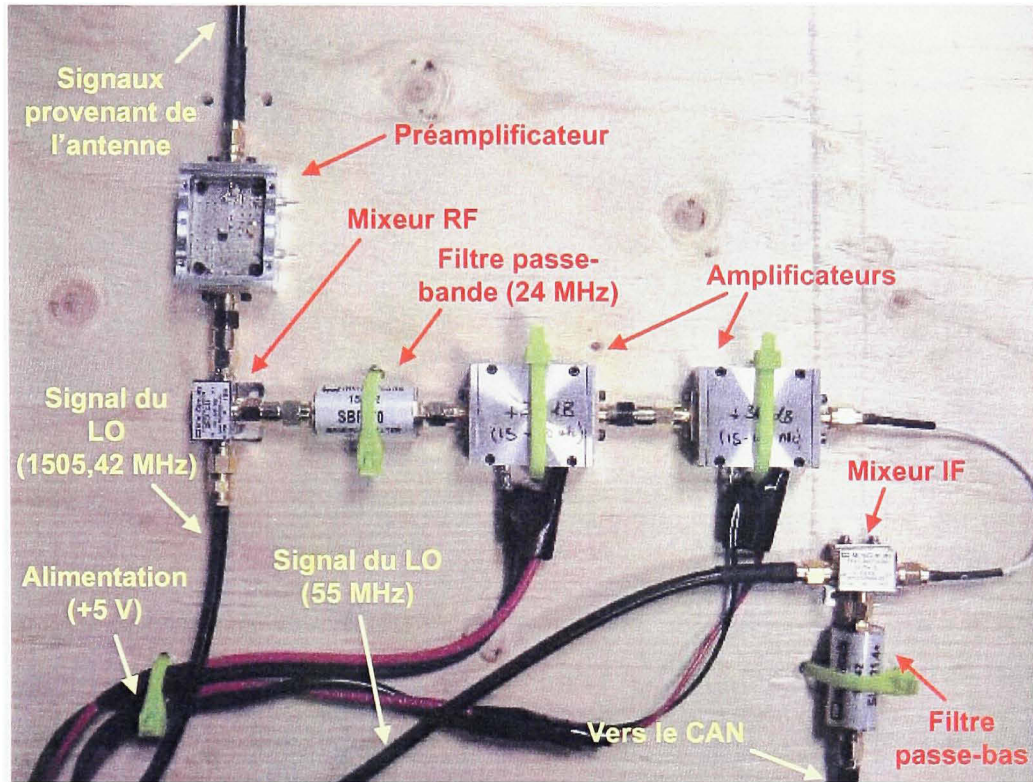
l'annexe 7 pour un schéma détaillé). Une photographie de la version préliminaire de la tête RF est ensuite présentée à la Figure 3.16.



**Figure 3.15 Schéma de la tête RF développée.**  
Tiré de (Sauriol et Landry, 2007)

La tête RF est principalement constituée de deux étages de conversion en fréquence. Un premier étage sert à abaisser la fréquence du signal de 1575,42 MHz à 70 MHz, puis un second de 70 MHz à 15 MHz. Le choix de ces fréquences repose principalement sur la facilité de trouver des composants électroniques à ces mêmes fréquences. La raison qui a motivé l'emploi de deux étages de conversion est que les filtres passe-bande RF bon marché possèdent une trop grande largeur de bande et qu'ils ne coupent pas de façon suffisamment abrupte à l'extérieur de cette même bande. La solution la plus simple demeurerait donc d'ajouter un second filtre passe-bande beaucoup plus sélectif en IF. Ce filtre, centré sur 70 MHz et d'une largeur de bande de 24 MHz, empêche ainsi qu'un phénomène de repliement spectral ne se produise lors de la conversion de RF à IF. Or, le choix d'une bande passante de 24 MHz peut sembler curieuse à priori puisque la largeur de bande des signaux GPS L1 C/A n'est que de 20,46 MHz. Le travail de conception a toutefois été fait dans l'intention d'inclure les signaux Galileo E1, dont la largeur de bande est justement de 24 MHz (ESA, 2006).

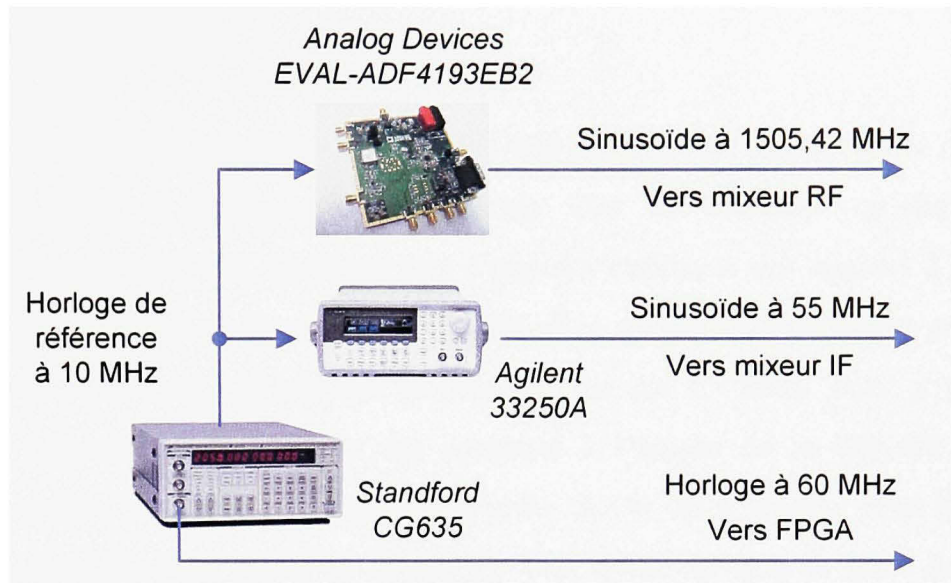




**Figure 3.16 Photographie de la version préliminaire de la tête RF.**

Un troisième et dernier filtre est utilisé de manière à rejeter les composantes à haute fréquence produites lors de la seconde conversion en fréquence. Ce dernier agit par la même occasion comme filtre antirepliement pour les CAN. Deux oscillateurs locaux (LO – *Local Oscillator*) sont aussi requis pour faire fonctionner la tête RF. Une carte d'évaluation de chez Analog Devices (EVAL-4193EB2, soit un circuit de PLL surmonté d'un VCO CLV1525E-LF de chez Z-Comm) fait office de premier LO en générant une sinusoïde à 1505,42 MHz. Le second LO est quant à lui un générateur de signaux arbitraires Agilent 33250A réglé pour produire une sinusoïde à 55 MHz. Les deux LO sont synchronisés et pilotés par une horloge Stanford CG635 à faible gigue de phase ( $< 1$  ps RMS *phase jitter*) (Stanford Research Systems, 2003). Un signal à 60 MHz est aussi extrait de l'horloge Stanford CG635 de manière à piloter les CAN et le FPGA. La Figure 3.17 illustre ces branchements de façon synthétisée.





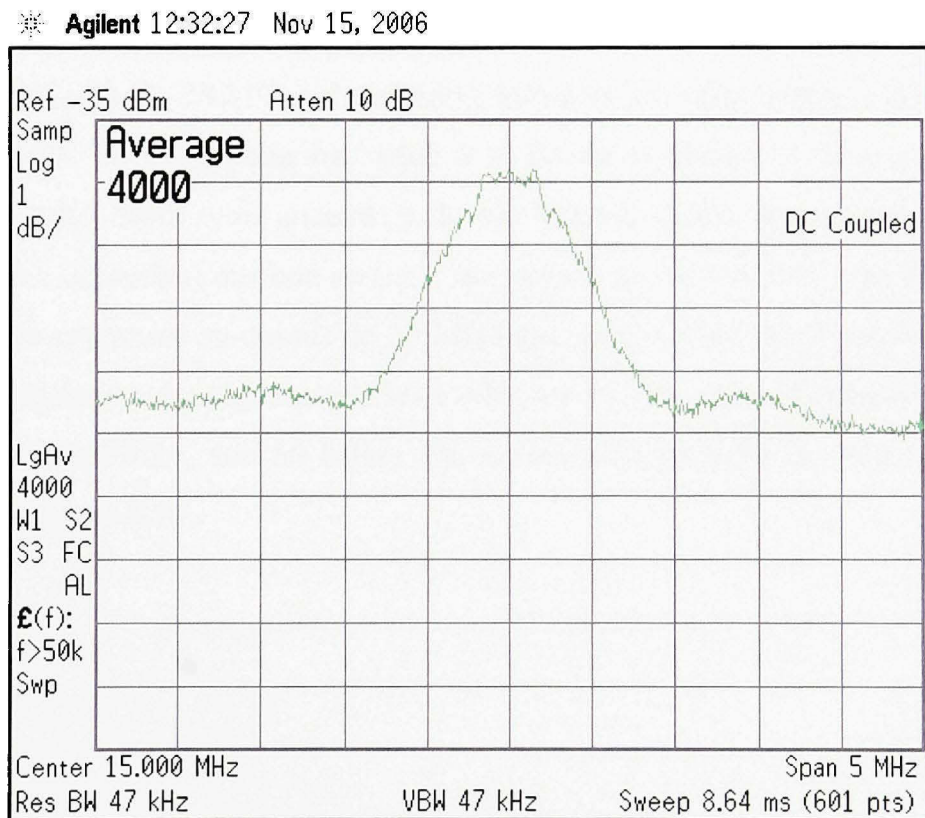
**Figure 3.17** *Schéma de branchement des horloges et LO.*

Afin d'amplifier les signaux, quatre amplificateurs de 18 dB ont été mis en cascade pour obtenir un gain total de 72 dB. Ces amplificateurs ont été montés deux par deux dans des boîtiers d'aluminium usinés à l'ÉTS de façon à minimiser les interférences, tels que montrés à la Figure 3.16. Un préamplificateur, composé d'un filtre passe-bande de 50 MHz de largeur de bande et d'un LNA (*Low Noise Amplifier*) de 14 dB de gain, a aussi été développé de manière à améliorer la qualité des signaux. Le préamplificateur n'est toutefois pas nécessaire si une antenne active est utilisée. Une source d'alimentation à 5 volts a d'ailleurs été connectée à l'entrée de la tête RF de manière à pouvoir alimenter une antenne active (voir annexe 7 pour plus de détails).

Une particularité de la tête RF développée est qu'elle est très flexible. Elle peut ainsi opérer à n'importe quelle fréquence située aux alentours de 1 à 2 GHz grâce à un simple ajustement de l'oscillateur RF (le premier oscillateur de la chaîne). En réglant ce LO à 1505,42 MHz, on arrive donc à traiter les signaux GPS sur la bande L1. On peut de la même manière traiter les signaux sur la bande L2 en abaissant le LO à 1157,6 MHz (et à 1106,45 MHz pour la fréquence L5). Puisque le préamplificateur que nous avons développé est conçu pour opérer seulement à la fréquence L1, cette flexibilité n'est donc atteinte qu'avec une antenne active pour l'instant.

### 3.4.2 Test de la tête RF

Le but de ce projet n'étant pas de développer une tête RF de haute performance, nous nous en sommes donc tenus à quelques tests de base afin de s'assurer qu'elle fonctionne adéquatement. Un premier test consistait tout d'abord à vérifier si des signaux à la fréquence L1 pouvaient être transmis au travers de la tête RF avec un gain suffisamment grand et si les signaux obtenus à la sortie étaient bel et bien centrés sur 15 MHz. Pour y parvenir, un générateur de signaux sinusoïdaux a été connecté à l'entrée de la tête RF, suivi d'un analyseur de spectre à sa sortie. Un gain de l'ordre de 70 dB a pu être ainsi déterminé en incluant le préamplificateur dans la chaîne (il faut noter ici que le gain de 14 dB du préamplificateur sert essentiellement à compenser pour les pertes d'insertion de chacune des composantes de la tête RF – voir annexe 7).

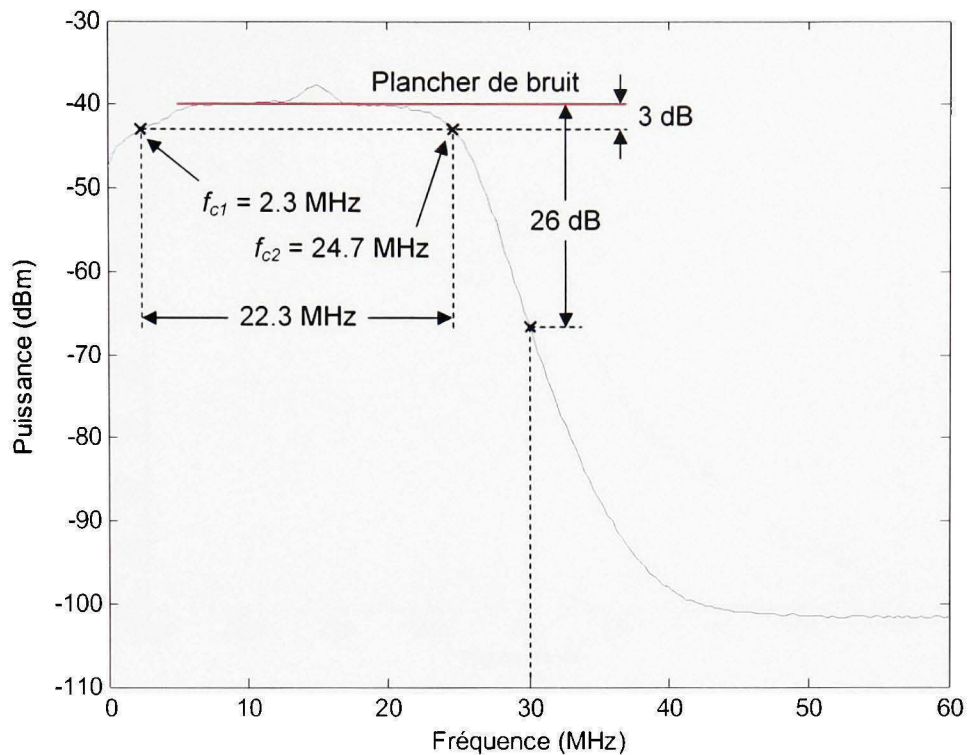


**Figure 3.18** *Lobe GPS central mesuré à la sortie de la tête RF.*

Le test a ensuite été répété en remplaçant le générateur de signaux par un simulateur de signaux GPS GSS7700 de la compagnie Spirent. Puisque les signaux GPS sont naturellement situés sous le plancher de bruit ambiant, la puissance des signaux a dû être réglée au maximum en sortie du simulateur afin de pouvoir la mesurer. Malgré cette contrainte, les tests se sont avérés un succès, tel qu'en témoignent la Figure 3.18 et la Figure 3.19. En examinant la Figure 3.18, on discerne facilement le lobe central du spectre GPS, qui dépasse de 3 à 4 dB le plancher de bruit. Le lobe en question a bien la largeur de 2 MHz (2,046 MHz pour être exact) prévue par la théorie, et on arrive à distinguer deux lobes secondaires d'une largeur de 1 MHz de part et d'autre. Le signal est aussi bien centré sur 15 MHz, ce qui confirme le bon fonctionnement de la double conversion en fréquence.

En examinant le spectre fréquentiel de la tête RF sur une plage de fréquences plus grande, il est aussi possible de mesurer la bande passante de la tête RF. La Figure 3.19 montre très clairement cette bande passante, qui s'étend de 2,3 à 24,7 MHz. La largeur de bande de la tête RF n'est donc pas de 24 MHz tel que prévu initialement, mais plutôt de 22,3 MHz. Cette différence est due au filtre passe-bas situé à la fin de la chaîne et dont la fréquence de coupure est de 24,5 MHz (voir annexe 7). Le fait d'avoir choisi une fréquence de coupure aussi basse n'est cependant pas une erreur. Ceci permet au contraire de rejeter un maximum de composantes spectrales au-dessus de 30 MHz afin qu'il n'y ait pas de repliement spectral lors de la numérisation des signaux par les CAN (dont la fréquence d'échantillonnage est la même que celle du FPGA, soit 60 MHz). On mesure ainsi un rejet de 26 dB à 30 MHz tel qu'affiché à la Figure 3.19.

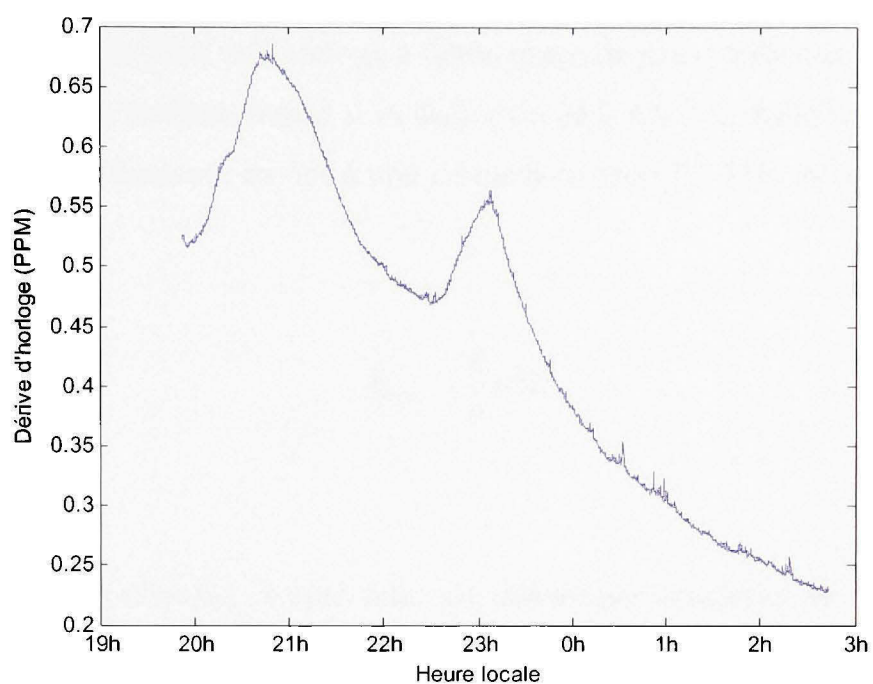




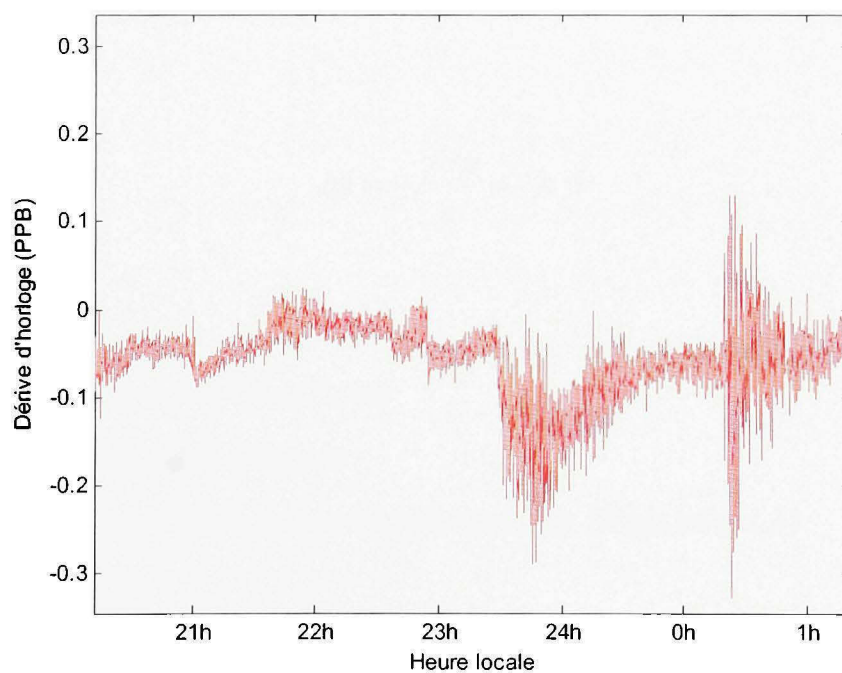
**Figure 3.19** *Mesure de la bande passante de la tête RF.*

Pour le troisième et dernier test, nous avons mesuré la dérive de l'horloge Stanford CG635 employée pour la tête RF. Or, nous avons vu au chapitre 2 que la dérive d'horloge d'un récepteur de navigation peut être obtenue en sortie de la solution de navigation. Cette dérive a donc pu être mesurée à l'aide du récepteur GPS développé une fois celui-ci complété et fonctionnel. Les résultats du test, effectué durant la nuit du 17 au 18 avril 2007, sont présentés à Figure 3.20. On peut y voir que la dérive d'horloge demeure inférieure à 0,7 PPM et qu'elle chute considérablement au fur et à mesure que le temps progresse. Ceci est dû au fait que l'horloge n'est pas compensée en température, température qui est d'ailleurs abaissée la nuit dans les locaux de l'ÉTS de manière à économiser de l'énergie durant la saison froide.

Pour des fins de comparaison, nous avons refait le même test en remplaçant la Stanford par une horloge au rubidium 8040 de Symmetricom. Les résultats, présentés à la Figure 3.21, démontrent sans équivoque la précision et la stabilité de cette horloge dont la dérive absolue demeure inférieure à 0,33 PPB (1 PPM = 1000 PPB).



**Figure 3.20** *Dérive de l'horloge Stanford CG635 durant la nuit du 17 au 18 avril 2007.*



**Figure 3.21** *Dérive de l'horloge Symmetricom 8040 durant la nuit du 18 au 19 avril 2007.*

L'utilisation d'une horloge au rubidium n'est toutefois pas nécessaire au sein d'un récepteur de navigation. Dans les faits, une horloge à faible gigue de phase telle que la Stanford permet d'obtenir d'excellents résultats même si sa dérive est de l'ordre de quelques PPM. Une erreur d'horloge de 1 PPM équivaut en fait à une erreur d'environ 1,6 kHz au niveau des mesures de Doppler, tel que :

$$\dot{b}_{PPM} = \frac{\dot{b}}{c} \times 10^6 \quad (3.19)$$

où :

- $\dot{b}$  est la dérive d'horloge en m/s telle que donnée par la solution de navigation
- $c$  est la vitesse de la lumière
- $\dot{b}_{PPM}$  est la dérive d'horloge en PPM

Ainsi, une erreur de 1 PPM équivaut à une dérive d'environ 300 m/s à la sortie de la solution de navigation, qu'on convertit ensuite en décalage Doppler grâce à l'équation (2.6) vue au chapitre 2 :

$$\Delta f = -\frac{f \cdot \dot{b}}{c} = -\lambda \cdot \dot{b} \quad (3.20)$$

où :

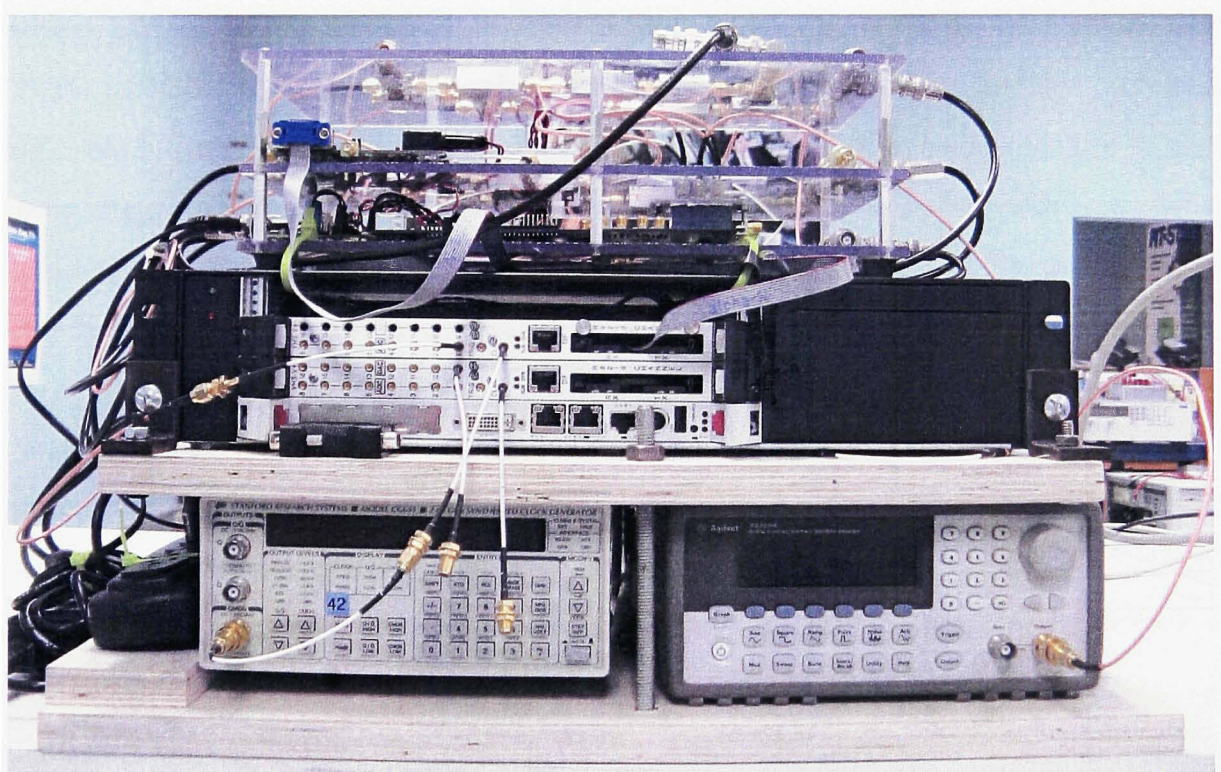
- $\Delta f$  est l'erreur Doppler équivalente
- $f$  est la fréquence de la porteuse
- $\lambda$  est la longueur d'onde de la porteuse (environ 19 cm pour L1)

Or, une dérive d'horloge aussi faible que 1 PPM n'a pas de réelle conséquence au niveau de la solution de navigation, cette dernière étant conçue pour la compenser. Il est toutefois important de la considérer lors de la recherche des signaux, un sujet abordé plus loin dans ce chapitre.



### 3.4.3 Évolution de la tête RF

La tête RF a subi quelques modifications au fil du projet. Parmi celles-ci, on note le remplacement des LO, qui étaient à l'origine deux générateurs de signaux très volumineux, par une PLL et un plus petit générateur de signaux. Ceci a contribué à rendre davantage portable et moins énergivore la plateforme de développement, facilitant du coup la réalisation des tests à l'extérieur. La tête RF a aussi été répliquée trois fois afin d'obtenir un total de quatre têtes RF fonctionnelles. Ces têtes RF pourront être employées ultérieurement pour des tests de précision appliqués à la géodésie ou encore afin de démoduler les signaux de navigation sur les diverses bandes de fréquences, telles que L2, L5/E5a ou E5b. La plateforme de développement finale, incluant les quatre têtes RF, est présentée à la Figure 3.22.



**Figure 3.22 Photographie de la plateforme de développement finale.**

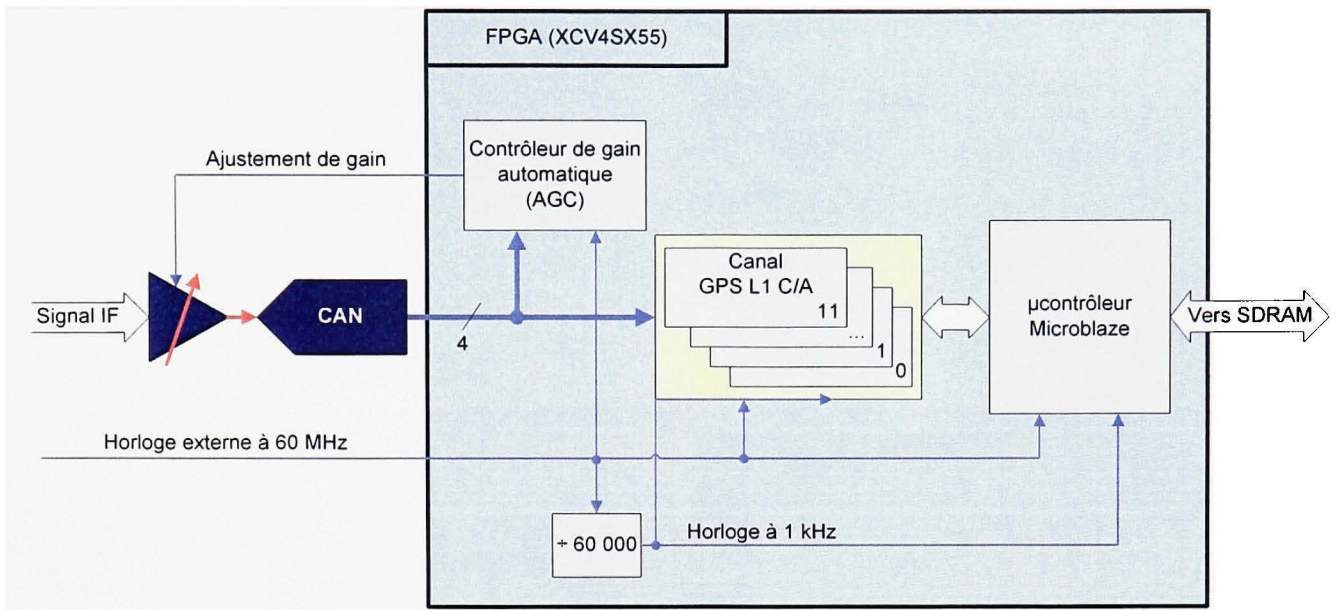
### 3.5 Mise en oeuvre des canaux de démodulation en fréquence intermédiaire

#### 3.5.1 Vue d'ensemble

Une fois les signaux GPS conditionnés par la tête RF et numérisés par un CAN, ils sont transmis au FPGA afin d'être démodulés. Un total de 12 canaux a donc été mis en œuvre au sein du FGPA de façon à démoduler les signaux L1 C/A provenant de tous les satellites GPS en vue. L'architecture développée est présentée de façon sommaire à la Figure 3.23. Il faut noter que le système opère à une fréquence nominale de 60 MHz plutôt que 105 MHz, cette dernière valeur étant la fréquence d'échantillonnage maximale des CAN. Cette limitation provient de contraintes de *timing* propres au microcontrôleur implanté dans le FPGA, et pourrait être repoussée en ajustant certains paramètres au niveau des outils de développement de Xilinx, tels que l'effort au placement ou au routage, par exemple. Un autre élément important à noter est que seulement quatre bits sont retenus en sortie du CAN plutôt que de 14. Dans les faits, il est possible de ne conserver que les quatre bits les plus significatifs (MSB - *Most Significant Bit*) d'un signal GNSS sans risquer d'en compromettre la qualité (on note une dégradation du SNR de 0,54 dB en ne gardant que deux bits et de 1,96 dB avec un seul bit) (Cloutier et al., 1999). Ceci permet de minimiser les ressources nécessaires pour chacun des canaux de démodulation et ainsi de rendre le récepteur beaucoup plus compact.

En examinant la Figure 3.23, on remarque trois autres composantes principales : un contrôleur de gain automatique, un microcontrôleur et un diviseur d'horloge. Le rôle du contrôleur de gain automatique (AGC – *Automatic Gain Control*) est en fait d'ajuster l'amplitude des signaux en entrée de façon à ce qu'elle demeure optimale. Le microcontrôleur a quant à lui comme tâche de réaliser des calculs et des opérations séquentielles trop contraignantes pour le FPGA. Ce microcontrôleur est un Microblaze de Xilinx. Un mécanisme d'interruption a été réglé au niveau du Microblaze pour accéder aux canaux de démodulation de façon régulière. Les interruptions sont quant à elles déclenchées par un signal à 1 kHz généré par un diviseur d'horloge (qui n'est rien de plus qu'un compteur modulo 60 000). Plus de détails seront apportés dans ce chapitre au sujet de l'AGC et du microcontrôleur.

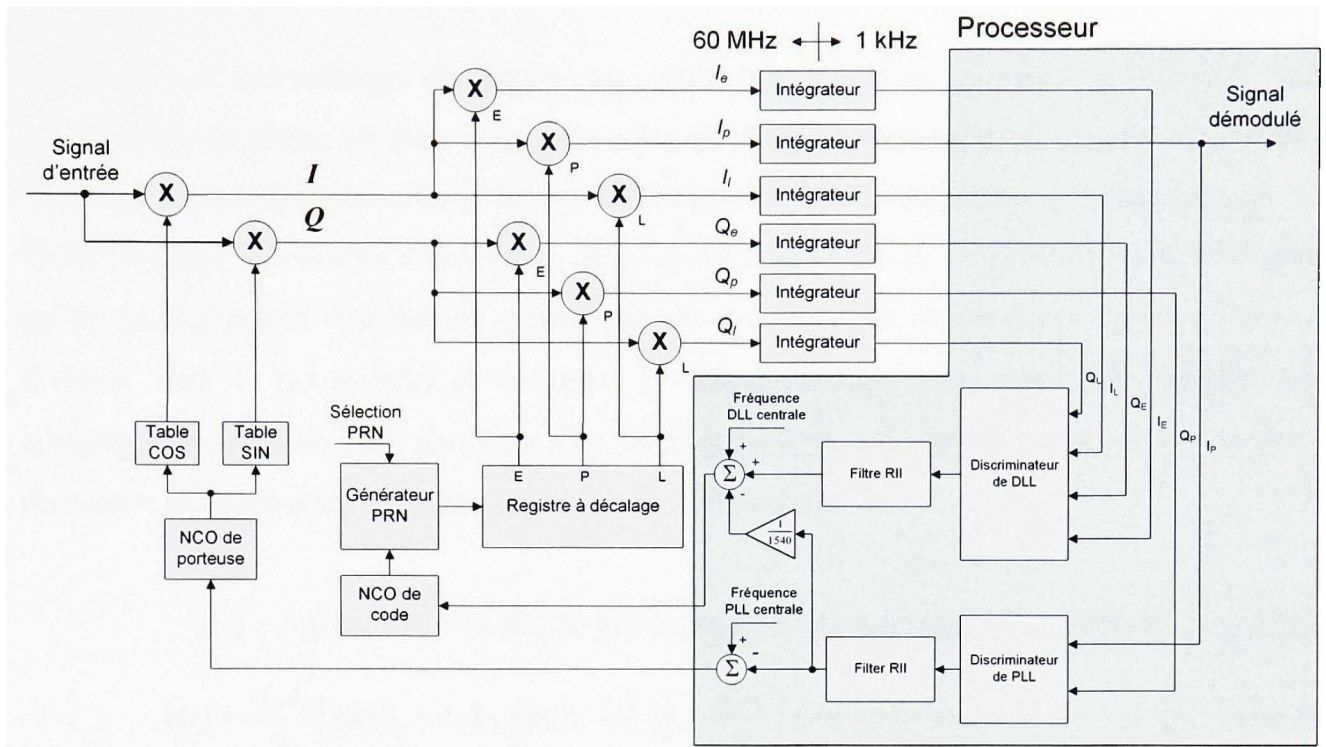




**Figure 3.23 Architecture IF générale.**

Les canaux GPS L1 C/A sont largement basés sur l'architecture de démodulation présentée dans (Kaplan, 1996, p. 123). Cette architecture repose en fait sur l'utilisation de trois corrélateurs (en avance, en phase et en retard) de type phase et quadrature afin de poursuivre à la fois le code et la porteuse des signaux GPS. La Figure 3.24 présente un canal de démodulation dans son ensemble. Le signal IF numérisé est ainsi multiplié par une copie en phase ( $I$ ) et en quadrature ( $Q$ ) de la porteuse. Chacun des signaux  $I$  et  $Q$  est à son tour multiplié par une version en avance ( $E - Early$ ), en phase ( $P - Prompt$ ) et en retard ( $L - late$ ) du code d'étalement. Ce code d'étalement est produit au niveau du récepteur à l'aide d'un générateur PRN puis décalé grâce à un registre à décalage. On obtient donc six signaux différents, soit  $I_e$ ,  $Q_e$ ,  $I_p$ ,  $Q_p$ ,  $I_l$  et  $Q_l$ , qui sont ensuite intégrés de façon numérique avant d'être transférés à un rythme de 1 kHz vers le processeur embarqué.





**Figure 3.24** Architecture d'un canal de démodulation.

Le processeur embarqué se charge ensuite de calculer les rétroactions nécessaires afin de minimiser les erreurs de phase entre le signal entrant et le signal régénéré. Des discriminateurs sont spécialement utilisés pour accomplir cette tâche, suivis de filtres afin d'en améliorer la réponse. Les rétroactions sont finalement envoyées à deux oscillateurs numériques, les NCO (*Numerically Controlled Oscillator*) de manière à refermer les boucles d'asservissement. Il y a donc deux boucles imbriquées l'une dans l'autre, soit la boucle à verrouillage de phase (PLL – *Phase Locked Loop*) dont le but est de poursuivre la phase de la porteuse, et la boucle à verrouillage de délai (DLL – *Delay Locked Loop*), qui sert à poursuivre la phase du code. Chacun des éléments énumérés ci-dessus est abordé plus en détail dans les sections qui suivent. Des schémas plus détaillés de l'architecture sont aussi disponibles à l'annexe 8.

### 3.5.2 Théorie sur les boucles à verrouillage de phase

La boucle à verrouillage de phase est constituée de trois éléments principaux : un comparateur de phase, un filtre de boucle et un oscillateur contrôlable en fréquence. Pour une PLL classique, telle que celle présentée à la Figure 3.25, l'oscillateur est généralement un VCO (*Voltage Controlled Oscillator*), alors que le comparateur de phase n'est rien d'autre qu'un multiplicateur. Ce dernier permet de mesurer l'erreur de phase  $x(t)$  entre le signal d'entrée  $S_i(t) = A_i \sin(2\pi f_i t)$  et le signal de référence  $S_r(t) = A_r \cos(2\pi f_r t)$ . Puisque des composantes parasites sont générées à haute fréquence par ce type de comparateur, un filtre de boucle est cependant nécessaire afin de s'en débarrasser :

$$x(t) = S_i(t) \cdot S_r(t) = A_i \sin(2\pi f_i t) \cdot A_r \cos(2\pi f_r t) = A_i \sin(\theta_i) \cdot A_r \cos(\theta_r) \quad (3.21)$$

$$x(t) = \frac{A_i A_r}{2} [\sin(\theta_i - \theta_r) + \sin(\theta_i + \theta_r)] = \frac{A_i A_r}{2} [\sin(\Delta\theta) + \sin(2\pi \{f_i + f_r\} t)] \quad (3.22)$$

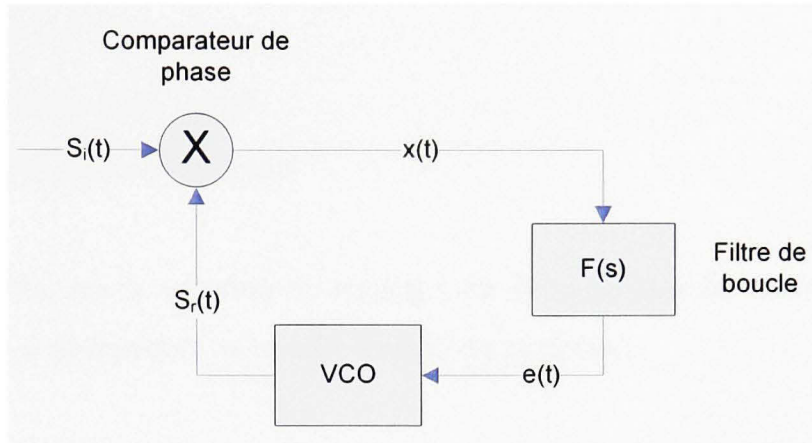
Si l'erreur de phase demeure petite, c'est-à-dire que  $\Delta\theta \ll 1$ , on peut alors approximer :

$$x(t) \approx \frac{A_i A_r}{2} [(\theta_i - \theta_r) + \sin(2\pi \{f_i + f_r\} t)] \quad (3.23)$$

$$x(t) \approx K [\Delta\theta + \sin(2\pi \{f_i + f_r\} t)] \quad (3.24)$$

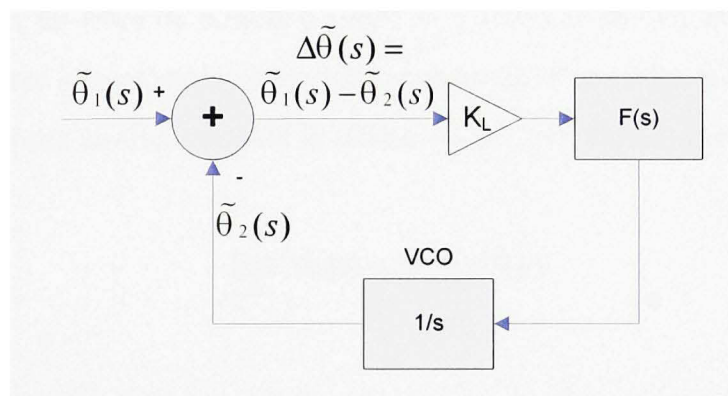
où :

- $K$  est une constante proportionnelle à l'amplitude des signaux
- $\Delta\theta$  représente l'erreur de phase entre les signaux  $S_i(t)$  et  $S_r(t)$
- $f$  est la fréquence des signaux  $S_i(t)$  et  $S_r(t)$



**Figure 3.25 Schéma bloc d'une PLL classique.**

À supposer que la condition énoncée précédemment soit respectée et que le filtre de boucle soit ajusté de façon à rejeter seulement la composante à haute fréquence, il devient alors possible de modéliser la PLL comme un système linéaire tel qu'illustré à la Figure 3.26. On pose alors un gain de boucle total égal à une constante  $K_L = K \cdot K_{VCO} \cdot K_F$ , où  $K_{VCO}$  et  $K_F$  sont les gains propres au VCO et au filtre de boucle respectivement.



**Figure 3.26 PLL linéarisée autour d'un point d'équilibre.**

Puisque le VCO prend en entrée une consigne de fréquence afin de générer une sortie de phase, sa fonction de transfert peut donc être modélisée par un intégrateur ( $1/s$ ), à savoir que :

$$2\pi f(t) = \frac{d\theta(t)}{dt} \Rightarrow \theta(t) = 2\pi \int f(t) dt \quad (3.25)$$



où :

- $\theta(t)$  est la phase d'un signal
- $f(t)$  est la fréquence d'un signal

En sachant que  $F(s)$  est la fonction de transfert du filtre de boucle, il est donc possible de calculer la fonction de transfert en boucle fermée du système :

$$H(s) = \frac{\tilde{\theta}_2(s)}{\tilde{\theta}_1(s)} = \frac{\text{Phase de sortie linéarisée}}{\text{Phase d'entrée linéarisée}} = \frac{K_L F(s)}{s + K_L F(s)} \quad (3.26)$$

Il est aussi possible de calculer la fonction d'erreur en boucle fermée :

$$\Delta\tilde{\theta}(s) = \tilde{\theta}_1(s) - \tilde{\theta}_2(s) = (1 - H(s))\tilde{\theta}_1(s) = \frac{s}{s + K_L F(s)}\tilde{\theta}_1(s) \quad (3.27)$$

On note ici que pour un filtre de boucle d'ordre  $N$ , l'ordre de la PLL en boucle fermée est de  $N+1$ . Si on s'intéresse à la réponse en régime permanent d'une PLL d'ordre  $N+1$ , il est alors possible d'avoir recours au théorème de la valeur finale, qui stipule que :

$$\lim_{t \rightarrow \infty} \Delta\theta(t) = \lim_{s \rightarrow 0} s \Delta\tilde{\theta}(s) \quad (3.28)$$

Ainsi, en supposant que le signal en entrée soit un saut de phase d'amplitude égal à  $A$ , c'est-à-dire que  $\tilde{\theta}_1(s) = A/s$ , et que le filtre soit d'ordre 0 (i.e. que la boucle est d'ordre 1), soit  $F(s) = 1$ , la réponse en régime permanent s'écrit alors :

$$\lim_{t \rightarrow \infty} \Delta\theta(t) = \lim_{s \rightarrow 0} s \frac{s}{s + K_L} \frac{A}{s} = \lim_{s \rightarrow 0} \frac{As}{s + K_L} = 0 \quad (3.29)$$

L'erreur en régime permanent pour une boucle d'ordre 1 à un saut de phase est donc nulle. Si on tente à nouveau l'expérience avec une rampe de phase dont la pente est égale à  $R$  (soit un saut de fréquence égal à  $R/2\pi$ ), c'est à dire que  $\tilde{\theta}_1(s) = R/s^2$ , on obtient cette fois-ci une erreur en régime permanent constante et égale à :

$$\lim_{t \rightarrow \infty} \Delta\theta(t) = \lim_{s \rightarrow 0} \frac{R}{s + K_L} = \frac{R}{K_L} \quad (3.30)$$

Enfin, si on appliquait une rampe de fréquence ( $\tilde{\theta}_1(s) = J/s^3$ ), on arriverait à démontrer qu'une PLL d'ordre 1 n'arrive pas à suivre le signal en entrée puisque l'erreur en régime permanent tend vers l'infini. Ceci n'est toutefois pas le cas d'une boucle d'ordre 2 dont le filtre est un intégrateur parfait, soit  $F(s) = 1/s$  :

$$\lim_{t \rightarrow \infty} \Delta\theta(t) = \lim_{s \rightarrow 0} s \frac{s}{s + K_L} \frac{J}{s^3} = \frac{J}{s^2 + K_L} = \frac{J}{K_L} \quad (3.31)$$

Enfin, en soumettant une boucle d'ordre 2 à un saut de fréquence, on peut démontrer que l'erreur en régime permanent est nulle :

$$\lim_{t \rightarrow \infty} \Delta\theta(t) = \lim_{s \rightarrow 0} s \frac{s}{s + K_L} \frac{R}{s^2} = \frac{sJ}{s^2 + K_L} = 0 \quad (3.32)$$

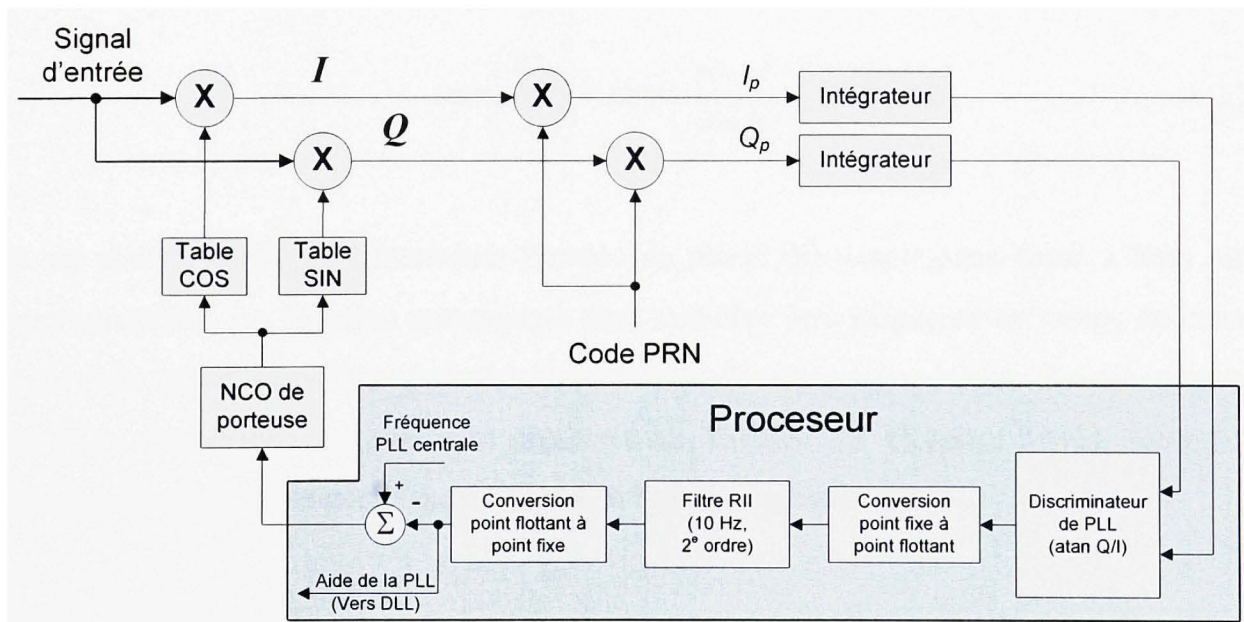
Ainsi, une boucle d'ordre 2 soumise à une rampe de fréquence (soit un stimulus d'ordre 3) présente une erreur constante en régime permanent, alors que l'erreur est nulle pour un saut de fréquence (un stimulus d'ordre 2). Ceci peut être généralisé pour n'importe quelle boucle PLL d'ordre  $N+1$  soumise à un stimulus d'ordre  $M$ , tel que :

- Si  $N+2 > M$ , alors  $\lim_{t \rightarrow \infty} \Delta\theta(t) = 0$
- Si  $N+2 = M$ , alors  $\lim_{t \rightarrow \infty} \Delta\theta(t) = E$ , où  $E$  est une constante
- Si  $N+2 < M$ , alors  $\lim_{t \rightarrow \infty} \Delta\theta(t) = \infty$

Cette propriété est exploitée dans la section suivante où les PLL appliquées au GPS sont abordées. Pour plus d'information sur la théorie englobant les PLL, le lecteur est invité à consulter (Best, 2003).

### 3.5.3 Conception de la boucle à verrouillage de phase GPS L1 C/A

La boucle à verrouillage de phase que nous avons employée dans le récepteur GPS peut sembler plus compliquée qu'une PLL classique. Dans les faits, deux signaux, un en phase et l'autre en quadrature, sont disponibles et un discriminateur de phase est nécessaire, tels que présentés à la Figure 3.27. Ceci permet toutefois de simplifier grandement les calculs.



**Figure 3.27** Schéma bloc de la boucle à verrouillage de phase développée.



Si on considère un signal  $S(t) = A \cos(2\pi f_i t)$  en entrée et deux signaux de référence provenant du NCO, soit  $S_I(t) = \cos(2\pi f_r t)$  et  $S_Q(t) = \sin(2\pi f_r t)$ , on obtient après multiplication (on néglige ici le code d'étalement qu'on suppose parfaitement enlevé lors de la démodulation) :

$$\begin{aligned} I(t) &= \cos(2\pi f_i t) \cos(2\pi f_r t) = \frac{A}{2} \left[ \cos(\Delta\theta) + \cos(2\pi \{f_r + f_i\} t) \right] \\ Q(t) &= \cos(2\pi f_i t) \sin(2\pi f_r t) = \frac{A}{2} \left[ \sin(\Delta\theta) + \sin(2\pi \{f_r + f_i\} t) \right] \end{aligned} \quad (3.33)$$

Puisque les composantes à haute fréquence sont rejetées lors du passage dans les intégrateurs, on obtient donc :

$$I_p = \frac{A}{2} \cos \Delta\theta, \quad Q_p = \frac{A}{2} \sin \Delta\theta \quad (3.34)$$

En employant la fonction arctangente comme discriminateur, on obtient finalement :

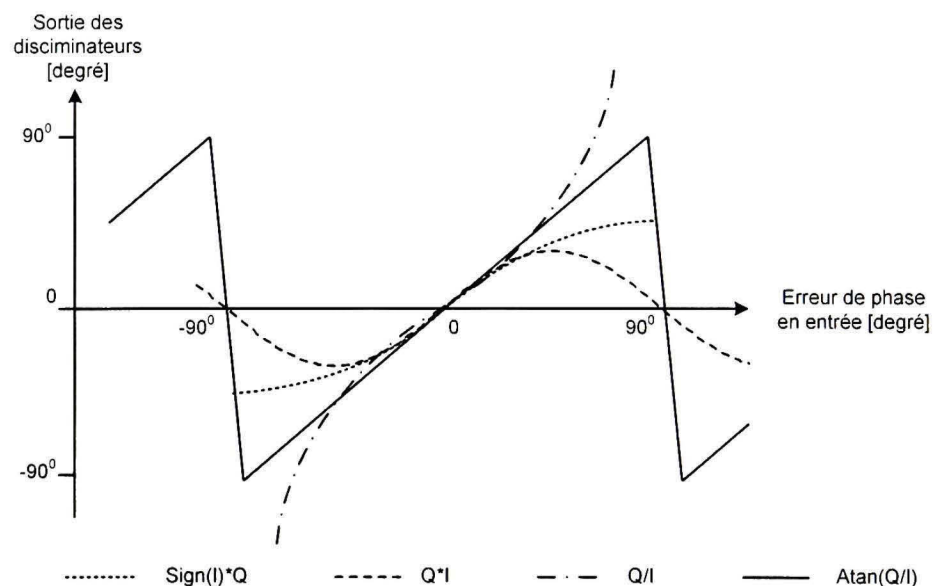
$$\text{atan} \left( \frac{Q_p}{I_p} \right) = \text{atan} \left( \frac{\sin \Delta\theta}{\cos \Delta\theta} \right) = \Delta\theta \quad (3.35)$$

Il est ainsi possible de déterminer l'erreur de phase  $\Delta\theta$  exacte sans avoir à faire aucune approximation. La fonction arctangente peut toutefois être exigeante en temps de calcul, et pour cette raison, plusieurs approximations ont été développées au niveau des discriminateurs de PLL. Ces discriminateurs sont présentés au Tableau 3.1 (Kaplan, 1996), suivi de leur courbes de réponse respectives à la Figure 3.28 (Chavagnieux, 2006).

Tableau 3.1  
Liste des principaux discriminateurs de PLL

Équation du discriminateur	Sortie du discriminateur	Particularités
$I_p \cdot Q_p$	$\sin 2\Delta\theta$	Optimal pour des SNR faibles Charge de calcul modérée Pente proportionnelle au carré de l'amplitude du signal
$\text{sign}(I_p) \cdot Q_p$	$\sin \Delta\theta$	Optimal pour des SNR élevés Charge de calcul modérée Pente proportionnelle à l'amplitude du signal
$Q_p / I_p$	$\tan \Delta\theta$	Charge de calcul supérieure Pente indépendante de l'amplitude du signal
$\text{atan}(Q_p / I_p)$	$\Delta\theta$	Optimal pour des SNR faibles et élevés Charge de calcul la plus importante Pente indépendante de l'amplitude du signal

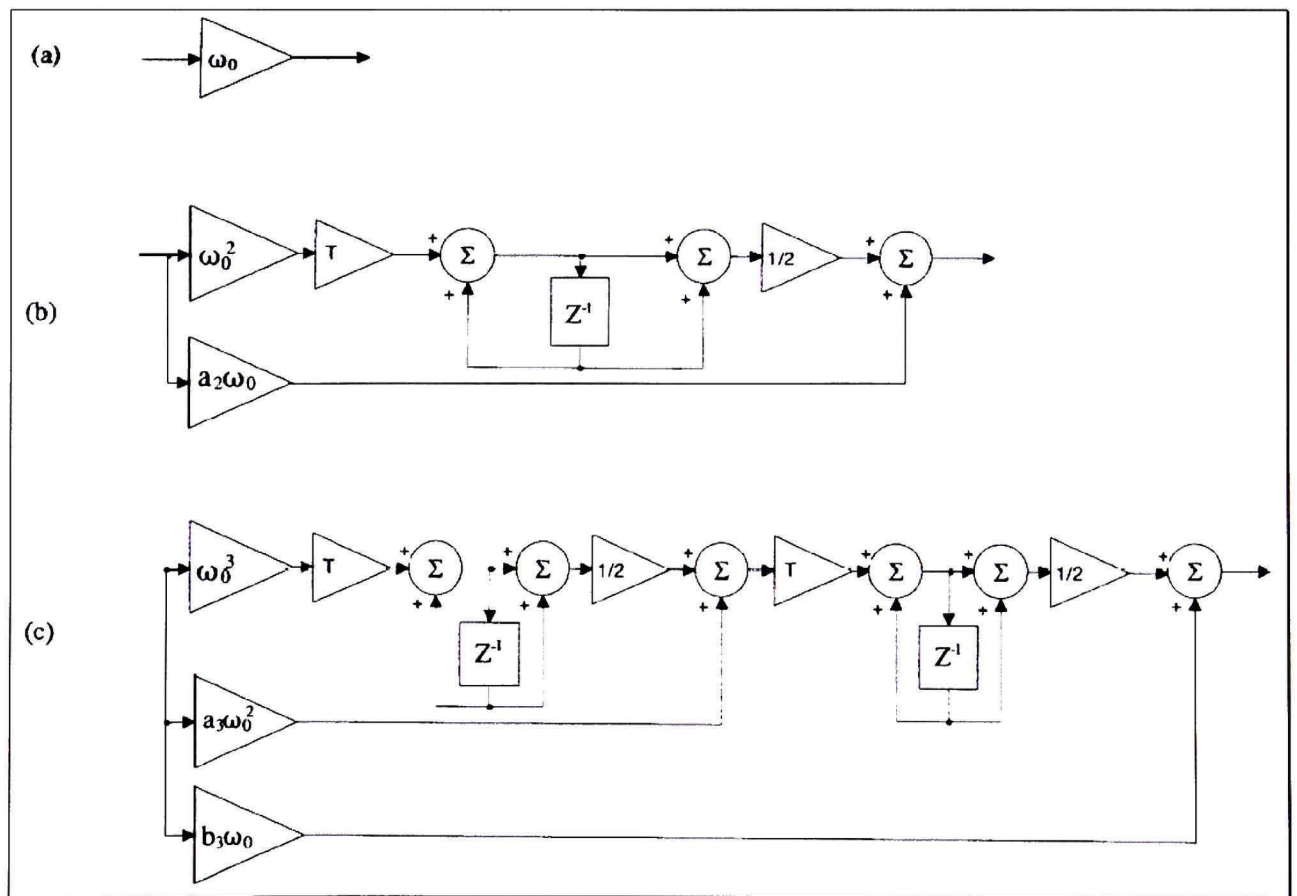
Pour la conception du récepteur GPS, notre choix s'est arrêté sur l'arctangente puisqu'il s'agit du discriminateur au maximum de vraisemblance. Afin de limiter la charge de calcul du processeur embarqué, nous avons toutefois programmé une fonction en assembleur en nous basant sur un algorithme CORDIC (*COordinate Rotation Digital Computer*). Les détails d'un tel algorithme peuvent être retrouvés dans (Andraka, 1998).



**Figure 3.28** Courbes de réponse des principaux discriminateurs de PLL.  
Tiré de (Chavagnieux, 2006)

Le choix d'un bon filtre de boucle joue aussi un rôle important au niveau de la PLL. Pour la conception du récepteur GPS, nous avons opté pour un filtre numérique à réponse impulsionnelle infinie (RII), compact et facilement implantable au sein d'un processeur embarqué. Puisque le Microblaze le permet, nous avons pu réaliser le filtre directement en point flottant, ce qui nous a permis de gagner du temps puisqu'il n'était plus nécessaire de s'occuper des problèmes de précision et de débordement associés aux calculs à point fixe. Une conversion des données de point fixe à point flottant s'est avérée toutefois nécessaire à l'entrée du filtre, suivi de l'opération inverse en sortie.

La Figure 3.29 (Kaplan, 1996) présente une série de filtres RII typiquement employés pour la conception de boucles de phase et de code GPS. Les paramètres associés à ces filtres sont succinctement présentés au Tableau 3.2.



**Figure 3.29** Schémas bloc de filtres de boucles numériques d'ordre 0 (a), d'ordre 1 (b) et d'ordre 2 (c).

Tiré de (Kaplan, 1996, p. 155)



Tableau 3.2  
Bande de bruit et coefficients de filtres de boucle numériques typiques

Ordre du filtre	Bande de bruit ( $B_n$ )	Coefficients typiques
0	$\frac{\omega_0}{4}$	$\omega_0$ $B_n = 0,25\omega_0$
1	$\frac{\omega_0(1+a_2^2)}{4a_2}$	$\omega_0^2$ $a_2\omega_0 = 1.414\omega_0$ $B_n = 0,53\omega_0$
2	$\frac{\omega_0(a_3b_3^2 + a_3^2 - b_3)}{4(a_3b_3 - 1)}$	$\omega_0^3$ $a_3\omega_0^2 = 1.1\omega_0^2$ $b_3\omega_0 = 2.4\omega_0$ $B_n = 0,7845\omega_0$

Le paramètre  $T$  affiché à la Figure 3.29 représente la période d'échantillonnage du filtre et il est directement lié à la période d'intégration des signaux. Dans le cas du GPS,  $T$  est généralement compris entre 1 et 20 ms inclusivement, ce qui correspond de façon respective à la durée d'une époque et à celle d'un bit de données. Une période  $T = 1$  ms a été retenue pour la PLL de manière à faciliter son implémentation en temps réel. Une intégration de plus de 1 ms nécessiterait au préalable une synchronisation avec les bits de données afin d'éviter que l'on n'intègre sur plusieurs bits consécutifs, ce qui rend instable la PLL. Or, il est impossible de se synchroniser avec les bits de données tant que la PLL n'est pas verrouillée.

Un filtre d'ordre 2 a été choisi pour la PLL de manière à rendre le récepteur robuste face aux perturbations dynamiques externes. Selon la théorie, un tel filtre permet de poursuivre une rampe de fréquence sans erreur ainsi qu'une parabole de fréquence avec une erreur constante. De façon pratique, cela signifie que la PLL est insensible aux accélérations externes et qu'elle peut encaisser un jerk maximal avant de décrocher. Ce jerk maximal est fonction de la bande de bruit du filtre ( $B_n$ ) et est intimement lié au ratio  $C/N_0$  (*Carrier to Noise Power Density Ratio*) des signaux. Or, en augmentant la bande de bruit du filtre, on augmente dans une certaine mesure la robustesse de la PLL face aux dynamiques externes. D'un autre côté, ceci entraîne une dégradation de la qualité des signaux démodulés dû au bruit thermique.

Typiquement, la bande de bruit se situe entre 2 et 18 Hz pour une PLL d'ordre 3 (Kaplan, 1996). Nous avons choisi une bande de bruit de 10 Hz pour notre filtre de PLL de manière à offrir un juste compromis entre les dynamiques pouvant être encaissées et la qualité des signaux. En considérant par exemple des signaux GPS de puissance moyenne (soit  $C/N_0 = 35$  dB Hz), on obtient selon (Kaplan, 1996, p. 165-166) une gigue de phase de l'ordre de  $4^\circ$  à  $1\sigma$  sur la porteuse. De plus, ceci permet à la PLL de résister à un jerk maximal de l'ordre de 4 G/s, où  $1\text{ G} = 9,81\text{ m/s}^2$ .

La robustesse et la précision de la PLL sont étudiées plus en détail au chapitre 6 alors que des tests en environnements simulé et réel y sont conduits. Les paramètres choisis pour la PLL sont résumés au Tableau 3.3 et peuvent être aussi retrouvés à l'annexe 1.

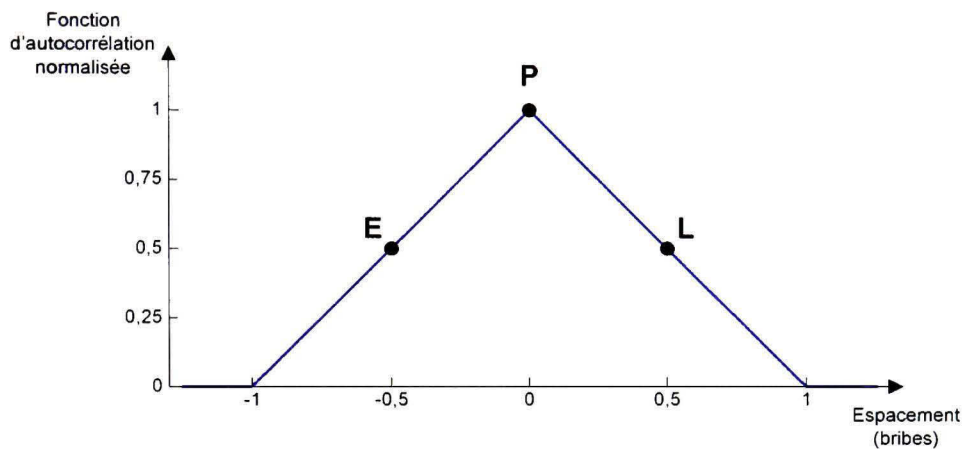
Tableau 3.3  
Résumé des paramètres de la PLL

Paramètres de la PLL	Valeurs
Période d'intégration ( $T$ )	1 ms
Ordre de la boucle ( $N+1$ )	3
Bande de bruit ( $B_n$ )	10 Hz
Discriminateur	atan(Q/I)

### 3.5.4 Conception de la boucle à verrouillage de délai GPS L1 C/A

La boucle à verrouillage de délai, telle que nous l'avons implantée dans le récepteur, se base sur des corrélateurs en avance (E) et en retard (L). Un délai de 0,5 bribe est inséré entre chacun de ces corrélateurs et le corrélateur en phase (P) lors de la phase initiale de poursuite, ce qui offre une grande plage dynamique à la DLL. Puisque le corrélateur en phase est en théorie synchronisé avec le code d'étalement, ce dernier reçoit donc un maximum de puissance du signal démodulé, alors que les corrélateurs en avance et en retard ne reçoivent que 50% de la puissance, tel qu'affiché à la Figure 3.30. Pour cette raison, l'espacement entre les corrélateurs est diminuée à seulement  $\pm 1/20^\circ$  de bribe une fois la DLL stabilisée, ce qui permet de diminuer le bruit au niveau de la DLL. Il faut noter qu'une telle manœuvre permet aussi d'améliorer la robustesse face aux multitrajets, à condition que la largeur de bande des

signaux en entrée soit large (10-20 MHz), ce qui est le cas de notre récepteur (Grewal, Weill et Andrews, 2001).



**Figure 3.30** *Disposition des corrélateurs en avance, en phase et en retard sur la fonction d'autocorrélation des signaux GPS L1 C/A.*

Afin de mesurer l'erreur de phase en bribes de la DLL,  $\Delta t_c$ , il s'agit de calculer la différence d'amplitude entre les corrélateurs E et L, telle que :

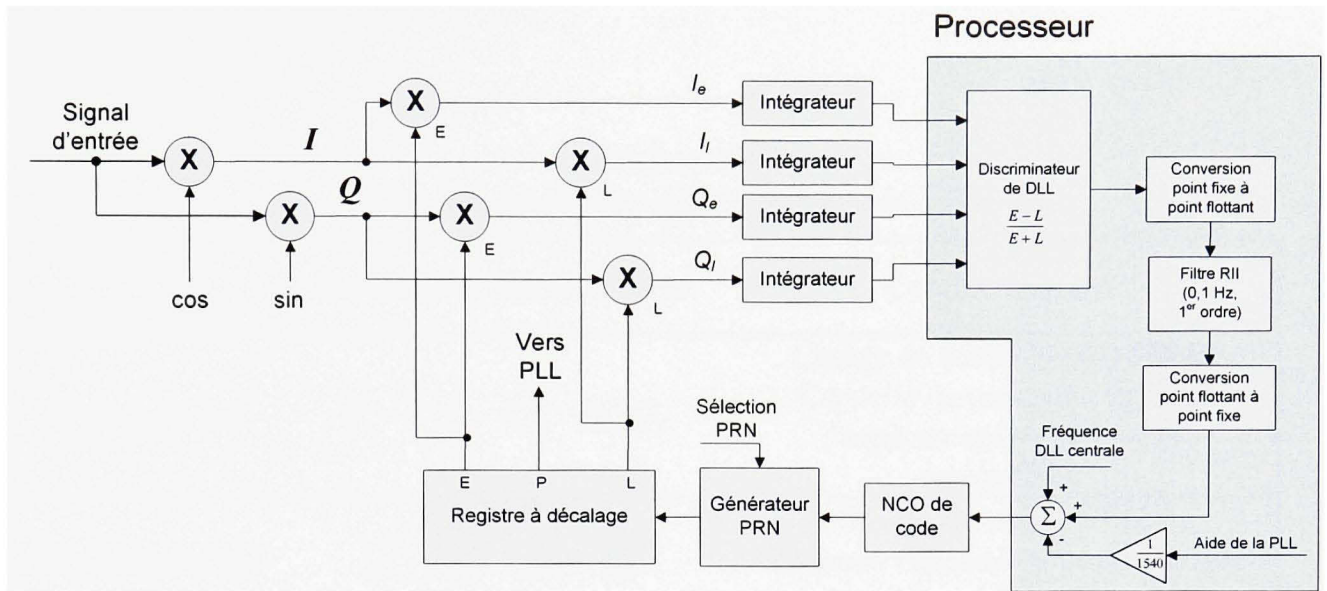
$$\Delta t_c \approx E - L \quad (3.36)$$

Le problème toutefois avec ce type discriminateur est que l'erreur calculée est dépendante de l'amplitude des signaux. Il s'agit donc de normaliser l'équation (3.36) afin d'obtenir en sortie du discriminateur la véritable erreur de phase :

$$\Delta t_c = \frac{E - L}{E + L} \quad (3.37)$$

On appelle ce discriminateur « enveloppe avance moins retard normalisée » (enveloppe AMRN). Puisqu'il s'agit d'un discriminateur de DLL au maximum de vraisemblance, ce dernier a été retenu pour le développement de la boucle PLL, présentée à la Figure 3.31.





**Figure 3.31** *Schéma bloc de la boucle à verrouillage de délai développée.*

On remarque à la Figure 3.31 que quatre signaux sont fournis en entrée du discriminateur de la DLL, soit  $I_e$ ,  $Q_e$ ,  $I_l$  et  $Q_l$ . Ces signaux peuvent être perçus comme des composantes vectorielles, dont l'angle représente l'erreur de phase de la PLL telle que vue précédemment. La norme de ces vecteurs représente quant à elle l'amplitude des signaux et se calcule :

$$E = \sqrt{I_e^2 + Q_e^2}, \quad P = \sqrt{I_p^2 + Q_p^2}, \quad L = \sqrt{I_l^2 + Q_l^2} \quad (3.38)$$

Le discriminateur enveloppe AMRN est donc mis en œuvre de la façon suivante :

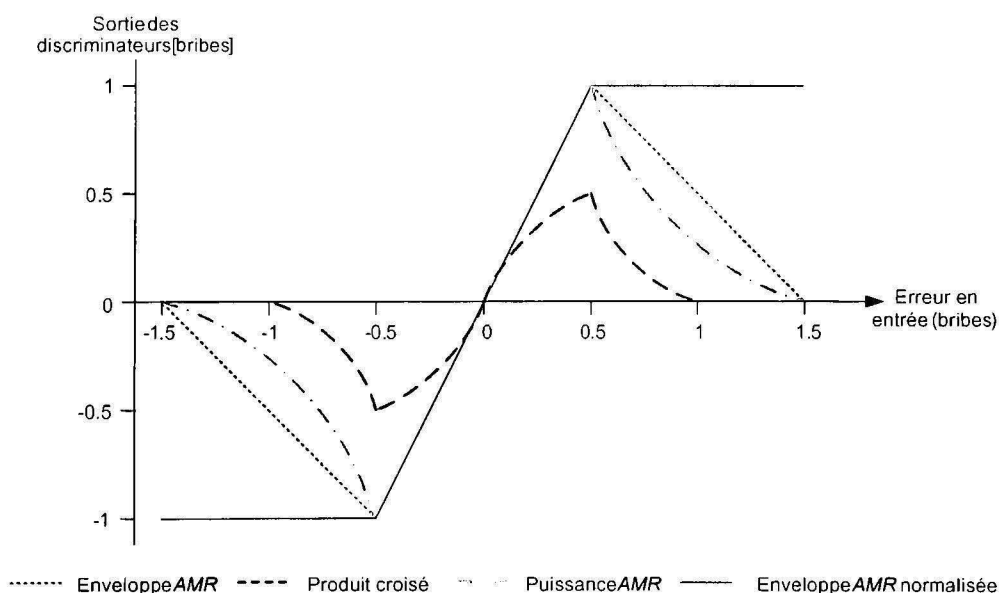
$$\Delta t_c = \frac{\sqrt{I_e^2 + Q_e^2} - \sqrt{I_l^2 + Q_l^2}}{\sqrt{I_e^2 + Q_e^2} + \sqrt{I_l^2 + Q_l^2}} \quad (3.39)$$

Le calcul de quatre racines carrées et d'une division est nécessaire, toutes des opérations relativement exigeantes en temps de calcul. À l'instar de la PLL, plusieurs approximations ont donc été développées afin de remplacer l'AMRN. Ces discriminateurs sont présentés au

Tableau 3.4 (Kaplan, 1996) suivi de leur courbes de réponse respectives à la Figure 3.32 (Chavagnieux, 2006).

Tableau 3.4  
Liste des principaux discriminateurs de DLL

Nom du discriminateur	Équation du discriminateur	Caractéristiques
Produit croisé	$(I_e - I_l)I_p - (Q_e + Q_l)Q_p$	Charge de calcul la plus faible Capacité de poursuite égale à l'espacement en brique des corrélateurs
Enveloppe AMR	$\sqrt{I_e^2 + Q_e^2} - \sqrt{I_l^2 + Q_l^2}$	Charge de calcul élevée Meilleure capacité de poursuite que le produit croisé
Puissance AMR	$(I_e^2 + Q_e^2) - (I_l^2 + Q_l^2)$	Charge de calcul modérée Caractéristiques semblables à l'enveloppe AMR
Enveloppe AMR normalisée	$\frac{\sqrt{I_e^2 + Q_e^2} - \sqrt{I_l^2 + Q_l^2}}{\sqrt{I_e^2 + Q_e^2} + \sqrt{I_l^2 + Q_l^2}}$	Charge de calcul la plus élevée Insensible à l'amplitude du signal Capacité de poursuite égale à 1,5 fois l'espacement en brique des corrélateurs



**Figure 3.32** *Courbes de réponse des principaux discriminateurs de DLL.*  
Tiré de (Chavagnieux, 2006)

Afin de minimiser la charge de calcul engendrée dans le microcontrôleur embarqué par le discriminateur enveloppe AMRN, un algorithme a été rédigé en assembleur afin de calculer les racines carrées. Les détails de cet algorithme peuvent être retrouvés dans (Turkowski, 1995). Une période d'intégration de 10 ms a aussi été choisie de manière à améliorer la réponse de la DLL et réduire davantage la charge de calcul. Contrairement à l'arctangente de la PLL, l'enveloppe AMRN exige d'élever au carré les signaux en phase et en quadrature, ce qui enlève du coup l'ambiguïté de signe liée à la transition des bits de données. Les signaux peuvent donc être intégrés sur une période plus longue par le biais d'une méthode appelée « intégration non cohérente » telle que :

$$S = \sqrt{I_0^2 + Q_0^2} + \sqrt{I_1^2 + Q_1^2} + \dots + \sqrt{I_{N-1}^2 + Q_{N-1}^2} = \sum_{i=0}^{N-1} \sqrt{I_i^2 + Q_i^2} \quad (3.40)$$

où :

- $I_i$  et  $Q_i$  sont les signaux en phase et en quadrature aux époques  $i = 0, 1, 2 \dots$
- $N$  est le nombre d'époques d'intégration non cohérente
- $S$  est le résultat d'une intégration non cohérente

La bande de bruit d'un filtre de DLL se situe autour de 1 à 2 Hz typiquement. Ces valeurs sont inférieures aux valeurs que l'on donne généralement au filtre de la PLL et s'explique par le fait que les dynamiques exercées sur le code sont moins grandes que sur la porteuse. Dans les faits, la fréquence du code d'étalement C/A est 1540 fois inférieure à celle de la porteuse L1. Les dynamiques sont donc aussi atténuées par un facteur de 1540 au niveau du code. D'autres perturbations, telles que la gigue de l'horloge du récepteur et les multitrajets, font cependant en sorte d'augmenter artificiellement les dynamiques au niveau de la DLL; ce qui explique qu'il n'y ait qu'un facteur de dix entre la bande de bruit de la PLL et celle de la DLL.

Deux filtres de boucle semblable à ceux de la PLL ont été retenus pour la DLL : un premier de bande de bruit  $B_n = 2$  Hz et un second pour lequel  $B_n = 0,1$  Hz. Considérant les faibles



dynamiques présentes sur le code, un ordre 1 a été choisi pour les deux filtres de afin d'accroître la stabilité de la DLL. Il faut toutefois noter qu'une aide externe provenant de la PLL est apportée au second filtre afin de compenser les dynamiques plus élevées. Le premier filtre est donc employé lors de la phase initiale de poursuite des signaux de manière à stabiliser rapidement la DLL. Le second filtre prend ensuite la relève afin de maximiser la précision de la DLL. Puisque les mesures de pseudo-distances proviennent directement de la phase du code, l'utilisation d'un second filtre plus restrictif permet ainsi d'en augmenter la précision. Les paramètres de la DLL sont résumés au Tableau 3.5.

Tableau 3.5  
Résumé des paramètres de la DLL

Paramètres de la DLL	Valeurs (phase initiale de poursuite)	Valeurs (poursuite avancée)
Période d'intégration ( $T$ )	10 ms	10 ms
Type d'intégration	Non cohérente	Non cohérente
Ordre de la boucle ( $N+1$ )	2	2
Bande de bruit ( $B_n$ )	2 Hz	0,1 Hz
Discriminateur	Enveloppe AMRN	Enveloppe AMRN
Espacement des corrélateurs	$\pm 1/2$ bribe	$\pm 1/20$ bribe

### 3.5.5 Réalisation des oscillateurs contrôlés numériquement (NCO)

Deux NCO sont présents dans chacun des canaux de démodulation GPS L1 C/A. Un premier sert à reproduire une version en phase et en quadrature de la porteuse et l'autre génère un signal d'horloge destiné au générateur de codes PRN. Le principe de base demeure toutefois le même dans les deux cas (voir Figure 3.33).

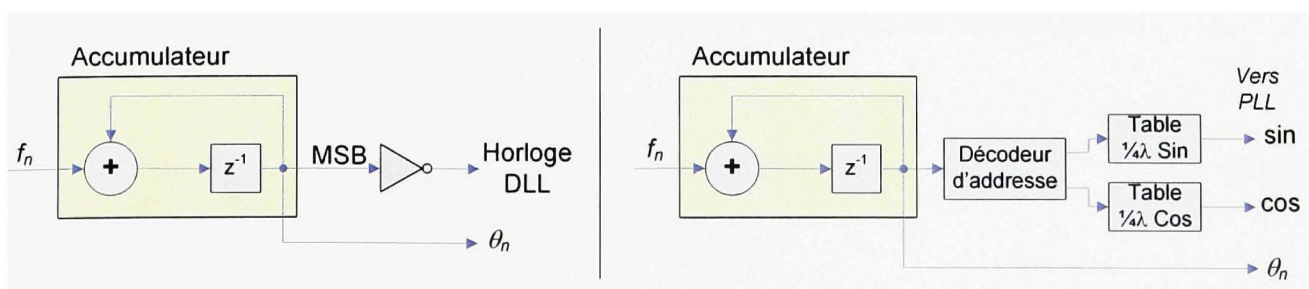


Figure 3.33 Schéma bloc des NCO de code (gauche) et de porteuse (droite).

Chaque NCO repose sur un accumulateur de phase, qui permet de convertir une consigne de fréquence en une sortie de phase. Puisque les NCO ont été implémentés directement dans le FPGA, il était donc plus facile d'employer des accumulateurs modulo  $2^N$ , où  $N$  représente le nombre de bits d'un accumulateur. Ceci nécessite simplement de convertir la véritable consigne en fréquence en une consigne machine, telle que :

$$f_n = \left\lfloor \frac{f}{f_s} 2^N \right\rfloor \quad (3.41)$$

où :

- $f_n$  est la fréquence normalisée numérique, soit la valeur transmise au NCO
- $f$  est la consigne en fréquence
- $f_s$  est la fréquence d'échantillonnage
- $\lfloor \rfloor$  représente l'opérateur d'arrondissement vers l'entier le plus près

Une formule très similaire est employée pour convertir la sortie de phase des NCO en radians :

$$\theta = 2\pi \frac{\theta_n}{2^N} \quad (3.42)$$

où :

- $\theta_n$  représente le contenu de l'accumulateur
- $\theta$  est la phase en radians

Prenons un exemple où  $N = 4$  bits,  $f_s = 100$  MHz et  $f = 25$  MHz. On calcule alors :

$$f_n = \left\lfloor \frac{25 \text{ MHz}}{100 \text{ MHz}} 2^4 \right\rfloor = \lfloor 0,25 \times 16 \rfloor = 4 \quad (3.43)$$

L'accumulateur de phase s'incrémentera donc de 4 à chaque coup d'horloge. En supposant que le NCO soit initialisé à 0, on obtiendra de façon consécutive : 0, 4, 8, 12 puis 0. Un signal périodique et de fréquence égale à  $f_s/4$ , soit 25 MHz est ainsi créé.

Dans le cas du NCO de code, nous nous sommes rendus compte après une série d'ajustements que sa résolution influençait sévèrement sur la précision de la DLL. Sa résolution a donc été fixée à  $N = 47$  bits de manière à limiter au minimum son impact sur la DLL. Cette résolution correspond à une dérive maximale en boucle ouverte de  $\delta x = 0,1$  mm/s, fixée *a priori* pour le NCO. Or, la longueur d'onde ( $\lambda$ ) du code C/A est d'environ 293 m :

$$\lambda = \frac{c}{f} \approx \frac{3 \times 10^8 \text{ m/s}}{1,023 \times 10^6 \text{ Hz}} = 293,3 \text{ m} \quad (3.44)$$

Ceci dit, 0,1 mm/s représente une dérive de phase  $\delta\theta$  d'environ 2  $\mu\text{rad/s}$  au niveau du code :

$$\delta\theta = 2\pi \frac{\delta x}{\lambda} = 2\pi \frac{0,1 \times 10^{-3} \text{ m/s}}{293,3 \text{ m}} = 2,15 \mu\text{rad/s} \quad (3.45)$$

Sachant que la fréquence d'échantillonnage du système est de 60 MHz, l'erreur de phase maximale  $\theta_e$  à chaque coup d'horloge est donc égale à :

$$\theta_e = \frac{\delta\theta}{f_s} = \frac{2,15 \times 10^{-6} \text{ rad/s}}{60 \times 10^6 \text{ s}^{-1}} = 35,8 \times 10^{-15} \text{ rad} \quad (3.46)$$

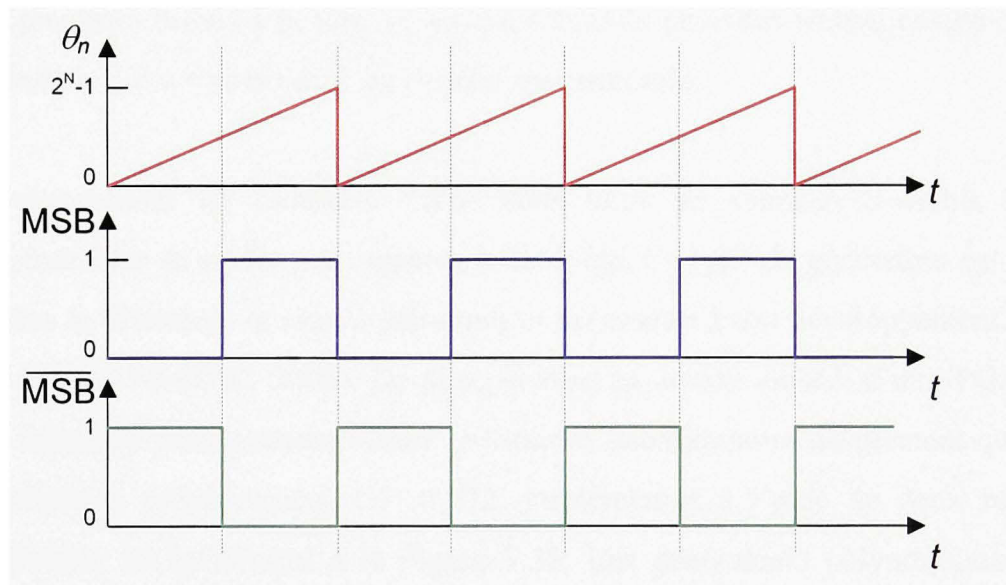
En fixant  $\theta_n = 1$  dans l'équation (3.42), on peut déterminer la résolution du NCO, telle que :

$$\frac{1}{2^N} = \frac{\theta_e}{2\pi} = 5,7 \times 10^{-15} \Rightarrow N = -\log_2(5,7 \times 10^{-15}) \approx 47 \text{ bits} \quad (3.47)$$



Pour ce qui est de la résolution du NCO de PLL, on sait que la longueur d'onde de la porteuse est 1540 fois plus petite que celle du code. Par conséquent, il est possible de garantir les mêmes spécifications que le NCO de code avec environ 11 bits en moins. Si on considère le fait que la fréquence d'échantillonnage de la PLL est dix fois plus élevée que celle de la DLL, il est alors permis de soustraire trois bits supplémentaires, pour une résolution finale de 33 bits.

Au niveau de la génération des signaux, seul le signe de la phase est conservé en sortie du NCO de code. Le signe étant représenté par le bit le plus significatif, il s'agit alors d'inverser le MSB de manière à générer un signal d'horloge adéquat, tel que montré à la Figure 3.34.



**Figure 3.34** Synthèse de l'horloge de DLL.

Dans le cas du NCO de porteuse, la phase de l'accumulateur est transmise à deux tables trigonométriques, soit une en cosinus et l'autre en sinus, de manière à régénérer les porteuses en phase et en quadrature. Ces tables sont d'un quart de longueur d'onde afin de minimiser les ressources nécessaires dans le FPGA. Puisque le signal IF entrant dans les canaux de démodulation est encodé sur 4 bits à complément à 2, les tables ont par ailleurs été limitées à 3 bits de résolution par 16 éléments de profond, pour un total de 48 bits par table. Un

décodeur est employé afin d'émuler des tables complètes de 4 bits de résolution par 64 éléments de profond. Ces caractéristiques sont résumées à l'annexe 1.

### 3.5.6 Réalisation du générateur de codes pseudo-aléatoires C/A

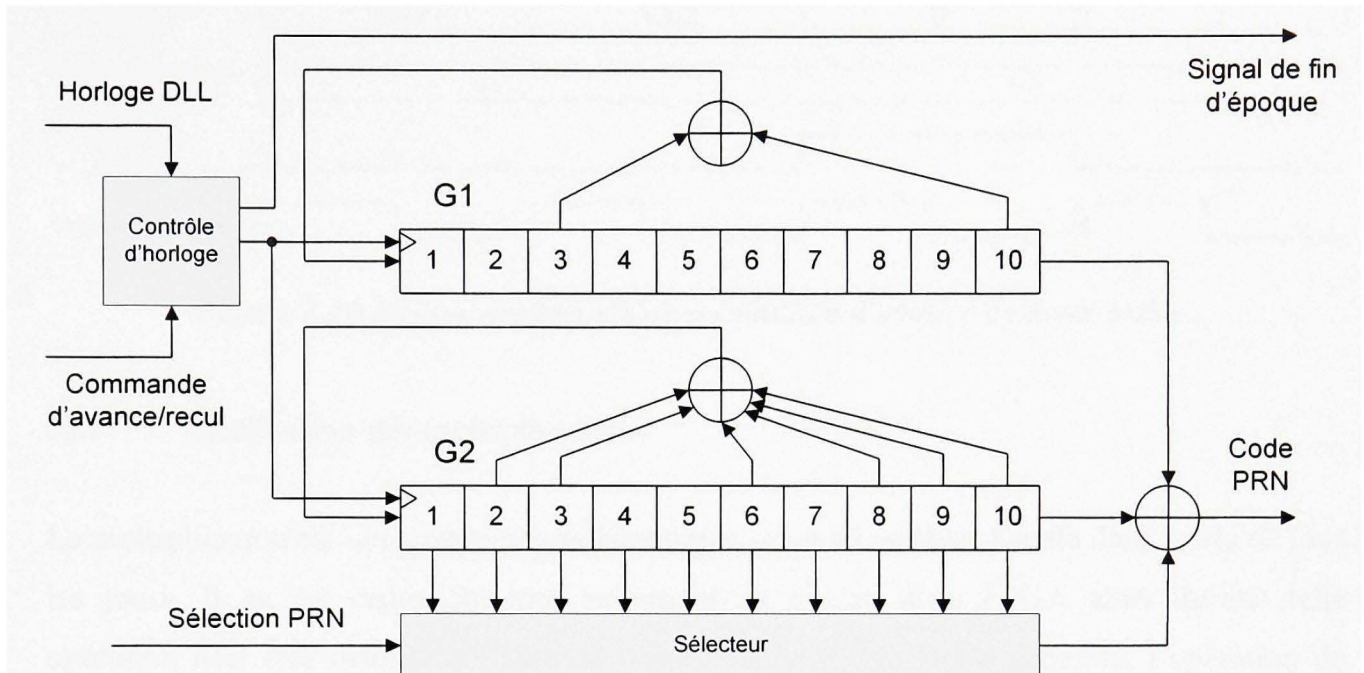
Plusieurs approches sont aujourd'hui possibles afin de régénérer les codes d'étalement pseudo-aléatoires au sein d'un récepteur. Les codes blocs en sont un exemple. Cette technique consiste à stocker les codes pseudo-aléatoires dans des blocs de mémoires pour y accéder ultérieurement lors de la démodulation des signaux. Ceci permet d'offrir une grande flexibilité, car des codes d'étalement peuvent être ajoutés ou modifiés tout au long de la vie d'un récepteur sans avoir à en modifier la structure. Les codes peuvent aussi être accédés à haut débit et plusieurs bribes à la fois, ce qui est très utile pour des techniques de recherche des signaux basées sur la transformée de Fourier, par exemple.

Pour le développement du récepteur GPS, nous nous en sommes toutefois tenus au traditionnel générateur de codes par registres à décalage. Ce type de générateur est très bien documenté dans la littérature et toute l'information nécessaire à son développement peut être trouvée dans (ICD-GPS-200C, 2000). De plus, sa mise en oeuvre au sein d'un FPGA est très simple et ne demande pratiquement aucune ressource. Mentionnons simplement qu'il s'agit de deux générateurs polynomiaux, G1 et G2, implémentés à l'aide de deux registres à décalage différents, tel qu'illustré à la Figure 3.35. Les générateurs polynomiaux sont les suivants :

$$\begin{aligned} G1 &= 1 + X^3 + X^{10} \\ G2 &= 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10} \end{aligned} \quad (3.48)$$

Les cases des deux registres doivent être initialisées à 1 au démarrage. Afin de générer un code PRN en particulier, deux branchements sont effectués à l'aide d'un sélecteur à la sortie du registre à décalage G2. Ces branchements sont détaillés dans (ICD-GPS-200C, 2000). Un

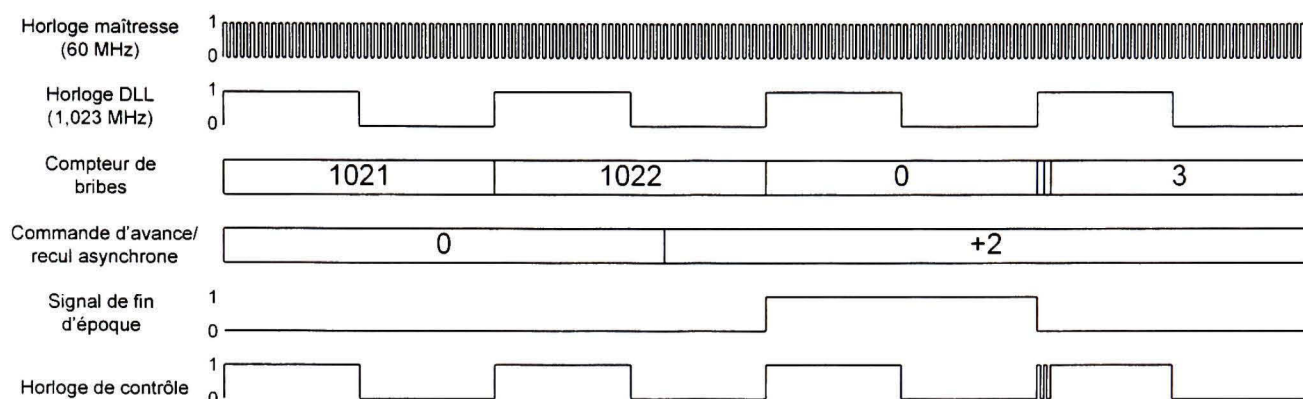
total de 37 codes PRN, appelés codes de Gold, peuvent ainsi être générés, dont cinq sont réservés pour des usages particuliers (on n'emploie donc que 32 codes différents).



**Figure 3.35 Générateur de codes pseudo-aléatoires GPS C/A.**

Afin de rendre le générateur de codes fonctionnel, nous avons ajouté un bloc de contrôle au niveau de l'horloge provenant de la DLL. Ce bloc de contrôle est utilisé lors de la phase de recherche des signaux et permet d'avancer d'une ou de deux bribes supplémentaires à chaque époque. Une série de pulses répétitifs sont ainsi générés au front montant de l'horloge de DLL afin de sauter rapidement d'une bricbe à l'autre, tel qu'affiché à la Figure 3.36. Il est aussi possible de reculer d'une bricbe à chaque époque, dans lequel cas le signal d'horloge provenant de la DLL est inhibé sur une période. Il faut noter qu'un compteur a été intégré au contrôleur d'horloge de manière à garder la trace des bribes et ainsi générer un signal de fin d'époque à la fin de chaque séquence de 1023 bribes.





**Figure 3.36** *Démonstration d'une commande d'avance de deux bribes.*

### 3.5.7 Réalisation des multiplicateurs

La multiplication est une opération mathématique qui peut sembler banale dans la vie de tous les jours. Il en est cependant tout autrement au niveau d'un FPGA alors qu'une telle opération doit être réalisée à l'aide de portes logiques. De façon générale, l'opération de multiplication est très gourmande en termes de ressources FPGA puisqu'elle demande beaucoup de portes logiques pour être réalisée. Fort heureusement, certaines familles de FPGA, telles que les Virtex II et IV de Xilinx, sont dotées de multiplicateurs intégrés, ce qui permet de réaliser une économie substantielle de ressources.

Certains FPGA, tels que le Virtex II 3000 employé lors de la phase initiale du projet, possèdent néanmoins un nombre très limité de multiplicateurs dédiés. Pour cette raison, nous avons développé des multiplicateurs spéciaux dont le rôle est de multiplier les signaux en phase et en quadrature (I et Q) aux codes PRN générés. Puisque les codes PRN sont représentés sur 1 bit seulement, la tâche de ces multiplicateurs, appelés « multiplicateurs 1xN bits », n'est en fait rien d'autre que d'inverser les signaux I et Q lorsque nécessaire.

Tableau 3.6  
Correspondance des niveaux logiques aux niveaux des signaux

Niveau logique	Niveau des signaux
1	-1
0	+1

Par convention, un bit de signe égal à 0 représente un nombre positif alors qu'un bit de signe égal à 1 représente un nombre négatif. Il en va de même pour les bribes du code PRN. Un tableau de correspondance présenté dans (ESA, 2006, p. 31) montre sans équivoque cette relation (voir Tableau 3.6). Les signaux entrant dans le multiplicateur 1xN bits doivent donc être inversés lorsque le code est égal à 1. Or, l'inversion d'un signal signé à complément à deux peut être effectuée ainsi :

$$-x = \bar{x} + 1 \quad (3.49)$$

Le multiplicateur 1xN bits se résume donc à un multiplexeur (MUX), un additionneur et un inverseur, tel que présenté à la Figure 3.37.

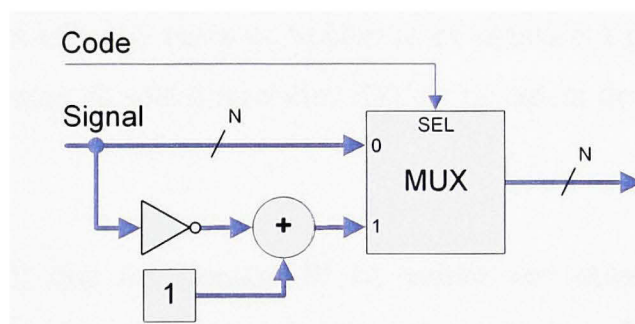
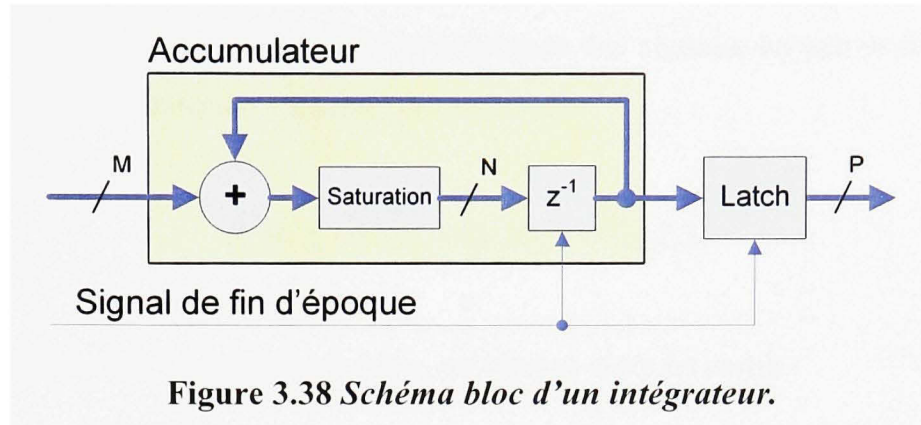


Figure 3.37 Schéma bloc d'un multiplicateur 1xN bits.

### 3.5.8 Réalisation des intégrateurs

Six intégrateurs sont présents dans chacun des canaux de démodulation. D'un point de vue numérique, ces composantes ne sont en fait rien d'autre que des accumulateurs équipés d'une mémoire *latch*. Puisque les intégrateurs sont déchargés à la fin de chaque époque du code C/A (i.e. que leur contenu est copié puis réinitialisé à 0), on leur confère souvent le nom anglais *Integrate and Dump* (I&D). Les intégrateurs réalisés dans le cadre de ce projet sont présentés à la Figure 3.38.



Afin d'éviter qu'il n'y ait de retournement, nous avons inclus un mécanisme de saturation au sein des accumulateurs. Il n'est toutefois pas souhaitable d'entrer en saturation car cela introduit une certaine instabilité au niveau de la DLL. La dimension des accumulateurs (en bits) doit donc être déterminée de manière à minimiser les risques de saturation. D'un autre côté, on doit limiter la taille de ces accumulateurs de manière à permettre l'implémentation d'un maximum de canaux au sein d'un même FPGA. Le calcul de la taille des accumulateurs est décrit ci-dessous.

Tout d'abord, on sait que les signaux IF en entrée des canaux de démodulation sont représentés sur 4 bits signés à complément à deux. Puisqu'ils sont multipliés par des répliques sur 4 bits signés de la porteuse, on obtient donc des signaux  $I$  et  $Q$  de dimension égale à  $M = 7$  bits (et non pas 8 car le bit de signe est dupliqué lors de la multiplication de deux nombres signés à compléments à deux). Les signaux avant intégration sont donc représentés par un nombre entier dont la plage est  $[-63, +63]$ . Si on suppose un signal GPS fort, c'est-à-dire que le ratio  $C/N_0 = 45$  dB Hz, on obtient alors un SNR en entrée égal à (on suppose que  $B_n = 22,3$  MHz, soit la largeur de bande de la tête RF) :

$$SNR = \frac{C}{N_0} \cdot \frac{1}{B_n} = 45 \text{ dB Hz} - 10 \log_{10} (22,3 \times 10^6 \text{ Hz}) = -28,5 \text{ dB} \quad (3.50)$$

Sachant que l'amplitude maximale d'un signal GPS + bruit en entrée des intégrateurs ( $A_n$ ) est de 63, on peut alors estimer l'amplitude d'un signal GPS seul ( $A_s$ ) grâce à la règle du pouce



suivante (on suppose qu'un AGC ajuste la puissance des signaux en entrée de manière à ce que  $A_n$  soit égal à une déviation standard des signaux) :

$$A_s \approx A_n \sqrt{SNR} = 63 \cdot 10^{-28.5 \cdot 20} = 2,37 \quad (3.51)$$

En fixant la période des intégrateurs à 1 ms, on obtient alors en sortie :

$$A_{out} = T f_s A_s = 10^{-3} \text{ s} \cdot 60 \times 10^6 \text{ Hz} \cdot 2,37 = 142,2 \times 10^3 \quad (3.52)$$

où :

- $A_{out}$  est l'amplitude des signaux en sortie des intégrateurs
- $T$  est la période d'intégration
- $f_s$  est la fréquence d'échantillonnage du système

L'intégrateur doit donc posséder un nombre de bits  $N$  minimal égal à :

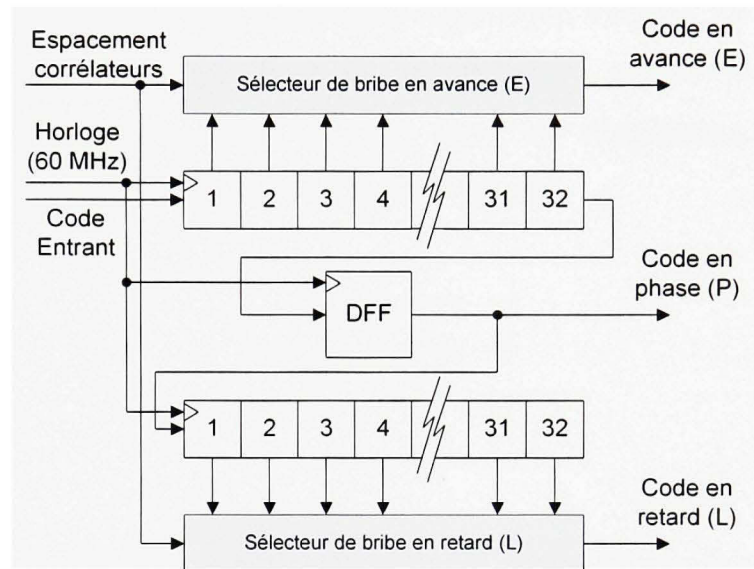
$$N = \log_2 (A_{out}) + 1 = \log_2 (142,2 \times 10^3) \approx 18 \text{ bits} \quad (3.53)$$

Par mesure de précaution, nous avons ajouté deux bits supplémentaires à l'accumulateur, pour un total de  $N = 20$ . Il faut toutefois noter que seuls les 16 bits les plus significatifs sont conservés afin d'être transmis au microcontrôleur. Ceci permet d'économiser des ressources au niveau du FPGA tout en simplifiant les opérations réalisées au niveau du microcontrôleur (qui n'opère que sur 32 bits).

### 3.5.9 Mise en place de corrélateurs à espacement variable

Il est intéressant pour un récepteur de navigation d'avoir accès à des corrélateurs à espacement variable. Ceci permet d'élargir la plage de recherche lors de l'acquisition des signaux tout en offrant un maximum de précision lors de la poursuite. Vu la fréquence

d'échantillonnage élevée du récepteur développé, ce type de corrélateur a pu être mis en place très facilement via l'implémentation de deux registres à décalage et d'une bascule D (DFF – *D Flip-Flop*), tel qu'illustré à la Figure 3.39.



**Figure 3.39** *Mise en place de corrélateurs à espacement variable.*

Grâce à un tel mécanisme, il est possible d'ajuster l'espacement entre les corrélateurs avec une résolution d'environ  $R = 1/60^e$  de brique, tel que :

$$R = \frac{f_c}{f_s} = \frac{1,023 \text{ MHz}}{60 \text{ MHz}} = 0,01705 \text{ brique} \quad (3.54)$$

où :

- $f_c$  est le taux du code C/A
- $f_s$  est la fréquence d'échantillonnage du système

L'utilisation de deux registres à décalage de 32 bits permet d'offrir un espacement maximal de  $\pm 0,55$  brique entre les corrélateurs, ce qui est suffisant lors de l'acquisition et de la phase initiale de poursuite.

### 3.6 Réalisation et test du contrôleur automatique de gain (AGC)

L'amplitude des signaux en entrée d'un récepteur de navigation varie continuellement dans le temps. Il est donc nécessaire d'ajuster le gain en entrée du CAN à l'aide d'un AGC de manière à profiter de toute la plage dynamique disponible. Puisque les signaux GPS reçus sont situés en dessous du niveau de bruit, on considère généralement les signaux IF numérisés par le CAN comme étant un bruit blanc de distribution gaussienne. Or, la déviation standard, ou valeur RMS, d'un tel bruit se calcule :

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} x^2(n-i)} \quad (3.55)$$

où :

- $\sigma$  représente la déviation standard d'un signal  $x(n)$  à distribution gaussienne pour  $N$  échantillons consécutifs

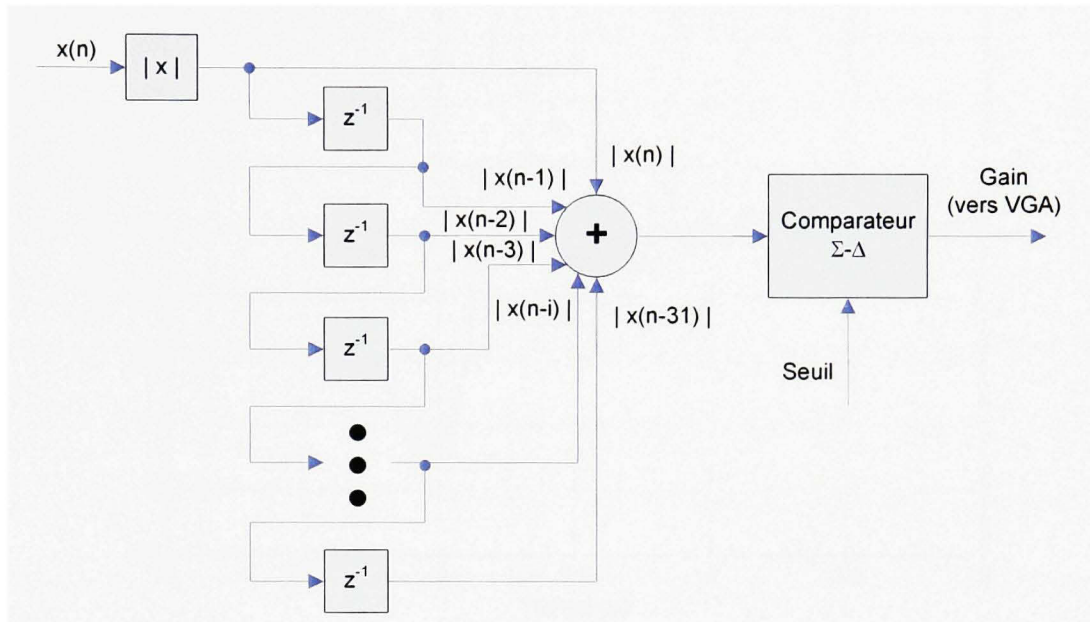
L'AGC développé dans le cadre de ce projet s'affaire dans un premier temps à estimer la valeur RMS du signal IF numérisé. Cette estimation est ensuite employée afin de commander adéquatement un amplificateur à gain variable (VGA – *Variable Gain Amplifier*) situé en amont du CAN (voir Figure 3.23). Une simplification a toutefois été apportée à l'équation (3.55) de manière à minimiser les ressources employées au sein du FPGA :

$$\sigma \approx \frac{1}{N} \sum_{i=0}^{N-1} |x(n-i)| \quad (3.56)$$

Un total de  $N = 32$  échantillons est employé afin d'estimer  $\sigma$ . Le résultat est ensuite transféré à un mécanisme d'ajustement du gain de type  $\Sigma\text{-}\Delta$  de manière à incrémenter le gain du VGA à chaque fois que  $\sigma$  est inférieur à un seuil prédéterminé, et vice-versa. Ce seuil a été fixé de façon empirique de manière à maximiser le ratio SNR des signaux GPS démodulés. Ceci revient à ajuster la puissance des signaux en entrée de manière à ce que leur déviation

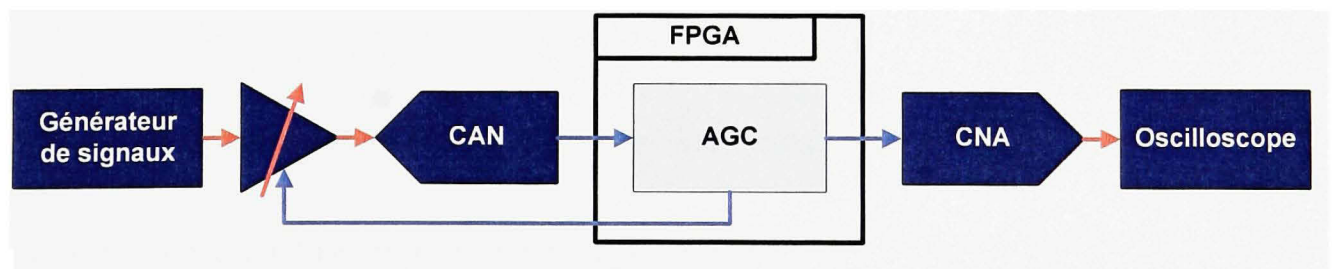


standard ( $\sigma$ ) corresponde à la plage dynamique du CAN. La Figure 3.40 illustre l'architecture interne de l'AGC.

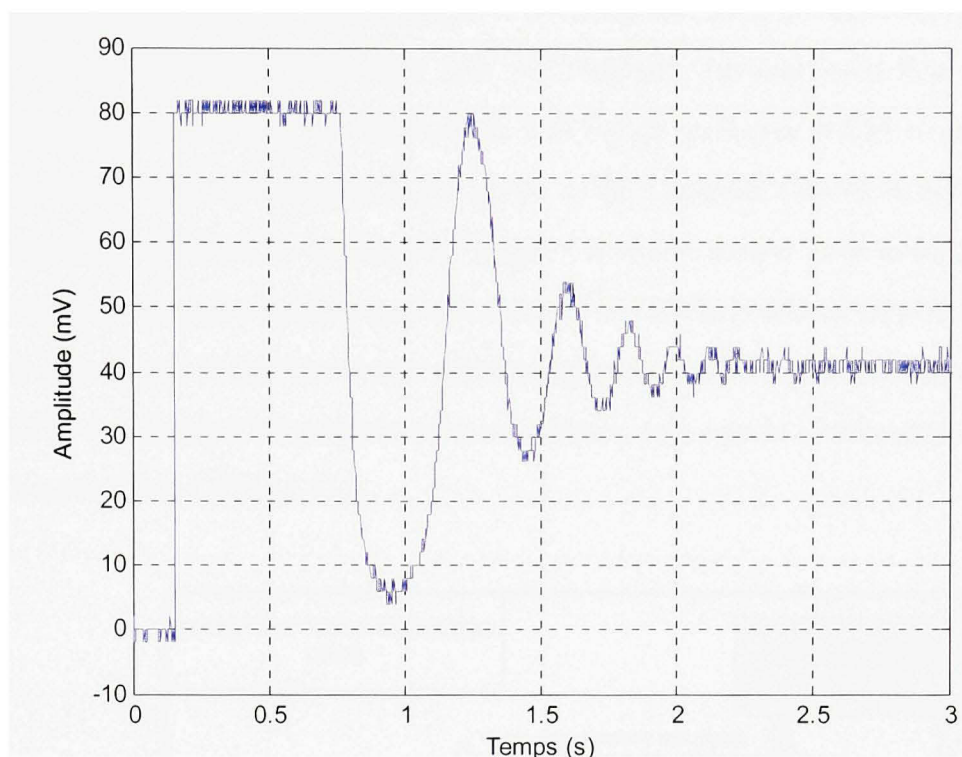


**Figure 3.40** Schéma bloc du contrôleur de gain automatique.

L'AGC a dû être testé séparément du reste du récepteur afin de s'assurer qu'il donne de bons résultats en temps réel. Pour y parvenir, une excitation de type échelon a été produite grâce à un générateur de signaux avant d'être injectée dans la carte de développement via un CAN. L'utilisation d'une carte fille comportant une série de CNA a ensuite permis de transférer le signal traité à un oscilloscope de manière à analyser la réponse temporelle de l'AGC (voir Figure 3.41). La réponse temporelle de l'AGC est présentée à la Figure 3.42.



**Figure 3.41** Schéma de branchements pour le test en temps réel de l'AGC.



**Figure 3.42 Réponse temporelle de l'AGC à un stimulus de type échelon.**

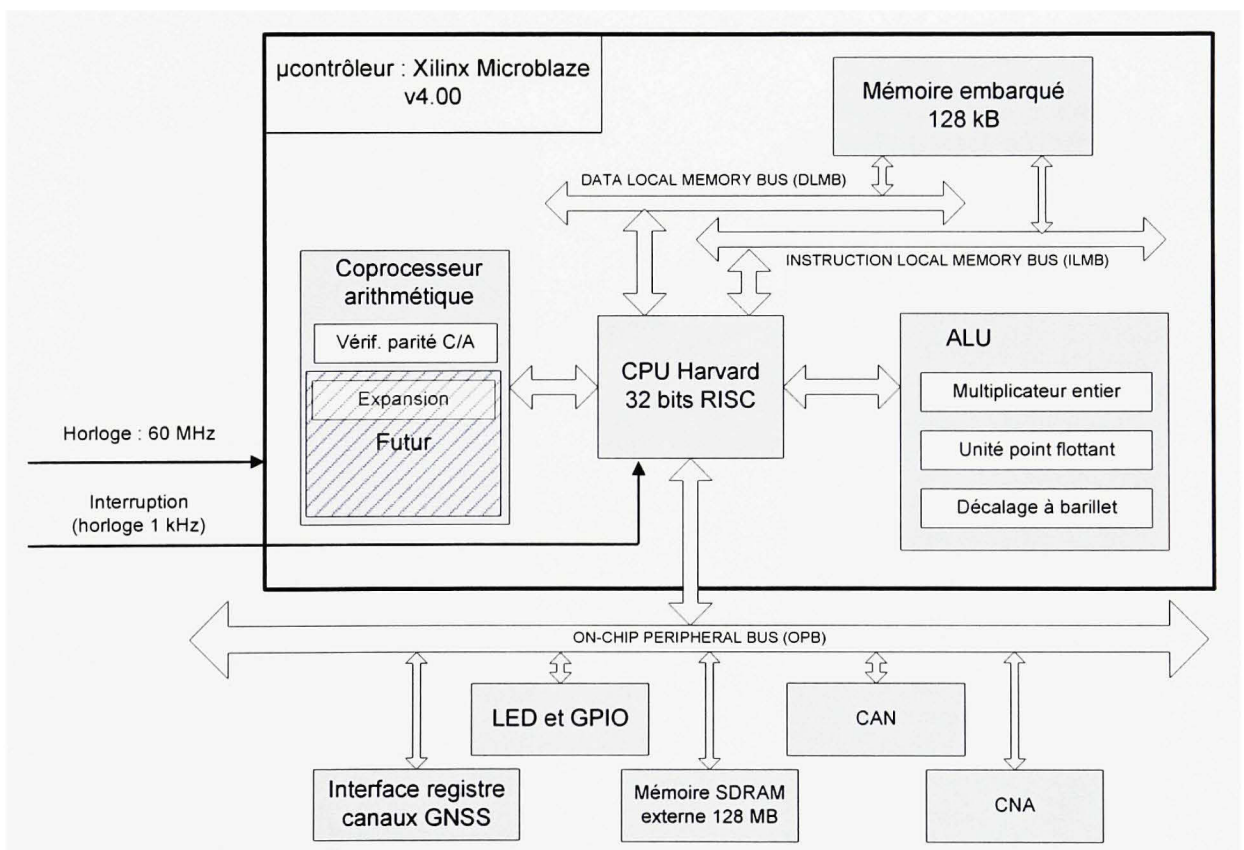
En examinant la Figure 3.42, on remarque que le gain du VGA sature initialement sur une période d'un peu plus de 0,5 s, suivi d'une série d'oscillations qui durent près de 1,5 s. Le temps total de stabilisation de l'AGC à une excitation de type échelon se situe donc autour de 2 s et est principalement dû au VGA employé, dont la réponse temporelle est lente. Un aussi grand délai de stabilisation ne dérange aucunement en présence de bruit blanc mais peut s'avérer problématique lorsque des interférences s'ajoutent aux signaux GPS reçus. Un VGA de plus haute performance serait donc nécessaire de manière à compenser plus rapidement les fluctuations générées par d'éventuelles interférences.

### 3.7 Implémentation d'un microcontrôleur embarqué

#### 3.7.1 Architecture interne du microcontrôleur

Un microcontrôleur Microblaze v4.00 de Xilinx a été implanté de façon logicielle dans le FPGA en utilisant une partie des ressources de ce dernier (l'utilisation des ressources est

abordée à la section 3.14). Ce microcontrôleur possède une unité de traitement centrale (CPU - *Central Processing Unit*) basée sur une architecture Harvard modifiée de type RISC (*Reduced Instruction Set CPU*). Il est doté de 128 kB de mémoire RAM embarquée, dont la moitié est présentement utilisée par le programme du récepteur GNSS. Une unité de calcul arithmétique et logique (ALU - *Arithmetic Logic Unit*) est fournie avec le microcontrôleur et permet de réaliser certaines opérations, telles que le calcul de nombres en point flottant. Nous avons aussi développé un coprocesseur arithmétique en VHDL, que nous avons par la suite greffé au CPU de manière à accélérer certains calculs, tels que la vérification de la parité du message de navigation GPS L1 C/A.

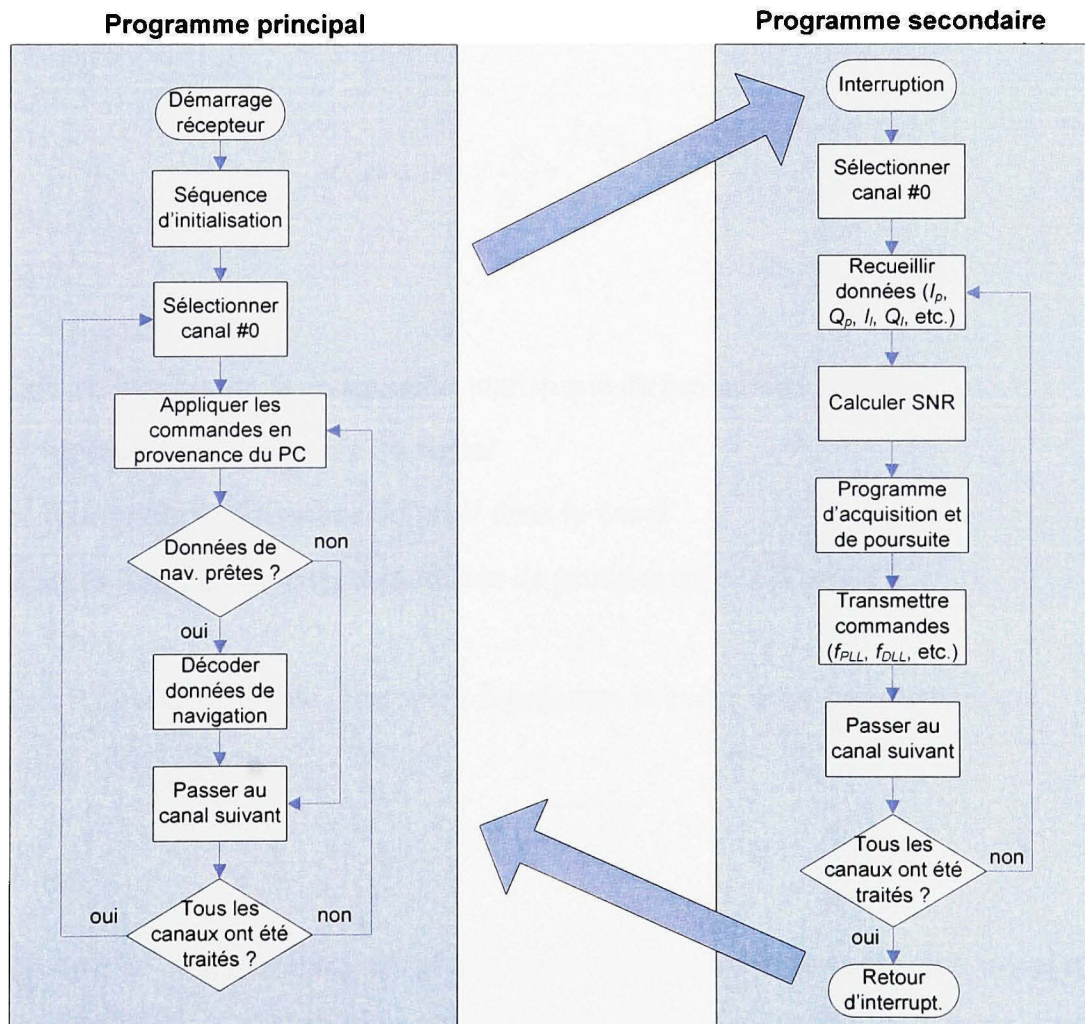


**Figure 3.43** Architecture générale du microcontrôleur embarqué.

Des pilotes ont été greffés au microcontrôleur afin d'accéder à une variété de ressources externes via un bus OPB (*On-Chip Peripheral Bus*) : CAN, CNA, mémoire SDRAM externe, diodes électroluminescentes (DEL), connecteur numérique GPIO (*General Purpose Input/Output*) et interface registre servant à accéder aux canaux de démodulation GNSS.



Le microcontrôleur embarqué possède deux programmes principaux, tels qu'affichés à la Figure 3.44. Un premier programme s'occupe d'initialiser le récepteur GNSS pour ensuite décoder en arrière-plan les messages de navigations reçus des satellites. Il s'occupe aussi d'appliquer les commandes provenant du PC (ex. : choix des satellites, redémarrage de l'acquisition, etc.). Un second programme est quant à lui appelé à toutes les millisecondes par le biais d'une interruption afin de réaliser l'acquisition et la poursuite des signaux GNSS. Plus de détails sont apportés à ce sujet à la section 3.9. Une partie du second programme sert aussi à mesurer le ratio SNR des signaux et est abordée à la section suivante. Mentionnons que les deux programmes traitent les canaux GPS de façon séquentielle (i.e. à tour de rôle), en débutant toujours par le canal #0 et en terminant par le canal #11.



**Figure 3.44** Ordinogramme des programmes du microcontrôleur embarqué.

### 3.8 Estimation du ratio SNR

Un certain nombre de méthodes ont été mises au point afin d'estimer le ratio SNR au sein des récepteurs de navigation. Une méthode particulièrement simple à implémenter est présentée dans (Parkinson et Spilker, 1996, p. 390-392). Le problème toutefois avec une telle approche est qu'elle nécessite qu'il y ait synchronisation avec le message de navigation afin de pouvoir estimer le ratio SNR. Or, il peut être utile d'avoir accès à cette estimation dès l'acquisition des signaux afin d'en faciliter le processus. Pour cette raison, nous avons entrepris de développer notre propre méthode d'estimation du SNR.

On sait que les signaux en sortie des corrélateurs suivent une distribution de probabilité de Rice définie telle que (Kaplan, 1996) :

$$f(x|v, \sigma) = \frac{x}{\sigma^2} e^{-\left(\frac{x^2+v^2}{2\sigma^2}\right)} I_0\left(i \frac{xv}{\sigma^2}\right) \quad (3.57)$$

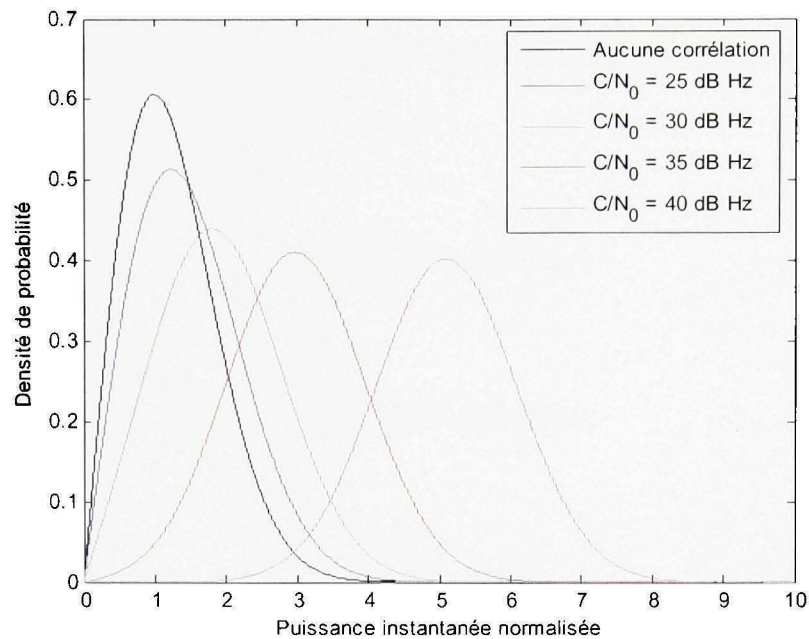
où :

- $f(x|v, \sigma)$  représente la distribution statistique de probabilité de Rice
- $v^2$  représente la puissance du signal
- $\sigma^2$  représente la puissance du bruit dans le canal
- $I_0$  est la fonction de Bessel modifiée de premier type et d'ordre 0

Il faut noter ici que la variable aléatoire  $x$  représente la sortie d'un corrélateur, soit :

$$x = \sqrt{I^2 + Q^2} \quad (3.58)$$

où  $I$  et  $Q$  sont les composantes en phase et en quadrature à la sortie des intégrateurs. La Figure 3.45 présente la distribution statistique en sortie des corrélateurs pour divers  $C/N_0$ , une puissance de bruit normalisée ( $\sigma^2 = 1$ ) et une bande de bruit de 1 kHz.



**Figure 3.45** *Distribution statistique en sortie des corrélateurs pour divers  $C/N_0$ .*

Le paramètre  $\sigma^2$  s'estime assez facilement, car pour un canal GPS non corrélé, le paramètre  $v^2 \approx 0$ . Par conséquent, la distribution statistique du bruit se réduit à une distribution de Rayleigh décrite par la fonction suivante (Wikipedia, 2007d) :

$$f(x|\sigma) = \frac{x}{\sigma^2} e^{\left(-\frac{x^2}{2\sigma^2}\right)} \quad (3.59)$$

Ceci est très intéressant au sens où la distribution de Rayleigh est bien connue et qu'elle se caractérise facilement. Par exemple, l'estimation au maximum de vraisemblance du paramètre  $\sigma^2$  est donnée par (Wikipedia, 2007b):

$$\sigma^2 = \frac{1}{2N} \sum_{i=1}^N x_i^2 \quad (3.60)$$



Une fois le paramètre  $\sigma^2$  estimé, on peut alors déterminer  $\upsilon^2$  d'une manière relativement semblable en partant de la définition du second moment statistique ( $\mu_2$ ) de la distribution de Rice, qui s'exprime (Wikipedia, 2007d) :

$$\mu_2 = \frac{1}{N} \sum_{i=1}^N x_i^2 = 2\sigma^2 + \upsilon^2 \quad (3.61)$$

Il s'agit alors d'isoler  $\upsilon^2$  en évaluant l'expression suivante :

$$\upsilon^2 = \frac{1}{N} \sum_{i=1}^N x_i^2 - 2\sigma^2 \quad (3.62)$$

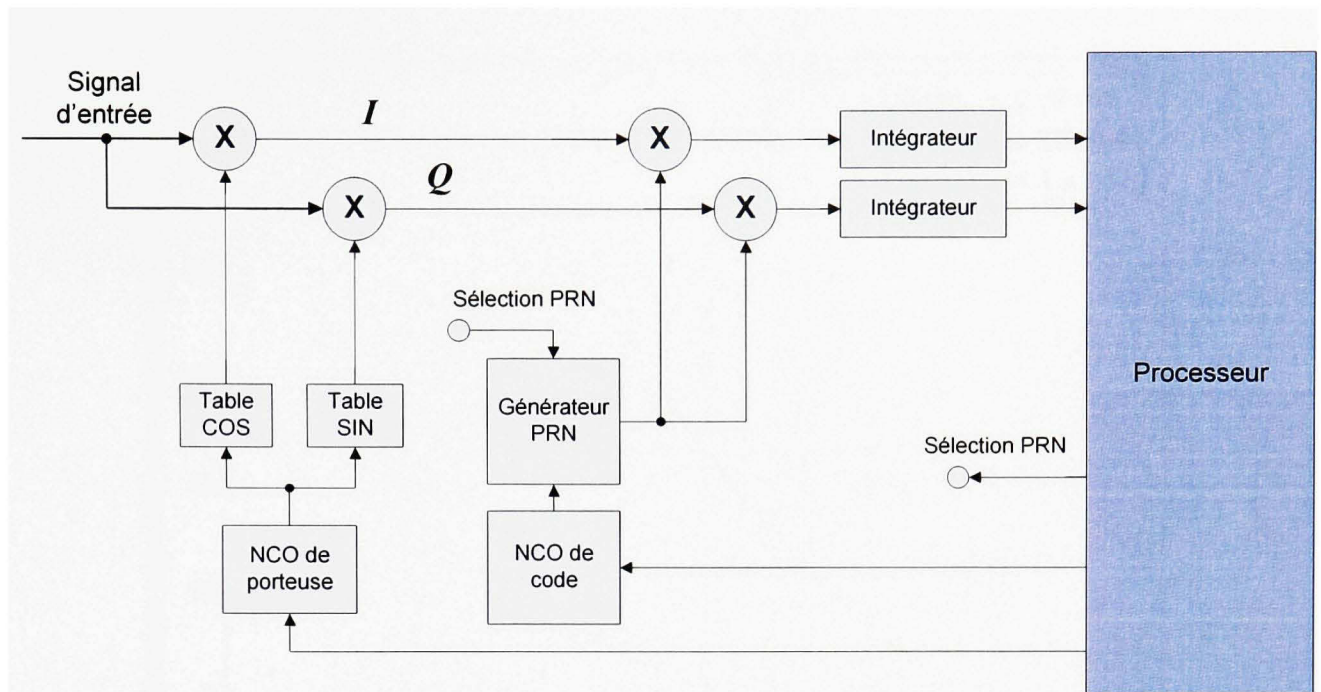
Finalement, le ratio SNR des signaux peut être calculé, tel que :

$$SNR = \frac{\upsilon^2}{\sigma^2} \quad (3.63)$$

Le ratio  $C/N_0$  peut aussi être estimé en dB, où  $T$  représente la période d'intégration des corrélateurs (Ward, 1996) :

$$C/N_0 = 10 \log_{10}(SNR) + 10 \log_{10}(T) \quad (3.64)$$

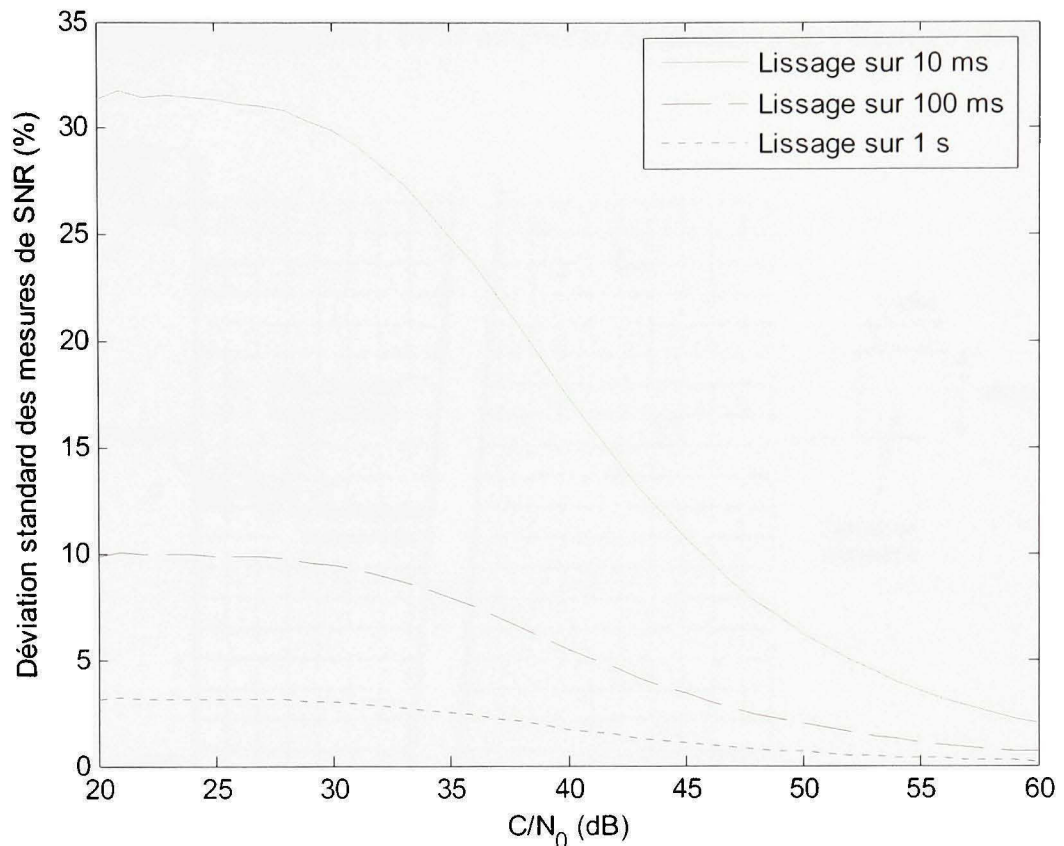
Afin de rendre concrète cette technique d'estimation du SNR, nous avons élaboré un 13<sup>e</sup> canal au sein du FPGA, appelé canal de calibration, et dont l'unique but est de mesurer la puissance du bruit  $\sigma^2$  (voir Figure 3.46). Pour y parvenir, nous avons dû faire en sorte qu'aucune corrélation ne puisse avoir lieu dans ce canal de calibration, ce qui est accompli en changeant la fréquence Doppler et le code PRN du canal à toutes les époques. Grâce à l'équation (3.60) et à un lissage sur dix secondes, il est alors possible de mesurer avec une grande précision la puissance du bruit en sortie des corrélateurs.



**Figure 3.46 Schéma bloc du canal de calibration.**

Afin d'estimer le ratio SNR, il s'agit ensuite d'estimer la puissance des signaux  $\psi^2$  en sortie des corrélateurs des véritables canaux de démodulation. Ceci est accompli grâce à l'équation (3.62). Il ne reste plus qu'à appliquer l'équation (3.63) pour obtenir une estimation du SNR. Il faut noter qu'un autre lissage sur une seconde est réalisé de manière à améliorer la précision des résultats.

Des tests par simulation ont permis de montrer que pour un lissage sur une seconde, la déviation standard des estimations de SNR demeure en dessous de 3%. Ces résultats sont aussi valides pour des ratios SNR très faibles (voir Figure 3.47). Un autre test est présenté au chapitre 6 dans le but de valider le fonctionnement de cette méthode d'estimation du SNR en temps réel.



**Figure 3.47** *Déviati   standard du ratio SNR due au bruit dans le canal.*

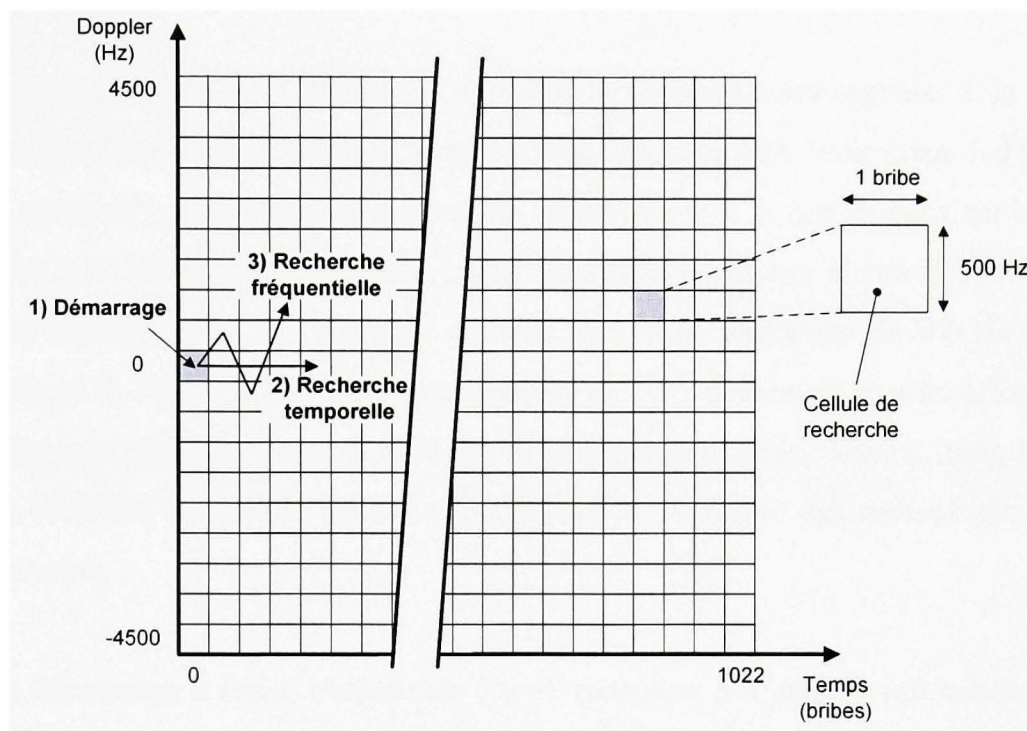
### 3.9 Mise en   uvre d'un m  canisme d'acquisition des signaux

#### 3.9.1 Principes g  n  raux sur l'acquisition des signaux

Avant de passer en mode de poursuite, un r  cepteur GPS doit d'abord faire l'acquisition des signaux. Concr  tement, cela exige de se synchroniser avec le code d'  talement tout en d  terminant le d  calage Doppler de signaux. Deux inconnues, une en temps et l'autre en fr  quence, doivent   tre ainsi r  solv  es. Or, la dimension du domaine de recherche, souvent per  u comme un pav      deux dimensions (voir Figure 3.48), varie beaucoup d'une application    l'autre. Il en va de m  me pour la taille de chacune des cellules de recherche. Pour une application typique    basse dynamique, qu'on pense    un utilisateur    pied ou en voiture par exemple, on peut toutefois s'en tenir    une plage de recherche fr  quentielle de  $\pm 4$  kHz telle que vue au chapitre 2. Il est aussi important de tenir compte de la d  rive de



l'horloge du récepteur. Ceci est accompli en ajoutant environ  $\pm 1,5$  kHz à la plage de recherche pour chaque tranche de 1 PPM au niveau de la dérive de l'horloge du récepteur.



**Figure 3.48** *Procédure d'acquisition séquentielle des signaux.*

La dimension des cellules de recherche dépend généralement de l'environnement dans lequel le récepteur doit opérer. Ainsi, pour un environnement très difficile, tel qu'à l'intérieur des bâtiments par exemple, on favorisera une recherche lente sur des cellules d'une dizaine de Hertz par un demi ou un quart de bribe seulement. Dans le cas du récepteur développé, nous avons préféré réaliser une acquisition rapide en choisissant des cellules de 500 Hz x 1 bribe, pour un total de 19 x 1023 cellules. La dimension finale du pavé de recherche est donc de  $\pm 4,5$  kHz par 1023 bribes.

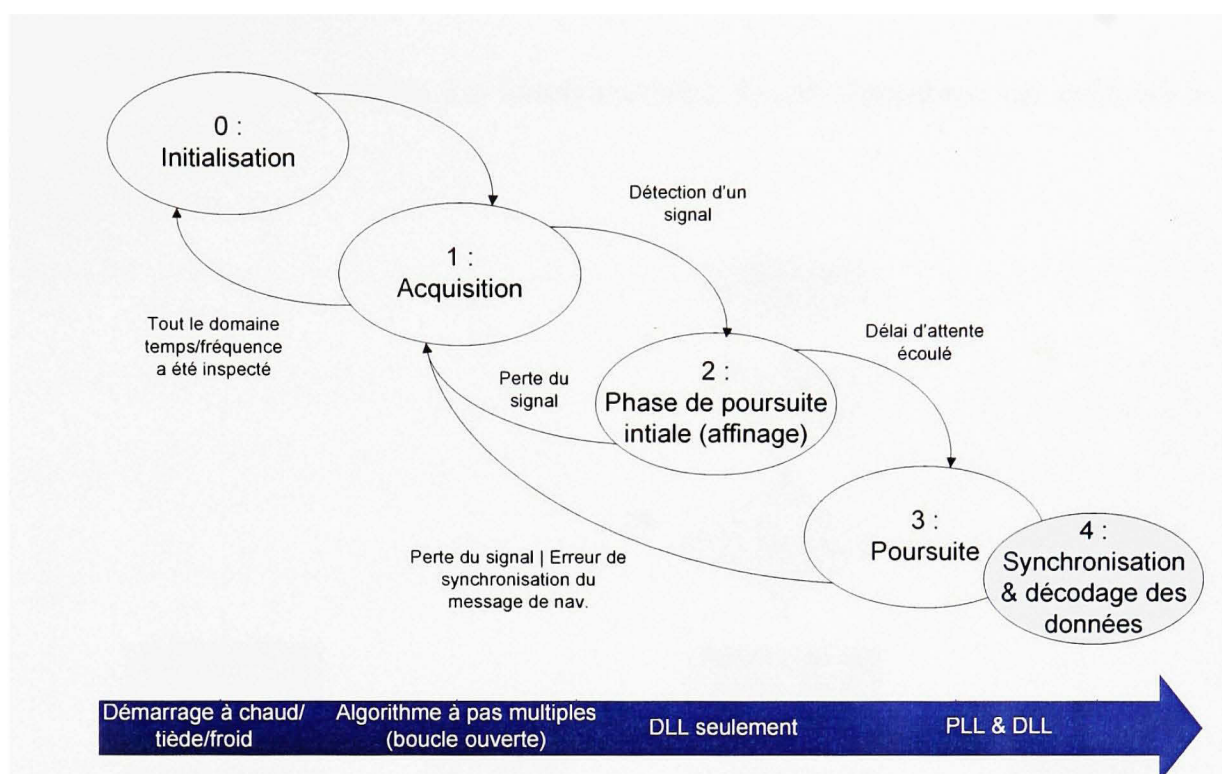
Du point de vue du mécanisme d'acquisition, nous avons opté pour une recherche de type séquentielle des signaux. Ce type de recherche, comparativement à une recherche en parallèle par transformée rapide de Fourier (FFT – *Fast Fourier Transform*), a le désavantage d'être lente puisque toutes les cellules doivent être inspectées une après l'autre afin de détecter un signal. Une recherche de type séquentielle est toutefois beaucoup plus facile à implémenter et

exige moins de ressource au niveau d'un FPGA qu'un algorithme par FFT. Pour plus d'information quant à l'acquisition par FFT, le lecteur est invité à consulter (Chavagnieux, 2006).

Le mécanisme d'acquisition développé démarre la recherche des signaux à la case (0,0), c'est-à-dire pour un Doppler nul et la première bribe du code C/A (voir étape 1, Figure 3.48). Les 1023 cases de la même ligne sont ensuite inspectées tour à tour en décalant le code C/A d'une bribe à la fois (étape 2, Figure 3.48). Advenant qu'aucun signal ne soit détecté, le mécanisme retourne alors à la première colonne tout en incrémentant de 500 Hz le décalage Doppler (étape 3, Figure 3.48). Un autre décalage de 1023 bribes est ensuite effectué, suite à quoi le décalage Doppler passe à -500 Hz si aucun signal n'est détecté, puis à 1000 Hz, -1000 Hz, 1500 Hz et ainsi de suite. Le mécanisme de détection des signaux est abordé à la section suivante.

Lors d'un démarrage à froid, c'est-à-dire que le récepteur n'a aucune information quant au temps ou à la position de l'utilisateur, ou encore que les éphémérides et l'almanach ne sont pas valides, le mécanisme d'acquisition doit en plus se charger de vérifier tour à tour pour chacun des 32 codes PRN valides du GPS. Lors d'un démarrage à tiède ou à chaud, la visibilité des satellites GPS est toutefois connue, ce qui permet d'accélérer le processus de recherche des signaux. Plus de détails sont apportés sur ce sujet au chapitre 5.

Une fois l'acquisition d'un signal accomplie, on peut alors passer à la phase initiale de poursuite des signaux, ou phase d'affinage (nommée *Pull-in* en anglais). Cette phase est essentielle avant de passer en mode de poursuite étant donné la plage dynamique très étroite de la PLL. Ceci est dû à la faible bande de bruit de la PLL, qui limite la capacité de recherche à 10 ou 30 Hz, tous ordres de boucle confondus. Vu la taille de 500 Hz de cellules de recherche, il est donc impératif de réaliser un premier dégrossissage. Cette étape est abordée en détail à la section 3.9.2.



**Figure 3.49** Diagramme d'états du programme d'acquisition et de poursuite.

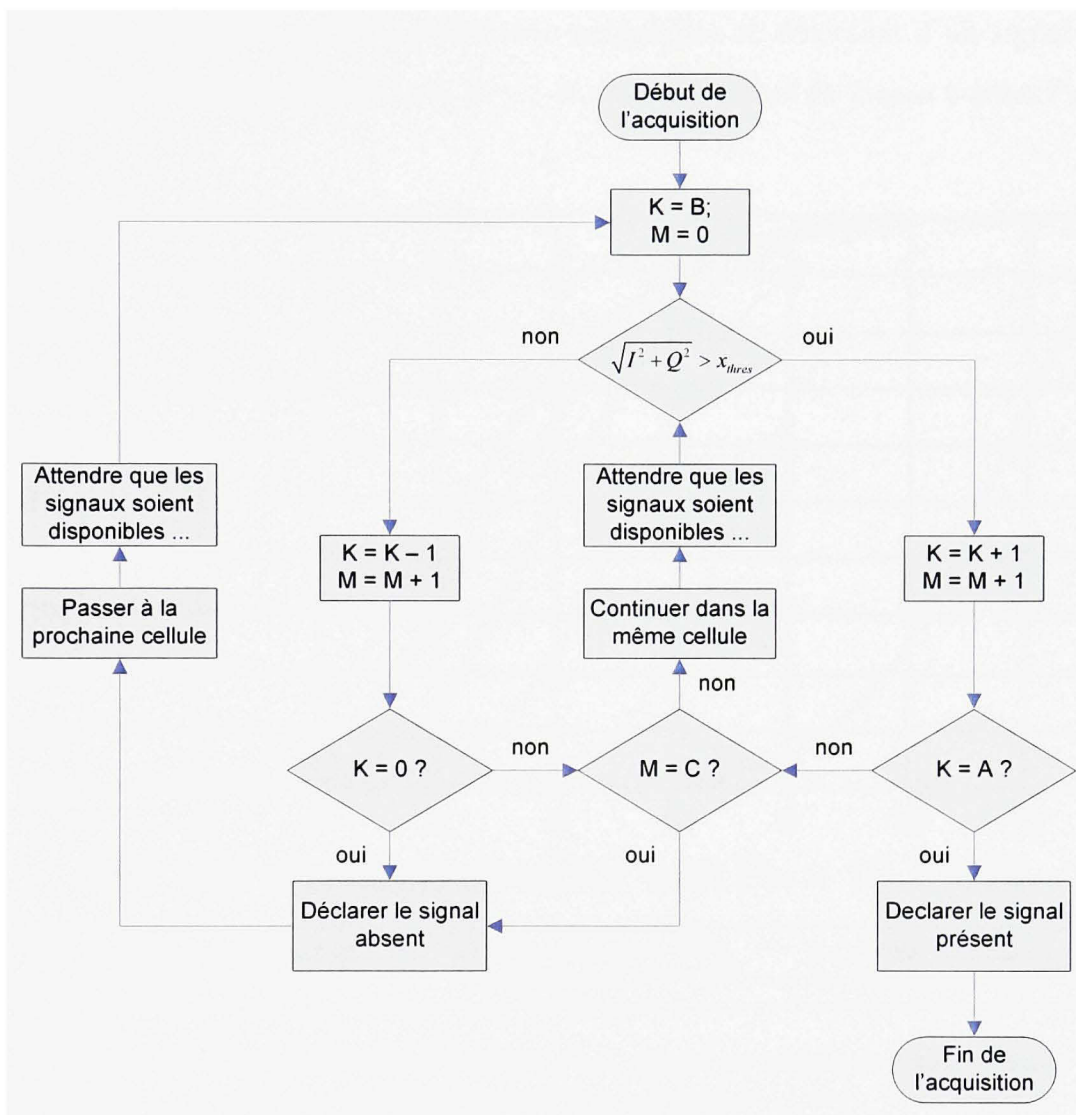
La Figure 3.49 présente chacun des états du programme d'acquisition et de poursuite. Les boucles en fonction à chacune des étapes y sont aussi affichées. On peut y voir que les étapes d'affinage et de poursuite possèdent chacune un retour sur l'étape d'acquisition en cas de perte des signaux ou de fausse acquisition. Un seuil minimal correspondant à un  $C/N_0$  de 30 dB Hz a été fixé dans les deux cas afin de vérifier la présence des signaux. Ce seuil a été fixé de façon arbitraire et pourrait éventuellement être abaissé de manière à poursuivre des signaux de plus faible puissance. On note qu'un retour au mode d'acquisition peut aussi être forcé lors de la poursuite advenant qu'une erreur soit détectée lors de la synchronisation au message de navigation. Ce mécanisme de synchronisation est abordé un peu plus loin.

### 3.9.2 Détection des signaux lors de l'acquisition

Plusieurs mécanismes sont proposés dans (Kaplan, 1996, p. 193-204) afin d'accélérer et d'améliorer la probabilité de détection des signaux lors de l'acquisition. Nous avons choisi parmi ceux-ci un algorithme à pas variables (appelé *Tong detector* ou *variable dwell time* en anglais) qui permet d'offrir un excellent compromis entre l'efficacité de recherche et la



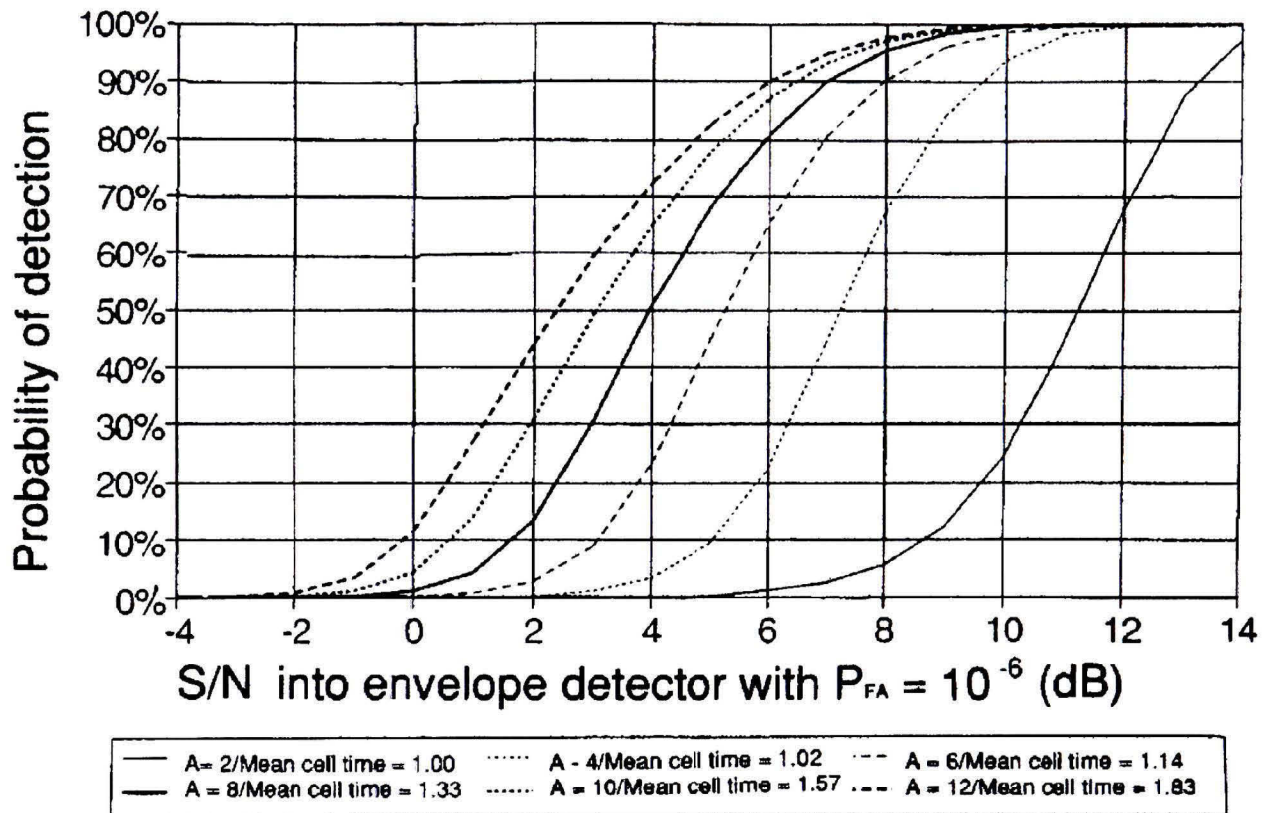
complexité d'implémentation. Le fonctionnement de cet algorithme est présenté à Figure 3.50.



**Figure 3.50** *Algorithme de détection à pas variables des signaux.*

L'algorithme à pas variables est un mécanisme de détection de type séquentiel car il doit inspecter les cellules de recherche une à une. Plusieurs tentatives de détection sont effectuées à chacune des cellules en comparant la sortie des corrélateurs à un seuil prédéterminé  $x_{thres}$ . Un compteur ( $K$ ) est incrémenté à chaque tentative réussie (i.e.  $\sqrt{I^2 + Q^2} > x_{thres}$ ) ou décrémenté dans le cas contraire. Un nombre minimal de tentatives ( $B$ ) est effectué avant de rendre un verdict final. Advenant que le compteur tombe à 0, ou encore qu'un nombre maximal de tentatives ( $C$ ) ait été effectué, le signal est alors déclaré non présent et il y a

passage à la cellule de recherche suivante. Si toutefois le compteur réussit à atteindre le seuil de détection ( $K = A$ ), le signal est déclaré présent et le processus d'acquisition se termine. La Figure 3.51 (Kaplan, 1996, p. 203) présente la probabilité de détection d'un signal en sortie du détecteur pour diverses valeurs de  $A$ ,  $B = 1$  et une probabilité de fausse alarme  $P_{FA} = 10^{-6}$ .



**Figure 3.51** *Probabilité de détection d'un signal en sortie de l'algorithme de recherche à pas variables*

Tiré de (Kaplan, 1996, p. 203)

Après plusieurs essais en temps réel à l'aide du récepteur développé, notre choix s'est arrêté sur les paramètres suivants qui offraient le meilleur compromis entre le temps de recherche et la capacité de détection des signaux :  $A = 8$ ,  $B = 1$  et  $C = 19$ . Le temps d'intégration a été maintenu à 1 ms et le seuil  $x_{thres}$  a été ajusté de manière à obtenir une probabilité de fausse alarme  $P_{FA} = 10^{-6}$ . D'après la Figure 3.51, le temps moyen écoulé à chaque cellule est donc de 1,33 ms, ce qui donne un temps de recherche maximal d'environ  $19 \times 1023 \times 1,33 \text{ ms} = 25,9 \text{ s}$  (on se rappelle la taille du domaine de recherche qui est  $19 \times 1023$  cellules). On note

aussi une probabilité de détection d'environ 12,5% pour un SNR de 2 dB en sortie des corrélateurs, ce qui équivaut à un  $C/N_0$  de 32 dB Hz d'après l'équation (3.64) :

$$C/N_0 = 10 \log_{10}(SNR) + 10 \log_{10}(T) = 2 \text{ dB} + 10 \log_{10}(10^{-3}) = 32 \text{ dB Hz} \quad (3.65)$$

Cette probabilité est toutefois quasi triplée grâce à une recherche en simultanée sur les trois corrélateurs E, P et L. De cette manière, la probabilité de détection se situe plutôt autour de 30% pour un ratio  $C/N_0$  de 32 dB Hz.

Nous avons fixé le seuil de détection à  $2,04\sigma$  de manière à ce que la probabilité de fausse alarme soit de  $10^{-6}$ , où  $\sigma$  est la valeur RMS du bruit dans les canaux. Or, ce seuil peut être relié à la probabilité de fausse alarme en sortie du mécanisme de détection grâce à l'équation suivante (Kaplan, 1996) :

$$P_{FA} = \frac{\left( \frac{1 - P_{fa}}{P_{fa}} \right)^B - 1}{\left( \frac{1 - P_{fa}}{P_{fa}} \right)^{A+B-1} - 1} \quad (3.66)$$

où  $P_{fa}$  représente la probabilité de fausse alarme par essai simple en sortie des corrélateurs. Cette dernière valeur est représentée par l'aire sous la courbe comprise entre les bornes  $[x_{thres}, \infty[$  d'une distribution de Rayleigh, telle qu'illustrée à la Figure 3.52, et se calcule ainsi (Wikipedia, 2007b) :

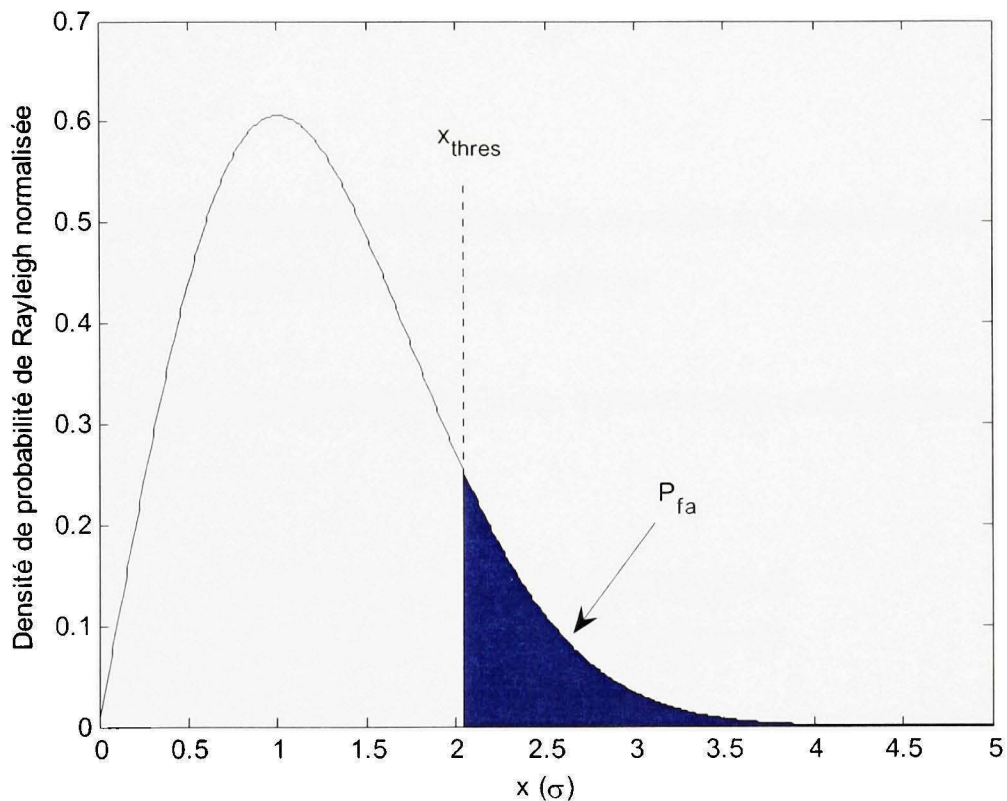
$$P_{fa} = \lim_{x_b \rightarrow \infty} \frac{1}{\sigma^2} \int_{x_{thres}}^{x_b} x e^{\frac{-x^2}{2\sigma^2}} dx \quad (3.67)$$

$$P_{fa} = \lim_{x_b \rightarrow \infty} \left[ e^{\frac{-x^2}{2\sigma^2}} \right]_{x_{thres}}^{x_b} = e^{\frac{-x_{thres}^2}{2\sigma^2}} - \lim_{x_b \rightarrow \infty} e^{\frac{-x_b^2}{2\sigma^2}} = e^{\frac{-x_{thres}^2}{2\sigma^2}} \quad (3.68)$$



En substituant  $x_{thres} = 2,04\sigma$  à l'équation précédente, on obtient alors  $P_{fa} = 0,125$ . On est alors en mesure de déterminer la valeur de  $P_{FA}$  grâce à l'équation (3.66) :

$$P_{FA} = \frac{\left(\frac{1-0,125}{0,125}\right)^1 - 1}{\left(\frac{1-0,125}{0,125}\right)^{8+1-1} - 1} = \frac{6}{7^8 - 1} \approx 10^{-6} \quad \text{C.Q.F.D.} \quad (3.69)$$



**Figure 3.52** *Détermination de la probabilité de fausse alarme par essai simple en sortie des corrélateurs ( $P_{fa}$ ).*

### 3.9.3 Phase initiale de poursuite des signaux

Deux raisons justifient l'ajout d'une étape entre les phases d'acquisition et de poursuite des signaux, la première étant de permettre à la DLL de verrouiller adéquatement. On favorisera donc une large bande de bruit au niveau de la DLL (2 Hz) ainsi que des corrélateurs

suffisamment éloignés ( $\pm 0,5$  bribe) lors de la phase de poursuite initiale. La seconde raison qui justifie une telle étape est qu'on doit affiner la fréquence Doppler obtenue lors de l'acquisition pour que la PLL puisse verrouiller. La technique la plus populaire pour y parvenir consiste à employer une boucle à verrouillage de fréquence (FLL – *Frequency Lock Loop*) dont la plage d'opération en fréquence est beaucoup plus large que celle d'une PLL. Le discriminateur de FLL au maximum de vraisemblance possède la fonction suivante (Kaplan, 1996):

$$\Delta f \approx \frac{\text{atan}(\text{cross} / \text{dot})}{2\pi T} \quad (3.70)$$

où :

- $\Delta f$  est l'erreur de fréquence entre la porteuse en entrée et la porteuse régénérée
- $T$  représente la période d'intégration des signaux

Les éléments *cross* et *dot* se calculent quant à eux à partir de deux échantillons successifs des signaux  $I_p$  et  $Q_p$  :

$$\begin{aligned} \text{cross} &= I_p(n-1) \cdot Q_p(n) - I_p(n) \cdot Q_p(n-1) \\ \text{dot} &= I_p(n) \cdot I_p(n-1) + Q_p(n) \cdot Q_p(n-1) \end{aligned} \quad (3.71)$$

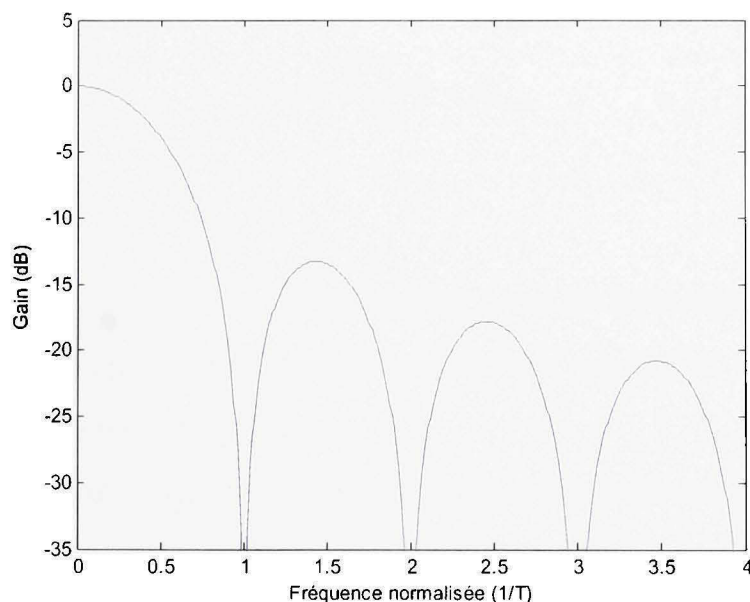
Ce discriminateur demeure toutefois une approximation et par conséquent une erreur est présente à sa sortie. Cette erreur est fonction de l'accélération radiale, c'est-à-dire l'accélération relative entre les satellites de navigation et le récepteur. L'affinage en fréquence devient donc incorrect en présence de dynamiques élevées. Un autre problème propre au discriminateur de FLL est que sa plage dynamique est limitée à  $\pm 1/4T$ , ce qui équivaut à une plage maximale de  $\pm 250$  Hz pour une période d'intégration de  $T = 1$  ms. Bien que la plage dynamique de la DLL ne soit que de  $\pm 2$  Hz au niveau du code, ceci équivaut à une plage d'environ  $\pm 3$  kHz au niveau de la porteuse, soit 12 fois la plage de la FLL. Résultat, il est très fréquent de voir la DLL verrouiller sur une des cellules en fréquence

adjacentes (ce qui se produit fréquemment lorsque le signal est fort), forçant du coup la FLL à verrouiller sur une « harmonique » plutôt que sur la porteuse. Puisque la PLL ne voit aucune différence entre ces harmoniques et la véritable porteuse, il faudra généralement attendre l'étape de synchronisation aux bits de données pour s'apercevoir de l'erreur, ce qui ralentit considérablement le processus d'acquisition des signaux.

Ce qu'on appelle « harmonique » est en fait une fréquence de battement produite par la différence entre la fréquence de la porteuse en entrée ( $f_i$ ) et celle de la porteuse régénérée ( $f_r$ ). Cette fréquence de battement ( $f_b$ ) s'exprime :

$$f_b = \pm(f_i - f_r) \quad (3.72)$$

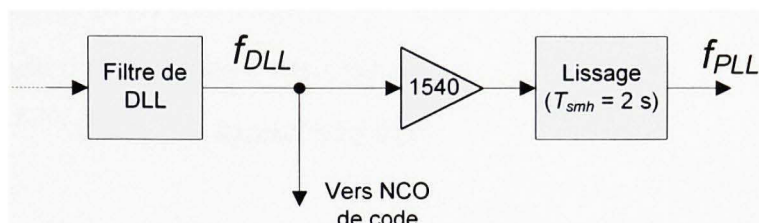
On emploie le terme « harmonique » car les fréquences de battement sont des multiples de  $1/2T$  (soit  $\pm 1,5/T$ ,  $\pm 2,5/T$ ,  $\pm 3,5/T$ , etc., où  $T$  est la période des intégrateurs). Ces harmoniques sont causées par le repliement spectral présent en sortie des intégrateurs. Ce repliement n'aurait toutefois pas lieu si des filtres plus sélectifs étaient choisis à la place des intégrateurs (voir la courbe de réponse spectrale des intégrateurs à la Figure 3.53).



**Figure 3.53** *Courbe de réponse spectrale normalisée en sortie des intégrateurs.*



Pour réaliser l'affinage, nous avons plutôt choisi d'estimer la fréquence de la porteuse à partir de celle du code. Le principal inconvénient d'une telle méthode est que le bruit en sortie de la boucle de code est amplifié par un facteur de 1540. Un lissage sur deux secondes permet toutefois d'éliminer le problème. Le mécanisme d'affinage en question est présenté à la Figure 3.54.



**Figure 3.54 Mécanisme d'affinage par dérivation de la fréquence DLL**

Un tel mécanisme n'est pas sujet au problème d'harmoniques abordé précédemment. De plus, il permet d'accélérer l'acquisition lorsque la puissance des signaux est forte en tirant profit de la grande plage dynamique de la DLL. Une certaine limitation existe toutefois quant à l'accélération maximale pouvant être prise par un tel mécanisme. Cette limite d'accélération peut être déterminée grâce à une règle du pouce tirée de (Kaplan, 1996), telle que :

$$3\sigma_f + \frac{R \cdot T_{smh}}{2} < \Delta f_{\max} \quad (3.73)$$

où :

- $\sigma_f$  est la déviation standard en Hz du bruit en sortie du mécanisme d'affinage
- $R$  représente l'amplitude d'un stress d'accélération, exprimé en Hz/s
- $T_{smh}$  représente la période de lissage
- $\Delta f_{\max}$  est l'erreur en fréquence maximale pouvant être prise par la PLL

Or, la déviation standard du bruit en sortie du filtre de DLL ( $\sigma_{DLL}$ ) se calcule (Kaplan, 1996) :

$$\sigma_{DLL} = \sqrt{\frac{2d^2 B_n}{c/n_0} \left[ 2(1-d) + \frac{4d}{T c/n_0} \right]} \quad (3.74)$$

où :

- $d$  est l'espacement en bribe entre les corrélateurs E, P et L
- $B_n$  est la bande de bruit de la DLL
- $T$  est la période d'intégration de la DLL
- $c/n_0 = 10^{0,1 \times C/N_0}$  ( $C/N_0$  est exprimé en dB)

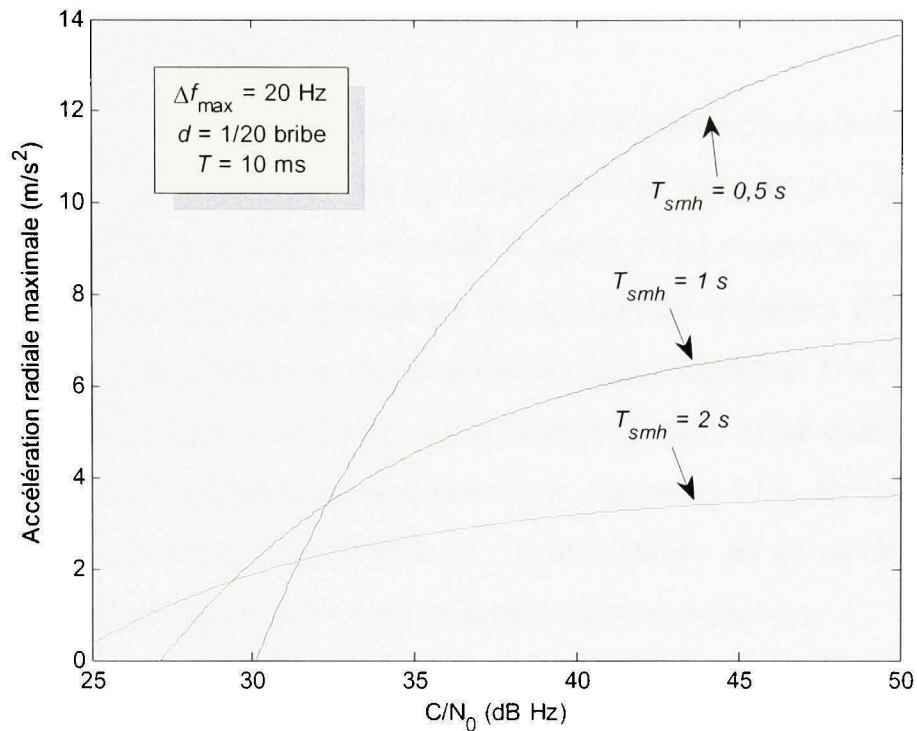
Sachant que l'amplitude du bruit avant lissage est égale à  $1540 \sigma_{DLL}$  et que la bande de bruit en sortie du lissage est de  $B_n = 1/T_{smh}$ , on obtient finalement :

$$\sigma_f = 1540 \sqrt{\frac{2d^2}{T_{smh} c/n_0} \left[ 2(1-d) + \frac{4d}{T c/n_0} \right]} \quad (3.75)$$

On voit en examinant cette formule qu'une nette amélioration est apportée en rapprochant les corrélateurs. Pour cette raison, nous employons un espacement de  $d = 1/20^e$  de bribe dès la stabilisation de la DLL. Ceci nous permet d'obtenir  $\sigma_f = 3,37$  Hz pour un ratio  $C/N_0 = 30$  dB Hz, un temps de lissage de deux secondes et une période d'intégration de 10 ms. En supposant que la plage dynamique de la PLL soit de  $\Delta f_{\max} = 20$  Hz, on calcule grâce à l'équation (3.73) un stress maximal  $R = 9,9$  Hz/s, ce qui correspond à :

$$a = R \lambda_{L1} = 9,9 \text{ Hz/s} \times 0,19 \text{ m} = 1,9 \text{ m/s}^2 \quad (3.76)$$

L'accélération radiale ne doit donc pas dépasser  $a = 1,9 \text{ m/s}^2$  lors de la phase initiale de poursuite. La Figure 3.55 présente les accélérations radiales maximales pouvant être encaissées par le mécanisme d'affinage en fonction de la période de lissage et du ratio  $C/N_0$ .



**Figure 3.55** *Accélération radiale maximale pouvant être encaissée au cours de la phase initiale de poursuite des signaux.*

### 3.10 Synchronisation au message de navigation

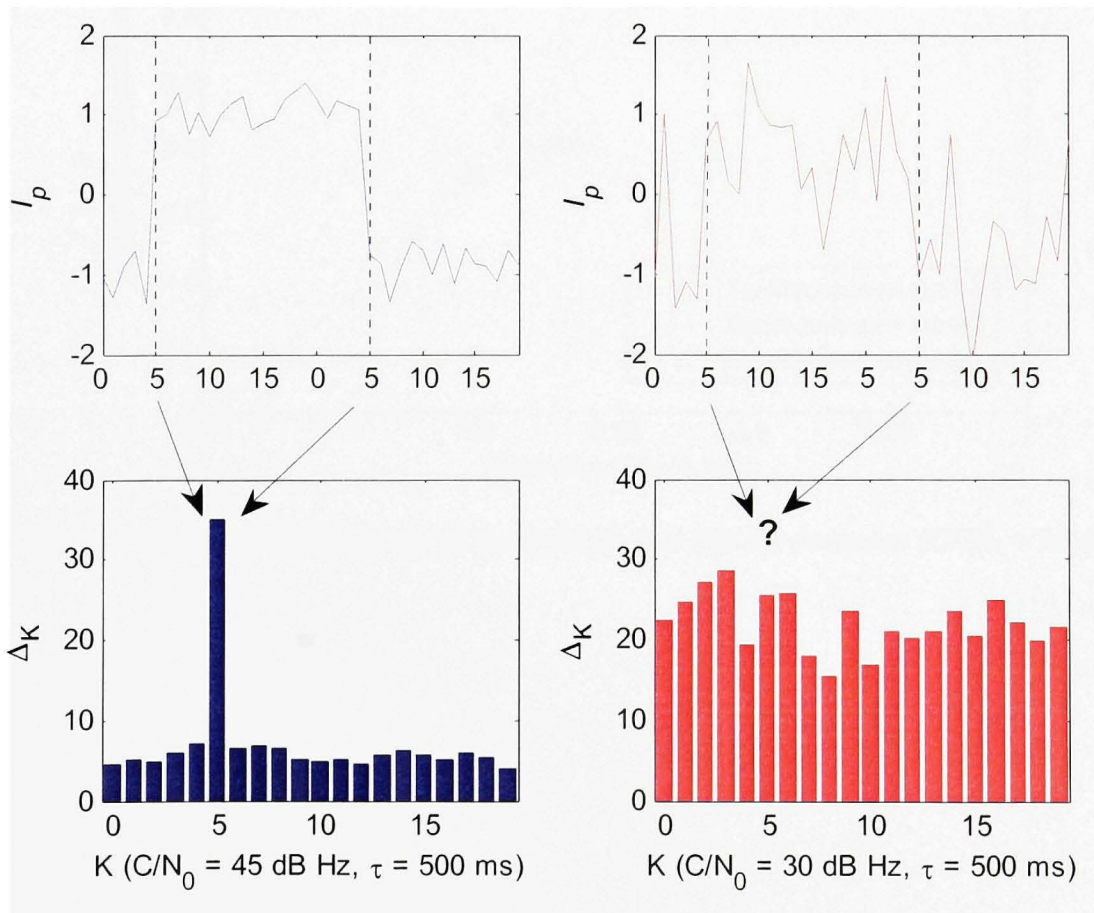
Un des inconvénients du GPS est que la période de répétition du code C/A (1 ms) n'est pas la même que celle des bits de données (20 ms). Ceci limite la période des intégrateurs à 1 ms durant les phases d'acquisition et de poursuite initiale des signaux et empêche de profiter de tout le gain d'étalement disponible. Une synchronisation aux bits de données doit donc être réalisée avant de pouvoir augmenter la période des intégrateurs et décoder correctement le message de navigation. Cette synchronisation ne peut toutefois être accomplie qu'en phase de poursuite.

La technique de synchronisation que nous avons employée consiste à détecter les transitions des bits de données sur le canal en phase du corrélateur en phase ( $I_p$ ). Une fois la PLL verrouillée et stabilisée, un compteur (K) est initialisé à 0. La différence absolue ( $\Delta_K$ ) entre chacun des échantillons du signal  $I_p$  est ensuite calculée, telle que :



$$\Delta_K = |I_p(n) - I_p(n-1)| \quad (3.77)$$

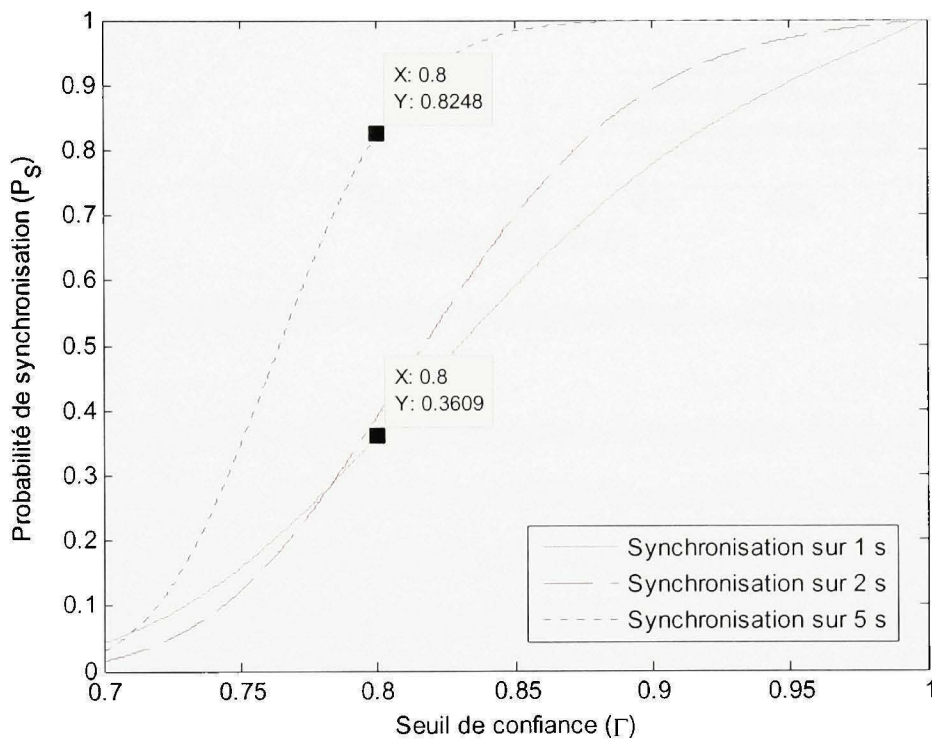
La fonction « valeur absolue » est essentielle à l'équation précédente étant donné que le signe de chacun des bits est *a priori* inconnu. Le compteur K est incrémenté à chaque époque du code C/A puis à nouveau réinitialisé à 0 lorsqu'il atteint 20 (il s'agit d'un compteur modulo 20). On accumule ainsi chacune des valeurs de  $\Delta_K$  dans 20 corbeilles différentes, à partir desquelles il est possible de détecter la transition des bits de données. Une méthode consiste alors à simplement prendre la corbeille ayant la plus grande valeur comme indicateur de transition, ce qui fonctionne relativement bien pour des ratios  $C/N_0$  élevés. Cette méthode s'avère néanmoins déficiente pour des ratios  $C/N_0$  plus faibles, tel qu'en témoigne la Figure 3.56 (une accumulation sur  $\tau = 500$  ms est effectuée dans les deux cas).



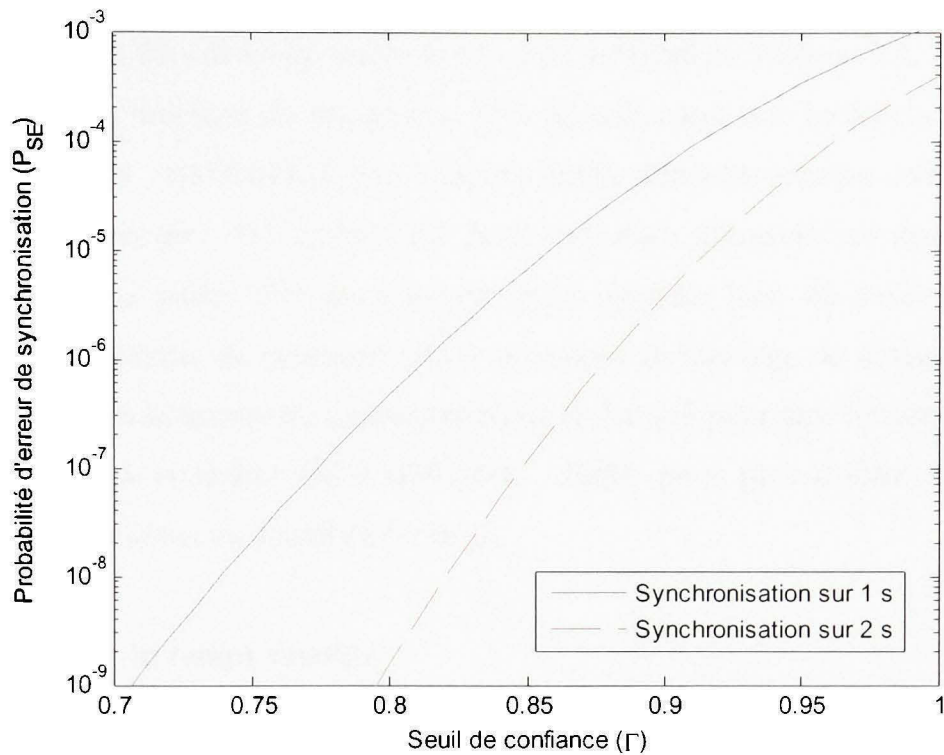
**Figure 3.56 Exemples de synchronisation aux bits de données par recherche de la transition la plus abrupte.**

Une méthode plus judicieuse consiste plutôt à comparer entre elles les corbeilles présentant les deux plus grosses valeurs, soit  $\Delta_{KMAX}$  et  $\Delta_{KMAX-1}$ . Un test de confiance est alors effectué à l'aide d'un seuil  $\Gamma$ , à savoir si la corbeille affichant la plus grosse valeur ( $\Delta_{KMAX}$ ) indique réellement l'époque où ont lieu les transitions des bits de données :

$$\frac{\Delta_{KMAX-1}}{\Delta_{KMAX}} \leq \Gamma \quad (3.78)$$



**Figure 3.57** *Probabilité de synchronisation aux bits de données ( $C/N_0 = 30$  dB Hz)*



**Figure 3.58 Probabilité d'erreur de synchronisation ( $C/N_0 = 30$  dB Hz).**

Nous avons choisi un seuil  $\Gamma = 0,8$  pour le récepteur de navigation ainsi qu'une période de recherche d'une seconde, ce qui permet de réduire la probabilité d'erreur de détection à  $P_{SE} = 4,8 \times 10^{-7}$  pour un  $C/N_0$  de 30 dB Hz (voir Figure 3.58). Advenant que le test de confiance échoue, les données sont alors accumulées durant une seconde supplémentaire pour être testées à nouveau, et ainsi de suite jusqu'à l'accomplissement de la synchronisation. On note que le temps de synchronisation moyen est inférieur à cinq secondes pour un ratio  $C/N_0 \geq 30$  dB Hz (le taux de succès est de 82,5% après cinq secondes pour un seuil  $\Gamma = 0,8$ , voir Figure 3.57). Nous avons tout de même limité le temps de synchronisation à un maximum de 30 secondes, suite auquel un retour à la phase d'acquisition est effectué, question de ne pas demeurer pris dans une boucle infinie. Les résultats présentés à la Figure 3.57 ainsi qu'à la Figure 3.58 ont été obtenus grâce à une simulation numérique du mécanisme de synchronisation.



Une fois la synchronisation aux bits de données effectuée, chacun des bits est intégré sur une période de 20 ms et décodé en dur selon le principe présenté au Tableau 3.6. Il s'agit alors de se synchroniser au message de navigation GPS en effectuant une recherche du préambule, dont la séquence est : 10001011. Il faut toutefois porter attention puisque cette séquence peut être aussi bien inversée : 01110100. Lors de la détection d'une de ces deux séquences, il suffit de vérifier la parité afin de s'assurer qu'il s'agisse bien du préambule. Une fois l'ensemble du processus de synchronisation accompli, le message de navigation est stocké mot par mot dans la mémoire du microcontrôleur embarqué pour être finalement décodé. Le lecteur est invité à consulter (ICD-GPS-200C, 2000) pour plus d'information quant au message de navigation et au calcul de la parité.

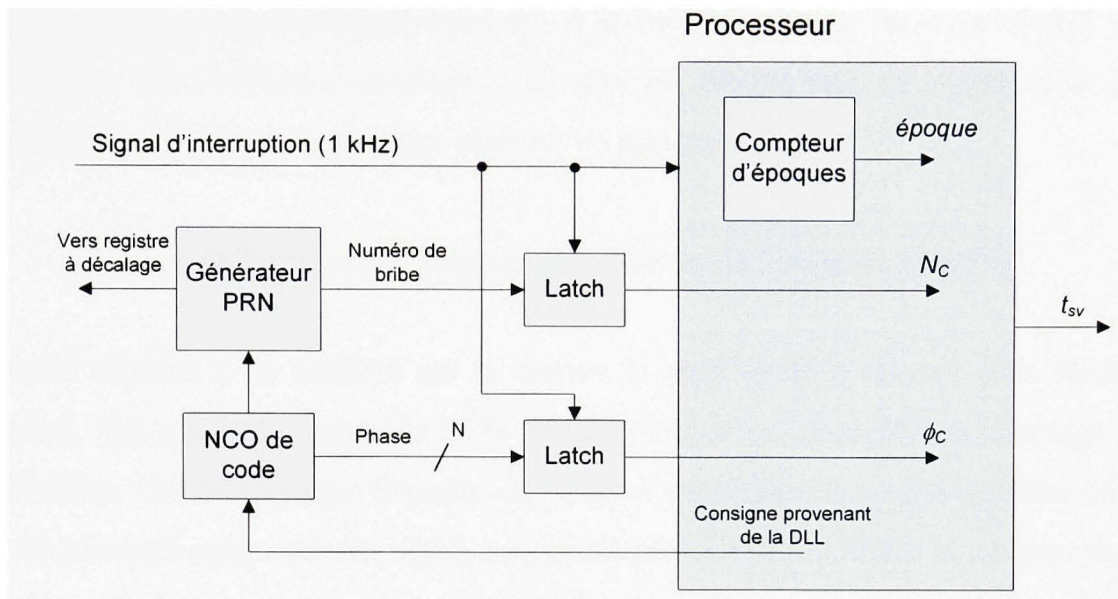
### 3.11 Mesure du temps satellite

Le temps d'émission d'un satellite,  $t_{SV}$ , est obtenu via la combinaison de plusieurs mesures différentes au sein d'un récepteur de navigation. La mesure la plus facile à obtenir est le TOW, transmis directement dans le message de navigation. Le principe consiste alors à synchroniser un compteur d'époques au TOW puis à l'incrémenter à toutes les époques afin de déterminer  $t_{SV}$  à 1 ms près. Puisque le TOW est présent au début de chaque sous-trame GPS C/A, il faut toutefois attendre environ six secondes suite à la synchronisation au message de navigation pour avoir accès à cette valeur. De plus, une attention toute particulière doit être apportée au fait que le TOW transmis dans le message est un multiple entier de six secondes. L'opération suivante doit donc être réalisée afin de déterminer correctement l'époque en cours (calculée ici en ms) :

$$\text{époque} = 6000 \cdot \text{TOW} \quad (3.79)$$

Afin d'obtenir plus de précision au niveau de  $t_{SV}$ , il s'agit ensuite d'extraire le numéro de bribe,  $N_C$ , ainsi que la phase du NCO de code,  $\phi_c$ , du canal de démodulation. Cette opération est réalisée de façon systématique pour tous les canaux de démodulation dès le déclenchement d'une interruption au niveau du microcontrôleur. Des mémoires de type *latch*

sont employées dans le FPGA pour sauvegarder les mesures avant qu'elles ne soient transmises au microcontrôleur, telles qu'illustrées à la Figure 3.59.



**Figure 3.59** *Extraction des mesures pour le calcul de  $t_{SV}$ .*

En ayant accès à un compteur d'époques ainsi qu'aux mesures  $N_c$  et  $\phi_c$ , il est alors possible de calculer directement la valeur de  $t_{SV}$  en secondes, telle que :

$$t_{SV} = \frac{1}{1000} \left[ \text{époque} + \frac{1}{1023} \left( N_c + \frac{\phi_c}{2^N} \right) \right] \quad (3.80)$$

où  $N$  représente le nombre de bits récoltés du NCO de code. Ce dernier paramètre détermine la précision maximale de  $t_{SV}$ , soit  $\Delta t_{SV}$  :

$$\Delta t_{SV} = \frac{1}{f_c \cdot 2^N} \quad (3.81)$$

où  $f_c$  est la fréquence du code d'étalement, soit 1,023 MHz pour GPS C/A. Puisque  $t_{SV}$  est employé pour le calcul des pseudo-distances, on peut alors exprimer  $\Delta t_{SV}$  en mètres ( $\Delta \rho$ ) en la multipliant par la vitesse de la lumière,  $c$  :

$$\Delta\rho = \frac{c}{f_c \cdot 2^N} \quad (3.82)$$

Au niveau du récepteur développé, nous avons limité le nombre de bits récoltés à  $N = 20$  afin de minimiser les ressources employées au sein du FPGA tout en offrant une précision maximale d'environ 0,3 mm pour les mesures de pseudo-distances.

### 3.12 Mesure de la pseudo-distance, de l'ADR et de la vitesse relative

La vitesse relative à un satellite est la mesure la plus facile à obtenir d'un récepteur de navigation. Tel que vu au chapitre 2, la vitesse relative est découlée du décalage Doppler d'un satellite. Or, ce décalage Doppler est directement obtenu en sortie du filtre de PLL. Il suffit de convertir cette mesure grâce à l'équation suivante pour obtenir la vitesse relative :

$$\dot{\rho} = -\Delta f \frac{c}{f} = -\Delta f \lambda \quad (3.83)$$

où :

- $\dot{\rho}$  représente la vitesse relative à un satellite
- $\Delta f$  est le décalage Doppler mesuré sur la porteuse
- $c$  est la vitesse de la lumière
- $f$  est la fréquence nominale de la porteuse
- $\lambda$  est la longueur d'onde nominale de la porteuse

La mesure de pseudo-distance n'est pas tellement plus compliquée à obtenir. Tout d'abord, une référence de temps suffisamment précise est nécessaire. Cette référence est facilement obtenue en dérivant initialement l'horloge du récepteur du temps d'émission d'un satellite,  $t_{SV}$ . Étant donné que les délais de propagation entre les satellites GPS et un utilisateur au sol s'étendent d'environ 60 ms à 80 ms, une référence de temps ( $t$ ) précise à environ 20 ms peut donc être obtenue lors du démarrage en initialisant l'horloge du récepteur telle que :



$$\text{Si démarrage, alors : } t \leftarrow t_{sv} + 70 \text{ ms} \quad (3.84)$$

Afin de déterminer les mesures de pseudo-distances ( $\rho$ ) de chacun des satellites, il s'agit par la suite de calculer la différence entre l'horloge du récepteur et le temps des satellites, telle que :

$$\rho = c(t - t_{sv}) \quad (3.85)$$

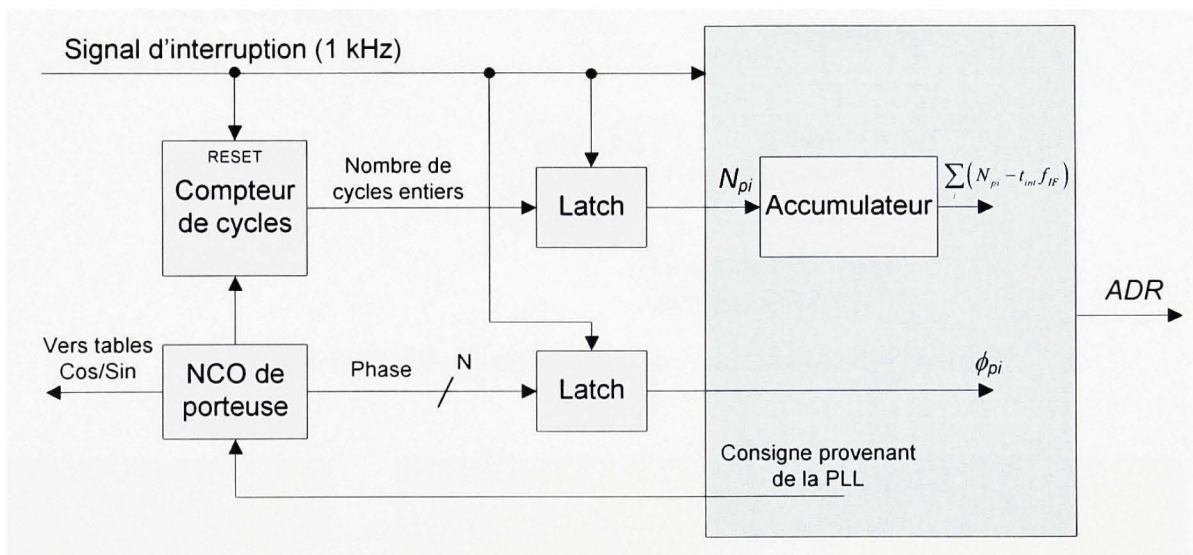
L'*Accumulated Doppler Range* (ADR) consiste quant à elle en une estimation de la pseudo-distance à partir de la phase de la porteuse plutôt que celle du code. L'utilisation des ADR plutôt que des pseudo-distances au niveau de la solution de navigation permet d'obtenir une position beaucoup plus précise, typiquement de l'ordre du centimètre. Ceci s'explique par le fait que la longueur d'onde de la porteuse (soit  $\approx 19$  cm pour L1) est beaucoup plus petite que celle du code (soit  $\approx 293$  m pour le code C/A). Une telle solution de navigation est toutefois beaucoup plus compliquée à réaliser puisqu'elle demande de résoudre l'ambiguïté de phase présente sur la porteuse. De façon générale, il est nécessaire d'opérer en mode différentiel pour obtenir une position grâce à l'ADR (Misra et Enge, 2006).

Comme son nom l'indique, l'ADR consiste à accumuler la fréquence Doppler de la porteuse afin d'obtenir une estimation de la pseudo-distance. Étant donné qu'une ambiguïté est présente sur la phase de la porteuse, on démarre cette accumulation de façon arbitraire au moment de la poursuite des signaux d'un satellite. Puisque des erreurs de quantification peuvent être introduites au niveau du NCO de PLL, il est toutefois préférable de déterminer l'ADR directement à partir de la phase du NCO plutôt qu'en sortie des filtres de PLL. Pour y parvenir, il s'agit d'ajouter un compteur en sortie du NCO afin de déterminer le nombre de cycles complets effectués par la porteuse entre deux interruptions consécutives du microcontrôleur (voir Figure 3.60). L'ADR peut ensuite être déterminé par l'équation suivante :

$$ADR = -\lambda \left[ \frac{\phi_{pi}}{2^N} + \sum_i (N_{pi} - t_{int} f_{IF}) \right] \quad (3.86)$$

où :

- $N_{pi}$  est la sortie du compteur de cycles de porteuse à l'instant  $i = 0, 1, 2, \dots$
- $t_{int}$  est la période des interruptions, soit 1 ms
- $f_{IF}$  est la fréquence de la porteuse en IF, soit 15 MHz
- $\phi_{pi}$  est la phase du NCO de PLL à l'instant  $i$
- $N$  est le nombre de bits recueillis au niveau de la phase du NCO de PLL
- $\lambda$  est la longueur d'onde nominale de la porteuse



**Figure 3.60** *Extraction des mesures pour le calcul de l'ADR.*

### 3.13 Mesure en temps réel de la fonction de cross-corrélation

Pour des fins de recherches, il peut être très pratique de mesurer en temps réel la fonction de cross-corrélation d'un canal de démodulation GNSS. Cette fonction de cross-corrélation, aussi appelée « pic » de corrélation, porte de l'information utile concernant les multitrajets par exemple. L'analyse de ces pics peut ainsi servir d'outil afin de développer de nouvelles techniques de mitigation des multitrajets.

Nous avons tiré avantage de la grande fréquence d'échantillonnage des canaux de démodulation développés afin de mesurer les pics de corrélations. Nous avons ainsi implémenté un corrélateur supplémentaire au sein de chacun des canaux de démodulation afin de balayer en continu un intervalle de  $-1$  à  $+1$  bribe par rapport au corrélateur en phase. Tel que mentionné précédemment, la résolution des corrélateurs à espacement variable est d'environ  $1/60^\circ$  de bribe. La résolution au niveau du balayage étant la même, nous obtenons donc approximativement 120 points de mesure au niveau du pic de corrélation. La Figure 3.61 présente l'architecture d'un tel corrélateur développé à l'aide d'un registre à décalage de 128 bits. Le fonctionnement de ce corrélateur est présenté à La Figure 3.62.

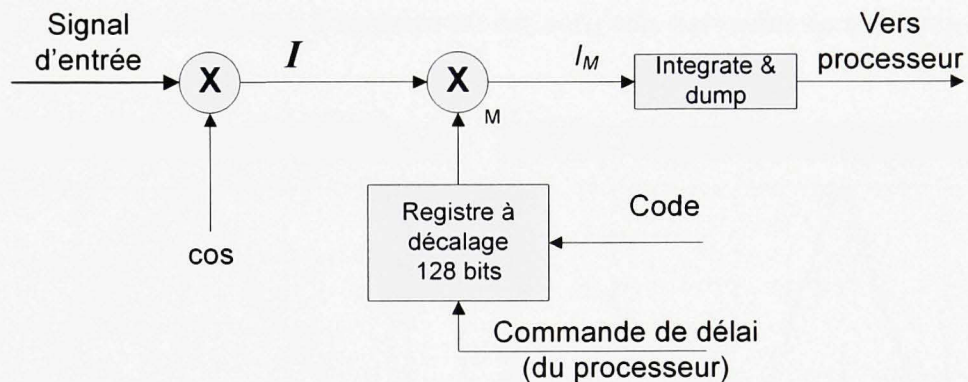


Figure 3.61 *Diagramme bloc du corrélateur de mesure.*

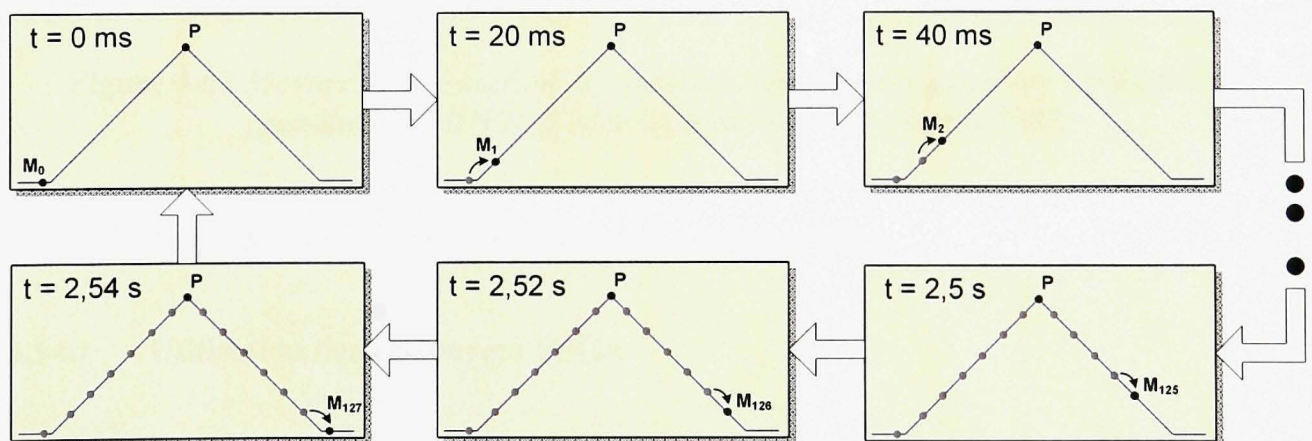
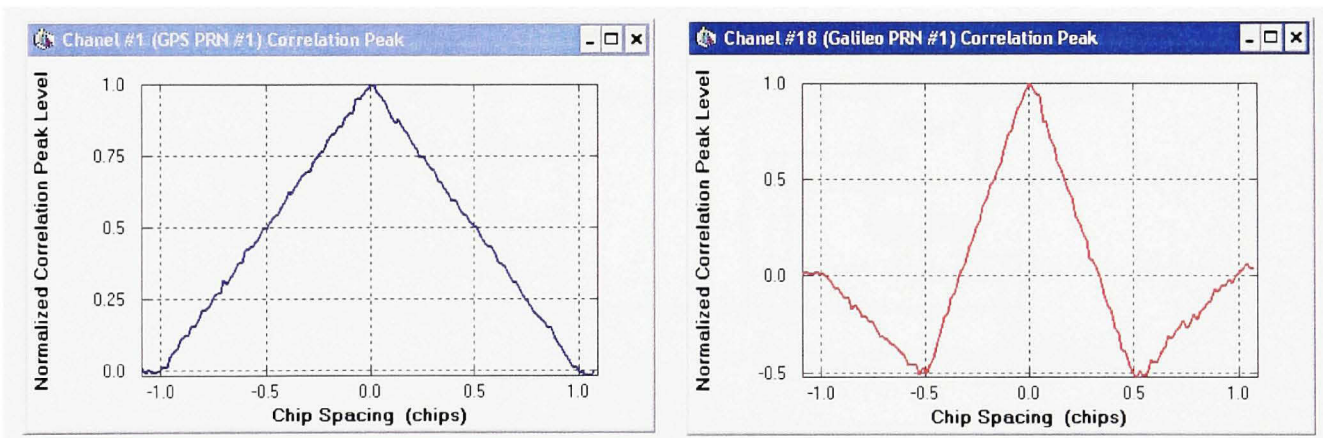


Figure 3.62 *Principe de fonctionnement du corrélateur de mesure.*

Il faut noter que seul le canal en phase ( $I_M$ ) est implémenté au niveau du mécanisme de mesure. Ceci permet dans un premier temps d'économiser des ressources au niveau du



FPGA. De plus, ceci permet de mesurer le véritable pic de corrélation puisqu'aucune mise au carré n'est nécessaire, contrairement aux autres corrélateurs. Il faut toutefois attendre la stabilisation de la PLL pour que les mesures en sortie du corrélateur soient correctes. Étant donné qu'une moyenne est effectuée pour chacun des points de mesure, il est aussi nécessaire d'attendre la synchronisation aux bits de données. La durée de cette moyenne peut être ajustée en temps réel par l'utilisateur grâce à l'interface graphique développée au niveau du PC. Mentionnons pour terminer qu'une normalisation est effectuée lors de l'affichage en temps réel afin que le maximum de corrélation soit positif et unitaire. La Figure 3.63 présente des exemples de pics de corrélations pour un satellite GPS et le satellite GIOVE-A de Galileo, captés à l'extérieur le 26 mai 2007 (une moyenne sur 10 secondes a été employée dans les deux cas, ce qui équivaut à environ 80 échantillons par point de mesure).



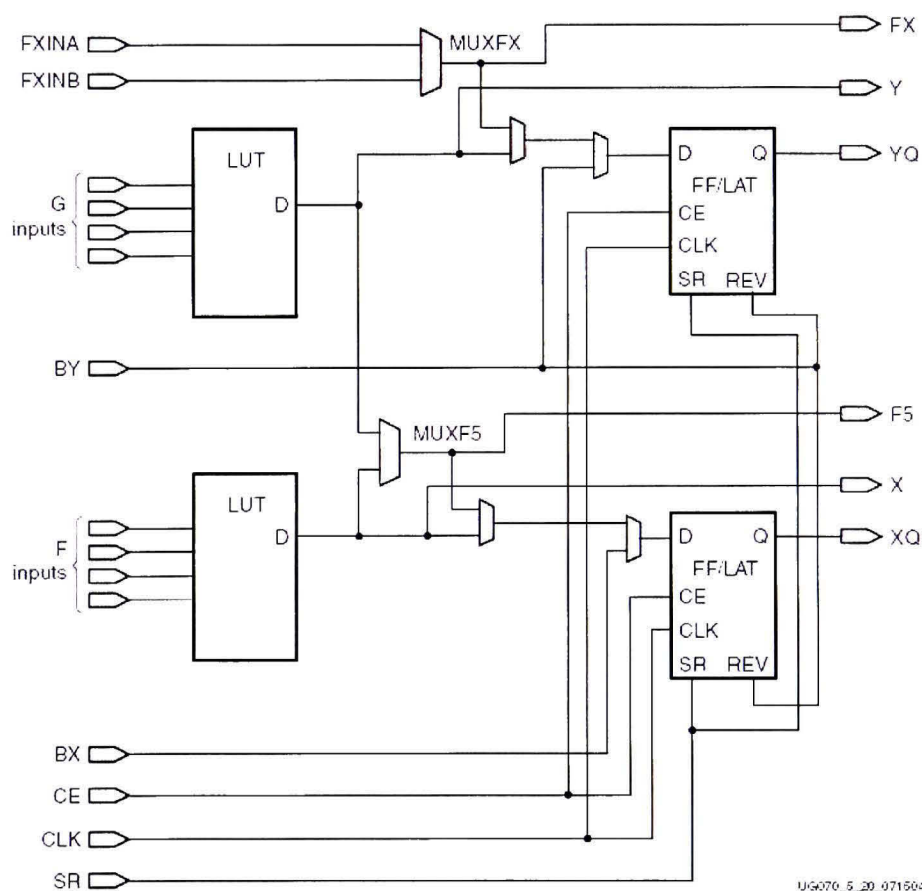
**Figure 3.63** *Mesure de la fonction de cross-corrélations des satellites GPS PRN #1 (gauche) et GIOVE-A (droite) à Montréal le 26 mai 2007.*

### 3.14 Implémentation en temps réel au sein d'un FPGA

#### 3.14.1 Utilisation des ressources FPGA

Contrairement à ce que l'on peut penser, l'implémentation d'un design au sein d'un FPGA ne s'effectue pas vraiment à l'aide de portes logiques (ex. : portes OU, ET, OU exclusif, ...). Des tables de correspondance (LUT – *Look-Up Table*), sont plutôt employées afin de mettre en place des fonctions mathématiques et logiques plus ou moins complexes. Une ou plusieurs

LUT sont ainsi encapsulées à l'intérieur de cellules, aussi appelées « cellules logiques » ou *slices*, auxquelles on ajoute généralement des bascules D (DFF) ainsi que des multiplexeurs (MUX). La Figure 3.64 (Xilinx, 2005b) présente l'intérieur d'une cellule logique pour un Virtex-4 de Xilinx, soit le FPGA employé pour la mise en œuvre du récepteur. Chaque FPGA contient un certain nombre de ces cellules logiques, chacune d'elle étant reliée aux autres par le biais d'un réseau d'interconnexions complexe. Les FPGA comprennent aussi plusieurs variétés de primitives, telles que des multiplicateurs ou des blocs de mémoire RAM (*Random Access Memory Block* - BRAM), par exemple. Ces primitives permettent de développer des architectures plus évoluées sans toutefois avoir à recourir à des composants externes supplémentaires.



**Figure 3.64** Architecture simplifiée d'une cellule logique de Virtex-4 de Xilinx.

Tiré de (Xilinx, 2005b)

Les outils de développement de Xilinx permettent de générer une vaste gamme de rapports. Grâce à ces rapports, nous avons pu déterminer le nombre de ressources employées au sein du FPGA par le récepteur de navigation développé. Le Tableau 3.7 présente de façon détaillée la consommation des ressources par chacun des canaux de démodulation GPS L1 C/A ainsi que pour l'ensemble du récepteur. La répartition des LUT est ensuite illustrée de façon graphique à la Figure 3.65 ainsi qu'à la Figure 3.66.

Tableau 3.7  
Utilisation des ressources FPGA par le récepteur GPS L1 C/A

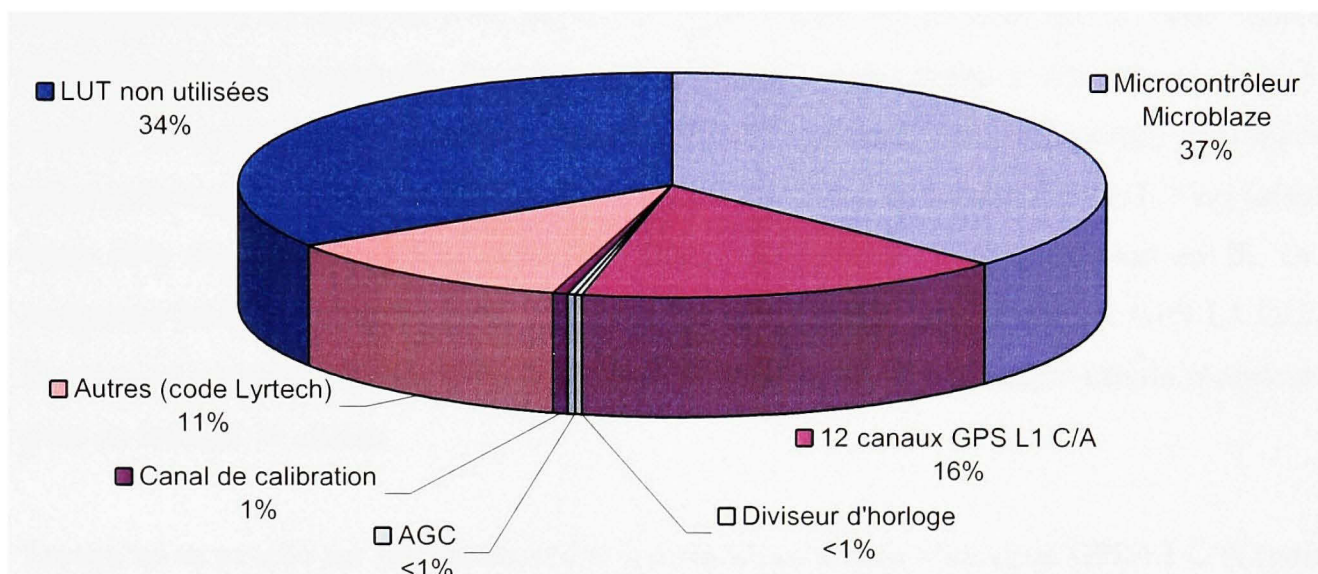
	LUT	DFF	BRAM	DSP48
Microcontrôleur Microblaze	18 338	10 490	73	7
Diviseur d'horloge	77	34	0	0
AGC	164	205	0	0
Canal de calibration	327	182	0	2
Autres (code Lyrtech)	5 595	3 373	3	1
Canal GPS L1 C/A	660	741	1	3
- Multiplicateurs	4	0	0	2
- Mult. 1xN bits	56	49	0	0
- Intégrateurs	385	245	0	0
- NCO (PLL)	87	53	0	0
- NCO (DLL)	2	20	0	1
- Générateur de code	47	44	0	0
- Registres à décalage	33	26	1	0
- Tampons de mesures ( <i>Latches</i> )	14	304	0	0
Total (12 canaux GPS L1 C/A)	32 421	23 176	88	46
Ressources totales (Virtex-4 SX-55)	49 152	49 152	320	512
Ressources employées (%)	66,0	47,2	27,5	9,0
Portes logiques équivalentes*	4,8 millions			

\* Estimation donnée par les outils de Xilinx

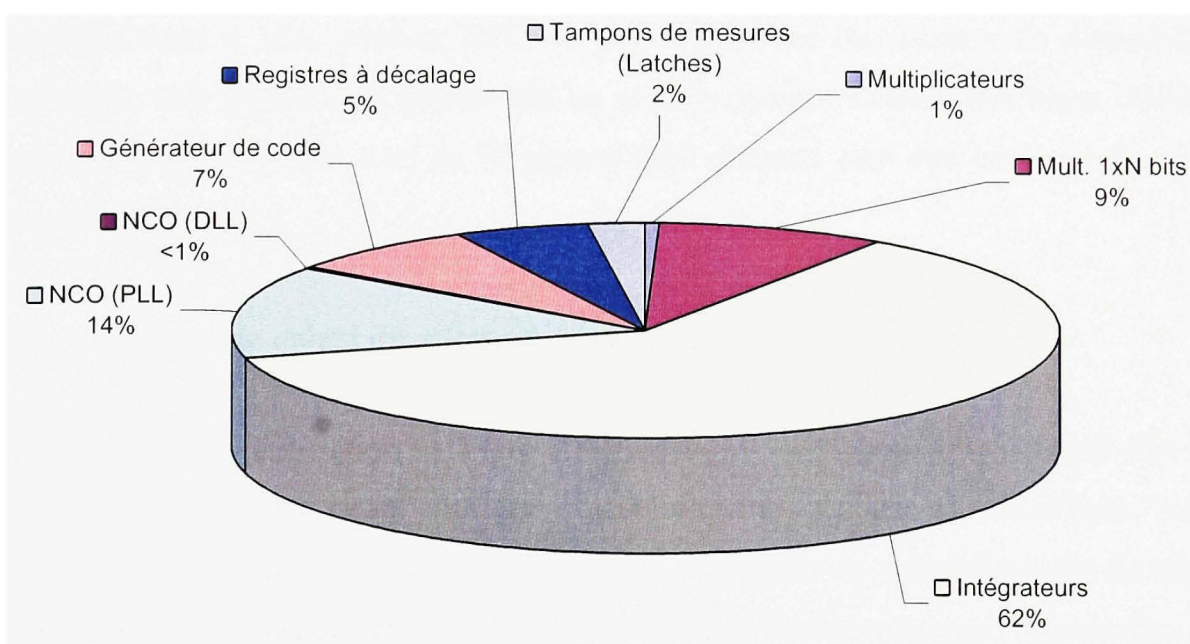
Un examen attentif du Tableau 3.7 révèle que les ressources les plus exploitées par le récepteur sont les LUT avec une utilisation totale de 66,3%, suivis des DFF et des BRAM avec 47,3% et 27,5% d'utilisation respectivement. Enfin, seulement 9% de tous les blocs DSP48 disponibles sont employés. Ces blocs, aussi nommés *Xtreme Digital Signal Processing Slices*, sont propres aux familles de FPGA Virtex-4 et Virtex-5 de Xilinx. Chacun



des blocs DSP48 est en fait composé d'un multiplicateur de 18 x 18 bits ainsi que d'un accumulateur de 48 bits.



**Figure 3.65 Répartition totale des LUT pour le récepteur GPS L1 C/A.**



**Figure 3.66 Répartition des LUT au niveau d'un canal de démodulation GPS L1 C/A.**

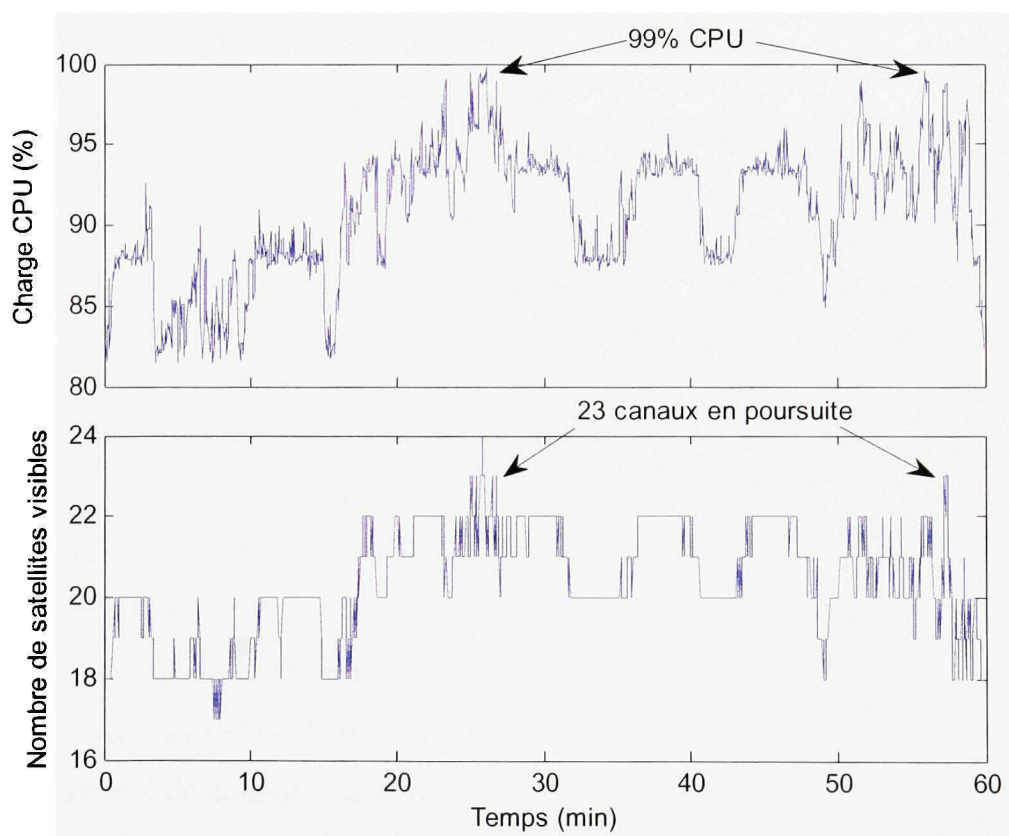
En observant la Figure 3.65, on constate que l'AGC, le diviseur d'horloge ainsi que le canal de calibration occupent un espace négligeable au niveau du FPGA (soit environ 2% des ressources totales). La majorité des ressources est plutôt employée par le microcontrôleur embarqué, avec un total de 37% des LUT. Il faut aussi mentionner que le code source développé par la compagnie Lyrtech (qui regroupe essentiellement des pilotes pour le contrôle des périphériques externes, tels que la SDRAM ou le bus PCI) occupe un espace relativement grand au sein du FPGA avec un total de 11% d'utilisation des LUT. Ceci laisse donc 50% des ressources pour l'implémentation des canaux de démodulation en IF. Or, seulement 16% de toutes les LUT servent à l'implémentation des 12 canaux GPS L1 C/A. Environ 24 canaux GPS supplémentaires pourraient ainsi être ajoutés au niveau du récepteur, pour un total de 36 canaux.

Lorsqu'on se penche sur la répartition des ressources au niveau d'un canal GPS L1 C/A (voir Figure 3.66), on découvre que 62% des ressources sont employées par les intégrateurs. Ce nombre pourrait être réduit de façon drastique en ayant recours à des blocs DSP48 afin d'implémenter les accumulateurs. L'expérience a été tentée avec le NCO de DLL et s'est avérée un succès puisque ce dernier occupe moins de 1% des ressources allouées à un canal (comparativement à 14% pour le NCO de PLL). La taille des canaux de démodulation pourrait donc être réduite d'au moins 50% en ayant seulement recours aux blocs DSP48 au niveau des intégrateurs. Un total de 72 canaux GPS pourrait ainsi être envisagé au sein du FPGA employé.

### **3.14.2 Charges de calcul du microcontrôleur embarqué**

Un grand travail d'optimisation a dû être effectué au niveau du logiciel embarqué afin de le rendre suffisamment performant pour être exécuté en temps réel par un Microblaze. Tel que mentionné précédemment, des fonctions ont été développées en assembleur afin de réaliser des opérations telles que le calcul de la racine carrée et de l'arctangente. Néanmoins, avec approximativement 500 instructions par appel, ces fonctions consomment encore la majeure partie du temps de calcul du Microblaze. Considérant le fait que ce dernier opère à 60 MHz

et qu'il traite une instruction par cycle d'horloge (Xilinx, 2005a), on ne compte donc que 60 000 instructions par interruption de 1 ms. Puisque deux racines carrées et une arctangente sont nécessaires pour le calcul des discriminateurs de DLL et de PLL, on obtient alors pour 12 canaux GPS L1 C/A environ 18 000 instructions par interruption, ce qui représente en pratique 30% de la charge de calcul maximale. Environ 20% supplémentaires sont quant à eux associés au reste du programme d'acquisition et de poursuite, pour une charge de calcul totale de 50%. Des tests effectués à l'aide de 24 canaux GPS ont permis de déterminer qu'un maximum de 23 canaux peuvent être ainsi traités en simultanée par le Microblaze (voir Figure 3.67). Cette valeur a été obtenue en mesurant le temps de calcul lors de chaque interruption et en comparant ce temps à la durée maximale de 1 ms de la routine d'interruption.



**Figure 3.67** Charge de calcul du Microblaze pour un récepteur à 24 canaux.



Plusieurs solutions sont possibles afin d'augmenter le nombre maximal de canaux pouvant être traités par le microcontrôleur embarqué. La première consiste à effectuer les calculs de la racine carrée et de l'arctangente directement au sein du FPGA à l'aide d'un accélérateur matériel (tel que celui employé pour le calcul de la parité du message de navigation GPS C/A). Une autre solution consiste plutôt à augmenter la fréquence d'horloge du microcontrôleur afin de réaliser un plus grand nombre d'instructions par unité de temps. Finalement, la dernière solution serait de remplacer le microcontrôleur par un véritable DSP (*Digital Signal Processor*). Un excellent candidat serait le DSP LEON développé par le groupe Gaisler Research. Le lecteur est invité à consulter (Gaisler Research, 2007) pour plus d'information concernant ce DSP.

### 3.15 Conclusion et recommandations

Dans ce chapitre, nous avons abordé la démodulation des signaux GPS L1 C/A en débutant par un bref survol des signaux CDMA. La structure des signaux GPS C/A a ensuite été détaillée, suivie par une description générale du récepteur développé. La conception et la réalisation de la tête RF et des canaux de démodulation IF ont ensuite été présentées tour à tour. Le fonctionnement du microcontrôleur embarqué a aussi été décrit, de même que plusieurs techniques de mesure des signaux, dont celles du ratio SNR et des pseudo-distances. Enfin, le nombre de ressources employées au niveau du FPGA ainsi que la charge de calcul du microcontrôleur embarqué ont été analysés.

Les résultats présentés montrent que l'architecture développée au niveau du FPGA est plutôt compacte et qu'elle permet l'implémentation de nombreux canaux de démodulation GPS (jusqu'à 36 au total) au sein d'un FPGA Virtex-4 SX-55 de Xilinx. Des modifications mineures permettraient même de doubler le nombre maximal de canaux pour un total de 72. Le nombre maximal de canaux pouvant être pris en charge par le microcontrôleur embarqué étant seulement de 23, des améliorations devront toutefois être apportées à ce niveau avant de pouvoir bénéficier de toute la capacité offerte par le FPGA. Voici un résumé des autres

améliorations qui pourraient être éventuellement apportées aux segments RF et IF du récepteur :

- ajouter un AGC plus performant au niveau de la tête RF;
- caractériser plus en détail la tête RF au niveau du bruit de phase;
- caractériser la tête RF pour d'autres fréquences (ex. : L2, L5, etc.);
- augmenter fréquence d'échantillonnage du système afin de permettre l'acquisition de signaux plus large bande, tels que E5 de Galileo;
- ajouter un signal d'avertissement en cas de saturation des intégrateurs;
- développer un mécanisme d'acquisition par FFT.

Une miniaturisation du récepteur serait aussi de mise afin de faciliter la réalisation de tests à l'extérieur (le récepteur se situe actuellement aux alentours de 20 kg et consomme environ 200 W d'électricité).

Une description détaillée des canaux Galileo est réalisée au chapitre suivant. Le chapitre 5 termine quant à lui la présentation du récepteur développé en abordant la partie logicielle implantée au niveau de l'ordinateur PC.

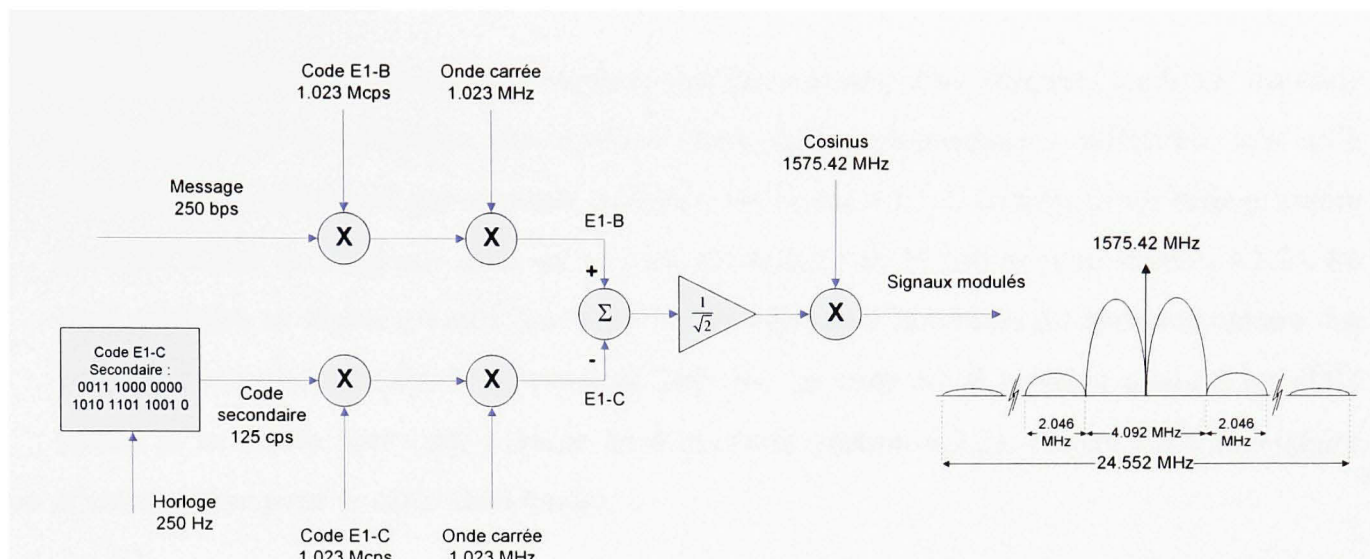
## CHAPITRE 4

### DÉMODULATION EN TEMPS RÉEL DES SIGNAUX GALILEO

#### 4.1 Structure des signaux Galileo E1

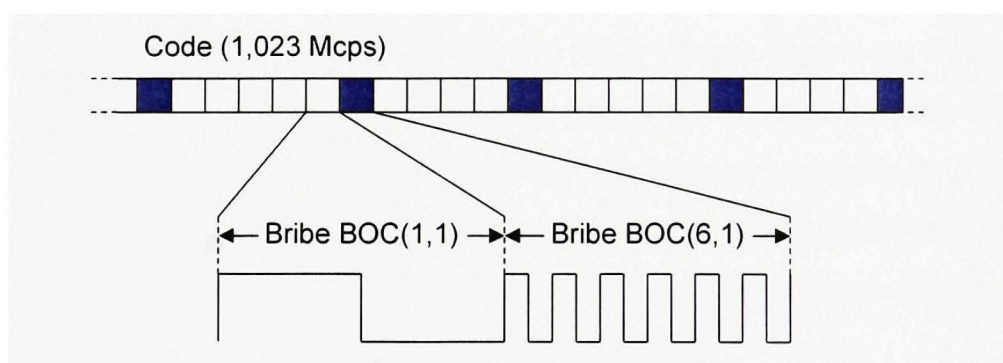
De la même manière que pour le GPS, les signaux Galileo E1 sont basés sur une modulation de type DS-CDMA. La fréquence du code d'étalement est aussi de 1,023 MHz et la porteuse est à la fréquence L1. Une modulation de type BOC est cependant employée de manière à limiter les interférences entre les signaux Galileo E1 et GPS L1, facilitant ainsi l'interopérabilité des deux systèmes. Plus précisément, on a recours à une modulation BOC(1,1) dans le cas du satellite GIOVE-A (ESTEC, 2007). Cela signifie que les signaux sont modulés par une sous-porteuse carrée de fréquence égale au code, soit 1,023 MHz. Le premier chiffre d'une telle notation indique la fréquence de la sous-porteuse carrée, alors que le second indique celle du code d'étalement (chacune des fréquences étant un multiple de 1,023 MHz). Le spectre fréquentiel d'un signal modulé BOC(1,1) ne présente aucune énergie au niveau de la porteuse. Deux lobes principaux sont plutôt situés de part et d'autre de cette dernière, tels qu'illustrés à la Figure 4.1. Dans le cas de Galileo E1, les centres de ces lobes coïncident avec les zéros spectraux des signaux GPS L1 C/A. La puissance des signaux Galileo E1 est sensiblement la même que celle des signaux GPS L1 C/A, soit de -155,7 dBW au minimum en ciel dégagé (ESTEC, 2007).





**Figure 4.1** *Schéma de modulation et spectre fréquentiel des signaux Galileo E1.*

Tel que mentionné au chapitre 1, une modulation MBOC devrait plutôt être employée pour les futurs satellites Galileo. Selon toute vraisemblance, la variante TMBOC(6,1,2/11) (*Time Multiplexed BOC*) devrait être employée, ce qui signifie que la sous-porteuse carrée à 1,023 MHz serait remplacée deux bribes sur 11 par une sous-porteuse à six fois la fréquence, soit 6,138 MHz. La sous-porteuse résultante est illustrée à la Figure 4.2. Une telle technique de modulation a pour but de réduire l'impact des multitrajets au niveau de la solution de navigation (Hein et al., 2006). Pour plus de détails sur les perspectives de modulation des signaux Galileo E1, le lecteur est invité à consulter (Stansell, 2006).



**Figure 4.2** *Exemple de modulation TMBOC(6,1,2/11).*

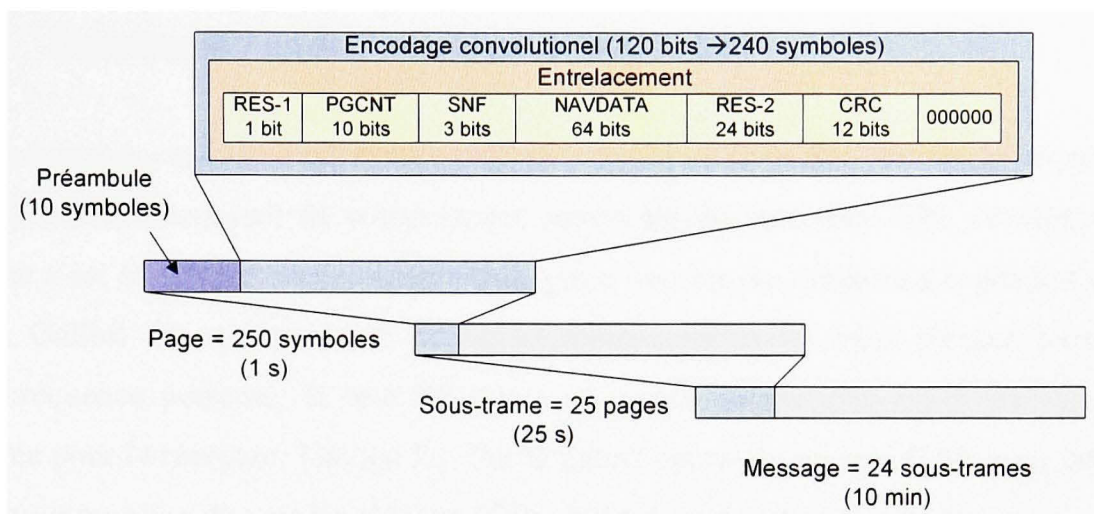
Une particularité des signaux Galileo E1 est que deux canaux sont employés : un canal de données (E1-B) et un canal pilote (E1-C). Puisqu'aucune donnée n'est modulée sur le canal

pilote, ceci permet d'intégrer les signaux sur des périodes très longues, facilitant du coup l'acquisition et la poursuite des signaux dans des environnements difficiles, tels qu'à l'intérieur des bâtiments par exemple. Le code d'étalement E1-C comprend un code primaire de 8184 bribes de longueur ainsi qu'un code secondaire de 25 bribes (voir section 4.3.3). Le code primaire se répète à toutes les 8 ms, soit la période d'une bribe du code secondaire. La période totale du code E1-C est donc de 200 ms. Le code E1-B possède quant à lui 4092 bribes et se répète après une période de 4 ms (voir section 4.3.3). Aucun code secondaire n'est employé pour le canal de données.

Les bits de données sur le canal E1-B possèdent une période de 4 ms et sont parfaitement synchronisés avec le code d'étalement. Les données sont regroupées en pages de 120 bits, chacune comprenant :

- Deux champs réservés (RES-1 et RES-2) de 1 et 24 bits respectivement
- Un compteur de page (PGCNT) sur 10 bits
- Trois drapeaux (SNF) servant à indiquer les changements au niveau du message de navigation
- 64 bits de données de navigation (NAVDATA)
- Un CRC (*Cyclic Redundancy Check*) de 12 bits

La séquence « 000000 » est finalement ajoutée à la fin de chaque page avant d'être entrelacée puis encodée de façon convolutionnelle. Une page encodée possède un total de 240 symboles de données plus dix symboles de préambule et est transmise sur une période d'une seconde (voir Figure 4.3). On compte 25 pages dans chacune des sous-frames et 24 sous-frames dans un message de navigation complet. Un message de navigation complet contient donc 600 pages pour une durée de transmission totale de dix minutes.



**Figure 4.3 Segmentation du message de navigation Galileo E1.**

Les quatre premières pages de chaque sous-trame servent à transmettre les éphémérides ainsi que les paramètres de correction d'horloge et ionosphérique. Les cinquième et sixième pages contiennent quant à elle de l'information concernant le temps d'émission du message de navigation (TOW). Enfin, les cinq dernières pages sont employées pour l'almanach (voir Figure 4.4) (ESTEC, 2007). Il faut noter que les pages 7 à 20 sont des pages réservées pour les développements futurs de Galileo. Il est possible de consulter (ESTEC, 2007) pour plus d'information concernant le message de navigation de GIOVE-A.

0s		25s		550s		575s		
Time	Page #	Subframe 1	Page #	Subframe 2	Page #	Subframe 23	Page #	Subframe 24
0 s	1	Ephemeris	26	Ephemeris,	551	Ephemeris	576	Ephemeris,
1 s	2	(packets 1...4)	27	Clock correction,	552	(packets 1...4)	577	Clock correction,
2 s	3		28	Iono correction	553		578	Iono correction
3 s	4		29	(packets 5...8)	554		579	(packets 5...8)
4 s	5	Time ref + etc	30	Time ref + etc	555	Time ref + etc	580	Time ref + etc
5 s	6	(packets 3 + 4)	31	(packets 1 + 2)	556	(packets 3 + 4)	581	(packets 1 + 2)
6 s	7		32		557		582	
7 s	8	Spare	33	Spare	557	Spare	583	Spare
...	...	...	...	...	...	...	...	...
18 s	19		44		569		594	
19 s	20		45		570		595	
20 s	21		46		571		596	
21 s	22	See additional	47	See additional	572	See additional	597	See additional
22 s	23	description	48	description	573	description	598	description
23 s	24		49		574		599	
24 s	25		50		575		600	
One frame on		E1-B		...				

**Figure 4.4 Structure des sous-trames du message de navigation Galileo E1.**  
Tiré de (ESTEC, 2007)



## **4.2 Prototypage rapide d'un récepteur Galileo E1**

Lors du développement des canaux de réception Galileo E1, un des principaux objectifs était de réutiliser un maximum de composantes provenant du récepteur GPS développé. Cette approche s'est avérée très intéressante vu le grand nombre de similarités existantes entre les signaux Galileo E1 et GPS L1 C/A. Par exemple, puisque les deux signaux partagent la même fréquence porteuse, la tête RF que nous avons développée a pu être directement employée pour le récepteur Galileo E1. Par la même occasion, un seul CAN a pu être utilisé pour la numérisation de tous les signaux (GPS et Galileo) de façon simultanée.

La majorité des composantes développées au niveau du FPGA ont pu aussi être réutilisées, avec quelques variations afin de permettre la démodulation des signaux BOC. Il en est de même au niveau du microcontrôleur embarqué. Il faut toutefois noter qu'aucun mécanisme de décodage des données Galileo n'a été implémenté jusqu'à ce jour au niveau du microcontrôleur. Ceci est dû à l'encodage convolutionnel des données qui empêche l'extraction du message de navigation en temps réel. Le développement d'un décodeur de Viterbi (Proakis, 2001) devrait toutefois permettre de contourner le problème. Pour le moment, les données reçues du satellite GIOVE-A sont décodées en post-traitement à l'aide du logiciel Matlab.

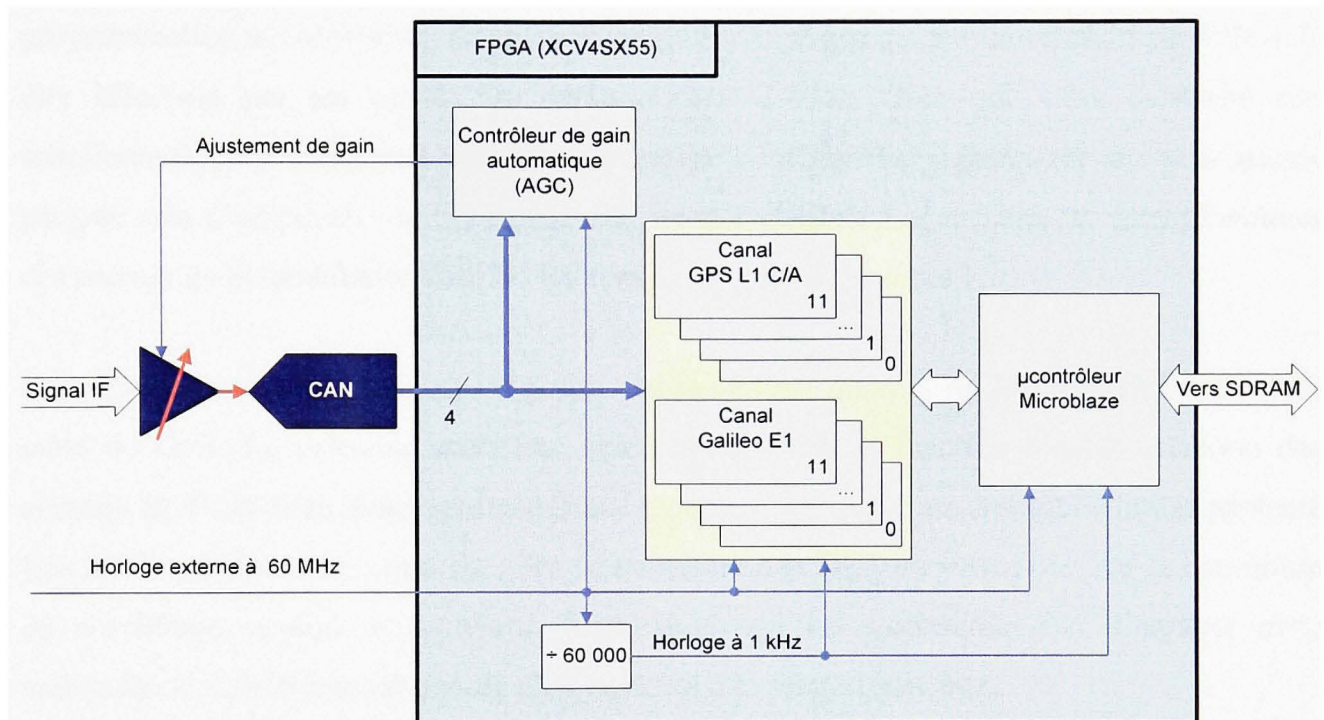
Les sections qui suivent décrivent les travaux réalisés au niveau du récepteur de navigation afin de démoduler les signaux Galileo E1. Une évaluation des ressources utilisées au sein du FPGA est aussi présentée à la fin de ce chapitre et permet de comparer la taille des canaux Galileo à ceux développés pour le GPS.

## **4.3 Mise en œuvre des canaux de démodulation en fréquence intermédiaire**

### **4.3.1 Vue d'ensemble**

Un total de 12 canaux de démodulation Galileo E1 ont été mis en œuvre au niveau du FPGA. Une fréquence d'échantillonnage de 60 MHz a une fois de plus été choisie et la résolution en

entrée des canaux a été fixée à 4 bits. En examinant la Figure 4.5, on remarque en fait que la seule véritable différence par rapport au récepteur GPS développé est l'ajout des canaux de démodulation Galileo. L'AGC, le diviseur d'horloge et le microcontrôleur sont donc demeurés les mêmes.

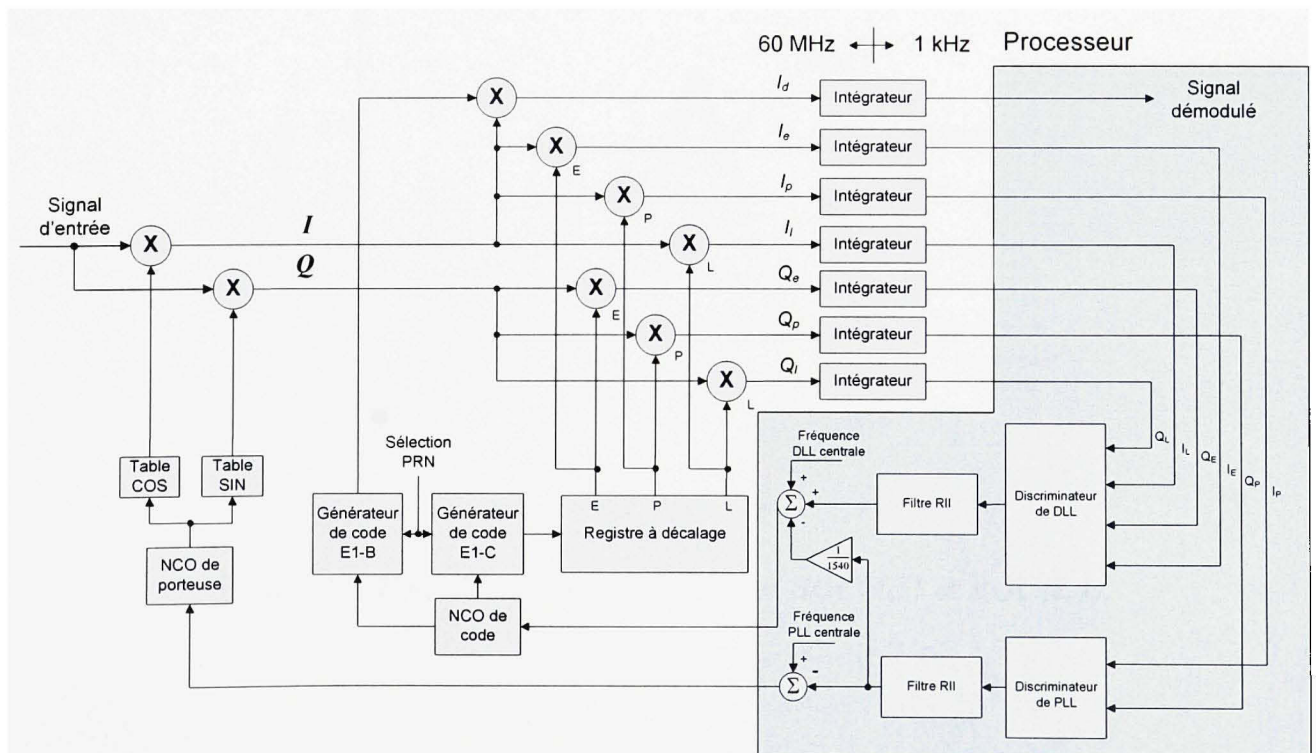


**Figure 4.5** *Architecture IF générale du récepteur GNSS développé.*

Les canaux Galileo développés sont très semblables à leurs homologues GPS. Tel que mentionné précédemment, un maximum de composantes ont pu être réutilisées, telles que les intégrateurs, les multiplicateurs et les NCO par exemple. De nouveaux générateurs de codes ont dû cependant être développés afin de reproduire les codes d'étalements E1-B et E1-C transmis par le satellite GIOVE-A. Six corrélateurs ont été implémentés au niveau du canal pilote afin de réaliser l'acquisition et la poursuite des signaux, et un corrélateur supplémentaire a été ajouté de manière à démoduler le message de navigation présent sur le canal de données. Ceci porte le nombre total de corrélateurs à sept tel qu'illustré à la Figure 4.6 (huit en incluant le corrélateur de mesure de la fonction de corrélation, corrélateur qui n'est pas affiché afin d'alléger la figure).

Le programme de poursuite implanté au niveau du microcontrôleur embarqué est exactement le même que celui des canaux GPS. Les mêmes discriminateurs ont été employés, à savoir l'arctangente pour la PLL et l'enveloppe AMRN pour la DLL. Les temps d'intégration et les bandes de bruit des filtres de boucle sont aussi demeurés identiques aux canaux GPS (ces paramètres sont toutefois indépendants des canaux GPS et peuvent être modifiés par programmation si nécessaire). Pour y parvenir, ceci implique qu'une corrélation partielle a dû être effectuée par les canaux de démodulation Galileo. Bien que cette approche soit fonctionnelle, il y aurait éventuellement avantage à intégrer les signaux sur une plus grande période afin d'augmenter la robustesse des canaux Galileo. Les principales caractéristiques des canaux de démodulation Galileo E1 sont énumérées à l'annexe 1.

Le programme d'acquisition des signaux Galileo E1 est quant à lui légèrement différent de celui du GPS. Le principal problème venait du fait que la fonction d'autocorrélation des signaux BOC possède plusieurs maximums locaux. Il y a donc une ambiguïté qui se présente lors de l'acquisition des signaux. Afin de permettre à la DLL de verrouiller sur le maximum de corrélation central, nous avons donc développé une technique afin d'enrayer cette ambiguïté. Ce problème est abordé plus en détail à la section suivante.

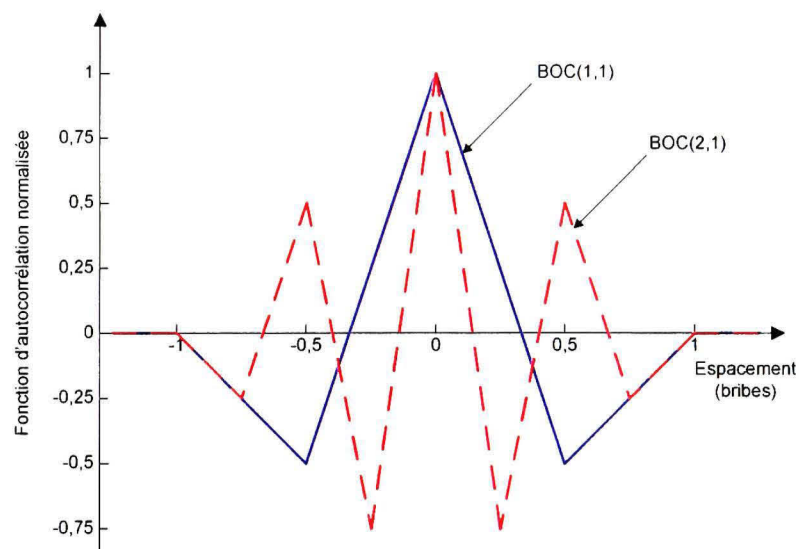


**Figure 4.6** Architecture d'un canal de démodulation Galileo E1.

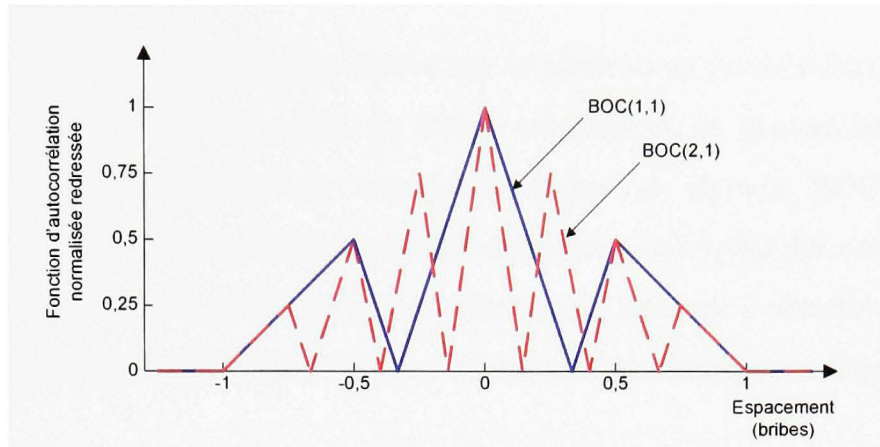


### 4.3.2 Conception de corrélateurs BOC universels

Tel que mentionné précédemment, le principal problème avec les signaux BOC est que leur fonction d'autocorrélation possède plusieurs maximums locaux (voir Figure 4.7 et Figure 4.8). Puisque le rôle de la DLL est de poursuivre le maximum de corrélation, il est donc fort probable que cette dernière se synchronise par erreur sur un des maximums adjacents au lieu du maximum de corrélation central. Dans le cas d'une modulation BOC(1,1), deux maximums adjacents sont présents; les probabilités de verrouiller sur un de ces maximums se situent donc autour de deux sur trois. Les probabilités sont encore plus grandes lorsque la fréquence de la sous-porteuse augmente par rapport à celle du code. On parle par exemple de six chances sur sept dans le cas d'une modulation BOC(2,1) (une telle modulation compte six maximums adjacents). Or, une telle erreur de verrouillage de la DLL se traduit par un sévère décalage des mesures de pseudo-distances et ceci se répercute directement au niveau de la solution de navigation. Ce décalage est de l'ordre de 147 m pour les signaux Galileo E1 transmis par GIOVE-A (soit la demi-longueur d'onde d'une bribe).

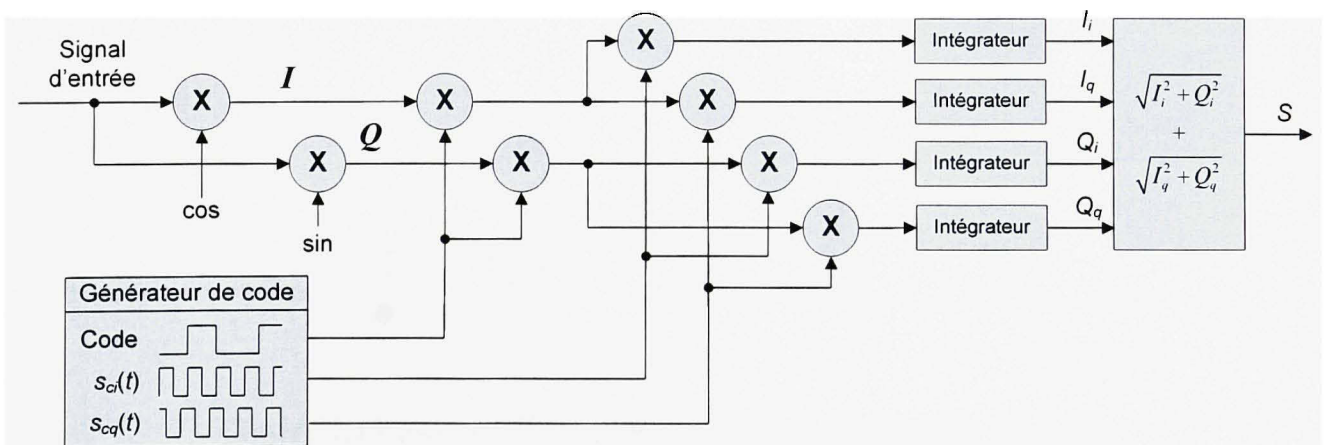


**Figure 4.7** Fonctions d'autocorrélation BOC(1,1) et BOC(2,1).



**Figure 4.8 Fonctions de cross-corrélation  $BOC(1,1)$  et  $BOC(2,1)$  telles que mesurées en sortie des corrélateurs.**

Plusieurs méthodes ont été développées afin d'éliminer ce problème d'ambiguïté de corrélation. La méthode la plus populaire est sans doute l'ajout de corrélateurs très en avance (VE – *Very Early*) et très en retard (VL – *Very Late*) telle que décrite dans (Borre, 2006). Bien que cette méthode semble fonctionner adéquatement, elle nécessite beaucoup de ressources supplémentaires pour implémenter les corrélateurs VE et VL. Or, ces ressources peuvent être considérées comme étant gaspillées puisque les corrélateurs en question ne servent que lors du verrouillage de la DLL. Afin de supprimer l'ambiguïté plus efficacement, nous avons donc entrepris de développer nos propres corrélateurs BOC.



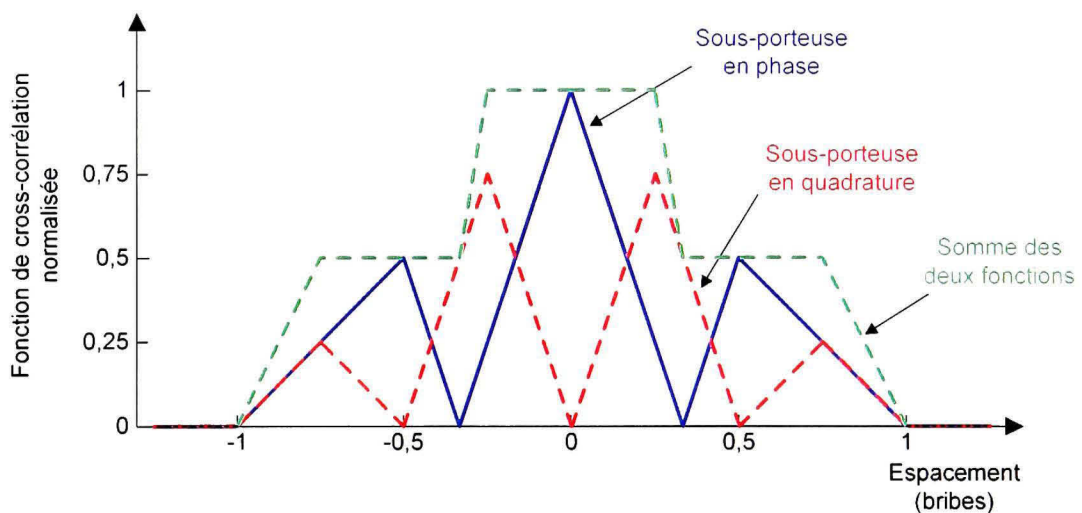
**Figure 4.9 Suppression de l'ambiguïté de corrélation par génération de sous-porteuses en phase et en quadrature.**

L'approche que nous avons employée repose sur la génération de sous-porteuses en phase,  $s_{ci}(t)$ , et en quadrature de phase,  $s_{cq}(t)$ , au niveau des canaux de démodulation Galileo E1 (voir Figure 4.9). En multipliant ces sous-porteuses par les signaux BOC(1,1) reçus, on obtient alors deux fonctions de cross-corrélation différentes en sortie des corrélateurs. Il est alors possible de combiner la sortie de ces corrélateurs de manière à obtenir une fonction de cross-corrélation sans ambiguïté, c'est-à-dire avec un seul maximum (voir Figure 4.10) :

$$S = \sqrt{I_i^2 + Q_i^2} + \sqrt{I_q^2 + Q_q^2} \quad (4.1)$$

où :

- $I$  et  $Q$  représentent les corrélateurs pour lesquels la porteuse est en phase et en quadrature respectivement
- Les indices  $i$  et  $q$  représentent les corrélateurs pour lesquels la sous-porteuse respective est en phase et en quadrature
- $S$  est un corrélateur BOC avec suppression d'ambiguïté



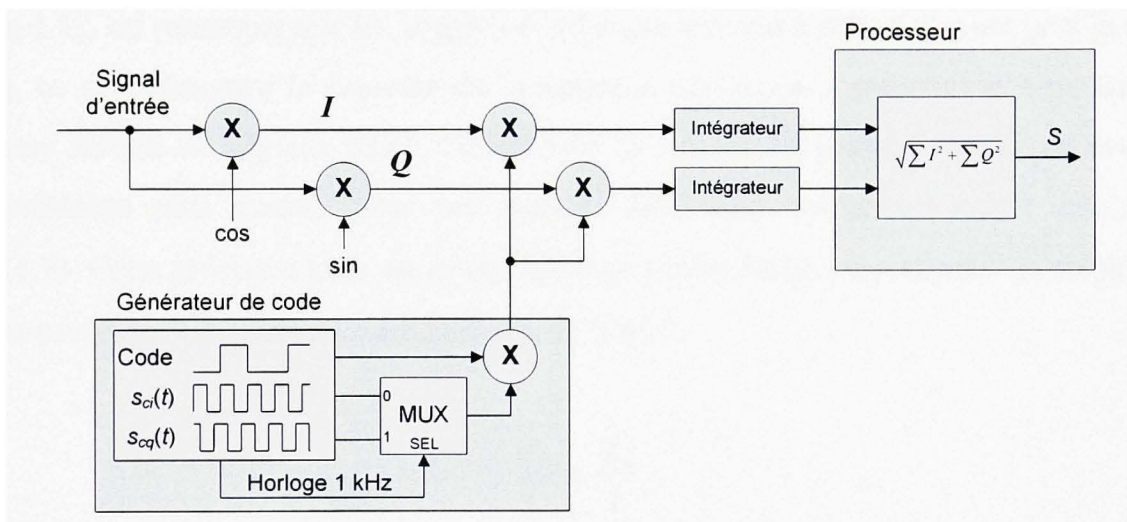
**Figure 4.10 Fonctions de cross-corrélation BOC(1,1) obtenues pour des sous-porteuses en phase et en quadrature.**

Une telle approche peut être aussi retrouvée dans (Ward, 2004). Elle nécessite néanmoins l'implémentation de 12 corrélateurs différents pour le canal pilote (quatre corrélateurs en



avance, quatre en retard et quatre en phase), ce qui est pire que pour la technique précédente avec les corrélateurs VE et VL. Nous avons donc adapté la méthode de manière à multiplexer temporellement la sous-porteuse carrée. Ainsi, la sous-porteuse en phase est employée environ une milliseconde sur deux afin de multiplier le signal reçu, alors que la sous-porteuse en quadrature est employée l'autre moitié du temps. On arrive ainsi à réduire de moitié le nombre de corrélateurs nécessaire (voir Figure 4.11). L'équation (4.1) se réécrit alors :

$$S = \sqrt{\sum I^2 + \sum Q^2} \quad (4.2)$$



**Figure 4.11** *Multiplexage des sous-porteuses en phase et en quadrature.*

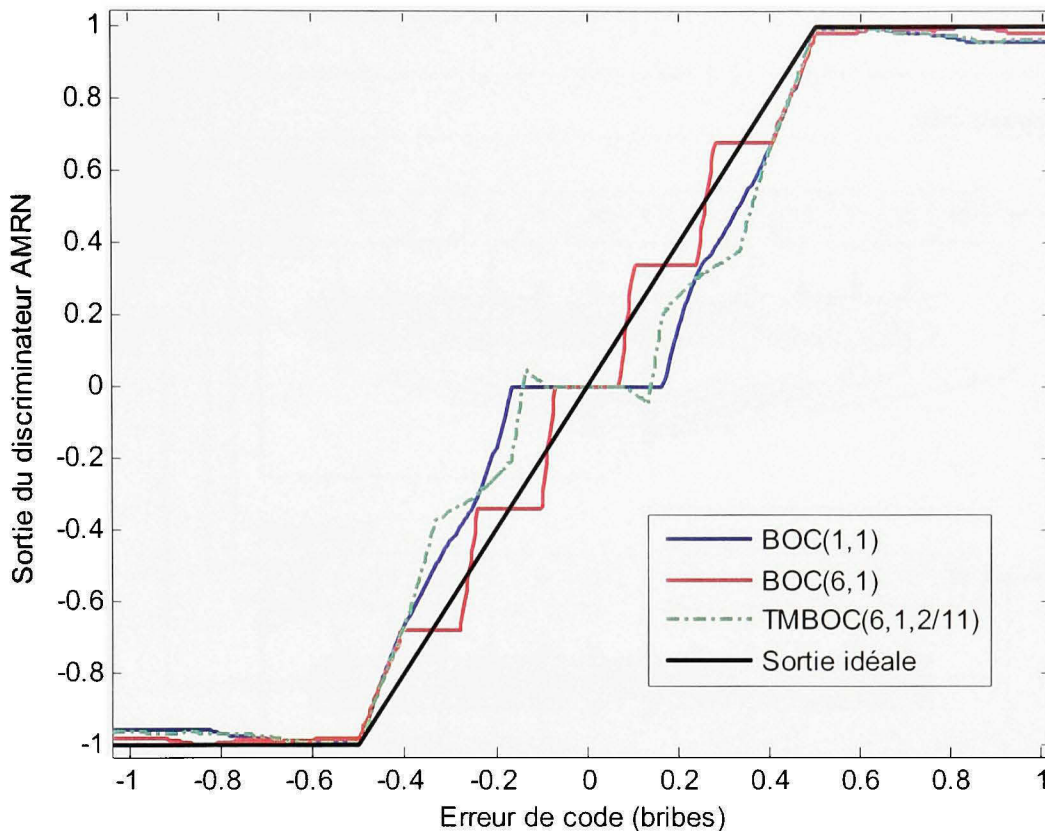
L'un des inconvénients du nouveau corrélateur développé est que le bruit est doublé à leur sortie. Ceci entraîne donc une dégradation de 3 dB du ratio SNR après démodulation. Un autre problème est que la fonction de cross-corrélation résultante forme un plateau central pour les signaux BOC(1,1). Or, ce plateau diminue sérieusement la précision de la DLL pour des corrélateurs espacés de  $\pm 0,5$  bribe, et le problème s'empire pour des espacements inférieurs. Ces deux problèmes n'ont toutefois pas de répercussion sérieuse au niveau des canaux de démodulation puisque nous n'utilisons les corrélateurs développés que lors de la phase initiale de poursuite des signaux. Les corrélateurs en question ne sont donc employés que dans le but de centrer la DLL sur le maximum principal de corrélation. Une fois cette opération accomplie, il est possible de poursuivre normalement le code à l'aide de

corrélateurs conventionnels espacés de  $\pm 0,25$  bribe ou moins (des corrélateurs espacés de  $\pm 1/20^\circ$  de bribe sont employés dans notre cas).

Bien que les corrélateurs développés visaient originalement à démoduler les signaux BOC(1,1) de GIOVE-A, une simulation a permis de montrer qu'ils pouvaient poursuivre à peu près n'importe quelle forme de signaux BOC sans avoir à subir de modification (d'où l'appellation « corrélateurs BOC universels »). La Figure 4.12 montre la sortie d'un discriminateur de type enveloppe AMRN appliqué aux corrélateurs développés. Les réponses à des signaux BOC(1,1), BOC(6,1) et TMBOC(6,1,2/11) y sont présentées. En examinant la Figure 4.12, on remarque que les courbes de réponse arrivent à suivre d'assez près la réponse idéale, ce qui démontre la capacité du corrélateur développé à poursuivre sans ambiguïté plusieurs formes de signaux BOC. Considérant la largeur du plateau central, on note aussi une précision plus grande pour les signaux BOC(6,1) comparativement aux signaux BOC(1,1). Cette précision ( $\Delta c$ ) est en réalité proportionnelle au rapport entre la fréquence de la sous-porteuse ( $f_{sc}$ ) et de celle du code ( $f_c$ ), telle que :

$$\Delta c \propto \frac{f_{sc}}{f_c} \quad (4.3)$$

Un ratio  $f_{sc}/f_c$  plus élevé permettra ainsi à la DLL de se positionner de façon plus juste sur le maximum de corrélation principal, ce qui est un avantage des corrélateurs BOC développés.

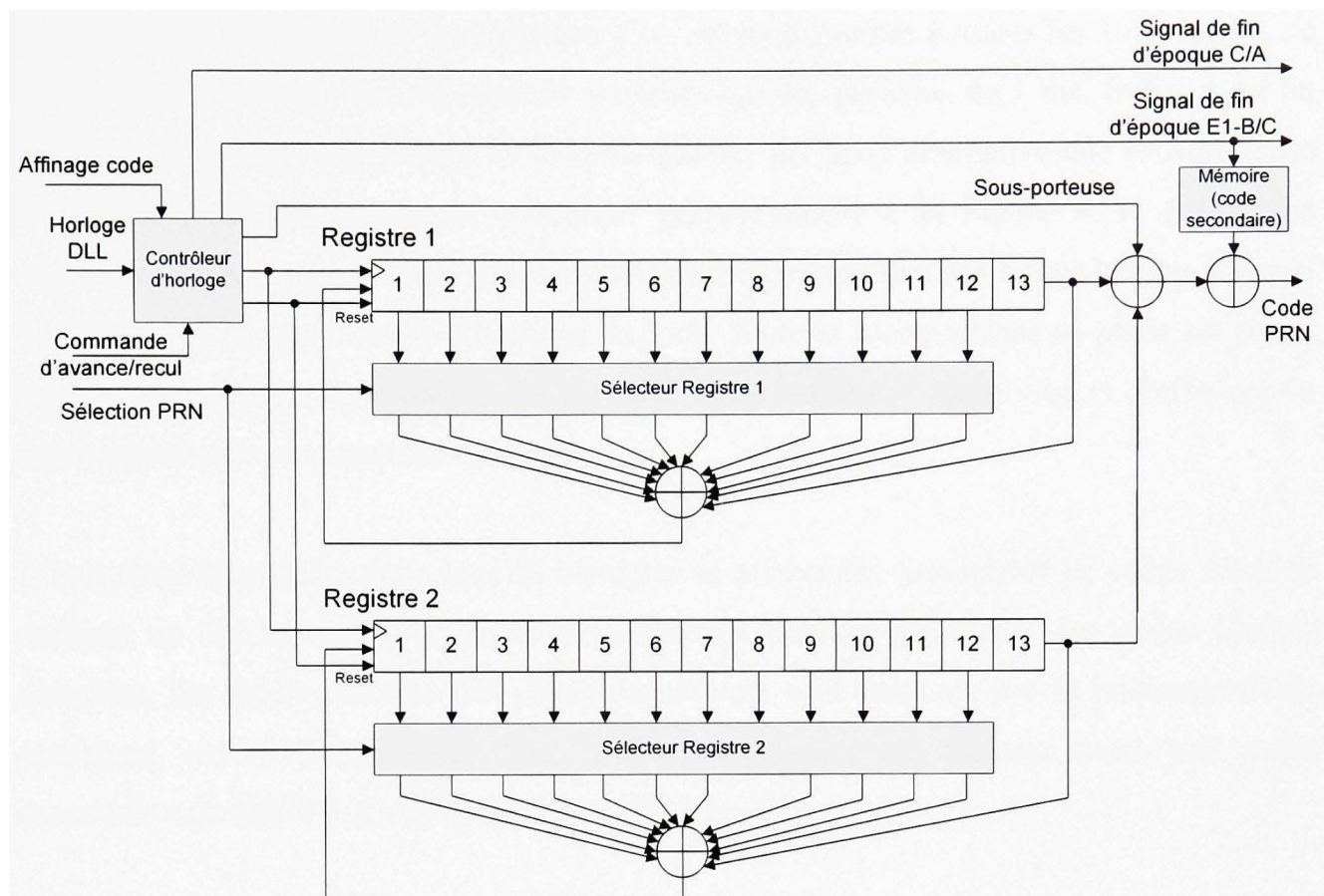


**Figure 4.12** *Erreur en sortie du discriminateur enveloppe AMRN lorsqu'appliqué aux corrélateurs BOC universels développés.*

### 4.3.3 Réalisation des générateurs de codes pseudo-aléatoires Galileo E1

Pour la génération des codes d'étalements Galileo E1-B et E1-C, nous avons opté pour une solution similaire à celle développée pour le GPS. Ainsi, deux registres à décalage de 13 bits de long sont nécessaires, en plus de deux sélecteurs distincts (voir Figure 4.13). Ces sélecteurs ont pour rôle de déterminer les rétroactions de chacun des registres en fonction du numéro PRN sélectionné. Dans le cas du satellite GIOVE-A, les rétroactions sont affichées au Tableau 4.1 (ESTEC, 2007). Une des particularités des générateurs de code Galileo E1-B et E1-C est que les registres doivent être réinitialisés de façon systématique à la fin de chacune des séquences pseudo-aléatoires. Ceci se produit à la fin des 4092 bribes de Galileo E1-B ainsi que des 8184 bribes du code primaire de Galileo E1-C. Les valeurs à l'initialisation sont aussi présentées au Tableau 4.1 pour le satellite GIOVE-A.





**Figure 4.13 Générateurs de codes pseudo-aléatoires Galileo E1-B/C.**

**Tableau 4.1**  
Paramètres des générateurs de code du satellite GIOVE-A

Signaux	Rétroactions des registres	Réinitialisation des registres	Longueur des codes (bribes)	Codes secondaires
E1-B				
Registre 1	4,5,7,9,10,13	1111111111111	4092	-
Registre 2	5,6,7,8,12,13	1101110000011		
E1-C				
Registre 1	1,3,4,13	1111111111111	8194	011100000
Registre 2	4,5,7,9,10,13	1100110000011		001010110 1100100

Tel que vu à la Figure 4.13, un module « contrôleur d'horloge » est présent au niveau des générateurs de codes Galileo développés, ceci encore une fois dans le but de permettre une avance ou un recul du code régénéré lors de la recherche des signaux. Ce contrôleur génère un signal de fin d'époque, dont la période est de 4 et 8 ms respectivement pour les canaux

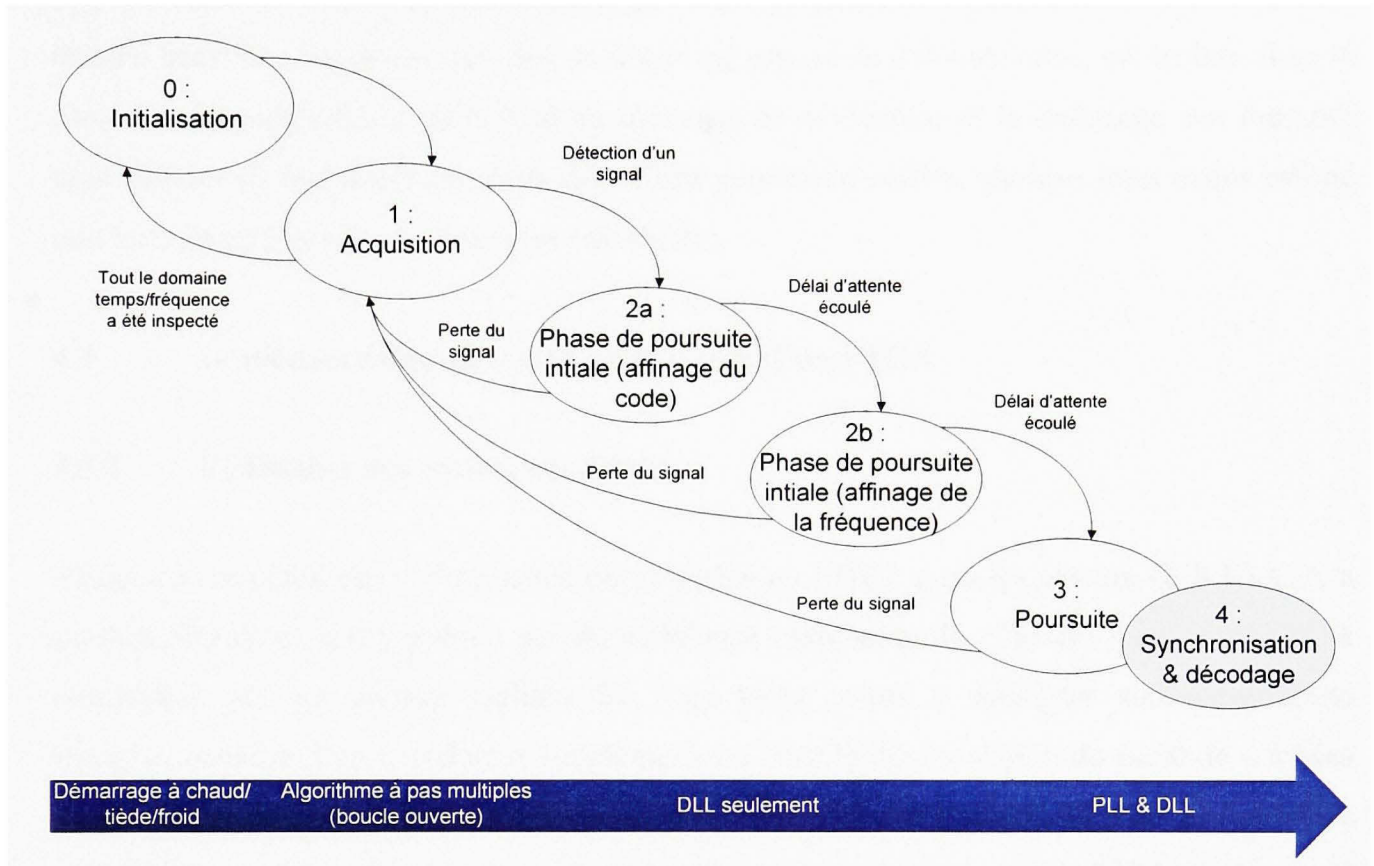
E1-B et E1-C. Un signal de fin d'époque C/A est aussi produit à toutes les 1023 bribes, ce qui permet de réaliser des corrélations partielles sur des périodes de 1 ms. Enfin, c'est au contrôleur d'horloge que revient la tâche de générer de façon alternative une sous-porteuse en phase et en quadrature, tel qu'illustré précédemment à la Figure 4.11. Ainsi, une alternance entre les deux sous-porteuses est réalisée à toutes les 1023 bribes lorsque le canal de démodulation est en mode d'affinage du code. Seule la sous-porteuse en phase est par la suite générée lors de la poursuite des signaux. Les processus d'acquisition et d'affinage du code sont abordés aux sections suivantes.

Il faut noter la présence d'un bloc de mémoire au niveau des générateurs de codes. Ce bloc contient en fait les codes secondaires nécessaires pour la génération des codes Galileo complets. De façon concrète, les codes secondaires sont transmis par le microcontrôleur embarqué, qui se charge à son tour de les synchroniser aux signaux reçus. Les codes secondaires de GIOVE-A sont présentés au Tableau 4.1.

#### **4.4 Particularités du mécanisme d'acquisition des signaux Galileo E1**

Vu la simplicité d'implémentation du mécanisme d'acquisition des signaux GPS C/A, nous avons choisi de garder le même pour l'acquisition des signaux Galileo. Or, la longueur des codes d'étalement Galileo E1-C a posé un problème à ce niveau. En ne considérant que les codes primaires par exemple, on parle alors d'un temps de recherche de  $1,33 \times 8 \text{ ms} \times 8192 \text{ bribes} \times 19 \text{ cases Doppler}$ , pour un total de plus de 27 min (la constante 1,33 est liée au temps passé dans chacune des cellules de recherche - voir chapitre 3, section 3.9.2). Nous avons donc été obligés de réduire la période d'intégration à 1 ms plutôt que 8 ms afin d'obtenir un temps de recherche raisonnablement petit (soit 3,5 minutes). Étant donné que GIOVE-A est le seul satellite Galileo présentement en orbite, nous avons aussi fait en sorte de rechercher le signal de GIOVE-A sur les 12 canaux Galileo en parallèle. Chacun des canaux se voit donc assigner une case Doppler différente afin d'obtenir un temps de recherche final d'environ 29 secondes. Une approche par FFT serait toutefois suggérée une

fois la constellation Galileo déployée afin d'accélérer la recherche des signaux et d'accroître la sensibilité du récepteur.



**Figure 4.14** *Diagramme d'état du programme d'acquisition et de poursuite Galileo.*

Une étape supplémentaire a dû être ajoutée au programme d'acquisition et de poursuite des signaux Galileo afin de permettre à la DLL de se centrer sur le maximum de corrélation central. Cette étape, appelée « affinage du code », s'insère directement entre les étapes d'acquisition et d'affinage à proprement dit (renommée « affinage de la fréquence » afin d'éviter toute confusion) du programme d'acquisition et de poursuite (voir Figure 4.14). Le programme se déroule de la même manière que pour le GPS une fois l'affinage du code complété.

Puisque les bits de données ont la même période que la séquence pseudo-aléatoire E1-B, le mécanisme de synchronisation au message de navigation est donc beaucoup plus simple que pour le GPS. L'approche que nous avons choisie consiste en fait à rechercher le préambule



au sein du flot de symboles reçus. Or, on sait que le préambule se répète à tous les 250 symboles exactement. Puisque les données peuvent prendre l'apparence du préambule, l'astuce consiste alors à vérifier la périodicité des préambules détectés. Si un préambule est détecté trois fois de suite et que chacun d'eux est espacé de 250 symboles, on déclare alors le canal de démodulation synchronisé au message de navigation et le décodage des données peut débuter (il faut noter que deux détections pourraient suffire, quoique nous ayons estimé que trois garantissent une robustesse maximale).

## **4.5 Implémentation en temps réel au sein d'un FPGA**

### **4.5.1 Utilisation des ressources FPGA**

Puisque la majorité des composantes développées en VHDL pour les canaux GPS L1 C/A a pu être réutilisée, il n'y a donc pas de différence marquante du côté des ressources FPGA employées par les canaux Galileo E1. Les seuls points à souligner sont sans doute l'implémentation d'un corrélateur supplémentaire pour la démodulation du canal de données E1-B, ce qui a fait grimper les ressources au niveau des intégrateurs, en plus de l'ajout de complexité au niveau du générateur de codes. On compte ainsi un total de 820 LUT par canal Galileo, comparativement à 660 par canal GPS, ce qui équivaut à une augmentation de 24,2% des ressources (voir Tableau 4.2).

La Figure 3.65 présente la distribution des ressources FPGA au niveau du récepteur GPS et Galileo hybride développé, alors que la Figure 3.66 montre la répartition des ressources au sein d'un canal de démodulation Galileo.

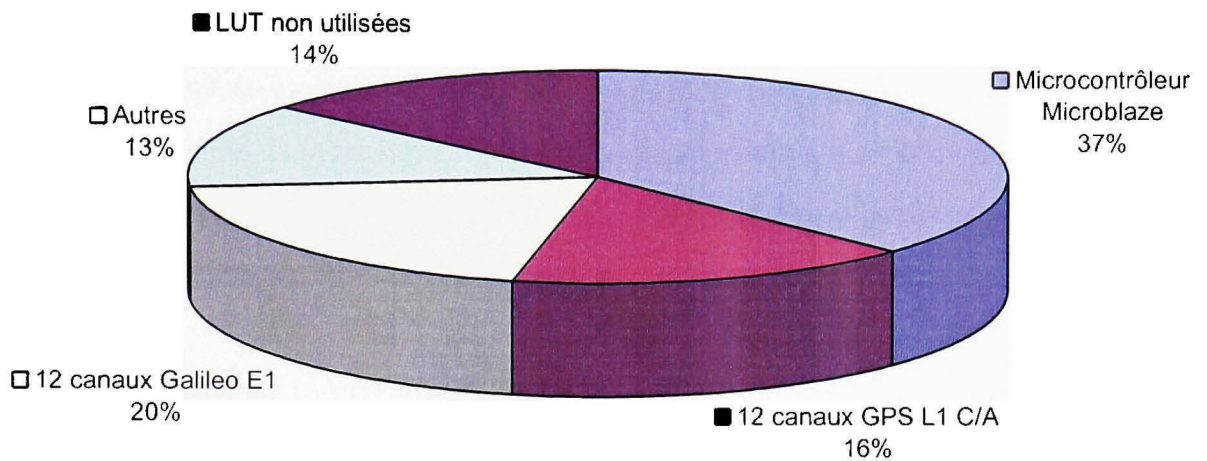
Puisque les canaux Galileo sont un peu plus volumineux que leurs homologues GPS, ils occupent donc 20% de toutes les LUT disponibles au sein du FPGA Virtex-4 SX-55 utilisé, comparativement à 16% pour les canaux GPS (voir Figure 3.65). Seulement 14% des ressources totales sont ainsi encore disponibles après l'implémentation des 12 canaux Galileo, ce qui permettrait d'ajouter 8 canaux Galileo supplémentaires ou 10 canaux GPS. Un examen de la Figure 3.66 permet néanmoins de constater que 54% des ressources d'un

canal sont allouées aux intégrateurs. Une réduction d'au moins 50% de la taille des canaux serait donc encore possible tel que décrit au chapitre 3, permettant du coup d'implémenter plus de 60 canaux GNSS au sein du FPGA employé.

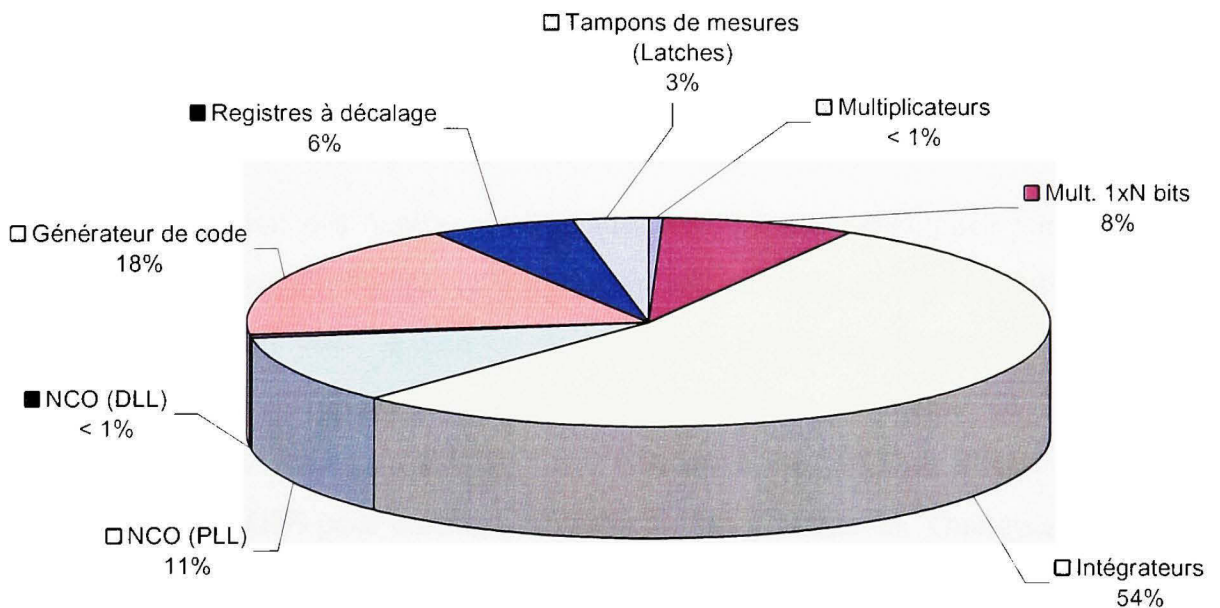
Tableau 4.2  
Utilisation des ressources FPGA par le récepteur GPS et Galileo

	LUT	DFF	BRAM	DSP48
Microcontrôleur Microblaze	18 338	10 490	73	7
Diviseur d'horloge	77	34	0	0
AGC	164	205	0	0
Canal de calibration	327	182	0	2
Autres (code Lyrtech)	5 595	3 373	3	1
Canal GPS L1 C/A	660	741	1	3
Canal Galileo E1	820	901	1	3
- Multiplicateurs	4	0	0	2
- Mult. 1xN bits	64	56	0	0
- Intégrateurs	440	280	0	0
- NCO (PLL)	87	53	0	0
- NCO (DLL)	3	21	0	1
- Générateur de code	148	99	0	0
- Registres à décalage	50	53	1	0
- Tampons de mesures ( <i>Latches</i> )	24	339	0	0
Total (12 canaux GPS + 12 Galileo)	42 261	33 988	100	82
Ressources totales (Virtex-4 SX-55)	49 152	49 152	320	512
Ressources employées (%)	86,0	69,1	31,3	16,0
Portes logiques équivalentes*	7,2 millions			

\* Estimation donnée par les outils de Xilinx



**Figure 4.15 Répartition totale des LUT pour le récepteur hybride GPS et Galileo.**



**Figure 4.16 Répartition des LUT au niveau d'un canal de démodulation Galileo E1.**

#### 4.5.2 Charges de calcul du microcontrôleur embarqué

La charge de calcul exigée par les canaux Galileo au niveau du microcontrôleur est relativement semblable à celle des canaux GPS. Cela est dû au fait que les programmes d'acquisition et de poursuite sont pratiquement semblables dans les deux cas. Le nombre maximal de canaux pouvant être exécutés en mode poursuite est donc de 23 en simultanée,



tel que mentionné au chapitre 3. Or, ceci ne pose pas problème pour le moment puisqu'il n'y a qu'un seul satellite Galileo en orbite. Les optimisations décrites au chapitre 3 devront cependant être mises en œuvre afin de garantir une opération adéquate du récepteur lorsque la constellation Galileo sera complète (ou dès que d'autres signaux situés sur les autres fréquences de radionavigation seront intégrés à l'architecture actuelle).

#### **4.6 Conclusion et recommandations**

Dans ce chapitre, la structure des signaux Galileo E1 transmis par le satellite GIOVE-A a été présentée de façon détaillée. La mise en œuvre des canaux de démodulation Galileo E1 a ensuite été abordée en mettant l'accent sur les nouveautés apportées afin de permettre l'acquisition et la poursuite des signaux BOC. La possibilité de réutiliser plusieurs composantes déjà développées pour les canaux de démodulation GPS L1 C/A a aussi été discutée à plusieurs reprises.

Grâce à la réutilisation des composantes et des programmes développés pour le récepteur GPS, nous avons pu mettre en œuvre très rapidement une série de canaux Galileo E1 au sein du FPGA employé. De plus, la quantité de ressources FPGA employées par de tels canaux est demeurée relativement petite, avec une augmentation de moins de 25% des LUT comparativement aux canaux GPS L1 C/A. Il a donc été possible d'intégrer 12 canaux Galileo au récepteur GPS pour créer un récepteur GNSS à 24 canaux. Quelques optimisations permettraient même éventuellement d'augmenter le nombre de canaux GNSS à plus de 60 au total. Pour y parvenir, les mêmes recommandations que celles émises au chapitre 3 s'appliquent, à savoir qu'il faut remplacer les accumulateurs présents au sein des intégrateurs par des blocs DSP48.

Certaines améliorations devraient aussi être apportées aux canaux de démodulation afin de profiter de tout le potentiel offert par Galileo. Parmi ces améliorations, on note le développement d'un mécanisme d'acquisition par FFT afin de permettre une acquisition rapide des signaux. Une augmentation du temps d'intégration lors de l'acquisition et de la

poursuite devrait être aussi envisagée afin d'augmenter la robustesse des canaux Galileo et permettre la navigation en milieux difficiles, tels qu'en forêt ou à l'intérieur des bâtiments par exemple.

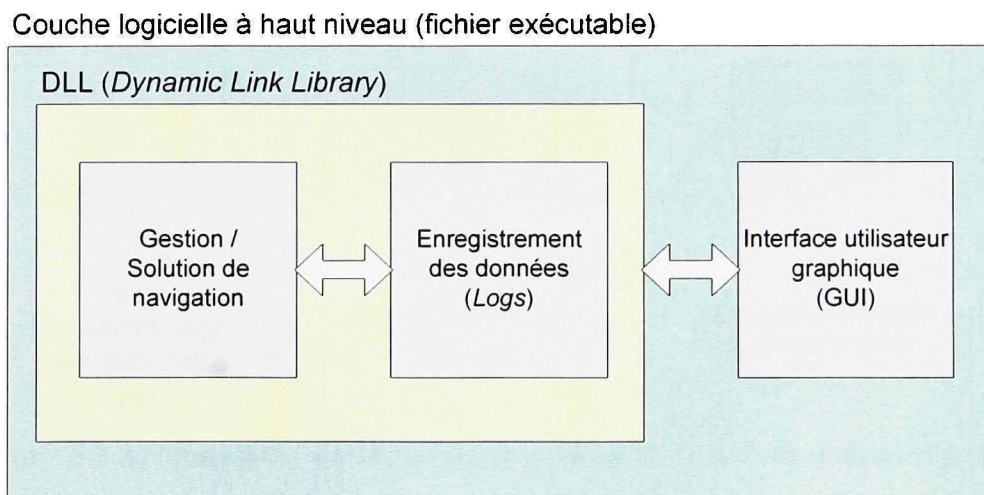
Le chapitre qui suit termine la présentation du récepteur développé en abordant la partie logicielle à haut niveau implantée au sein de l'ordinateur PC. Le chapitre 6 présente quant à lui les tests effectués à l'aide du récepteur ainsi que les performances de ce dernier. Bien que le satellite GIOVE-A ne transmette aucun message valide pour le moment, quelques tests sont aussi présentés afin de montrer le bon fonctionnement des canaux Galileo développés.

## CHAPITRE 5

### ARCHITECTURE LOGICIELLE À HAUT NIVEAU

#### 5.1 Vue d'ensemble de la couche logicielle à haut niveau

Le calcul d'une solution de navigation relativement précise exige d'être réalisé à l'aide d'algorithmes à double précision (64 bits). Pour plus de facilité, il est généralement souhaitable d'avoir recours à des algorithmes à point flottant plutôt qu'à point fixe, spécialement pour le calcul de la position des satellites à partir des éphémérides, tel que vu au chapitre 2. Puisque le microcontrôleur embarqué au sein du FPGA, le Microblaze, n'offrait pas cette capacité de calcul en point flottant de double précision, nous avons donc préféré transférer toutes les informations recueillies au niveau des canaux de démodulation vers un ordinateur de type PC. Un programme, appelé « couche logicielle à haut niveau », y est alors exécuté de manière à calculer la solution de navigation en temps réel. Le schéma bloc illustré à la Figure 5.2 présente les principaux segments de la couche logicielle à haut niveau.

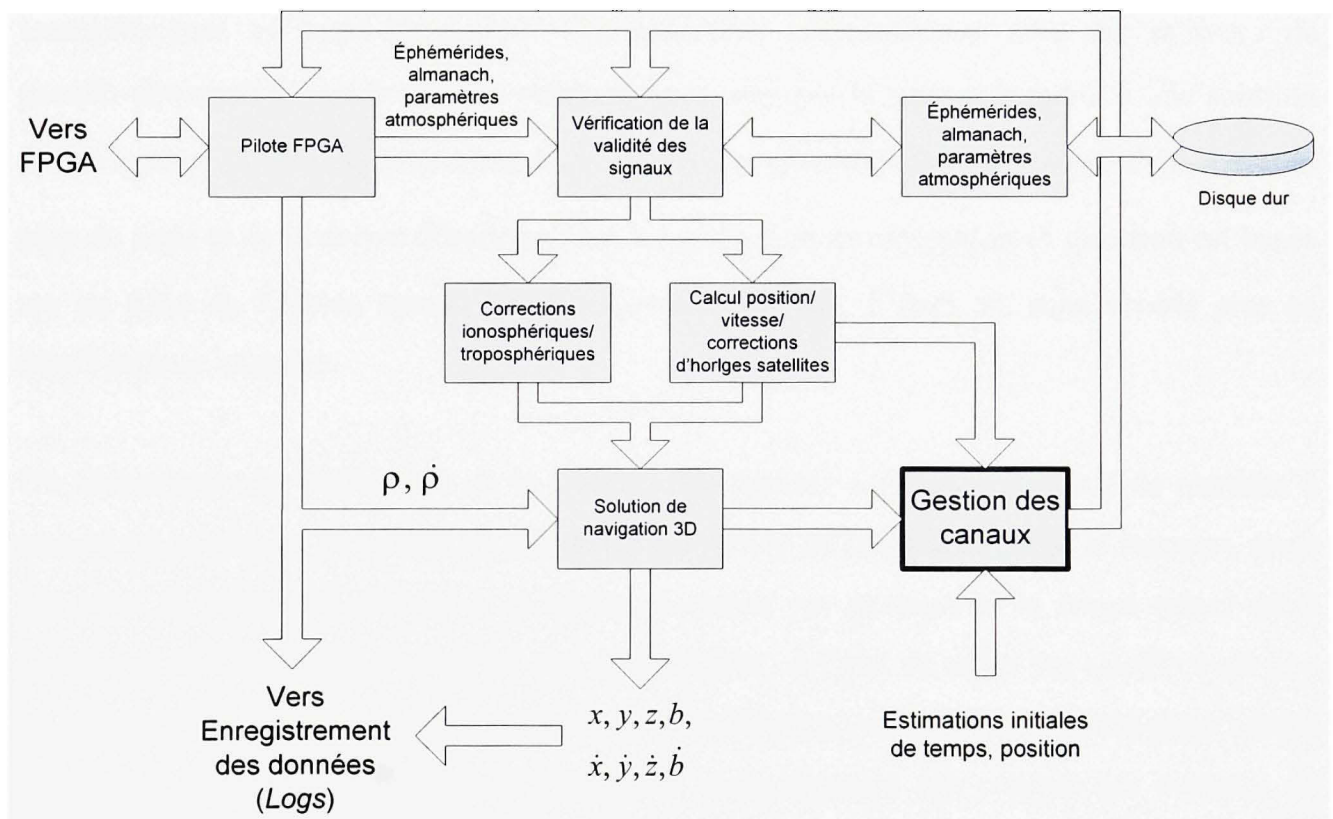


**Figure 5.1** Schéma bloc de la couche logicielle à haut niveau.

La couche logicielle consiste à la base en un fichier exécutable faisant office d'interface graphique utilisateur (GUI – *Graphical User Interface*). Cette interface a été développée en



langage Visual Basic (VB), un langage de haut niveau permettant de réaliser rapidement des interfaces utilisateurs conviviales (plus de détails sont donnés à l'annexe 2 concernant l'interface graphique). Vu la puissance de calcul relativement restreinte du VB, l'interface fait donc appel à une bibliothèque de liens dynamiques (DLL – *Dynamic Link Library*) programmée en langage C. Les opérations réalisées par cette DLL sont d'une part de gérer le récepteur de navigation en temps réel et de calculer une solution de navigation. Ce segment est illustré plus en détail à la Figure 5.2. La DLL a aussi pour rôle de collecter toutes les informations reçues et calculées par le récepteur et de les sauvegarder sur le disque dur du PC (cette action est appelée *log* en anglais – voir l'annexe 9 pour une description des fichiers de *log* pouvant être générés).



**Figure 5.2 Schéma bloc du segment de gestion et de solution de navigation.**

En examinant la Figure 5.2, on remarque qu'un module de gestion des canaux s'occupe de piloter la majeure partie de la couche logicielle à haut niveau. Ce module se base initialement sur une estimation de la position et du temps du récepteur afin de déterminer grossièrement la

position des satellites. On parlera alors de démarrage à froid, à tiède ou à chaud tout dépendamment de la validité de ces estimés ainsi que de la disponibilité des éphémérides et des almanachs sauvegardés dans la mémoire du récepteur. Une série de commandes est par la suite acheminée au FPGA en fonction du type de démarrage effectué. Plus de détails sont donnés à la section suivante concernant le module de gestion des canaux et les types de démarrages.

Un second module, appelé « pilote FPGA », s'occupe de transmettre les commandes reçues du module de gestion vers le FPGA. Son rôle est aussi d'extraire du FPGA les éphémérides, almanach et paramètres atmosphériques téléchargés afin de calculer la position et la vitesse des satellites, en plus d'une série de paramètres de corrections (corrections d'horloges, ionosphériques et troposphériques). Ces éléments, conjointement avec les mesures de pseudo-distances ( $\rho$ ) et de vitesses relatives ( $\dot{\rho}$ ), sont par la suite acheminés à une solution de navigation 3D afin de déterminer la position ( $x, y, z$ ) et la vitesse du récepteur ( $\dot{x}, \dot{y}, \dot{z}$ ), en plus du biais et de la dérive d'horloge ( $b, \dot{b}$ ). La solution de navigation en question est basée sur un filtre de Kalman étendu (EKF - *Extended Kalman Filter*), un sujet abordé plus en détail dans ce chapitre.

Un mécanisme de vérification de la validité des signaux a été mis en place de manière à assurer un minimum de qualité au niveau de la solution de navigation. Pour le moment, seuls des tests de base sont réalisés à ce niveau, à savoir par exemple si le temps actuel est à l'intérieur de l'intervalle de validité des éphémérides, si l'état de santé des satellites est bon ou encore si un drapeau d'alerte est levé. Les vérifications en question ne sont donc basées que sur le contenu du message de navigation et ne tiennent pas compte des mesures de pseudo-distances et de vitesses relatives. En incluant ces mesures, il serait éventuellement possible de développer un mécanisme de vérification autonome de l'intégrité des signaux (RAIM – *Receiver Autonomous Integrity Monitoring*) afin d'assurer une plus grande fiabilité au niveau du récepteur de navigation développé. Le lecteur est invité à consulter (Parkinson et Spilker, 1996) pour plus d'information sur le sujet.



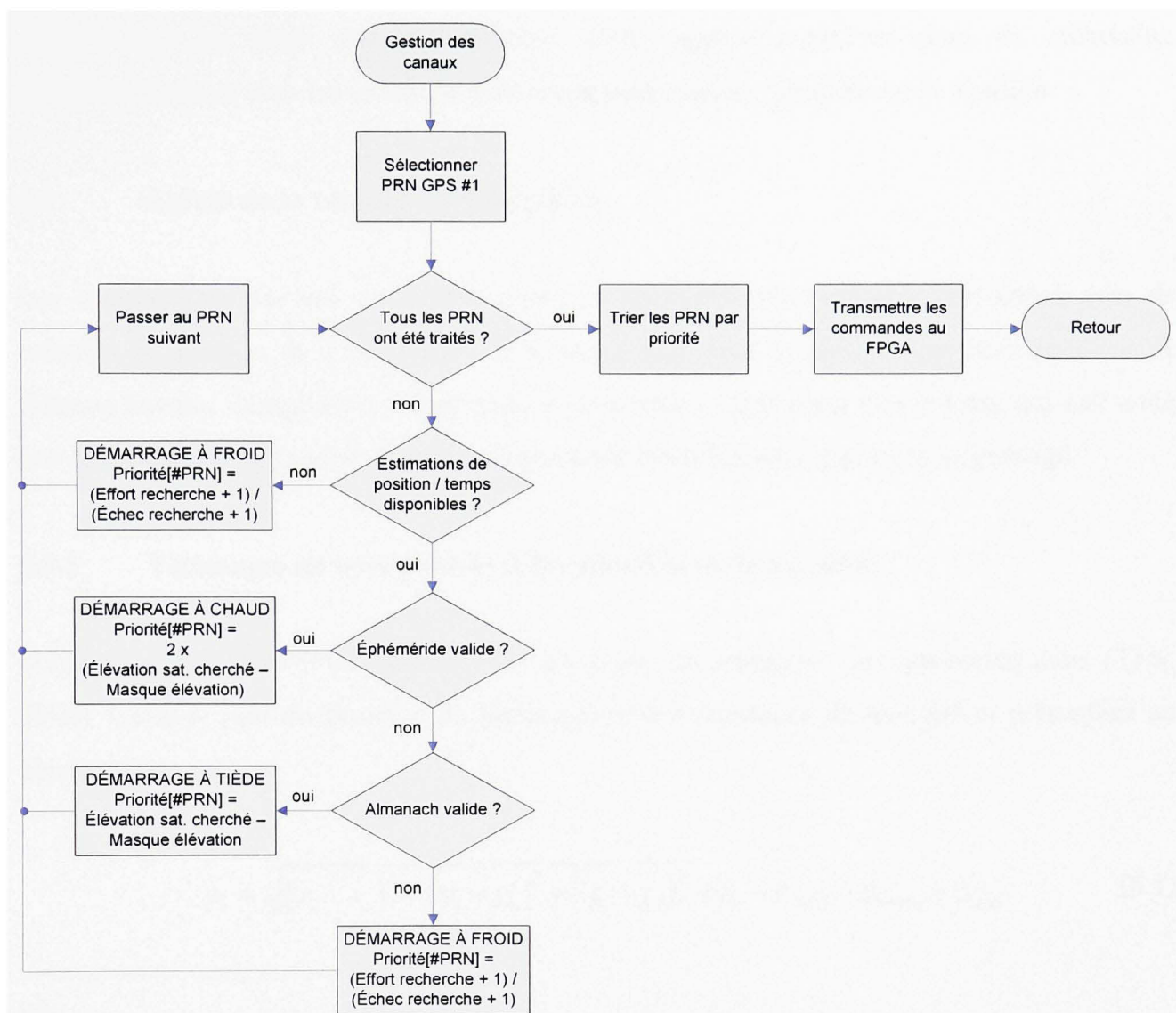
## 5.2 Gestion des canaux GNSS

Le mécanisme de gestion des canaux se base sur deux paramètres principaux lors du démarrage du récepteur : une estimation de la position et une estimation du temps. L'estimation de temps provient directement de l'horloge du PC, alors que la dernière position calculée par le récepteur est employée pour estimer la position lors du démarrage. Bien sûr, ces deux paramètres peuvent être aussi ajustés par un utilisateur de manière à accélérer la phase d'acquisition du récepteur. Grâce à ces estimations, il est alors possible de déterminer si les éphémérides et l'almanach sauvegardés dans la mémoire du récepteur (c'est-à-dire sur le disque dur du PC) sont toujours valides. Typiquement, on parlera d'un intervalle de validité de quatre heures pour les éphémérides et de six mois pour l'almanach.

Advenant que les éphémérides ou l'almanach soient valides, il devient alors possible d'estimer la position des satellites par rapport à celle du récepteur. Une estimation des décalages Doppler des satellites peut être aussi réalisée afin d'accélérer l'acquisition des signaux. De façon générale, tous les satellites au dessus d'un certain masque d'élévation, que nous avons fixé à  $5^\circ$  au dessus de l'horizon, sont recherchés par le récepteur, mais une priorité est accordée à ceux au zénith advenant que le nombre de satellites visibles dépasse le nombre de canaux disponibles.

Si les éphémérides et l'almanach ne sont pas disponibles, ou encore que l'horloge ou la position du récepteur au démarrage soit invalide, le mécanisme de gestion des canaux procédera plutôt par une recherche systématique de tous les satellites. On qualifie cette opération de démarrage à froid, par opposition au démarrage à chaud ou à tiède lorsque les éphémérides ou l'almanach sont disponibles. Mentionnons ici qu'aucun consensus n'existe au niveau des termes démarrages « à chaud » et « à tiède » et que leur définition peut dévier d'un fabricant à l'autre. Nous avons donc choisi de définir le démarrage à tiède comme étant une phase d'acquisition initiale des signaux GNSS aidée des informations provenant de l'almanach, alors que le démarrage à chaud est une phase d'acquisition des signaux aidée des informations provenant des éphémérides (donc plus précise et rapide).





**Figure 5.3** *Ordinogramme simplifié du mécanisme de gestion des canaux.*

La Figure 5.3 illustre le fonctionnement simplifié du mécanisme de gestion des canaux. On peut y voir qu'une priorité deux fois plus grande est attribuée aux satellites dont l'éphéméride est valide comparativement aux satellites dont la position est estimée à partir de l'almanach. On note aussi que deux paramètres sont employés afin de déterminer la priorité des satellites dont le démarrage est à froid. Le premier paramètre est l'effort de recherche, soit un nombre compris entre 0 et 4 et fixé initialement par l'utilisateur afin de forcer la recherche de satellites en particulier. Le second paramètre est le nombre d'échecs de recherche, soit un compteur qui s'incrémente à chaque fois que le pavé temps/fréquence a été balayé au

complet par un canal de démodulation. Cette approche permet ainsi de rechercher systématiquement tous les satellites dont on ne peut estimer initialement la position.

### 5.3 Calcul de la solution de navigation

Les sections qui suivent présentent quelques méthodes pouvant être employées afin de calculer la solution de navigation. La solution que nous avons retenue étant le filtre de Kalman étendu, l'emphase est donc mise à ce niveau. Il faut noter que le texte qui suit a été tiré directement de (Sauriol, 2006) avec quelques modifications apportées au passage.

#### 5.3.1 Technique de base pour la détermination de la position

Une méthode simple pour résoudre les équations de navigation est présentée dans (Tsui, 2000). Celle-ci part du principe de linéarisation des équations de navigation présentées au chapitre 2 :

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_u - \Delta_{ionoi} - \Delta_{tropoi} + \Delta_{clki} \quad (5.1)$$

où :

- $x_i, y_i$  et  $z_i$  représentent la position des satellites  $i = 1, 2, 3$  et 4,
- $\rho_i$  est la pseudo-distance du  $i^{\text{ème}}$  satellite
- $x_u, y_u, z_u$  représente la position du récepteur
- $b_u$  est le biais d'horloge du récepteur
- $\Delta_{ionoi}, \Delta_{tropoi}$  et  $\Delta_{clki}$  sont de façon respective les corrections ionosphériques, troposphériques et d'horloges associées au  $i^{\text{ème}}$  satellite

En linéarisant cette équation autour d'un point d'équilibre quelconque, on obtient alors :

$$\delta \rho_i = \frac{(x_i - x_u) \delta x_u + (y_i - y_u) \delta y_u + (z_i - z_u) \delta z_u}{\sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}} + \delta b_u \quad (5.2)$$

où  $\delta \rho_i$ ,  $\delta x_u$ ,  $\delta y_u$ ,  $\delta z_u$  et  $\delta b_u$  sont les équivalents linéarisés des paramètres  $\rho_i$ ,  $x_u$ ,  $y_u$ ,  $z_u$  et  $b_u$ . Cette équation peut ensuite être représentée sous une forme matricielle de la façon suivante :

$$\alpha_{i1} = \frac{x_i - x_u}{r_i} \quad \alpha_{i2} = \frac{y_i - y_u}{r_i} \quad \alpha_{i3} = \frac{z_i - z_u}{r_i} \quad (5.3)$$

$$\text{où : } r_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2}$$

$$\begin{bmatrix} \delta \rho_1 \\ \delta \rho_2 \\ \delta \rho_3 \\ \delta \rho_4 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta b_u \end{bmatrix} \quad (5.4)$$

$$\therefore \delta \mathbf{\rho} = \mathbf{H} \delta \mathbf{x} \quad (5.5)$$

La solution de l'équation (5.5) s'obtient donc en inversant la matrice  $\mathbf{H}$ , généralement appelée « matrice de géométrie » ou « matrice d'observabilité » :

$$\delta \mathbf{x} = \mathbf{H}^{-1} \delta \mathbf{\rho} \quad (5.6)$$

Dû à la linéarisation des équations, il n'est pas possible d'obtenir une solution de navigation directement par l'équation (5.6). On procédera plutôt de façon itérative en fixant arbitrairement un seuil d'erreur  $\delta e_{seuil}$  tel que décrit par l'équation suivante. On arrêtera tout simplement les calculs lorsque l'erreur  $\delta e$  deviendra inférieure ou égale à  $\delta e_{seuil}$  :



$$\delta e = \sqrt{\delta x_u^2 + \delta y_u^2 + \delta z_u^2 + \delta b_u^2} \leq \delta e_{seuil} \quad (5.7)$$

La procédure itérative se déroule comme suit :

0. Mesurer les pseudo-distances  $\rho_{mesure}$ .
1. Initialiser  $x_u, y_u, z_u$  et  $b_u$  à une valeur probable de position et de biais d'horloge. Il est possible de partir du centre de la Terre, soit (0,0,0), et de fixer un biais d'horloge nul si aucune estimation n'est disponible. Ceci a toutefois comme conséquence d'augmenter le nombre d'itérations nécessaires.
2. Calculer les distances  $\rho_i$  à partir de l'équation (5.1) et des valeurs fixées de  $x_u, y_u, z_u$  et  $b_u$ . La différence ( $\rho_{mesure} - \rho_i$ ) donnera  $\delta\rho_i$ .
3. Calculer les coefficients de la matrice **H** à l'aide de l'équation (5.3).
4. Calculer  $\delta x_u, \delta y_u, \delta z_u$  et  $\delta b_u$  à partir de l'équation (5.6).
5. Calculer l'erreur  $\delta e$  à partir de l'équation (5.7). Si l'erreur est inférieure ou égale au seuil fixé, soit  $\delta e \leq \delta e_{seuil}$ , alors on arrête la procédure. La solution de l'équation de navigation est alors approximativement égale à  $x_u, y_u, z_u$  et  $b_u$ . On doit poursuivre à l'étape suivante dans le cas échéant.
6. Additionner  $\delta x_u, \delta y_u, \delta z_u$  et  $\delta b_u$  aux valeurs de  $x_u, y_u, z_u$  et  $b_u$  respectivement de manière à obtenir les valeurs initiales de position et d'erreur d'horloge pour la prochaine itération.
7. Retourner à l'étape 2.

La méthode présentée ici est intéressante car elle est relativement simple à réaliser et qu'elle ne demande pas une grande charge de calcul. Néanmoins, une telle méthode est loin d'être optimale puisqu'elle ne repose que sur quatre satellites de positionnement. Pour obtenir de meilleurs résultats, on utilisera plutôt une approche basée sur la méthode des moindres carrés, telle que présentée à la section suivante.

### 5.3.2 Solution de navigation par la méthode des moindres carrés

Essentiellement, la méthode des moindres carrés présentée dans la littérature (Tsui, 2000) est la même que celle décrite ci-dessus, à la différence que tous les  $n$  satellites visibles sont employés pour effectuer les calculs. Il s'agit donc de faire les calculs présentés précédemment pour  $i$  allant de 1 jusqu'à  $n$ ,  $n$  étant supérieur à quatre. On obtient donc une série de  $n$  équations linéaires pour lesquelles il n'y a toujours que quatre inconnues. Il est alors possible de calculer la matrice pseudo-inverse de  $\mathbf{H}$  afin d'obtenir une solution meilleure qu'avec seulement quatre satellites (Tsui, 2000) :

$$\delta \mathbf{x} = [\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T \delta \mathbf{p} \quad (5.8)$$

En bref, pour résoudre les équations de navigations via la méthode des moindres carrés, il s'agit de répéter la procédure décrite précédemment pour la technique de base en remplaçant l'équation (5.6) par l'équation (5.8).

La méthode des moindres carrés est sans doute celle le plus fréquemment employée car elle est très facile à implémenter et qu'elle permet de calculer une solution de navigation meilleure puisque basée sur tous les satellites en vue. Un des défauts de cette méthode réside cependant dans le fait qu'un certain nombre d'itérations doit être réalisé à chaque fois que la solution de navigation est mise à jour. Bien que de façon générale moins de dix itérations soient nécessaires pour que l'algorithme converge (Tsui, 2000), la charge de calcul peut s'avérer exigeante pour une implémentation en temps réel, particulièrement dans les applications à hautes dynamiques, telles qu'en avion par exemple. La méthode des moindres carrés est donc généralement limitée aux applications en statique, tel qu'en géodésie par exemple. L'utilisation d'un filtre de Kalman, présenté à la section suivante, est habituellement préférable dans la plupart des applications car il tient compte de la dynamique d'un système. De plus, le filtre de Kalman permet d'intégrer facilement les données de plusieurs sources différentes, rendant du coup possible le développement de récepteurs de

navigation hybrides plus complexes, tels que GPS et Galileo, GPS et GLONASS ou encore GPS, Galileo et GLONASS,

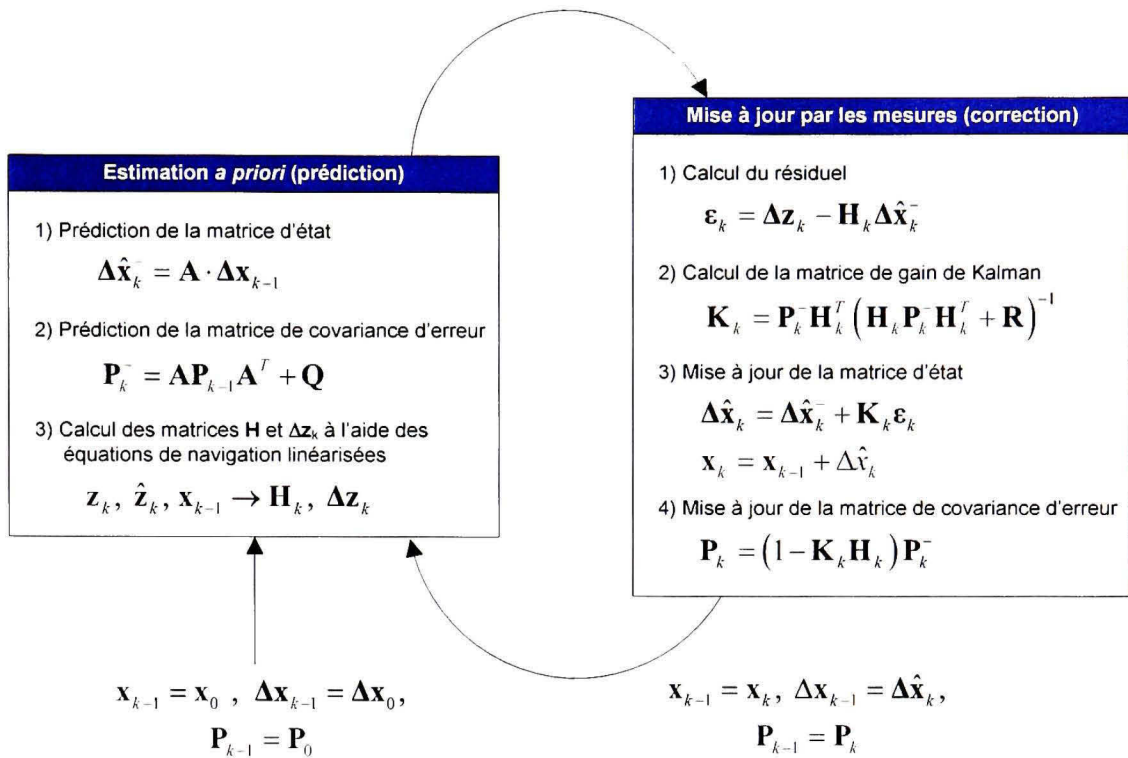
### 5.3.3 Solution de navigation à l'aide du filtre Kalman étendu (EKF)

Le filtre de Kalman est un outil de traitement des signaux extrêmement puissant permettant de combiner à la fois la dynamique et la statistique d'un système. Il s'agit en fait d'un estimateur optimal pour tous les systèmes discrets sujets à un bruit blanc de distribution gaussienne à leurs entrées (Maybeck, 1979). Il faut cependant noter que le filtre de Kalman est un outil dédié aux systèmes linéaires. Puisque les équations de navigation sont non linéaires, elles doivent donc être linéarisées autour d'un point d'équilibre afin d'être résolues, tel que vu précédemment. Dans ce cas bien précis, on dit du filtre de Kalman qu'il est étendu (EKF) afin de résoudre les systèmes non linéaires.

Essentiellement, le filtre de Kalman est un algorithme récursif dans lequel on cherchera tout d'abord à estimer l'état futur d'un système à partir de son état actuel. On corrigera dans un second temps cette estimation à l'aide de mesures externes et d'informations quant à aux statistiques de ces mesures. Le filtre de Kalman est donc un algorithme de type prédicteur-correcteur (PECE) tel qu'illustré à la Figure 5.4. Les principales matrices employées au sein du filtre sont les suivantes :

- $\mathbf{x}_k$ , la matrice d'état à l'instant  $k = 0, 1, 2, \dots$
- $\mathbf{z}_k$ , la matrice de mesure à l'instant  $k$
- $\mathbf{A}$ , la matrice de transition d'état
- $\mathbf{H}_k$ , la matrice d'observabilité à l'instant  $k$
- $\mathbf{P}_k$ , la matrice de covariance d'erreur à l'instant  $k$
- $\mathbf{Q}$ , la matrice de covariance du bruit de processus
- $\mathbf{R}$ , la matrice de covariance du bruit des mesures
- $\mathbf{K}_k$ , la matrice de gain de Kalman à l'instant  $k$
- $\varepsilon_k$ , la matrice des résiduels à l'instant  $k$





**Figure 5.4 Schéma synthèse du filtre de Kalman étendu.**

Dans le texte qui suit, seules les équations servant à l'élaboration du filtre de Kalman au sein du récepteur de navigation sont présentées. Le lecteur est donc invité à consulter (Brown et Hwang, 1997; Grewal et Andrews, 2001) pour plus d'information quant au fondement de ces équations.

Ainsi, dans le cas du filtre de Kalman développé, la matrice  $\mathbf{x}_k$  a été fixée comme étant :

$$\mathbf{x}_k = \begin{bmatrix} x_{uk} & \dot{x}_{uk} & y_{uk} & \dot{y}_{uk} & z_{uk} & \dot{z}_{uk} & b_{uk} & \dot{b}_{uk} \end{bmatrix}^T \quad (5.9)$$

où :

- $(x_{uk}, y_{uk}, z_{uk})$  représente la position du récepteur à l'instant  $k$
- $(\dot{x}_{uk}, \dot{y}_{uk}, \dot{z}_{uk})$  représente la vitesse du récepteur à l'instant  $k$
- $b_{uk}$  et  $\dot{b}_{uk}$  sont respectivement les biais et dérive d'horloge à l'instant  $k$

La matrice  $\mathbf{z}_k$  est quant à elle représentée par :

$$\mathbf{z}_k = [\rho_{1k} \quad \dot{\rho}_{1k} \quad \rho_{2k} \quad \dot{\rho}_{2k} \quad \cdots \quad \rho_{nk} \quad \dot{\rho}_{nk}]^T \quad (5.10)$$

où :

- $\rho_{ik}$  représente la mesure de pseudo-distance provenant du canal  $i$  l'instant  $k$
- $\dot{\rho}_{ik}$  représente la mesure de vitesse relative provenant du canal  $i$  l'instant  $k$
- $n$  est le nombre total de canaux de démodulation

On fixe  $\mathbf{x}_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$  lors d'un démarrage à froid du récepteur, tandis que la dernière position calculée ( $x_e, y_e, z_e$ ) peut être employée lors d'un démarrage à tiède ou à chaud, telle que :  $\mathbf{x}_0 = [x_e \quad 0 \quad y_e \quad 0 \quad z_e \quad 0 \quad 0]^T$ . Il faut noter que les matrices  $\mathbf{x}_k$  et  $\mathbf{z}_k$  ne sont pas directement employées au niveau du filtre de Kalman; on a plutôt recours aux matrices linéarisées  $\Delta\mathbf{x}_k$  et  $\Delta\mathbf{z}_k$ , définies telles que :

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_k \quad (5.11)$$

$$\Delta\mathbf{z}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \quad (5.12)$$

où  $\hat{\mathbf{z}}_k$  est une prédiction des mesures réalisée à partir de la matrice d'état :

$$\hat{\mathbf{z}}_k = [\hat{\rho}_{1k} \quad \hat{\dot{\rho}}_{1k} \quad \hat{\rho}_{2k} \quad \hat{\dot{\rho}}_{2k} \quad \cdots \quad \hat{\rho}_{nk} \quad \hat{\dot{\rho}}_{nk}]^T \quad (5.13)$$

Avant de présenter le calcul de chacun des éléments de  $\hat{\mathbf{z}}_k$ , introduisons tout d'abord certaines relations importantes, en débutant par :

$$r_{ik} = \sqrt{(x_{uk} - x_{ik})^2 + (y_{uk} - y_{ik})^2 + (z_{uk} - z_{ik})^2} \quad (5.14)$$

où :

- $r_{ik}$  est la distance géométrique entre le récepteur et le  $i^{\text{ème}}$  satellite à l'instant  $k$
- $(x_{ik}, y_{ik}, z_{ik})$  représente la position du  $i^{\text{ème}}$  satellite calculée à partir des éphémérides

Mentionnons aussi que :

$$\alpha_{xik} = \frac{x_{uk} - x_{ik}}{r_{ik}} \quad \alpha_{yik} = \frac{y_{uk} - y_{ik}}{r_{ik}} \quad \alpha_{zik} = \frac{z_{uk} - z_{ik}}{r_{ik}} \quad (5.15)$$

$$\dot{\alpha}_{xik} = \frac{\dot{x}_{uk} - \dot{x}_{ik}}{r_{ik}} \quad \dot{\alpha}_{yik} = \frac{\dot{y}_{uk} - \dot{y}_{ik}}{r_{ik}} \quad \dot{\alpha}_{zik} = \frac{\dot{z}_{uk} - \dot{z}_{ik}}{r_{ik}} \quad (5.16)$$

où :

- $\alpha_{xik}$ ,  $\alpha_{yik}$  et  $\alpha_{zik}$  sont appelées « lignes de visée »
- $\dot{\alpha}_{xik}$ ,  $\dot{\alpha}_{yik}$ ,  $\dot{\alpha}_{zik}$  sont les dérivées temporelles des lignes de visée
- $(\dot{x}_{ik}, \dot{y}_{ik}, \dot{z}_{ik})$  représente la vitesse du  $i^{\text{ème}}$  satellite à l'instant  $k$

Chacun des  $2n$  éléments de  $\hat{\mathbf{z}}_k$  se calcule donc :

$$\hat{\rho}_{ik} = r_{ik} + b_u + T\dot{b}_u - \Delta_{ionoik} - \Delta_{tropoik} + \Delta_{clkik} \quad (5.17)$$

$$\dot{\rho}_{ik} = \alpha_{xik}\dot{x}_{ik} + \alpha_{yik}\dot{y}_{ik} + \alpha_{zik}\dot{z}_{ik} + \dot{b}_{uk} \quad (5.18)$$

où :

- $\Delta_{ionoik}$ ,  $\Delta_{tropoik}$  et  $\Delta_{clkik}$  sont de façon respective les corrections ionosphériques, troposphériques et d'horloges associées au  $i^{\text{ème}}$  satellite à l'instant  $k$
- $T$  représente la période d'échantillonnage du filtre de Kalman (attention ici à ne pas confondre avec l'indice «  $^T$  » qui indique une transposition de matrice)

Une fois la matrice  $\Delta \mathbf{z}_k$  calculée, il s'agit d'évaluer l'estimation à *priori*  $\Delta \hat{\mathbf{x}}_k^-$  telle que :



$$\Delta \mathbf{x}_k^- = \mathbf{A} \cdot \Delta \mathbf{x}_{k-1} \quad (5.19)$$

Il faut mentionner ici que  $\Delta \mathbf{x}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . La matrice de transition  $\mathbf{A}$  d'état se définit quant à elle :

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.20)$$

Ceci revient à estimer la position et le biais d'horloge du récepteur à l'instant  $k$  en réalisant une extrapolation à partir de la vitesse du récepteur et de la dérive d'horloge à l'instant  $k-1$ . On suppose évidemment que la vitesse et la dérive demeurent constantes, ce qui n'est pas toujours vrai. Ces estimations pourraient donc être améliorées en ayant accès à des mesures d'accélération, ces dernières provenant de centrales inertielles (IMU – *Inertial Measurement Unit*), par exemple. Le lecteur est invité à consulter (Grewal, Weill et Andrews, 2001) pour plus d'information sur le sujet.

L'étape suivante consiste à calculer la matrice résiduelle,  $\varepsilon_k$ , d'une manière semblable à celle vue précédemment pour la méthode des moindres carrés :

$$\varepsilon_k = \Delta \mathbf{z}_k - \mathbf{H}_k \Delta \hat{\mathbf{x}}_k^- \quad (5.21)$$

On remarque ici la matrice d'observabilité  $\mathbf{H}$ , qui est construite de la façon suivante à l'aide des équations de navigation linéarisées :

$$\mathbf{H}_k = \begin{bmatrix} \alpha_{x1k} & 0 & \alpha_{y1k} & 0 & \alpha_{z1k} & 0 & 1 & 0 \\ \dot{\alpha}_{x1k} & \alpha_{x1k} & \dot{\alpha}_{y1k} & \alpha_{y1k} & \dot{\alpha}_{z1k} & \alpha_{z1k} & 0 & 1 \\ \alpha_{x2k} & 0 & \alpha_{y2k} & 0 & \alpha_{z2k} & 0 & 1 & 0 \\ \dot{\alpha}_{x2k} & \alpha_{x2k} & \dot{\alpha}_{y2k} & \alpha_{y2k} & \dot{\alpha}_{z2k} & \alpha_{z2k} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{xnk} & 0 & \alpha_{ynk} & 0 & \alpha_{znk} & 0 & 1 & 0 \\ \dot{\alpha}_{xnk} & \alpha_{xnk} & \dot{\alpha}_{ynk} & \alpha_{ynk} & \dot{\alpha}_{znk} & \alpha_{znk} & 0 & 1 \end{bmatrix} \quad (5.22)$$

Jusqu'à présent, nous n'avons abordé que la partie dynamique de la solution de navigation. La dimension statistique est principalement représentée par la matrice de covariance d'erreur,  $\mathbf{P}_k$ , dont la valeur *a priori*  $\mathbf{P}_k^-$ , se calcule :

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (5.23)$$

La matrice de covariance d'erreur initiale,  $\mathbf{P}_0$ , est généralement obtenue à partir d'une estimation pessimiste de la matrice des résiduels  $\varepsilon_0$ . Dans notre cas, nous nous sommes basés sur la matrice  $\mathbf{P}_0$  présentée dans (Kaplan, 1996, p. 400) puis l'avons ajustée via une série de tests par simulation. Nous avons alors obtenu la matrice suivante qui permet au filtre de Kalman de converger en un temps raisonnablement court (< 10 itérations) :

$$\mathbf{P}_0 = \begin{bmatrix} 4 \times 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 \times 10^5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^4 \end{bmatrix} \quad (5.24)$$

On remarque la présence de la matrice de covariance de bruit de processus,  $\mathbf{Q}$ , à l'équation (5.23). Cette matrice donne une certaine information au filtre de Kalman quant au bruit

introduit dans la solution de navigation par le filtre lui-même. Ce bruit est généralement déterminé à l'aide d'un second filtre de Kalman, un processus plutôt fastidieux. Une analyse statistique présentée dans (Brown et Hwang, 1997) permet de simplifier le problème en posant :

$$\mathbf{Q} = \begin{bmatrix} S_p \frac{T^3}{3} & S_p \frac{T^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ S_p \frac{T^2}{2} & S_p T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & S_p \frac{T^3}{3} & S_p \frac{T^2}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & S_p \frac{T^2}{2} & S_p T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_p \frac{T^3}{3} & S_p \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_p \frac{T^2}{2} & S_p T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_f T + S_g \frac{T^3}{3} & S_g \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & S_g \frac{T^2}{2} & S_g T \end{bmatrix} \quad (5.25)$$

où  $S_p$ ,  $S_f$  et  $S_g$  représentent l'amplitude spectrale du bruit-blanc associée au calcul de la position, de l'erreur d'horloge et de la dérive de l'horloge respectivement. Les valeurs suivantes ont été choisies après une série de tests car elles permettaient d'obtenir de bons résultats en temps réel :

$$S_p = 10^9 \quad S_f = 10^5 \quad S_g = 10^4$$

Une fois la matrice  $\mathbf{P}_k^-$  calculée, il s'agit de procéder avec le calcul de la matrice de gain de Kalman,  $\mathbf{K}_k$  :

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left( \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R} \right)^{-1} \quad (5.26)$$



où  $\mathbf{R}$  est la matrice de covariance de bruit de mesure. Celle-ci est estimée au mieux des connaissances disponibles en tirant profit du fait que les erreurs sur les mesures de pseudo-distances et de vitesses relatives ne sont pas corrélées entre elles. Ceci fait donc en sorte d'obtenir la matrice diagonale  $n \times n$  suivante :

$$\mathbf{R} = \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_p^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} \quad (5.27)$$

où  $\sigma_p$  est la déviation standard du bruit associé aux mesures de pseudo-distances. Le paramètre  $\sigma_v$  représente quant à lui la déviation standard du bruit associé aux mesures de vitesses relatives. Bien qu'en pratique les deux paramètres soient de l'ordre de 2 m et 0,01 m/s respectivement pour les signaux GPS L1 C/A, nous avons pu les fixer à 1 m et 0,3 m/s sans pour autant remarquer de différence en sortie du filtre de Kalman. Une amélioration pourrait toutefois être apportée au niveau de la précision de la solution de navigation en ajustant de façon dynamique la matrice  $\mathbf{R}$  en fonction du  $C/N_0$  des signaux et de l'élévation des satellites (ce qui est relativement facile à réaliser puisque le  $C/N_0$  et l'élévation des satellites sont *a priori* connus du récepteur).

La dernière étape consiste à mettre à jour les estimations *a priori* à l'aide des mesures. On parle alors d'estimations *a posteriori* des matrices d'état ( $\Delta \hat{\mathbf{x}}_k$ ) et de covariance d'erreur ( $\mathbf{P}_k$ ). Ces estimations sont calculées grâce à la matrice  $\mathbf{K}_k$ , telles que :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{e}_k \quad (5.28)$$

$$\mathbf{P}_k = (1 - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (5.29)$$

Le vecteur d'état  $\mathbf{x}_k$  peut alors être mis à jour grâce à l'équation (5.11) avant de passer à l'itération suivante ( $k+1$ ). Nous invitons le lecteur à consulter la Figure 5.4 pour un schéma synthèse du filtre de Kalman développé.

### 5.3.4 Modifications apportées au filtre de Kalman étendu

Un problème dont il faut tenir compte au niveau du filtre de Kalman est que le nombre de satellites visibles peut varier au fil du temps. Une des approches possibles afin de pallier ce problème consiste à faire varier la taille des matrices en fonction du nombre de satellites inclus au niveau de la solution de navigation. Une telle approche est toutefois complexe à mettre en œuvre et n'apporte aucune valeur ajoutée lors d'une implémentation en temps réel puisque le processeur doit être de toute façon capable de traiter un nombre maximal de satellites en simultané. Or, on pourrait penser minimiser la consommation électrique du processeur en procédant ainsi, ce qui est généralement souhaitable au sein d'applications embarquées. Cette réduction risque cependant d'être négligeable face à la charge de calcul exigée pour déterminer la position des satellites, sans compter la charge supplémentaire requise pour la gestion de la taille des matrices (voir section 5.4).

La technique que nous avons employée est beaucoup plus simple puisque la taille des matrices demeure fixe. Cette technique se base sur la théorie des filtres robustes, telle que décrite dans (Deergha Rao, Swamy et Plotkin, 2004). La théorie se résume rapidement de la façon suivante : le filtre de Kalman accorde aux mesures (la matrice  $\mathbf{z}_k$ ) une confiance inversement proportionnelle à la variance qui leur est attribuée. Il est donc possible de rendre plus robuste le filtre en pondérant la matrice de covariance de bruit de mesure  $\mathbf{R}$  par une seconde matrice  $\mathbf{D}$ , de manière à rejeter les mesures présentant une erreur aberrante. Cette matrice  $\mathbf{D}$  se définit ainsi :

$$\mathbf{R}' = \mathbf{R} \cdot \mathbf{D} = \mathbf{R} \cdot \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_n \end{bmatrix} \quad (5.30)$$

où :

- $\mathbf{R}'$  est la matrice  $\mathbf{R}$  pondérée
- $d_i$  est le coefficient de pondération associé aux canaux  $i = 1, 2, 3, \dots, n$

La procédure inverse doit aussi être appliquée au niveau de la matrice  $\mathbf{\varepsilon}_k$ , telle que :

$$\mathbf{\varepsilon}'_k = \mathbf{\varepsilon}_k \cdot \mathbf{D}^{-1} = \mathbf{\varepsilon}_k \cdot \begin{bmatrix} d_1^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_1^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_2^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_2^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_n^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_n^{-1} \end{bmatrix} \quad (5.31)$$

De façon générale, la théorie des filtres robustes se base sur la taille des résiduels pour déterminer les éléments de la matrice  $\mathbf{D}$ . Notre approche consiste plutôt à fixer les coefficients de cette matrice en fonction de la validité et de la visibilité des satellites. Si un satellite est jugé apte à être inclus au sein de la solution de navigation, nous fixons alors le coefficient correspondant  $d_i = 1$ . Dans le cas inverse, nous fixons  $d_i = 10^{10}$  afin d'être certains que le filtre de Kalman ne lui accorde aucune confiance. Enfin, le passage de  $d_i = 10^{10}$  à  $d_i = 1$  est effectué de façon progressive de manière à ce que les nouveaux satellites ne fassent pas diverger la solution de navigation (l'introduction d'un nouveau satellite s'effectue sur



cinq itérations consécutives du filtre de Kalman;  $d_i$  prend alors les valeurs suivantes à tour de rôle: 256, 64, 16, 4 puis 1).

En plus de la matrice de pondération  $\mathbf{D}$ , nous avons aussi développé un mécanisme afin de s'assurer de la convergence du filtre de Kalman. Un test consiste alors à vérifier si les résiduels contenus dans la matrice  $\varepsilon_k$  sont inférieurs à un seuil minimal  $\varepsilon_{\min}$ , dans lequel cas nous considérons que le filtre a convergé, tel que :

$$\begin{aligned} \varepsilon_{ik} \leq \varepsilon_{\min} \forall i &\rightarrow \text{Le filtre de Kalman a convergé} \\ \varepsilon_{ik} > \varepsilon_{\min} &\rightarrow \text{Le filtre de Kalman a divergé} \end{aligned} \quad (5.32)$$

Le seuil  $\varepsilon_{\min}$  a été fixé à 10 km durant la phase initiale de convergence de la solution de navigation (soit durant les 20 premières itérations du filtre de Kalman) puis à 100 m par la suite. Des avertissements sont au même moment envoyés à l'utilisateur via l'interface graphique afin qu'il puisse connaître l'état de la solution de navigation. Cette dernière est enfin réinitialisée advenant une divergence du filtre de Kalman.

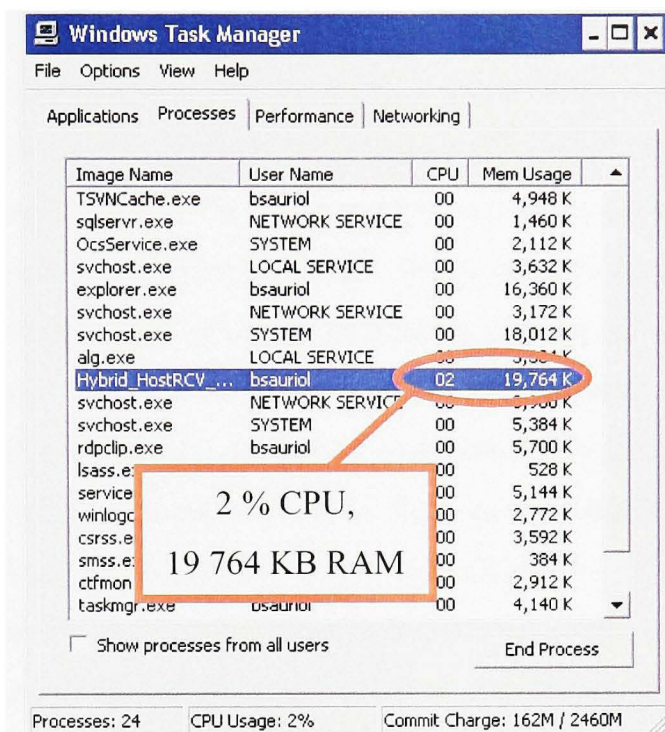
Le mécanisme de détection de convergence que nous avons développé s'est avéré fonctionnel, quoique nous ayons décelé certaines limitations lors des tests. Dans les faits, l'erreur de positionnement peut fluctuer au-delà de 100 m lorsque la disponibilité sélective (SA) est activée au niveau des signaux GPS (Kaplan, 1996). De telles fluctuations seraient alors interprétées comme étant une divergence du filtre de Kalman, ce qui est incorrect. Évidemment, il serait possible d'augmenter le seuil à 200 m par exemple, mais ceci risquerait d'augmenter le délai de réaction du mécanisme en cas de divergence. Plutôt que de procéder de façon absolue, les résiduels du filtre de Kalman devraient donc être comparés entre eux de façon relative plutôt qu'en absolu afin de déterminer correctement l'état du filtre de Kalman.

#### 5.4 Charge de calcul liée à l'architecture logicielle à haut niveau

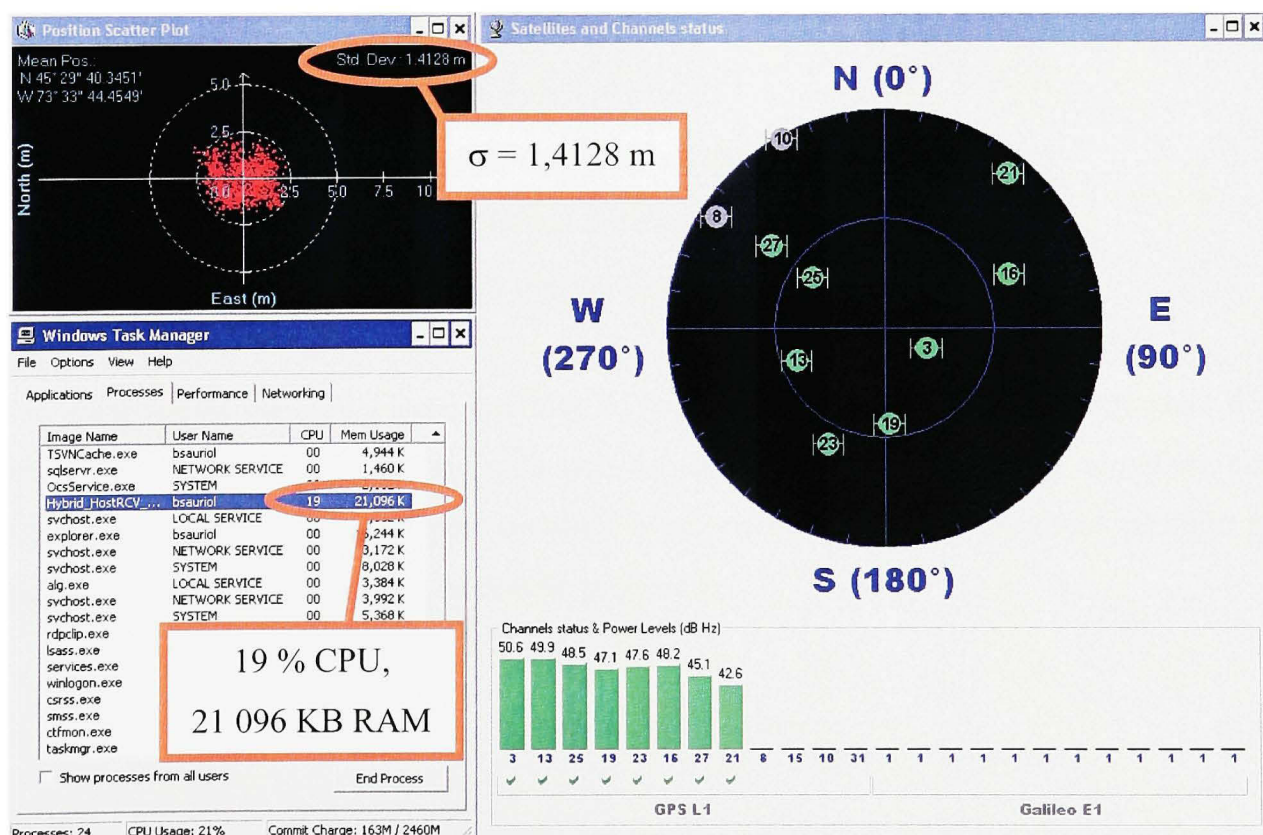
L'ordinateur PC employé pour l'implémentation du récepteur est muni de 1 GB de mémoire RAM et d'un processeur Pentium 4-M d'Intel. Bien que le système d'exploitation Windows XP opère en arrière-plan, la fréquence de 2,2 GHz du processeur fait en sorte de nous offrir une très grande puissance de calcul pour l'exécution de la couche logicielle à haut niveau. Afin de déterminer la charge de calcul liée à cette couche, nous avons exécuté une série de tests en temps réel une fois le récepteur complété et fonctionnel. Quelques-uns de ces tests sont présentés ci-dessous.

Dans un premier temps, nous avons ajusté la fréquence d'échantillonnage à 20 Hz au niveau de la solution de navigation du récepteur (il faut mentionner que la fréquence d'échantillonnage peut être ajustée par l'utilisateur via l'interface graphique – voir annexe 2). Nous avons ensuite connecté le récepteur à une antenne extérieure située sur le toit de l'ÉTS. Une fois le récepteur démarré, nous avons attendu que le filtre de Kalman ait convergé avant de prendre des mesures à l'aide du gestionnaire de tâche de Windows. La Figure 5.5 présente l'affichage du gestionnaire de tâche pour une solution de navigation calculée à l'aide de huit satellites.

Lors d'un second test, nous avons gardé la même configuration que celle décrite ci-dessus, à l'exception de la fréquence d'échantillonnage qui a été ajustée à 1 kHz. La Figure 5.6 montre une capture d'écran réalisée lors de ce test et qui permet de prouver le bon fonctionnement du récepteur, même à 1 kHz (une erreur de positionnement horizontale de l'ordre de 1,4 m à  $1\sigma$  a été mesurée après 10 minutes de tests). Une explication des résultats est présentée ci-dessous.



**Figure 5.5** Affichage du gestionnaire de tâche de Windows lors de l'exécution de la solution de navigation à 20 Hz.



**Figure 5.6** Capture d'écran lors de l'exécution de la solution de navigation à 1 kHz.



En examinant la Figure 5.5, on voit que plusieurs processus opèrent en arrière-plan au niveau de Windows. Parmi ces processus, on remarque le programme « Hybrid\_HostRCV.exe », soit la couche logicielle à haut niveau du récepteur développé. Un examen plus détaillé permet de constater que seulement 2% du processeur est employé par le récepteur et que 20 MB sont requis au niveau de la mémoire RAM. Or, la majeure partie de ces ressources est liée à l'interface graphique, la solution de navigation n'occupant qu'une partie infime du temps de calcul et de la mémoire RAM lorsqu'elle est exécutée à 20 Hz. La Figure 5.6 permet de prouver cette allégation, alors que seulement 19% du processeur est employé lorsque la solution de navigation opère à 1 kHz. Bien qu'à 1 kHz la charge de calcul exigée par la solution de navigation devrait être 50 fois plus élevée qu'à 20 Hz, on n'obtient qu'un ratio de 9,5 en pratique ( $19\% / 2\% = 9,5$ ). Les tests que nous avons réalisés ont en fait permis d'élaborer la formule empirique suivante afin d'estimer la charge de calcul de la solution de navigation :

$$\%CPU = 2,1 \times 10^{-3} \cdot n \cdot f_{nav} \quad (5.33)$$

où :

- $\%CPU$  est la charge de calcul exigée au niveau du processeur en %
- $n$  est le nombre de satellites inclus dans la solution de navigation
- $f_{nav}$  est la fréquence de mise à jour de la solution de navigation en Hz

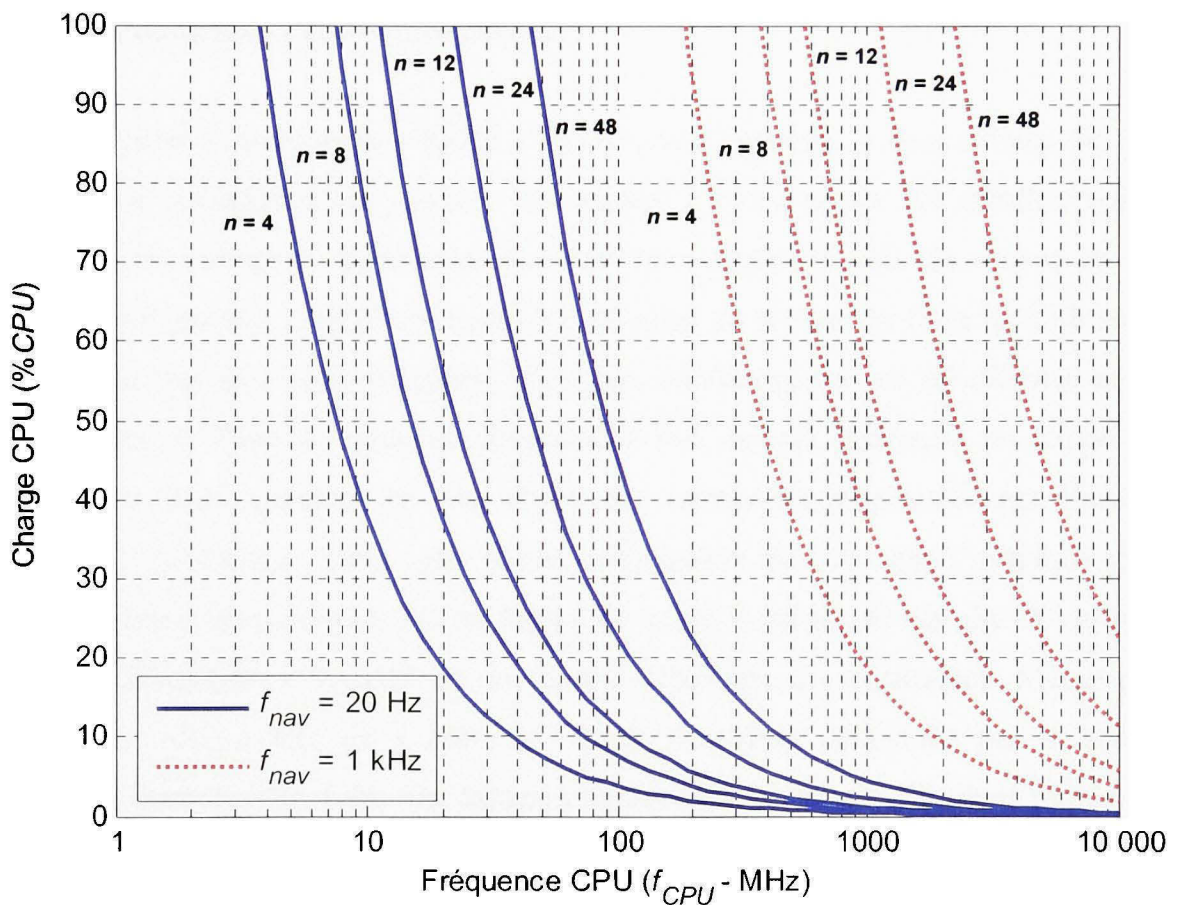
L'approximation suivante a aussi été mise au point afin d'obtenir un ordre de grandeur de la charge de calcul requise si on implantait la solution de navigation au sein d'un autre processeur à simple cœur (l'approximation n'est valide que si le processeur possède une unité de calcul en point flottant de double précision) :

$$\%CPU \approx \frac{4,68 \cdot n \cdot f_{nav}}{f_{CPU}} \quad (5.34)$$

où :

- $f_{CPU}$  est la fréquence du processeur exprimée en MHz

La charge CPU est illustrée de façon graphique à la Figure 5.7 pour des fréquences CPU comprises entre 1 MHz et 10 GHz, des fréquences de 20 Hz et 1 kHz au niveau de la solution de navigation et divers nombres de satellites ( $n = 4, 8, 12, 24$  et 48).



**Figure 5.7** Estimation de la charge CPU exigée par la solution de navigation.

Grâce à l'équation (5.33), on arrive à déterminer une consommation nette d'environ 17% du temps de calcul du processeur pour une solution de navigation comprenant huit satellites et une fréquence de mise à jour de 1 kHz (l'interface graphique consomme ainsi 2% de temps de calcul). En comparaison, la même solution pourrait être exécutée à 20 Hz sur un processeur de 8 MHz seulement, ce qui illustre la performance de la solution de navigation développée. Cela signifie qu'une solution de navigation à 20 Hz pourrait être entièrement

exécutée au sein du FPGA en remplaçant le Microblaze par un processeur LEON de Gaisler par exemple (processeur pouvant effectuer des calculs en point flottant et opérer à 100 MHz et plus dans un FPGA Virtex 4 de Xilinx) (Gaisler Research, 2007). Dans ce cas-ci, une charge de calcul de l'ordre de 8% serait nécessaire au niveau du processeur embarqué pour déterminer une solution à huit satellites ( $f_{CPU} = 100$  MHz), ce qui laisse suffisamment de ressources pour exécuter le programme d'acquisition et de poursuite par exemple.

## 5.5 Conclusion et recommandations

Dans le chapitre 5, nous avons abordé l'architecture logicielle à haut niveau du récepteur développé. Le mécanisme de gestion des canaux a entre autres été détaillé, suivi de la présentation de quelques algorithmes pouvant être employés afin de mettre en place la solution de navigation. L'emphasis a surtout été mise au niveau du filtre de Kalman, soit la technique retenue pour notre récepteur. Quelques améliorations ont été décrites au passage afin de rendre le filtre en question fonctionnel lors de son opération en temps réel. Le chapitre s'est enfin terminé par une étude des ressources employées par l'architecture logicielle à haut niveau. Dans cette étude, nous avons pu voir que l'interface graphique développée demandait environ 2% du temps de calcul total au niveau du PC, alors que la solution de navigation n'exigeait qu'une partie minuscule des ressources. Même pour une fréquence de mise à jour de 1 kHz, la charge de calcul demandée par la solution de navigation n'était que de 17%. Un espace mémoire d'environ 20 Mo était requis dans les deux cas.

Malgré le fait que la solution de navigation développée soit très performante, plusieurs améliorations restent à être apportées à ce niveau. Premièrement, mentionnons le mécanisme de détection de convergence du filtre de Kalman qui présente certaines limitations. Un mécanisme plus fiable devrait éventuellement être développé afin de vérifier correctement la convergence du filtre. Ensuite, l'ajout d'une matrice de covariance de bruit de mesure ( $\mathbf{R}$ ) adaptative a été soulevé dans ce chapitre, ce qui permettrait sans doute d'améliorer les résultats obtenus via le filtre de Kalman. L'implantation d'un mécanisme RAIM serait aussi



souhaitable afin de garantir une solution de navigation intègre en tout temps. Enfin, un point qui a été abordé plus tôt au chapitre 2 serait de développer une solution de navigation plus complète afin de permettre la navigation à partir de deux ou trois satellites seulement, ce qui n'est pas le cas actuellement (un minimum de quatre satellites est nécessaire).

Le prochain et dernier chapitre présente une série de tests effectués à l'aide du récepteur afin de caractériser les performances de ce dernier. La performance de la solution de navigation occupe une place importante au niveau de ces tests et elle y est abordée en détail.

## **CHAPITRE 6**

### **TESTS EN TEMPS RÉEL**

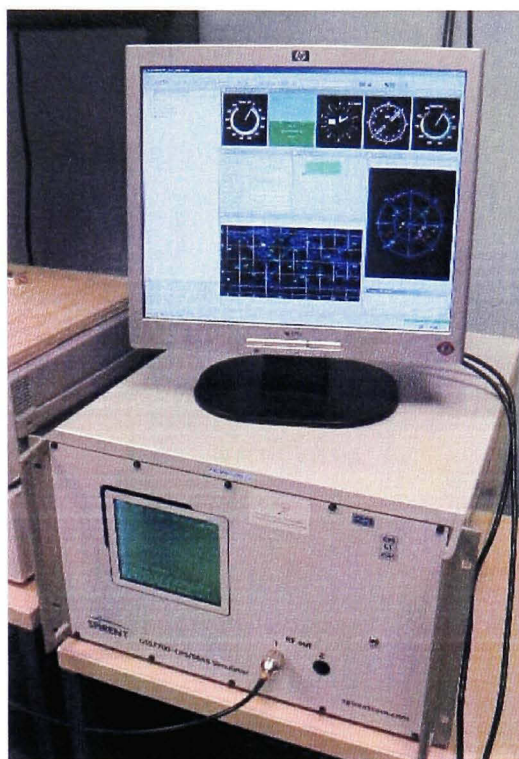
#### **6.1 Sommaire des tests réalisés**

Afin de valider son fonctionnement et d'en évaluer les performances, une batterie de tests a été réalisée lors du développement de notre récepteur de navigation. Vu le grand nombre de tests effectués, ce chapitre ne présente donc que les tests en temps réel les plus pertinents, effectués une fois le développement terminé. Le chapitre est divisé en deux volets principaux : les tests en environnements simulés et les tests en environnement réel.

Le premier volet débute par une étude du mécanisme de mesure du SNR développé au sein du récepteur (test #1). Deux tests sont ensuite présentés, un en mode statique (test #2) et l'autre en mode dynamique (test #3), de manière à caractériser la précision et les performances du récepteur. Tous les tests exposés au niveau du premier volet ont été réalisés à l'aide du simulateur de signaux GPS GSS7700 de Spirent, illustré à la Figure 6.1.

Le second volet poursuit par une évaluation du temps de démarrage du récepteur (test #4). Des tests en modes statique (test #5) et dynamique (test #6) sont aussi présentés de manière à caractériser le fonctionnement du récepteur en environnement réel. Un dernier test, réalisé en environnement difficile (test #7), complète finalement notre étude.

Il faut noter que les signaux transmis par le satellite GIOVE-A n'étaient pas valides au moment de la rédaction de ce document. Aucune solution de navigation GPS et Galileo hybride n'a donc pu être réalisée. Pour cette raison, une section a été ajoutée à ce chapitre afin de montrer la capacité du récepteur à capturer les signaux Galileo.



**Figure 6.1** *Simulateur GS7700 de Spirent.*

## **6.2 Étude du mécanisme de mesure du SNR (test #1)**

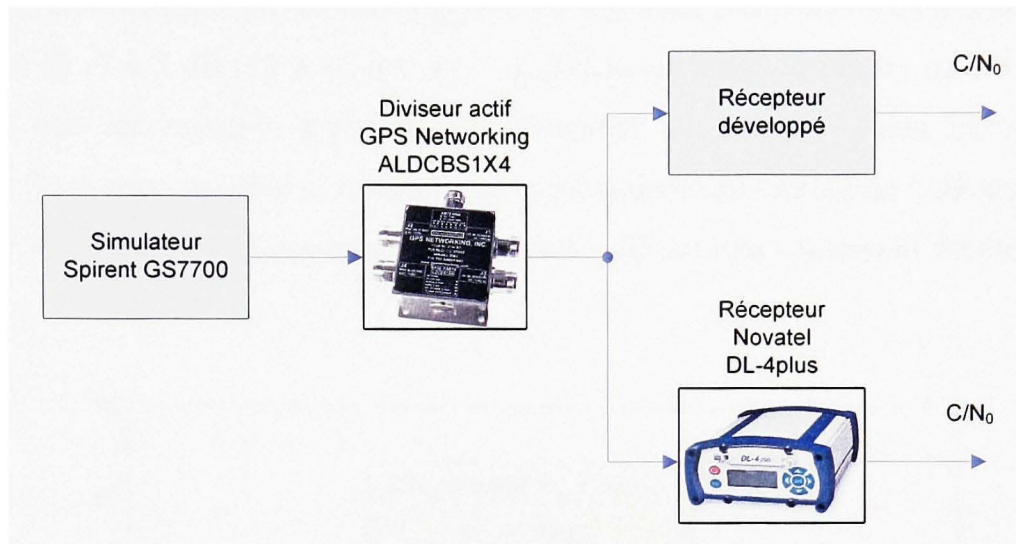
### **6.2.1 Méthodologie du test #1**

Nous avons élaboré un banc de test afin de déterminer la précision de notre mécanisme de mesure du SNR. Ce banc de test, illustré à la Figure 6.2, est basé sur un simulateur GS7700 de Spirent. Le signal provenant de ce dernier a été séparé grâce à un diviseur actif ALDCBS1X4 de marque GPS Networking (GPS Networking, 2007), ceci dans le but de l'acheminer à la fois au récepteur développé et à un récepteur DL-4plus de haute précision de Novatel (Novatel, 2007b). Des câbles coaxiaux rigoureusement identiques ont été employés afin d'assurer que la puissance des signaux soit la même à l'entrée des deux récepteurs.

Une fois le montage réalisé, nous avons exécuté un scénario en mode statique à partir du simulateur. Nous avons pris le soin d'ajuster la puissance de tous les signaux à une valeur nominale de 0 dB, valeur propre au simulateur. Nous avons par la suite réglé la puissance d'un seul signal GPS L1 à 20 dB et avons pris en note les valeurs de  $C/N_0$  retournées par les



deux récepteurs. Nous avons répété les mêmes opérations en diminuant la puissance du signal par incréments de 3 dB jusqu'à ce qu'un des deux récepteurs ne soit plus verrouillé. Les résultats obtenus sont présentés à la section suivante.



**Figure 6.2** *Montage pour l'étude du mécanisme de mesure du SNR.*

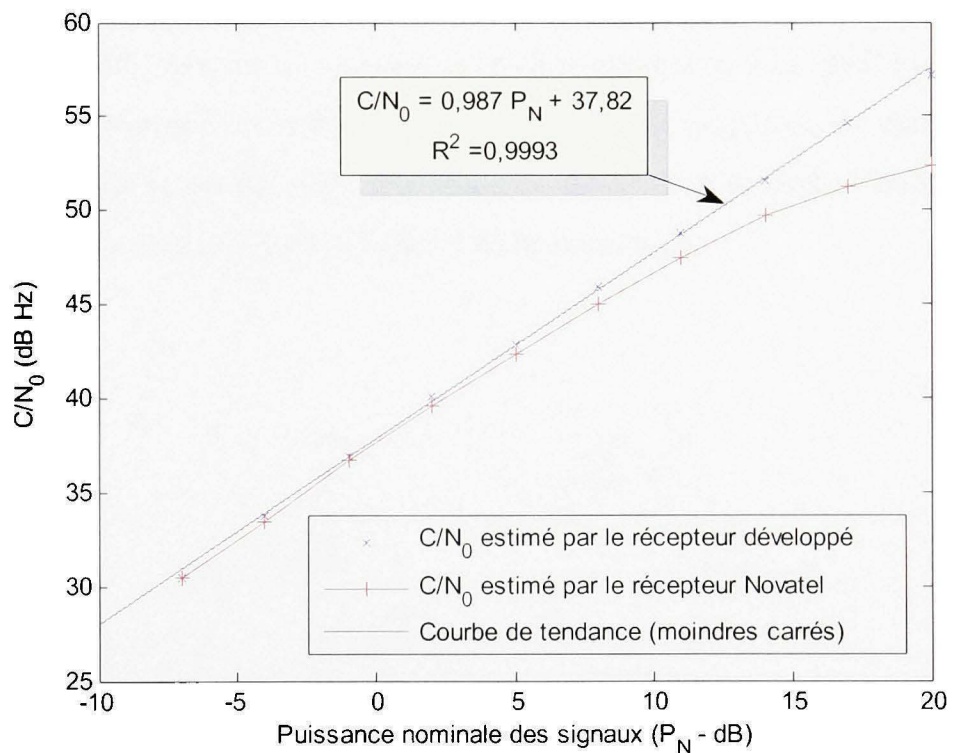
### 6.2.2 Résultats et analyse du test #1

Les résultats obtenus lors des tests sont présentés à la Figure 6.3. Il faut noter que les tests ont pu être réalisés jusqu'à une puissance minimale de -7 dB, ce qui correspond à un  $C/N_0$  d'environ 30 dB Hz au niveau des deux récepteurs. En dessous de cette puissance, le récepteur que nous avons développé semblait avoir de la difficulté à demeurer verrouillé au signal. Or, cette limite est très raisonnable pour un premier prototype si on la compare à celle du récepteur de Novatel, qui était de l'ordre de -10 dB (27 dB Hz) lors du test.

En examinant la Figure 6.3, on remarque que les mesures de  $C/N_0$  retournées par notre récepteur sont plutôt linéaires, comparativement à celles du récepteur de Novatel qui semblent saturer autour de 50 dB Hz. En faisant passer une droite de régression par les mesures de notre récepteur, il est possible de déterminer une pente quasi unitaire (soit 0,987), associée à un coefficient de corrélation  $R^2 = 0,9993$ . Ceci témoigne de la grande linéarité du mécanisme de mesure du SNR développé. Par souci de rigueur, nous avons consulté la fiche technique du simulateur, qui mentionne une précision de  $\pm 0,7$  dB au niveau de la puissance

des signaux générés (Spirent Communications, 2006). On peut donc supposer que la puissance des signaux en sortie du simulateur était également linéaire.

Finalement, on remarque que les mesures de  $C/N_0$  des deux récepteurs sont très similaires sur l'intervalle de -7 à 5 dB (30 à 42 dB Hz). La différence y est de l'ordre de 0,3 dB Hz. En supposant que les mesures provenant du récepteur de Novatel soient exactes sur cet intervalle, il est donc possible d'affirmer que le mécanisme de mesure du SNR développé est très précis. On parle alors d'une erreur inférieure à 1 dB sur tout l'intervalle étudié.



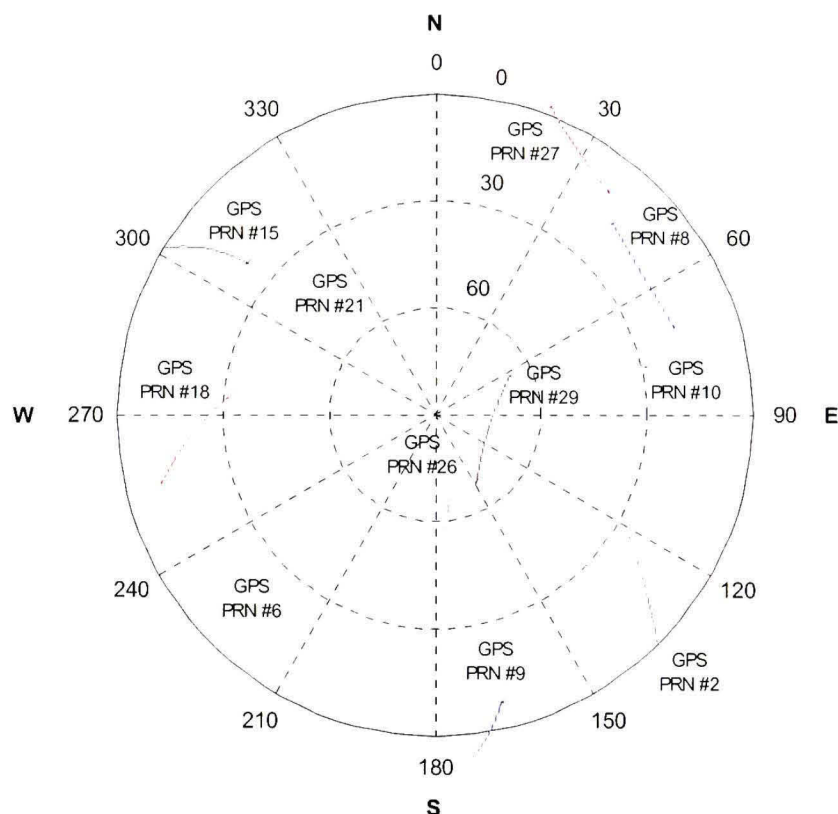
**Figure 6.3** *Évaluation de la linéarité du mécanisme de mesure de SNR.*

## 6.3 Test statique en environnement simulé (test #2)

### 6.3.1 Méthodologie du test #2

Afin d'avoir une idée de la précision de notre récepteur, nous avons testé ce dernier en le connectant à un simulateur GSS7700 de Spirent. Nous avons alors exécuté un scénario en mode statique en simulant un récepteur fixé au sol à Montréal. Il faut noter que la constellation de satellites GPS simulée était basée sur de véritables almanachs tirés du web (NAVCEN, 2007). Pour rendre le tout plus réaliste, nous avons aussi ajouté les délais ionosphériques et troposphériques à la simulation.

Les tests ont été effectués sur une période d'environ une heure. Les satellites visibles durant cette période sont présentés à la Figure 6.4 sous format de graphique en élévation et azimuth (le centre représente le zénith alors que le cercle extérieur représente l'horizon). Les croix indiquent la position des satellites vers la fin de la simulation.



**Figure 6.4** *Satellites visibles lors de la simulation en statique.*



### 6.3.2 Résultats et analyse du test #2

Puisque la position du récepteur est parfaitement connue du simulateur, nous avons été en mesure de déterminer les erreurs de positionnement horizontales ainsi que verticales de notre solution de navigation, mise à jour ici toutes les trois secondes (voir Figure 6.9 et Figure 6.10 respectivement). Ces erreurs de positionnement ont été obtenues en réalisant une rotation des coordonnées cartésiennes de la solution de navigation de sorte à obtenir un système de coordonnées locales. Ces coordonnées sont exprimées en terme de distances nord-sud (coordonnées N-S), ouest-est (coordonnées O-E) et verticale par rapport au point de référence entré dans le simulateur. Le lecteur est invité à consulter (Misra et Enge, 2006) pour plus de détails sur la technique de rotation des coordonnées.

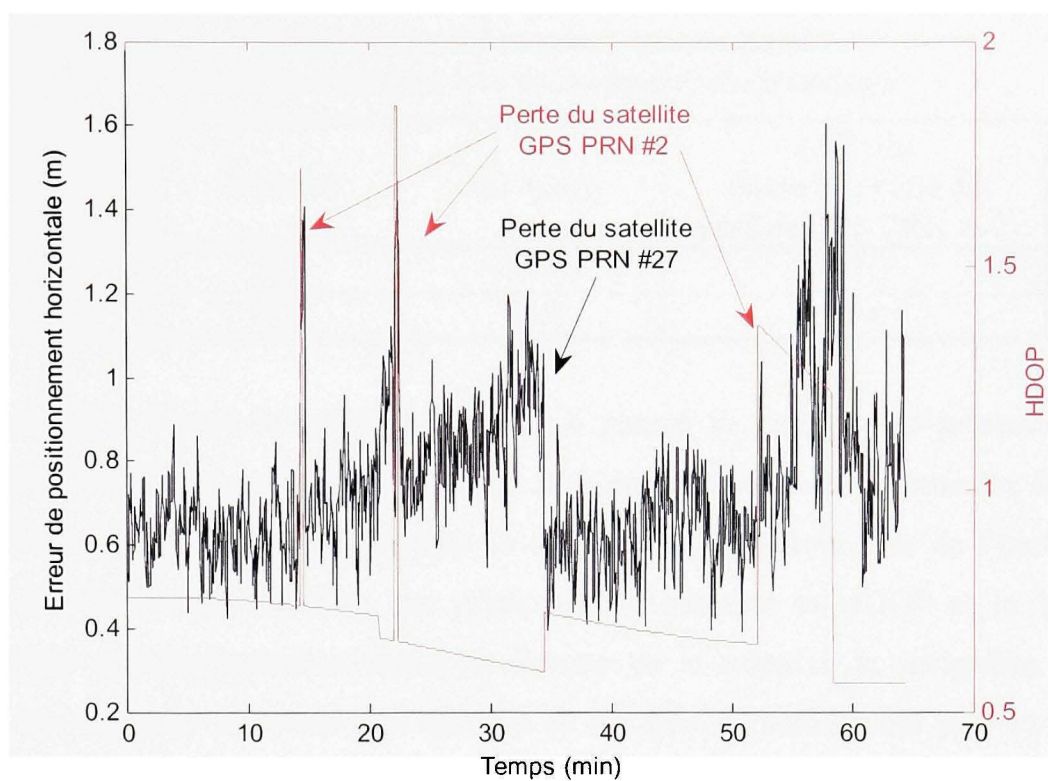
L'erreur de positionnement horizontale affichée à la Figure 6.5 a été calculée par trigonométrie, telle que :

$$\Delta H = \sqrt{\Delta N^2 + \Delta E^2} \quad (6.1)$$

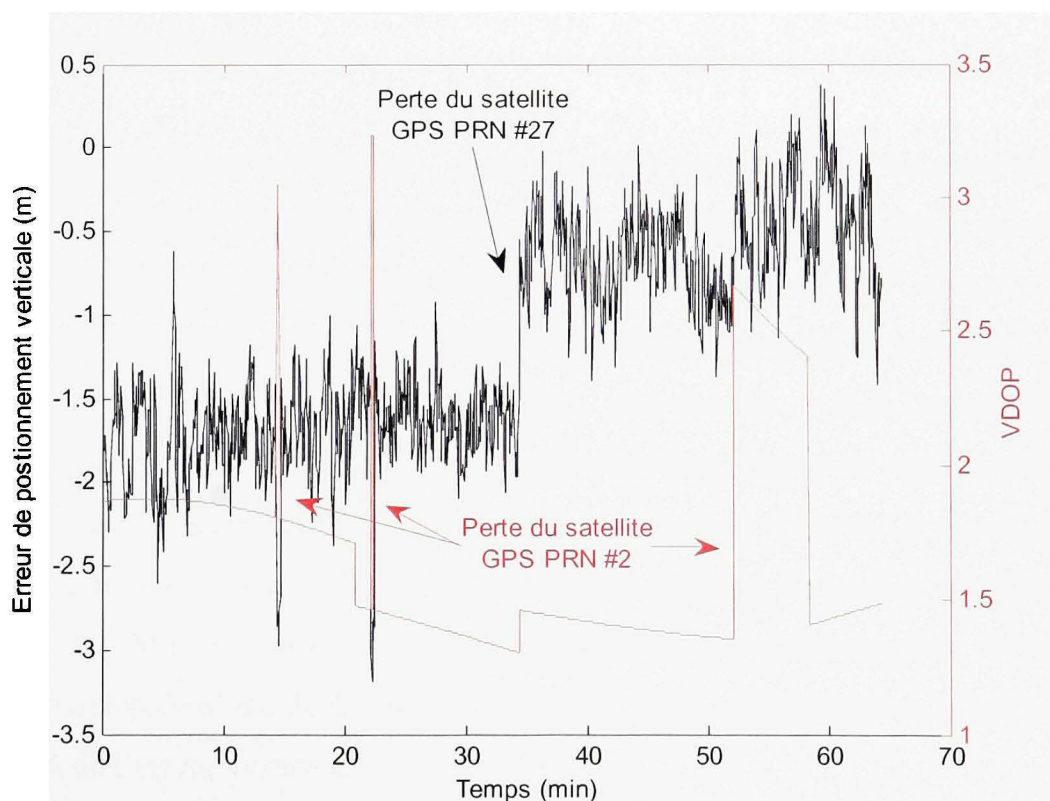
où :

- $\Delta H$  est l'erreur de positionnement horizontale
- $\Delta N$  est l'erreur de positionnement N-S
- $\Delta E$  est l'erreur de positionnement O-E

Les mesures de HDOP et de VDOP sont aussi affichées aux figures suivantes (se référer à la section 2.3.8 pour plus d'information au sujet des mesures de HDOP et de VDOP). La moyenne ( $\mu$ ) et la déviation standard ( $\sigma$ ) des erreurs horizontale et verticale ( $\Delta V$ ) sont quant à elles présentées au Tableau 6.1.



**Figure 6.5** *Erreur de positionnement horizontale et HDOP.*



**Figure 6.6** *Erreur de positionnement verticale et VDOP.*

Tableau 6.1  
Erreurs de positionnement moyennes et  
déviations standard lors de la simulation en statique

	$\Delta H$ (cm)	$\Delta V$ (cm)	$\Delta V$ (cm) (suite à la perte du satellite GPS PRN #27)
$\mu$	74,0	-117,0	-57,9
$\sigma$	18,7	79,4	31,6

Un examen de la Figure 6.5 et de la Figure 6.6 permet de constater la présence de deux « pics » entre les minutes 10 et 30. Ces pics sont dus aux pertes momentanées du satellite GPS PRN #2, ce qui est normal puisque ce dernier était situé tout près de l'horizon (voir Figure 6.4). Ces pertes ont alors fait grimper drastiquement la HDOP et la VDOP, se traduisant par une augmentation rapide de l'erreur de la solution de navigation. Ceci n'a toutefois pas empêché d'obtenir une solution de navigation relativement précise, avec une erreur horizontale de 74 cm en moyenne ainsi qu'une déviation standard de 18,7 cm (voir Tableau 6.1). L'erreur verticale est également intéressante, quoique supérieure à l'erreur horizontale (tel que prévu par la théorie), avec une erreur moyenne de -117 cm et une déviation standard de 79,4 cm.

Un examen plus détaillé de la Figure 6.5 permet de constater un phénomène curieux durant les 30 premières minutes : la HDOP diminue alors que l'erreur horizontale augmente. On note ensuite une correction abrupte de la solution de navigation vers la 35<sup>e</sup> minute. Ces deux phénomènes sont en fait liés au satellite GPS PRN #27 dont l'angle d'élévation est demeuré très faible ( $< 10^\circ$ ) tout au long de la simulation. Puisque les erreurs ionosphériques et troposphériques sont moins bien modélisées près de l'horizon, l'ajout du satellite #27 au sein de la solution de navigation a donc fait en sorte d'augmenter l'erreur de positionnement. L'erreur s'est ainsi automatiquement améliorée lors du passage du satellite sous le masque d'élévation de  $5^\circ$ . On note à partir de ce moment une erreur verticale moyenne de seulement -57,9 cm, accompagnée d'une déviation standard de 31,6 cm, ce qui représente une réduction d'environ 50% de l'erreur verticale.



Afin d'améliorer la solution de navigation, il serait donc avantageux d'augmenter le masque d'élévation à  $10^\circ$  plutôt que  $5^\circ$  afin d'être plus restrictif lors de la sélection des satellites. Une telle approche pourrait cependant limiter les performances du récepteur en environnement difficile, tel qu'en ville ou en forêt dense par exemple, alors que les satellites au zénith peuvent être occasionnellement bloqués. La technique privilégiée serait alors d'employer une matrice de pondération (**D**) au niveau du filtre de Kalman, tel qu'expliqué au chapitre 5, afin d'ajuster la matrice de covariance de bruit de mesure (**R**) en fonction de l'angle d'élévation des satellites. Une telle approche permettrait de limiter les erreurs induites par les satellites à basse élévation tout en ayant la possibilité d'employer ces derniers lors d'une obstruction des satellites au zénith.

## **6.4 Test dynamique en environnement simulé (test #3)**

### **6.4.1 Méthodologie du test #3**

Afin de valider le comportement du récepteur en déplacement, nous avons répété la même procédure que décrite à la section précédente. Un scénario en dynamique a toutefois été généré cette fois-ci de manière à simuler une trajectoire de test. La trajectoire en question, présentée à la Figure 6.7, est composée d'une série de manoeuvres différentes (montée, descente, virage, lignes droites), ponctuées d'accélérations variées, situées entre  $2 \text{ m/s}^2$  et  $1,5 \text{ G}$  ( $1 \text{ G} = 9,81 \text{ m/s}^2$ ). Il faut noter que la vitesse maximale était de  $200 \text{ m/s}$  et que les stress de jerk ont été limités à  $80 \text{ m/s}^3$ . Une représentation en 3D de la trajectoire est illustrée à la Figure 6.8.

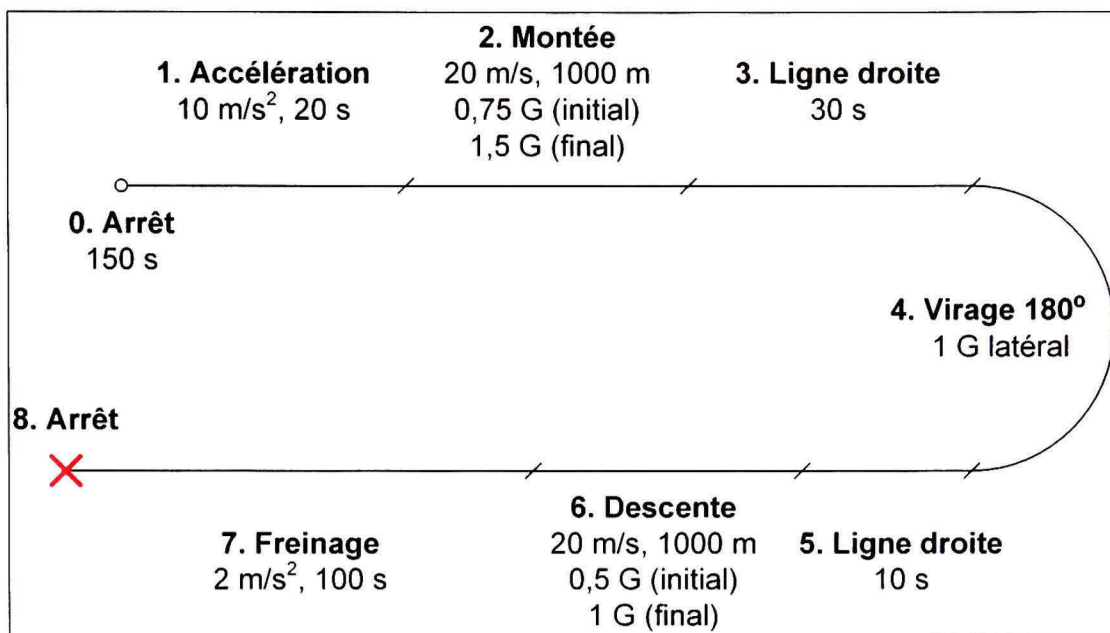


Figure 6.7 Détails de la trajectoire simulée.

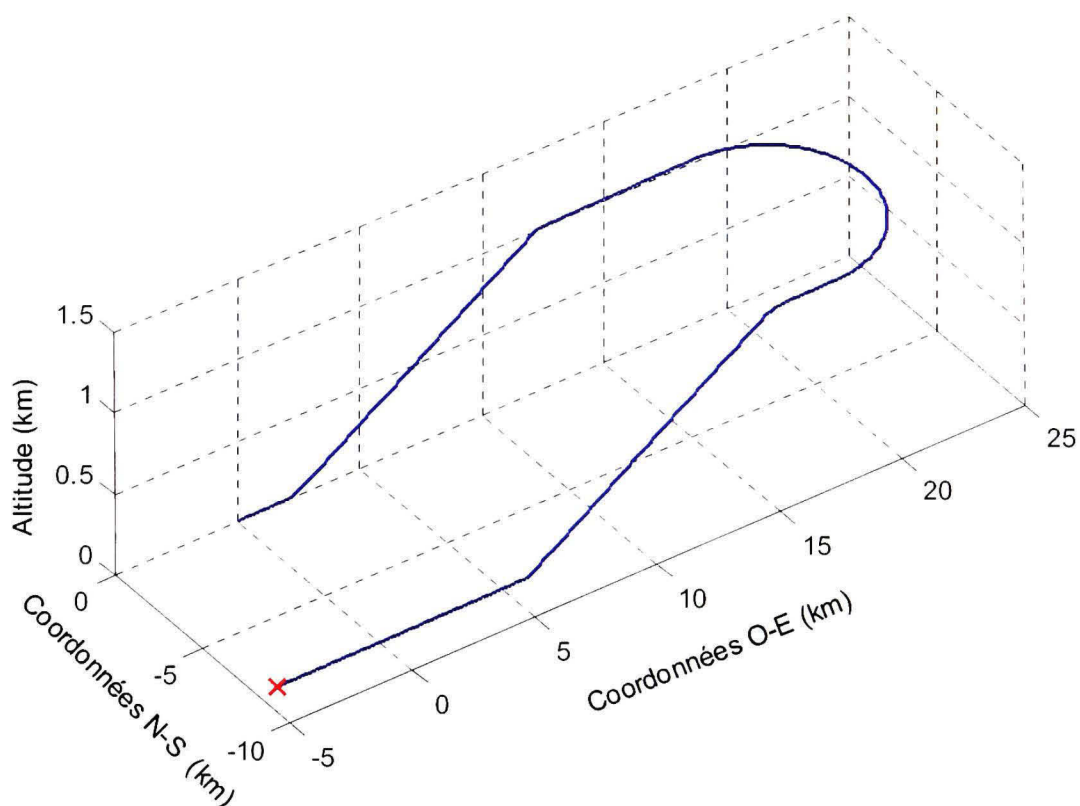


Figure 6.8 Représentation 3D de la trajectoire simulée.

### 6.4.2 Résultats et analyse du test #3

La position du récepteur étant connue lors de la simulation, nous avons donc une fois de plus été en mesure de déterminer précisément les erreurs en sortie de la solution de navigation (mise à jour ici à un taux de 20 Hz). La Figure 6.9 présente les erreurs de positionnement en 3D en fonction du temps alors que la Figure 6.10 montre l'erreur au niveau de la vitesse totale,  $\Delta v_{tot}$ , calculée telle que :

$$\Delta v_{tot} = \sqrt{(v_{rx} - v_x)^2 + (v_{ry} - v_y)^2 + (v_{rz} - v_z)^2} \quad (6.2)$$

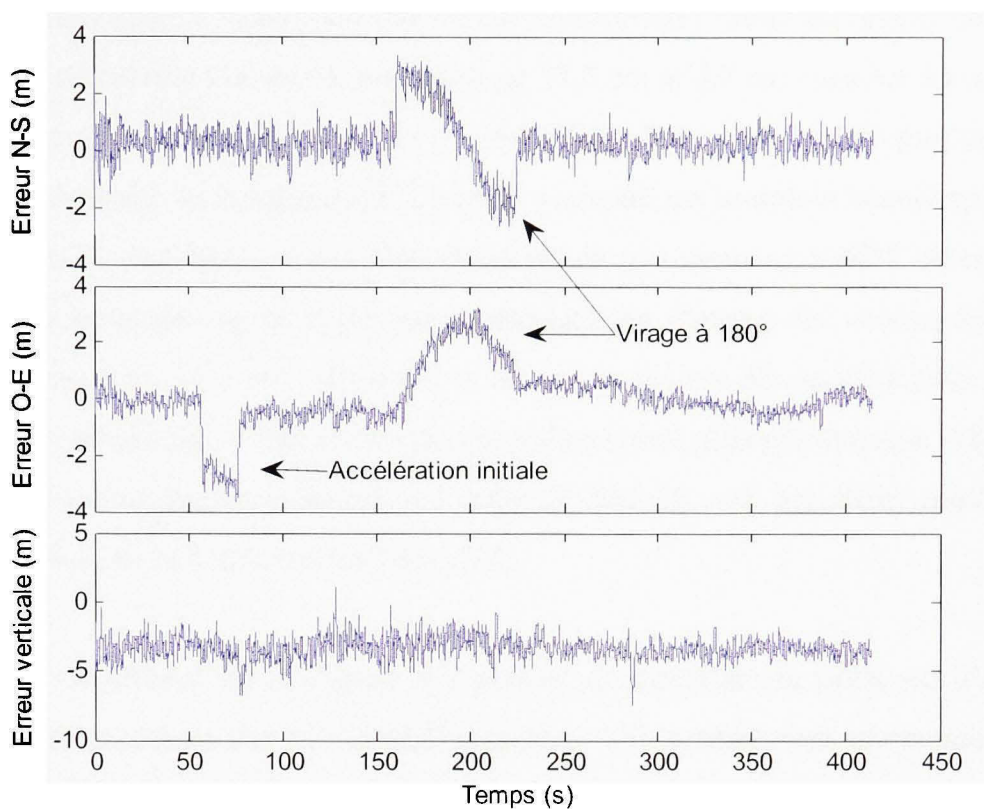
où  $v_{rx}$ ,  $v_{ry}$  et  $v_{rz}$  sont les vitesses en  $x$ ,  $y$  et  $z$  à la sortie du récepteur, tandis que  $v_x$ ,  $v_y$  et  $v_z$  sont les vitesses données par le simulateur.

Les erreurs de positionnement moyennes ainsi que leur déviation standard sont présentées au Tableau 6.3. À titre d'information, les HDOP et VDOP étaient d'environ 1 et 4 respectivement tout au long de la simulation. Il faut noter que les boucles de DLL de 0,1 Hz initialement développées se sont avérées incapables de poursuivre les signaux lorsque soumises à des accélérations de plus de 1 G. Il n'a donc pas été possible d'obtenir de résultat concluant à l'aide de ces boucles. Les résultats présentés ici ont plutôt été obtenus à l'aide de boucles de DLL indépendantes (i.e. sans aucune aide de la PLL) et de bande de bruit égale à 1 Hz.

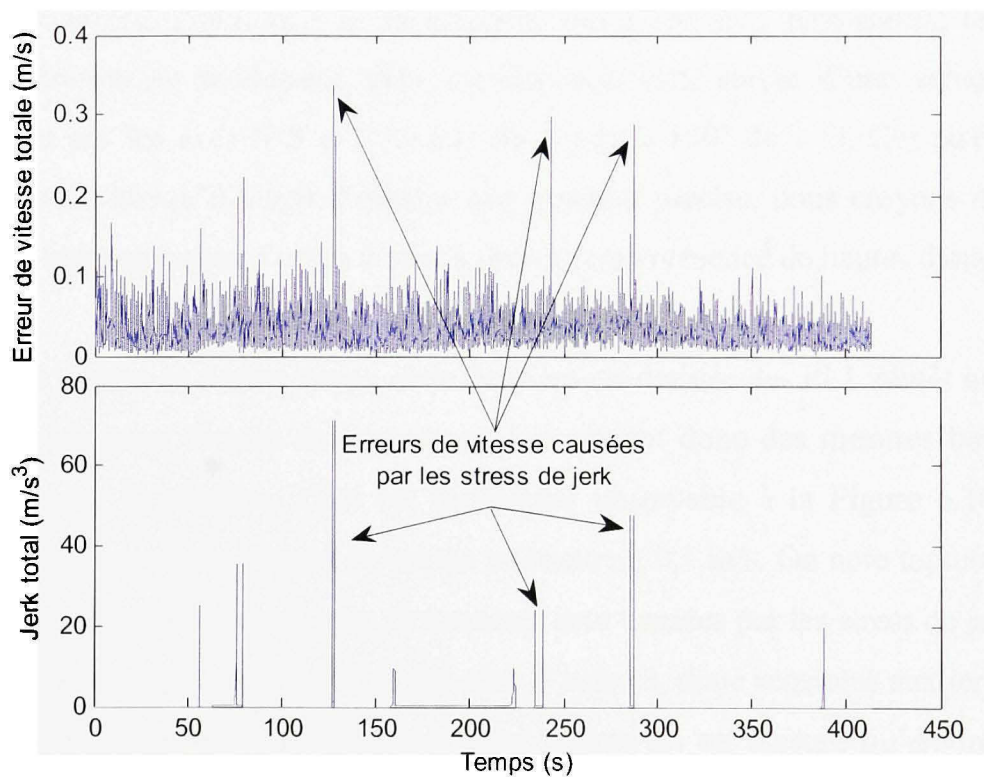
Tableau 6.2  
Erreurs de positionnement moyennes et  
déviations standard lors de la simulation en dynamique

	$\Delta H$ (N-S, cm)	$\Delta H$ (O-E, cm)	$\Delta V$ (cm)
$\mu$	27,5	4,2	-331,7
$\sigma$	77,3	104,4	87,0





**Figure 6.9** *Erreurs de positionnement nord-sud, ouest-est et verticale.*



**Figure 6.10** *Correspondance entre les erreurs de vitesse et les stress de jerk.*

En examinant la Figure 6.9, on note une erreur relativement faible au niveau des trois axes. Dans les faits, on calcule une erreur moyenne de 27,5 cm et 4,2 cm dans les directions N-S et O-E respectivement, ce qui est excellent en considérant les fortes dynamiques appliquées au récepteur tout au long de la trajectoire. L'erreur verticale est toutefois beaucoup plus grande avec -331,7 cm en moyenne, ce qui s'explique par le très mauvais VDOP obtenu lors de la simulation. Du point de vue de la déviation standard, on observe des valeurs très similaires sur les trois axes avec 77,3 cm, 104,4 cm et 87 cm au niveau des coordonnées N-S, O-E et verticales respectivement. Il faut noter que ces valeurs sont plus grandes que celles obtenues précédemment en statique (qui étaient de l'ordre de 20 à 30 cm), ceci étant une conséquence de l'augmentation de la bande de bruit des DLL.

Un examen plus attentif de la Figure 6.9 permet de constater la présence d'erreurs plus prononcées aux alentours des 50<sup>e</sup> et 150<sup>e</sup> secondes. Ces erreurs sont directement liées aux accélérations subies par le récepteur et au fait que les DLL sont de deuxième ordre. Des erreurs constantes et proportionnelles aux stress d'accélération apparaissent alors au niveau de la phase des DLL et se traduisent par un biais sur les mesures de pseudo-distances. On note ainsi une erreur d'environ 3 m en direction ouest lors de l'accélération initiale de 10 m/s<sup>2</sup> (l'accélération se produisant alors en direction est), suivie d'une erreur maximale d'environ 3 m sur les axes N-S et O-E lors du virage à 180° de 1 G. Ces erreurs peuvent s'avérer gênantes lorsqu'il s'agit d'obtenir une position précise, nous croyons donc que les boucles DLL indépendantes d'ordre 2 sont à proscrire en présence de hautes dynamiques.

Concernant la vitesse de déplacement, cette dernière est dérivée des PLL plutôt que des DLL. Vu la courte longueur d'onde de la porteuse, on obtient donc des mesures beaucoup plus précises que pour la position. Ceci est facilement observable à la Figure 6.10, alors que l'erreur sur la vitesse est la plupart du temps inférieure à 0,1 m/s. On note toutefois quelques erreurs ponctuelles dépassant 0,2 m/s. Ces erreurs sont causées par les stress de jerk subis par le récepteur et s'expliquent par les PLL qui sont d'ordre 3, donc sensibles aux jerks. Les PLL se comportent tout de même relativement bien puisqu'on ne mesure qu'environ 0,35 m/s d'erreur pour des jerks aussi grands que 70 m/s<sup>3</sup>.

En résumé, le récepteur de navigation développé semble se comporter relativement bien en hautes dynamiques. Les boucles de PLL et de DLL demeurent stables et la solution de navigation arrive à fournir une position assez précise. Les DLL devront toutefois être modifiées afin d'obtenir une précision équivalente à celle obtenue en statique. Deux options sont alors possibles : augmenter l'ordre des DLL ou corriger le mécanisme d'aide provenant des PLL. La première option offre sans doute la solution la plus robuste puisqu'elle ne dépend pas des PLL, généralement considérées comme le maillon faible d'un récepteur de navigation. Une bande de bruit relativement large doit toutefois être conservée dans ce cas-ci, faisant en sorte d'augmenter la quantité de bruit thermique au niveau des mesures de pseudo-distances. L'approche la plus intéressante demeure donc la seconde option puisqu'elle permet en théorie de diminuer la bande de bruit des DLL sans pour autant compromettre leur robustesse face aux dynamiques externes (Kaplan, 1996). Si on se fie à la théorie (et aux fiches techniques de certains manufacturiers qui mentionnent l'emploi de DLL de 0,01 Hz aidées par les PLL) (Novatel, 2007b), il devrait donc y avoir un problème au niveau du mécanisme d'aide aux DLL développé. Il serait alors suggéré de réaliser des vérifications plus approfondies à ce niveau.

## **6.5 Évaluation du temps de démarrage en environnement réel (test #4)**

### **6.5.1 Méthodologie du test #4**

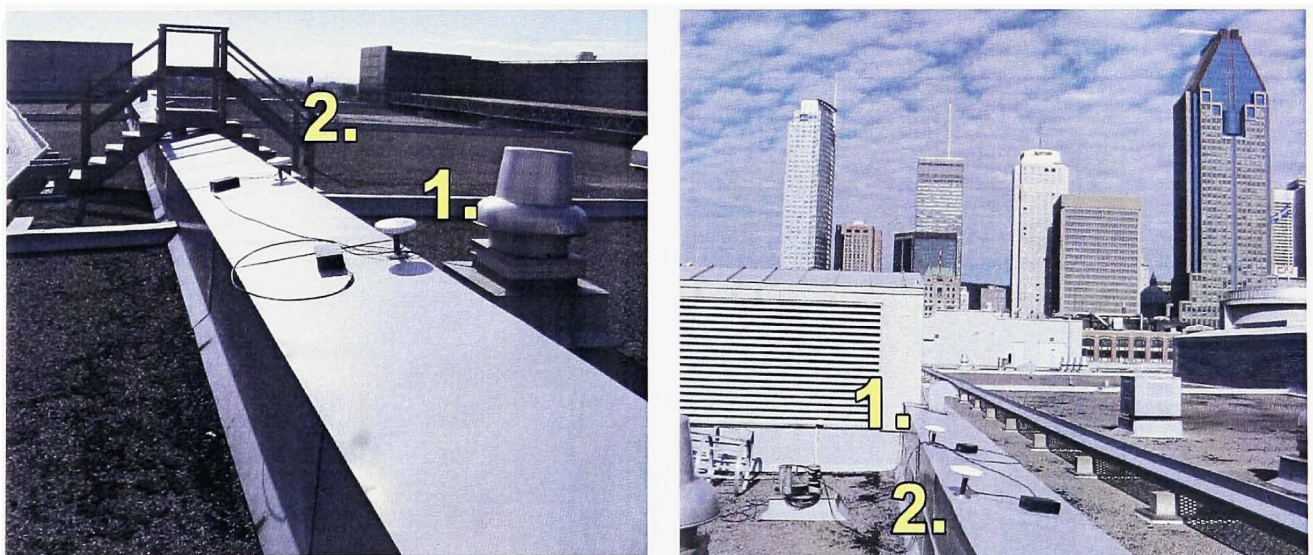
Une caractéristique importante d'un récepteur de navigation est son temps de démarrage. Ce temps (ou délai) de démarrage est généralement évalué entre les moments où le récepteur a été mis sous tension et celui où la première position a été obtenue. Dans le cas de notre récepteur, ce temps a plutôt été évalué à partir du moment où le FPGA a été programmé. Ce temps de programmation a donc été mesuré en plus des temps de démarrage suivants :

- à chaud : les éphémérides et l'almanach sont valides et une estimation du temps et de la position est disponible;
- à tiède : seul l'almanach est valide et une estimation du temps et de la position est disponible;
- à froid : aucune information n'est disponible.

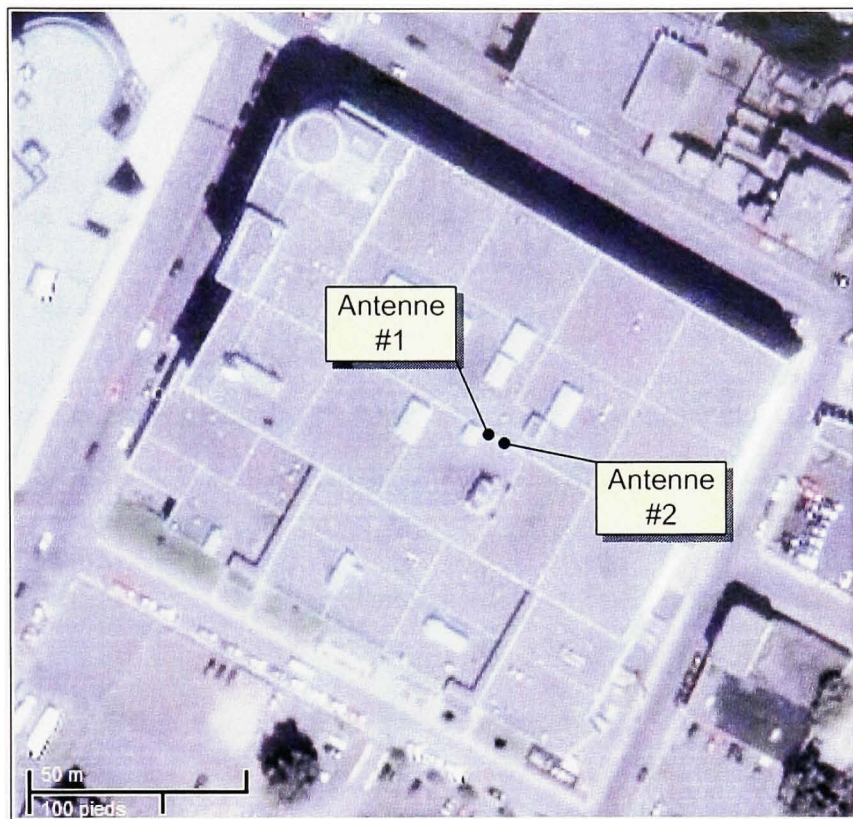


Les temps de démarrage ont été mesurés de façon manuelle à l'aide d'un chronomètre. Ceci implique que des erreurs de quelques dixièmes de secondes se sont insérées au niveau des mesures. De cinq à dix mesures consécutives ont toutefois été réalisées de manière à s'assurer de la reproductibilité des résultats.

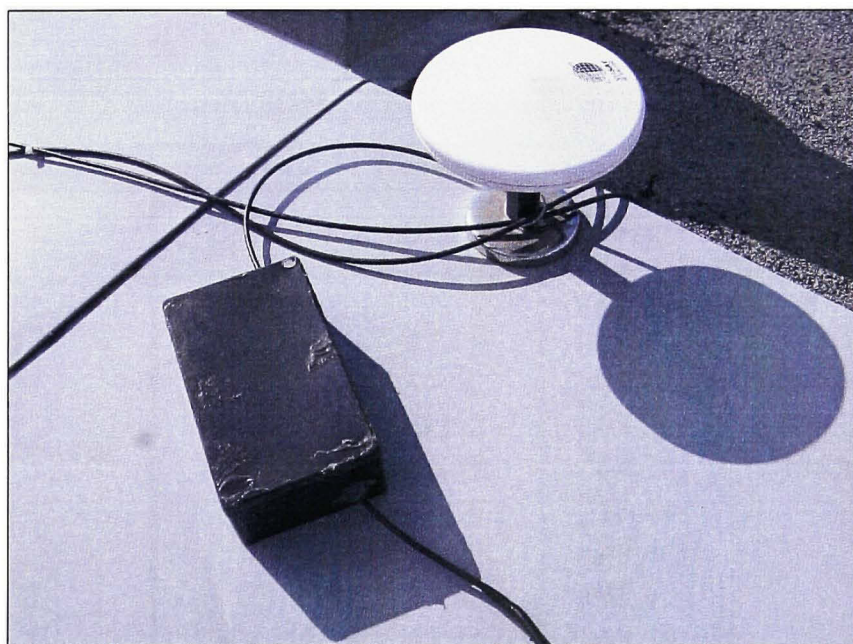
Les mesures ont été réalisées en environnement réel à l'aide d'une antenne située sur le toit de l'ÉTS. L'antenne #2 (voir Figure 6.11 et Figure 6.12) a été employée puisqu'elle offrait la vue la plus dégagée du ciel. Les antennes déployées sur le toit sont des antennes L1 actives AT575-75 de marque AeroAntenna Technology, de polarisation circulaire droite (RHCP – *Right Hand Circular Polarisation*) et de 10 MHz de largeur de bande. Puisqu'une grande distance sépare les antennes du récepteur (la longueur des câbles les reliant étant d'environ 38 m), un amplificateur de 30 dB (WS2901-30 de Wi-Sys Communications) (Wi-Sys, 2003) a été connecté à chacune des antennes afin de compenser les pertes dans les câbles. Les amplificateurs ont été scellés dans des boîtes étanches afin de les protéger des intempéries (voir Figure 6.13).



**Figure 6.11** Localisation des antennes GPS sur le toit de l'ÉTS;  
vues direction sud-est (gauche) et nord-ouest (droite).



**Figure 6.12** *Photographie satellite de l'ÉTS.*  
Tirée de Google Maps ([www.google.com/map](http://www.google.com/map))



**Figure 6.13** *Amplificateur (gauche) et antenne (droite).*



### 6.5.2 Résultats et analyse du test #4

Tel que prévu, nous avons eu une bonne visibilité du ciel lors des tests, avec au moins sept satellites GPS captés en tout temps. Le ratio  $C/N_0$  des signaux s'est avéré très fort avec des valeurs situées entre 40 et 50 dB Hz. Les divers temps de démarrage mesurés lors des tests sont présentés au Tableau 6.3.

En observant Tableau 6.3, on note que la programmation du FPGA requiert environ 11,5 secondes à elle seule. Ceci fait en sorte de ralentir considérablement le temps de démarrage du récepteur, qui est d'environ 27 s à chaud, 73 s à tiède et 136 s à froid en incluant le temps de programmation du FPGA. Des temps de démarrage beaucoup plus rapides pourraient ainsi être atteints en ne configurant le FPGA qu'une seule fois lors de la mise sous tension de la plateforme. On obtiendrait alors environ 16 s pour un démarrage à chaud, 62 s à tiède et 124 s à froid tel qu'indiqué au Tableau 6.3. Ces temps sont relativement similaires au récepteur DL-4plus de Novatel (employé entre autres à la section 6.2 - Étude du mécanisme de mesure du SNR) dont les temps de démarrage typiques sont de 30 s à chaud, 40 à tiède et 60 s à froid (Novatel, 2007b).

Tableau 6.3  
Mesures du temps de démarrage du récepteur

	Temps de démarrage (s)			
	Prog. FPGA	Chaud	Tiède	Froid
Mesures	11,64	17,71	58,45	133,03
	11,52	18,05	61,82	113,19
	11,35	15,48	69,82	107,79
	11,53	14,02	59,03	129,29
	11,52	14,53	64,24	138,19
	-	18,19	53,41	-
	-	12,90	49,60	-
	-	15,32	77,10	-
	-	15,44	68,72	-
	-	14,89	53,18	-
Moyenne	11,51	15,65	61,54	124,30
Écart type	0,09	1,69	8,15	11,75
Écart max.	0,29	5,29	27,50	30,40



Un autre fait intéressant est qu'un écart de 5,29 s existe entre le temps de démarrage à chaud le plus rapide et celui le plus lent. Cet écart est directement lié à la période de 6 s des sous-frames du message de navigation GPS L1 C/A. Or, afin de se synchroniser avec les messages de navigation, le récepteur doit au préalable recevoir une sous-frame entière de chacun des satellites. Puisque le récepteur démarre à un instant aléatoire, il est donc possible qu'un délai de 0 à 6 s vienne s'ajouter au temps de démarrage.

Un écart apparaît aussi au niveau des démarrages à tiède et à froid. On parle toutefois d'un temps d'environ 30 s dans les deux cas. Ceci s'explique par le téléchargement des éphémérides contenues au début de chacune des trames GPS L1 C/A. Par souci de simplicité, nous avons préféré télécharger chacune des trames en entier avant d'en extraire les éphémérides, ce qui nécessite au moins 30 s. Puisque le récepteur démarre à un instant aléatoire, il est toutefois nécessaire en pratique de télécharger l'équivalent d'une à deux trames afin d'en obtenir au moins une complète, ce qui requiert 45 s en moyenne. Un gain d'au moins 30 s pourrait alors être réalisé lors des démarrages à tiède et à froid en décodant, dès leur réception, les trois sous-frames contenant les éphémérides. Des temps de démarrage d'au plus 32 s à tiède et 94 s à froid pourraient alors être réalisés (soit 43 s à froid et 106 s à tiède en incluant le temps de programmation du FPGA).

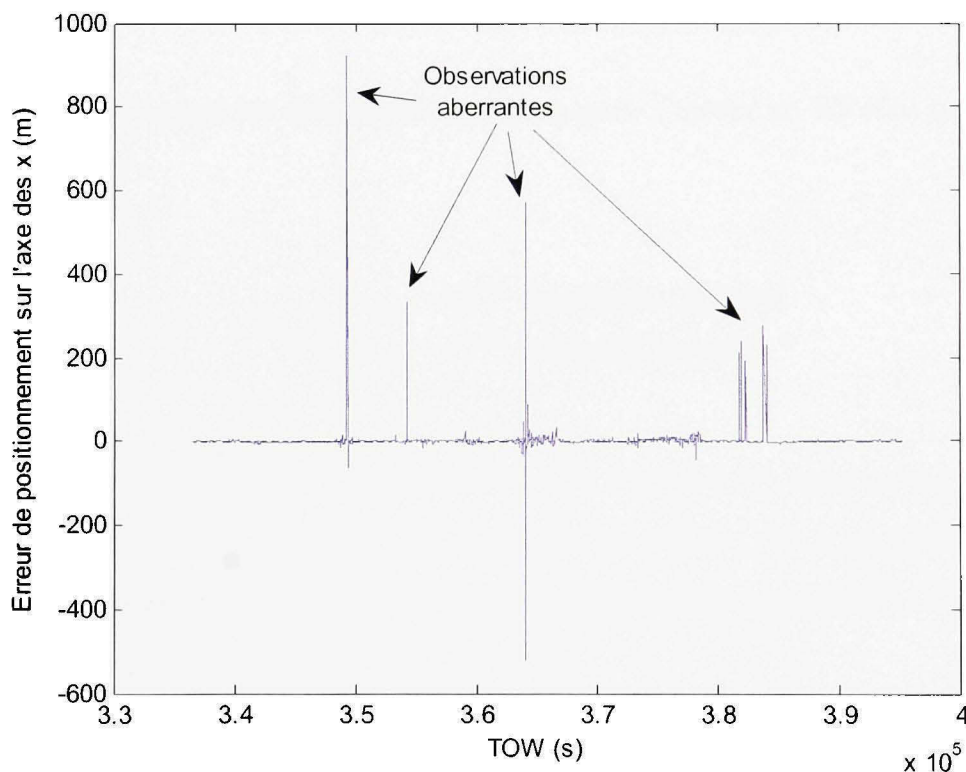
## **6.6 Test statique en environnement réel (test #5)**

### **6.6.1 Méthodologie du test #5**

Afin de quantifier la précision de notre récepteur en environnement réel, nous avons réutilisé le montage présenté à la section précédente. Nous avons toutefois utilisé un récepteur DL-4plus de Novatel afin de comparer les résultats obtenus par notre récepteur. Le diviseur présenté à la section 6.2 (Étude du mécanisme de mesure du SNR) a été employé de manière à distribuer les signaux aux deux récepteurs. Nous avons réglé la période de mise à jour des récepteurs à 3 s (i.e. la période de mise à jour des solutions de navigation) puis les avons laissé opérer durant au moins 16 heures. Les résultats obtenus pour chacun des récepteurs sont comparés à la section suivante.

### 6.6.2 Résultats et analyse du test #5

Lors des tests, un problème a été détecté au niveau de notre récepteur. Des sauts de position anormalement grands ont en fait été mesurés lors de l'ajout de certains satellites au sein de la solution de navigation. Ces sauts, visibles à la Figure 6.14, dépassent facilement les 200 m sur chacun des trois axes et s'étendent sur plusieurs échantillons consécutifs. Or, aucune observation du genre n'a été faite lors des tests précédents, ceci étant dû au fait que nous n'avions jamais réalisé de test sur de longues périodes (le phénomène semble ne s'être produit que six fois en 17 heures). Afin d'éliminer le problème (ou du moins en limiter l'impact), une transition plus douce devrait sans doute être considérée lors de l'ajout de nouveaux satellites dans la solution de navigation. Vu le développement rapide du filtre de Kalman, nous croyons aussi qu'il serait approprié de réviser l'algorithme de façon approfondie en mettant au point une série de tests rigoureux.



**Figure 6.14** *Présence de données aberrantes en sortie de la solution de navigation.*

Pour les besoins de l'analyse, nous avons tout de même réussi à contourner le problème en filtrant la position retournée par notre récepteur. Pour ce faire, nous avons tout d'abord estimé les erreurs de positionnement sur les trois axes ( $x$ ,  $y$  et  $z$ ) en calculant :

$$\Delta x_i = x_i - \bar{x}, \quad \Delta y_i = y_i - \bar{y}, \quad \Delta z_i = z_i - \bar{z} \quad (6.3)$$

où :

- $\Delta x_i$ ,  $\Delta y_i$  et  $\Delta z_i$  sont les erreurs estimées sur les trois axes à l'instant  $i = 0, 1, 2, \dots$
- $\bar{x}$ ,  $\bar{y}$  et  $\bar{z}$  sont les positions moyennes, calculées sur l'ensemble des données

Une fois les erreurs de positionnement estimées, nous avons calculé l'erreur en 3D ( $\Delta_{3D}$ ) de la solution de navigation, telle que :

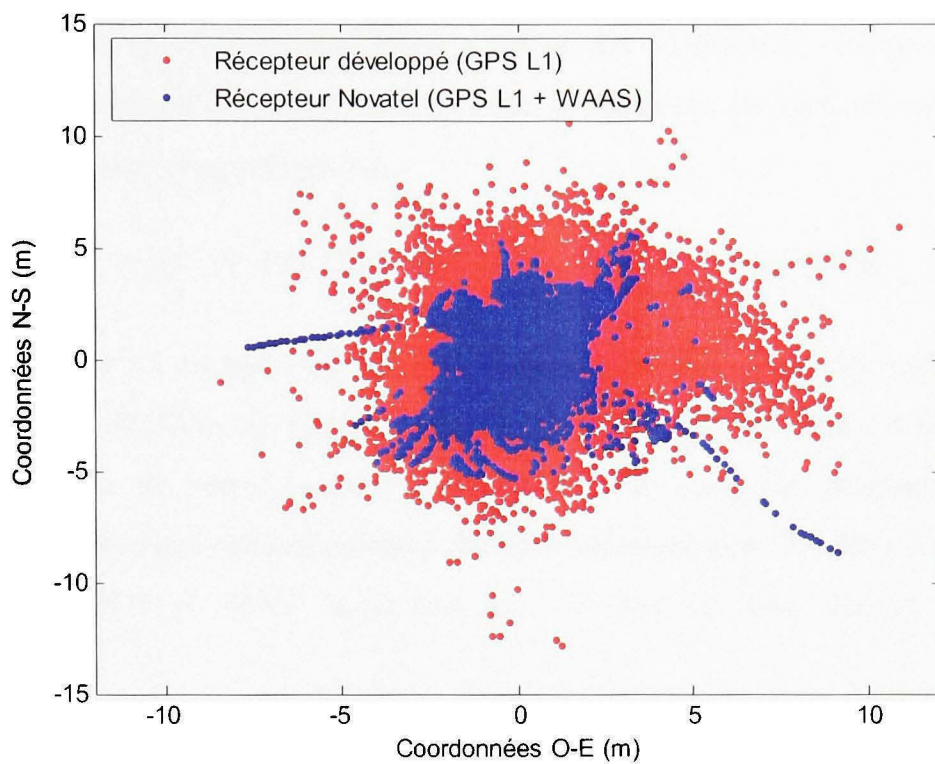
$$\Delta_{3Di} = \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2} \quad (6.4)$$

Nous avons ensuite rejeté les échantillons pour lesquels l'erreur en 3D était trop grande grâce au critère suivant :

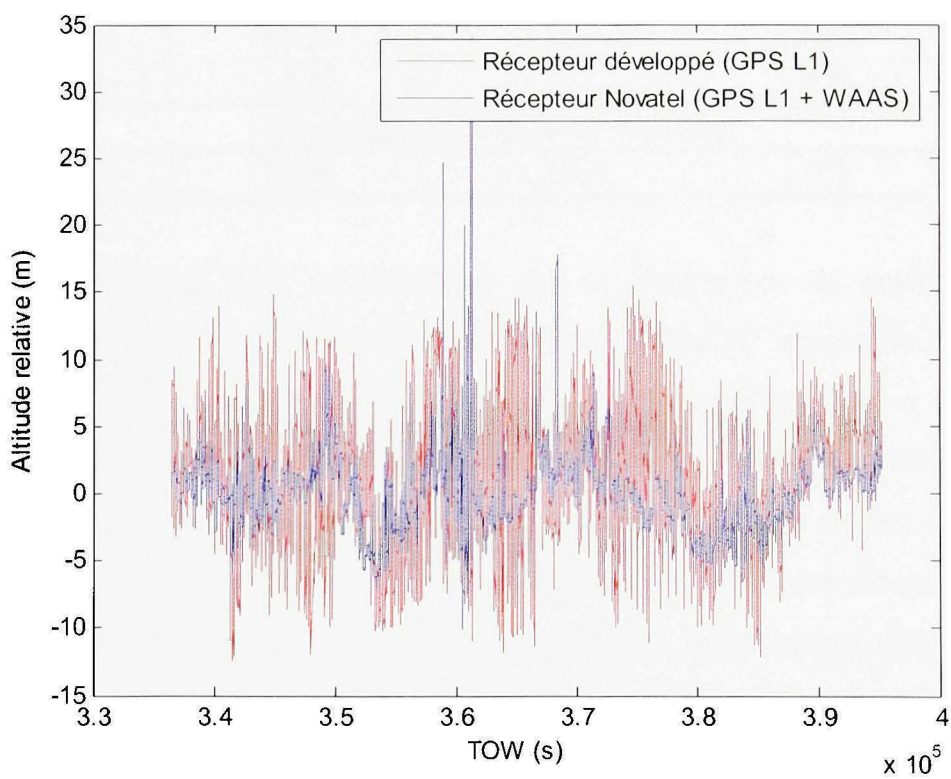
$$\Delta_{3Di} \begin{cases} < e_{\max} \rightarrow i^{\text{ème}} \text{ échantillon accepté} \\ \geq e_{\max} \rightarrow i^{\text{ème}} \text{ échantillon rejeté} \end{cases} \quad (6.5)$$

Pour les résultats présentés à la Figure 6.15 et à la Figure 6.16, nous avons fixé le seuil de rejet ( $e_{\max}$ ) de manière à éliminer les données aberrantes, sans toutefois dénaturer ou améliorer artificiellement les résultats de positionnement. Un seuil de 20 m s'est donc avéré judicieux dans ce cas-ci.





**Figure 6.15** *Distribution horizontale des solutions de navigation.*



**Figure 6.16** *Altitudes relatives en fonction du temps.*

Les résultats présentés ci-dessus ont été obtenus à partir de 21h30 UTC le 27 juin 2007, ce qui équivaut à un TOW d'environ  $3,366 \times 10^5$  s. La position de référence a été choisie comme étant la position moyenne retournée par le récepteur de Novatel, correspondant en terme de coordonnées géographiques à :

$$N\ 45^\circ\ 29'\ 40,3434'' \quad W\ 73^\circ\ 33'\ 44,4193'' \quad +6,9547\ m$$

L'altitude indiquée ici est par rapport à l'ellipsoïde WGS-84 (ce qui équivaut à une hauteur approximative de 38,47 m au dessus du géoïde EGM-96). Le Tableau 6.4 présente l'erreur moyenne en sortie de notre récepteur par rapport à ce point de référence. La déviation standard des mesures de positionnement y est aussi présentée pour les deux récepteurs. Il faut noter que les HDOP et VDOP moyennes tout au long des tests étaient de 0,9 et 2,73 respectivement.

Tableau 6.4  
Erreurs de positionnement moyennes et déviations standards lors des tests statiques en environnement réel

	$\Delta H$ (N-S, m)	$\Delta H$ (O-E, m)	$\Delta V$ (m)
<b>Récepteur développé</b>			
$\mu$	-0,035	0,404	1,226
$\sigma$	2,520	2,152	4,458
<b>Récepteur DL-4plus de Novatel</b>			
$\sigma$	1,580	1,042	2,728

En examinant la Figure 6.15, on remarque que la distribution de position horizontale (représentée par un nuage de points) de notre récepteur coïncide avec celle du récepteur de Novatel. On mesure ainsi une erreur moyenne de -3,5 cm sur l'axe N-S et de 40,4 cm sur l'axe O-E entre les deux distributions. Le nuage de point donné par notre récepteur est cependant plus grand que celui du récepteur de Novatel. On parle en fait d'une déviation standard d'environ 2,52 m et 2,15 m sur les axes N-S et O-E pour notre récepteur, par rapport à 1,58 m et 1,04 m environ pour le récepteur de Novatel. La précision obtenue à l'aide de notre récepteur demeure tout de même excellente, surtout lorsqu'on sait que le récepteur de Novatel bénéficie du WAAS pour corriger sa position. Or, la fiche technique de ce dernier mentionne une erreur CEP de 1,8 m sans WAAS (Novatel, 2007b), ce qui équivaut à une

déviations standard de 2,16 m horizontalement (la conversion entre CEP et RMS est présentée au chapitre 2).

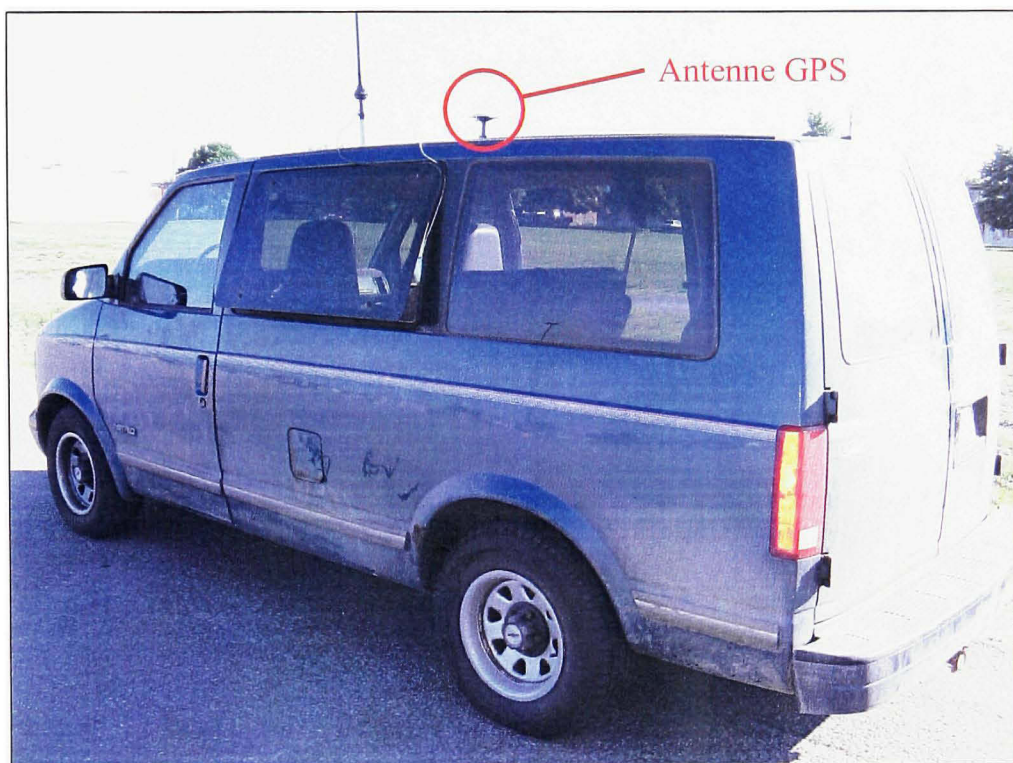
En examinant la Figure 6.16, on remarque que les mesures d'altitude provenant des deux récepteurs suivent sensiblement la même tendance. Un biais de 1,22 m fait cependant son apparition entre les mesures des deux récepteurs, ce qui s'explique sans doute par les corrections ionosphériques dont bénéficie le récepteur de Novatel (corrections fournies par le WAAS). On note tout de même l'apparition de quelques sauts brusques dans la solution de Novatel aux alentours de  $TOW = 3,6 \times 10^5$  s. Ces sauts, de l'ordre de 15 à 35 m, n'apparaissent pas dans notre solution puisqu'ils ont été retirés au préalable.

## **6.7 Test dynamique en environnement réel (test #6)**

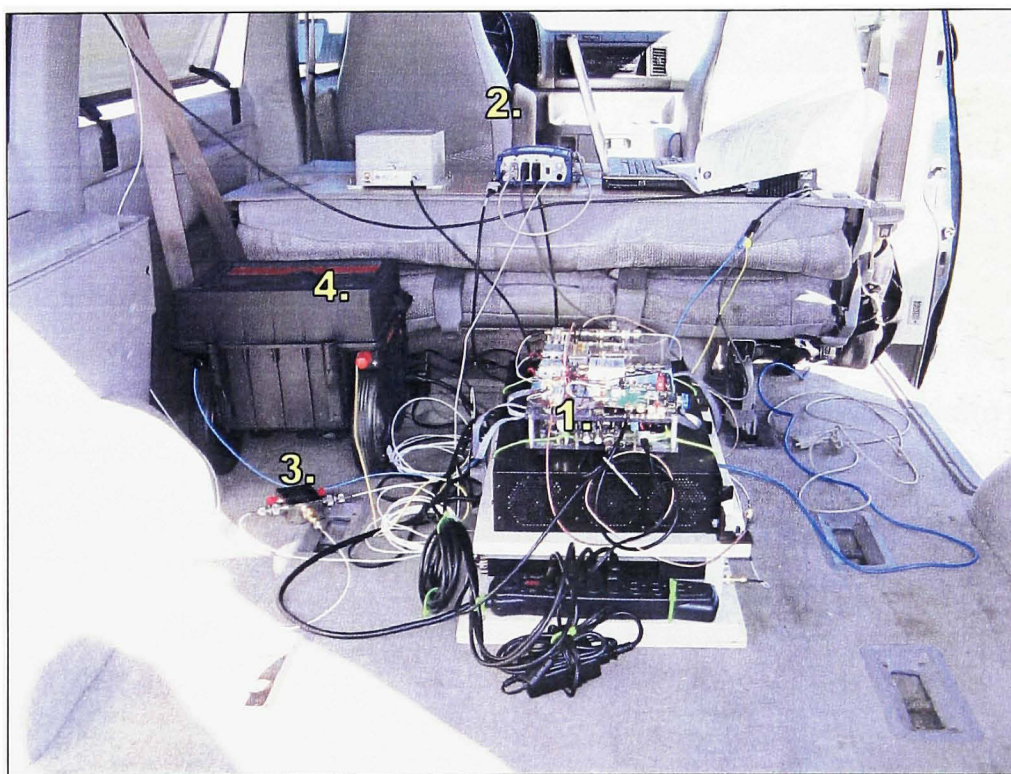
### **6.7.1 Méthodologie du test #6**

Afin de pousser plus loin l'étude de notre récepteur, nous avons réalisé des tests dynamiques en environnement réel. Pour ce faire, nous avons emprunté un véhicule de l'ÉTS (voir Figure 6.17), que nous avons surmonté d'une antenne GPS Omnistar d'Antcom Corporation (Modèle 5G01215A-XT-1, soit une antenne RHCP active en bande L1 et L2) (Antcom Corporation, 2001). Tel que montré à la Figure 6.18, nous avons chargé notre récepteur (1) et un récepteur DL-4plus de Novatel (2) à l'arrière du véhicule. Les signaux provenant de l'antenne ont pu ensuite être acheminés aux deux récepteurs grâce au diviseur (3) présenté à la section 6.2 (Étude du mécanisme de mesure du SNR). Enfin, une batterie de 12 V munie d'un onduleur (4) a été employée afin d'alimenter le montage final.

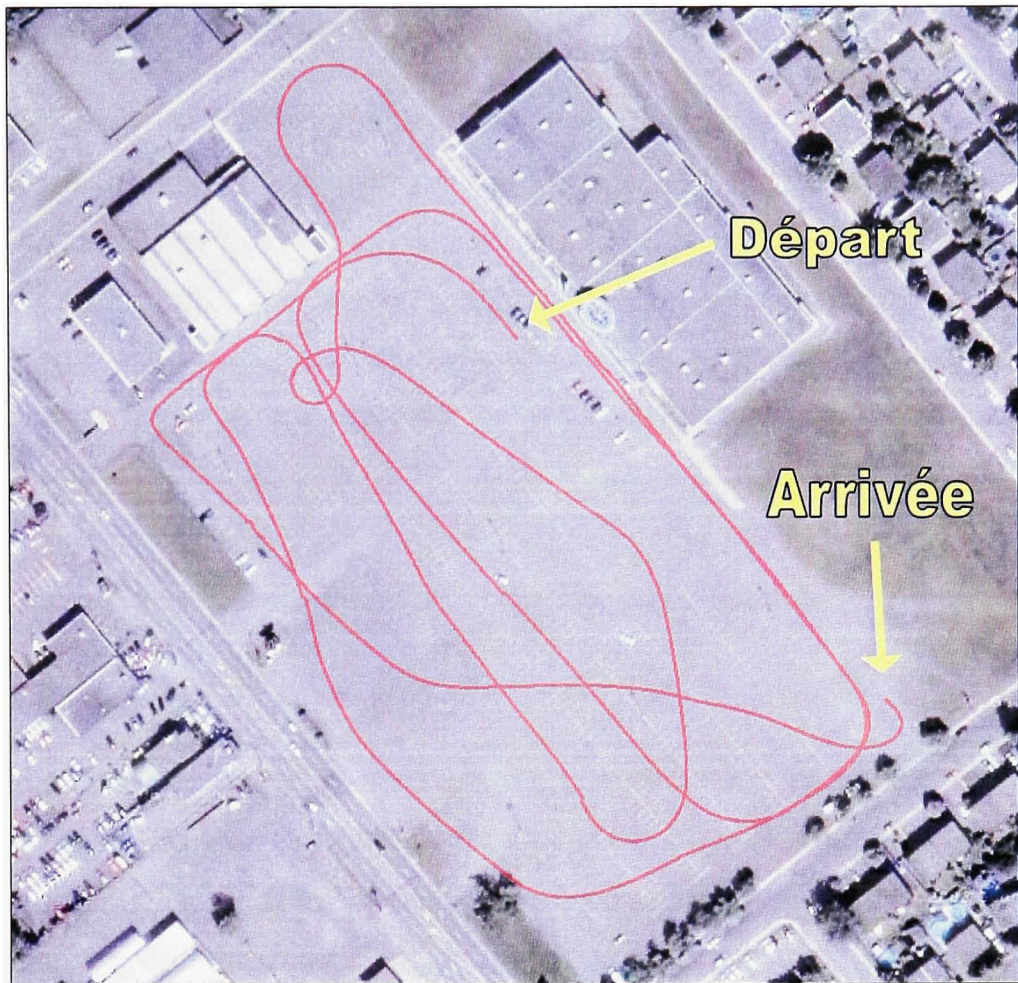




**Figure 6.17** Photographie du véhicule de test.



**Figure 6.18** Montage employé lors des tests dynamiques en environnement réel.



**Figure 6.19** *Trajectoire de test en environnement réel.*  
Image satellite tirée de Google Maps ([www.google.com/maps](http://www.google.com/maps))

Une fois le montage complété, nous nous sommes dirigés vers Laval pour entreprendre des tests dans un stationnement vacant (voir Figure 6.19). Une série de manœuvres y ont alors été réalisées, telles que des accélérations, des freinages et des virages. La trajectoire présentée à la Figure 6.19 provient de la solution de navigation du récepteur de Novatel. Une comparaison entre cette solution de navigation et celle obtenue par notre récepteur est présentée à la section suivante.

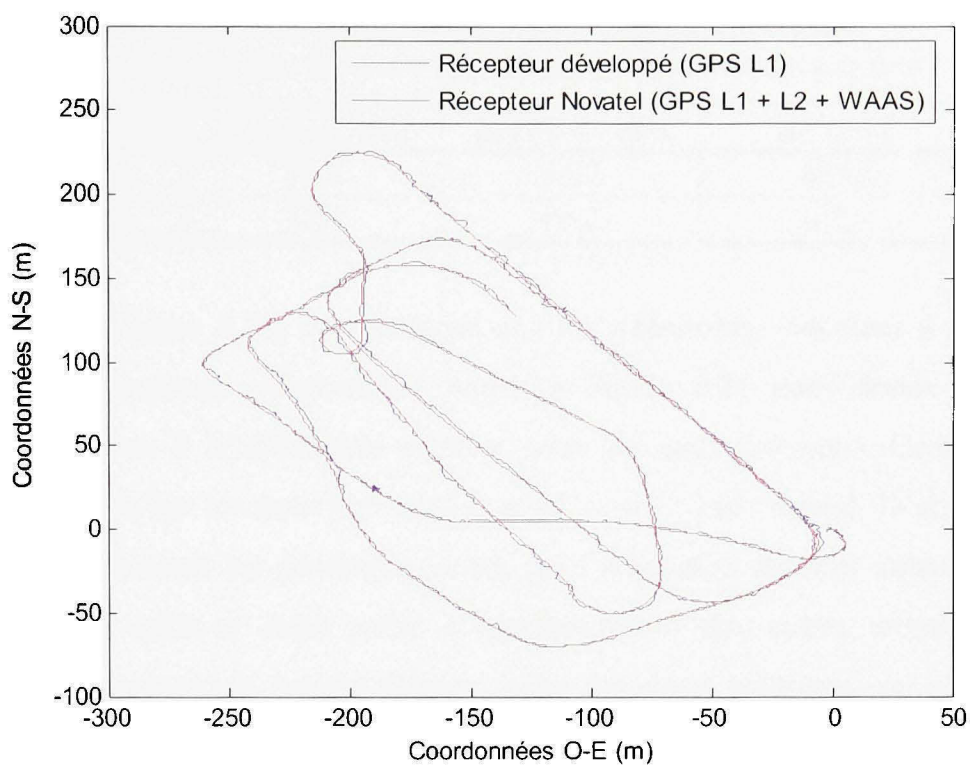


### 6.7.2 Résultats et analyse du test #6

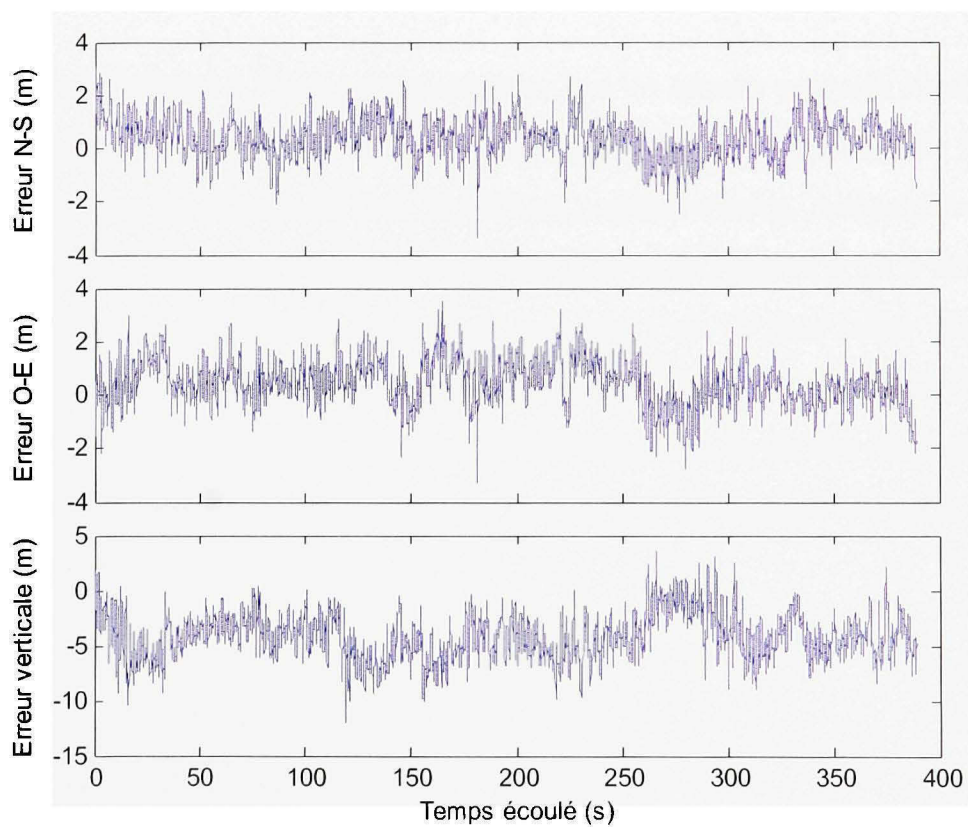
Quelques problèmes ont été éprouvés lors des tests. En effet, suite à une première série de tests, le 14 juin 2007, nous nous sommes aperçus que le récepteur de Novatel n'avait rien enregistré du tout. Nous avons donc été obligés de répéter une seconde fois les tests, cette fois-ci le 5 octobre 2007. Lors de la seconde série de tests, nous avons toutefois réalisé qu'il nous manquait un câble afin de connecter le diviseur de signaux. Étant déjà situés loin du laboratoire, nous avons dû nous débrouiller en connectant le récepteur de Novatel à une seconde antenne : une antenne GPS-702 de Novatel, très similaire à l'antenne Omnistar d'Antcom (Novatel, 2007a). Nous avons alors pris soin de placer les centres de phase des deux antennes très près l'un de l'autre ( $< 40$  cm) en les alignant dans l'axe de déplacement du véhicule.

La Figure 6.20 présente les solutions de navigation obtenues lors du second test en question. La Figure 6.21 illustre quant à elle la différence entre les solutions de navigation de deux récepteurs. Enfin, le Tableau 6.5 montre les différences moyennes ainsi que les déviations standards entre les solutions des deux récepteurs. Il faut noter que de six à neuf satellites étaient visibles en permanence, les satellites occasionnellement perdus étant ceux situés sous  $15^\circ$  d'élévation. Les HDOP et VDOP moyennes étaient quant à elles de 0,92 et 3,26, respectivement. Il faut aussi mentionner que le récepteur de Novatel opérait en mode GPS L1 + L2 + WAAS, lui conférant ainsi une précision de 0,8 m CEP (Novatel, 2007b) (ce qui équivaut à une déviation standard de 0,96 m horizontalement).





**Figure 6.20** *Superposition des solutions de navigation horizontales.*



**Figure 6.21** *Différence entre les solutions de navigation.*

Tableau 6.5  
Erreurs de positionnement moyennes et déviations  
standards lors des tests dynamiques en environnement réel

	$\Delta H$ (N-S, cm)	$\Delta H$ (O-E, cm)	$\Delta V$ (cm)
$\mu$	39,2	45,7	-415,6
$\sigma$	73,1	86,2	207,8

En examinant la Figure 6.20, on remarque que les trajectoires calculées à partir des deux récepteurs se superposent relativement bien. La Figure 6.21 nous donne cependant une meilleure idée quant à la différence existant entre les deux solutions. Cette différence ne dépasse jamais 4 m sur les axes N-S et O-E et 12 m sur l'axe vertical. D'ailleurs, on arrive difficilement à percevoir les décalages causés par l'utilisation de deux antennes différentes, la distance entre celles-ci étant petite comparativement aux autres erreurs. Or, le bruit thermique est la principale source d'erreur entre les deux solutions, ce qui pourrait être amélioré en diminuant la bande de bruit de nos DLL (nous avons ajusté la bande de bruit des DLL à 1 Hz lors de ce test – sans aide des PLL – de manière à accroître la robustesse du récepteur face aux dynamiques).

Un coup d'œil au Tableau 6.5 permet de constater que les erreurs sont très similaires à celles obtenues précédemment par simulation. On parle ainsi d'erreurs moyennes de 39,2 cm, 45,7 cm et -415,6 cm sur les axes N-S, O-E et vertical respectivement, comparativement à 27,5 cm, 4,2 cm et -331,7 cm pour la simulation dynamique (voir Tableau 6.2). Il en est de même avec les déviations standards qui sont de l'ordre de 1 m. Il faut noter que les erreurs verticales sont une fois de plus supérieures aux erreurs horizontales, ceci étant dû à la VDOP qui est plus grande que la HDOP. Les résultats obtenus lors de ce test demeurent toutefois très satisfaisants et permettent de démontrer le bon fonctionnement de notre récepteur en environnement réel.

## 6.8 Capture des signaux Galileo (test #7)

### 6.8.1 Méthodologie du test #7

Jusqu'à présent, aucune solution de navigation hybride (GPS et Galileo) n'a pu être calculée. Ceci s'explique par le fait que nous n'avons jamais réussi à capter de message de navigation valide en provenance du satellite GIOVE-A. Lors de tests effectués le 4 octobre 2007 par exemple, nous avons pu décoder les paramètres orbitaux suivants du message de navigation :

- $A = 29\,633$  km (demi-grand axe)
- $e = 7,8903 \times 10^{-4}$  (excentricité)

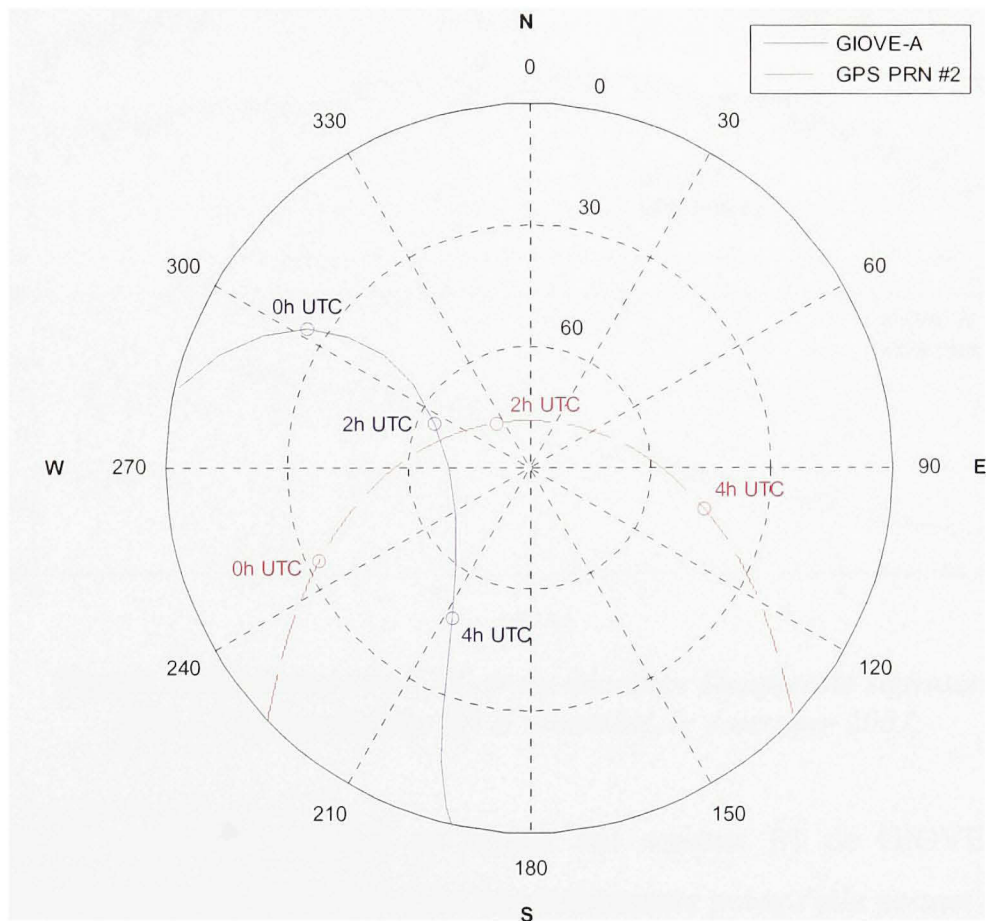
Bien que ces valeurs puissent sembler valides *a priori*, un examen plus détaillé du message de navigation a révélé que tous les drapeaux d'état de santé étaient à 0, ce qui signifie en bref que le message n'est pas valide (ESTEC, 2007). De plus, la valeur retournée pour le TOW était de l'ordre de  $5,43 \times 10^5$  s, ce qui correspond à un samedi. Le *hic*, c'est que les tests ont été effectués un jeudi.

Nous avons tout de même été en mesure d'exécuter certains tests afin de démontrer le bon fonctionnement de nos canaux de démodulation Galileo. Nous avons entre autres enregistré le ratio  $C/N_0$  et le décalage Doppler du satellite GIOVE-A lors de son passage. Ceci nous a permis de comparer les mesures obtenues pour le satellite Galileo à celles d'un satellite GPS dont l'orbite était alors similaire. Nous avons aussi pu bénéficier du mécanisme de mesure de l'enveloppe de corrélation afin de vérifier le verrouillage correct de la DLL (i.e. sur le maximum central de corrélation). Les tests ont été réalisés à l'aide du même montage que celui décrit à la section 6.5 (Évaluation du temps de démarrage en environnement réel). Les résultats obtenus sont présentés à la section suivante.



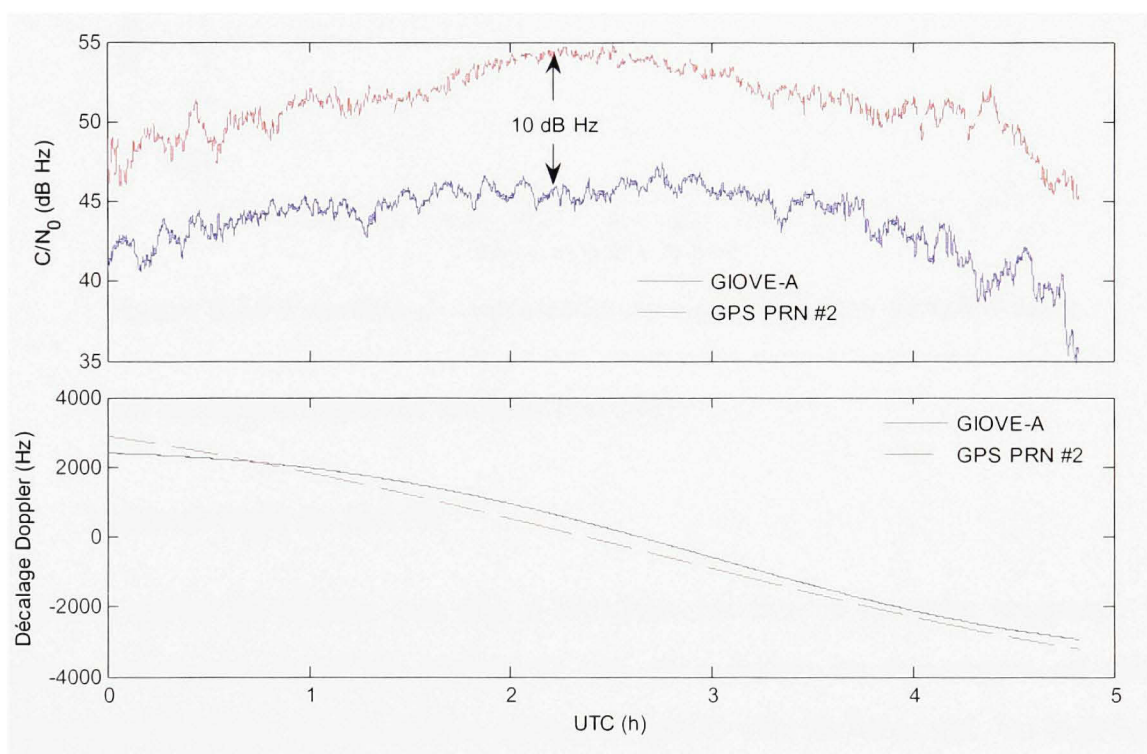
### 6.8.2 Résultats et analyse du test #7

Les tests ont été exécutés le 4 octobre 2007 de 0 h à 5 h UTC. Le satellite GPS PRN #2 a été choisi pour les comparaisons au satellite GIOVE-A puisqu'il présentait une orbite très similaire à ce dernier lorsqu'observé du sol à Montréal. Les trajectoires des deux satellites sont présentées à la Figure 6.22 sous forme de graphique en azimut et élévation. La trajectoire du satellite GPS PRN #2 a été calculée par notre récepteur alors que celle de GIOVE-A provient du site web CalSKY ([www.calsky.com](http://www.calsky.com)). Le  $C/N_0$  et le décalage Doppler des deux satellites sont quant à eux présentés à la Figure 6.23.



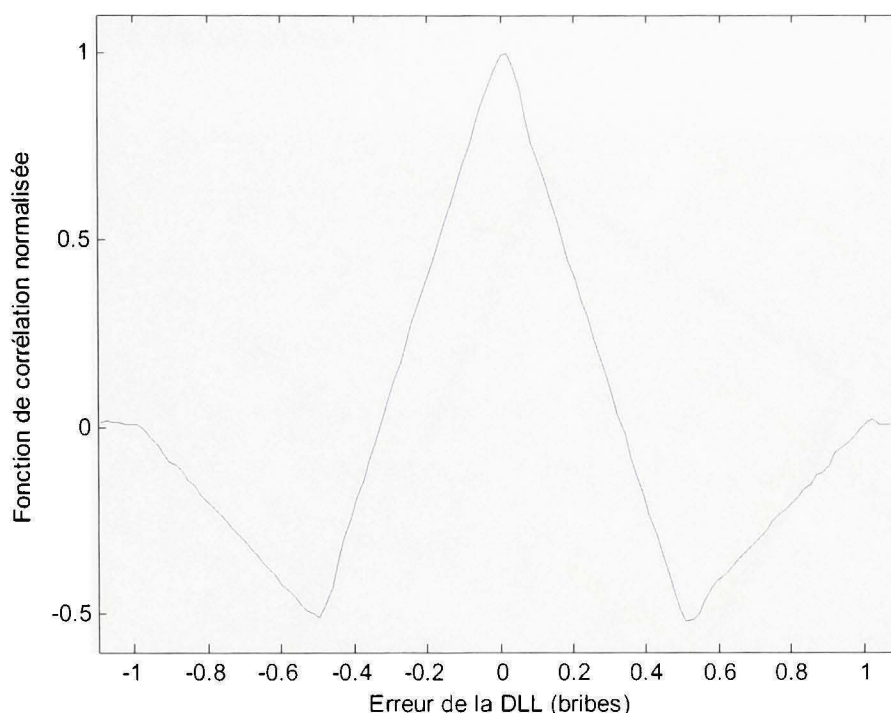
**Figure 6.22 Trajectoires des satellites GIOVE-A et GPS PRN #2 telles qu'observées à Montréal, le 4 octobre 2007.**

En examinant la Figure 6.23, on note un décalage maximal d'environ 10 dB Hz entre les  $C/N_0$  des deux satellites. Ce décalage semble être une particularité de GIOVE-A, dont la puissance d'émission serait réduite (il ne faut pas oublier que GIOVE-A est avant tout un satellite de test). On remarque ensuite des décalages Doppler très similaires au niveau des deux satellites, signe que les orbites sont relativement semblables. Chose encore plus intéressante, la courbe de décalage Doppler de GIOVE-A est lisse et continue, ce qui témoigne du bon fonctionnement des boucles à verrouillage de phase (et de code) des canaux Galileo.



**Figure 6.23** *Mesures du  $C/N_0$  et du décalage Doppler de signaux GPS L1 C/A et Galileo E1 à Montréal, le 4 octobre 2007.*

La fonction de corrélation mesurée au niveau des signaux E1 de GIOVE-A est aussi présentée à la Figure 6.24. Cette figure est très intéressante puisqu'elle permet de démontrer tout d'abord que le mécanisme de mesure de l'enveloppe de corrélation est fonctionnel. De plus, un examen détaillé permet de constater que la DLL est bien verrouillée sur le maximum central de corrélation. Par conséquent, la technique mise en place afin d'éliminer l'ambiguïté de corrélation des signaux BOC opère tel que prévu (ceci avait d'ailleurs été validé à plusieurs reprises lors de tests antécédents).



**Figure 6.24** *Fonction de corrélation du signal E1 reçu de GIOVE-A.*

## 6.9 Test en environnement difficile (test #8)

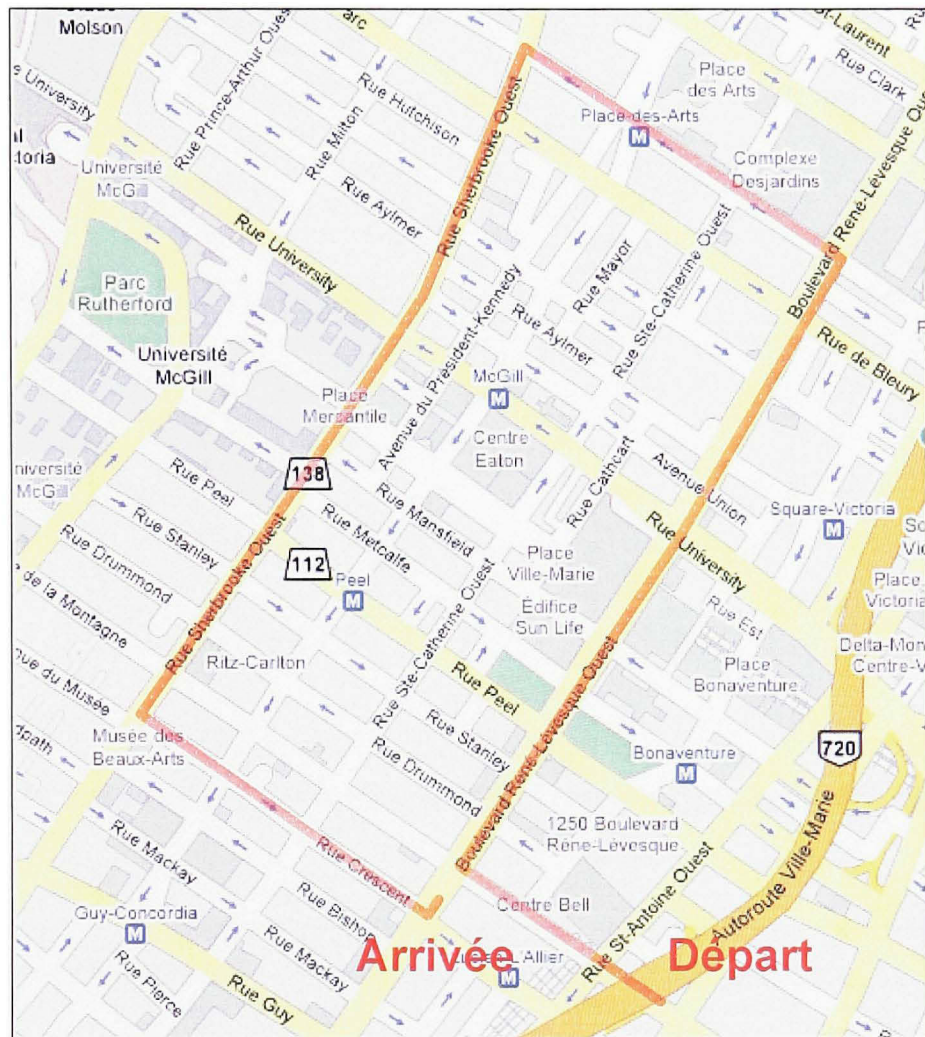
### 6.9.1 Méthodologie du test #8

Nous avons réalisé un dernier test afin d'identifier les limites de notre récepteur. Pour y parvenir, nous avons choisi de réaliser le test en question en dynamique au sein d'un environnement difficile, soit en canyon urbain au centre-ville de Montréal. Vu la présence de grands édifices et d'arbres en bordure du boulevard René-Lévesque Ouest et de la rue Sherbrooke, ces deux routes ont tout d'abord été retenues pour l'élaboration du trajet. Nous avons ensuite choisi les rues Jeanne-Mance et Crescent afin d'établir un quadrilatère de bonne dimension. Le trajet final est présenté à la Figure 6.25.

Le même véhicule ainsi qu'un montage identique à la section 6.7 (Test dynamique en environnement réel) ont été employés pour ce test, à l'exception du récepteur de Novatel qui n'a pas été utilisé. Mentionnons qu'une bande de bruit de 1 Hz a été choisie pour la DLL (sans aide de la PLL) afin de garantir un maximum de robustesse. La fréquence de mise à



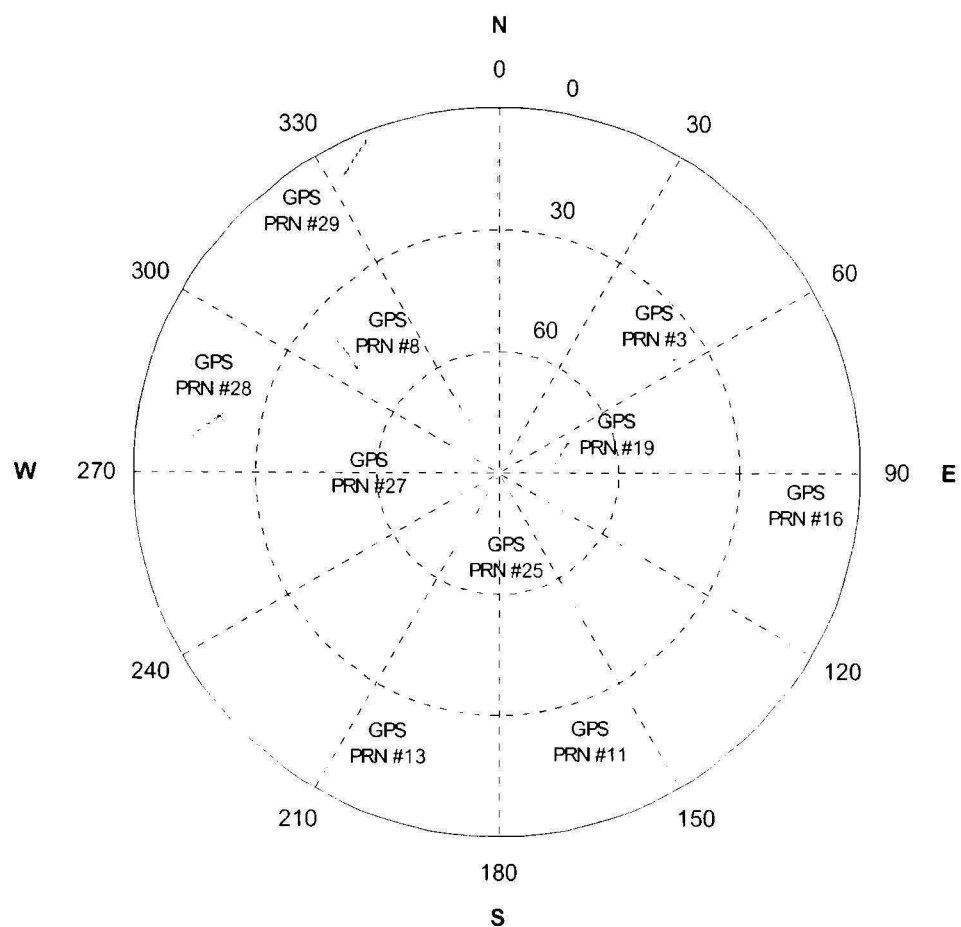
jour de solution de navigation a quant à elle été ajustée à 20 Hz. Les résultats obtenus lors du test sont présentés à la section suivante.



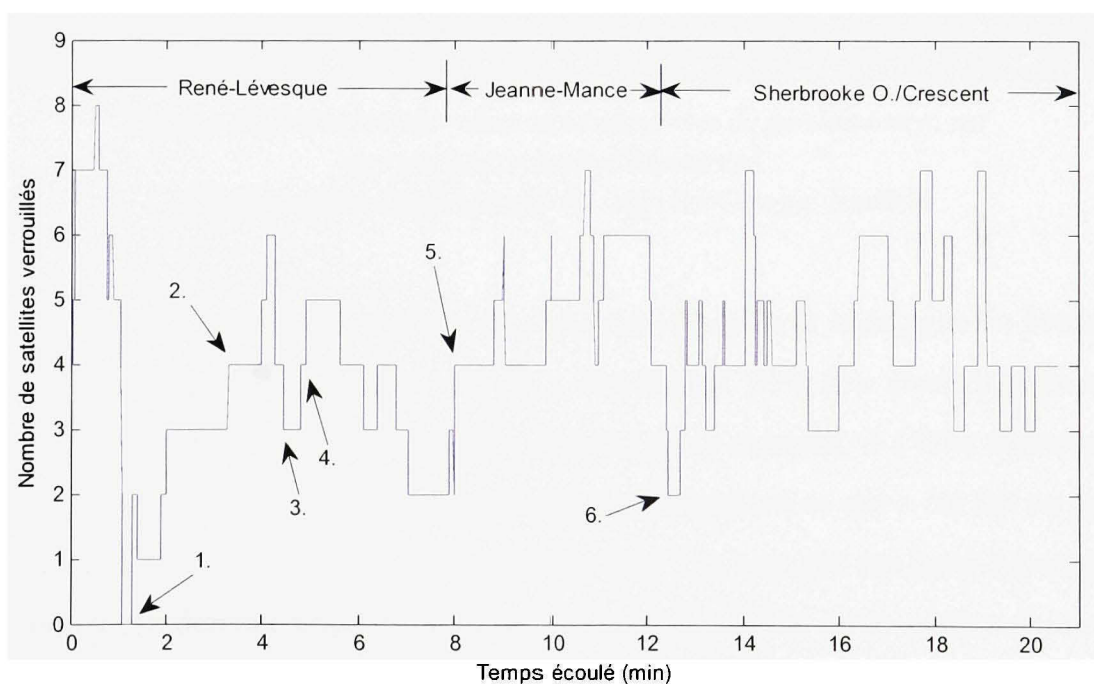
**Figure 6.25 Trajectoire de test en environnement difficile.**  
Carte tirée de Google Maps ([www.google.com/maps](http://www.google.com/maps))

### 6.9.2 Résultats et analyse du test #8

Le test en environnement difficile a été réalisé le mercredi 3 octobre 2007 entre 14h30 et 14h50, heure locale (soit de 18h30 à 18h50 UTC). Le trajet a donc été exécuté en près de 20 minutes au milieu d'une circulation dense et de constructions routières. Les satellites visibles durant la période de test sont présentés à la Figure 6.26, tandis que le nombre de satellites verrouillés par le récepteur est présenté à la Figure 6.27. Les résultats de positionnement obtenus à l'aide du récepteur sont finalement illustrés à la Figure 6.28 et à la Figure 6.29.

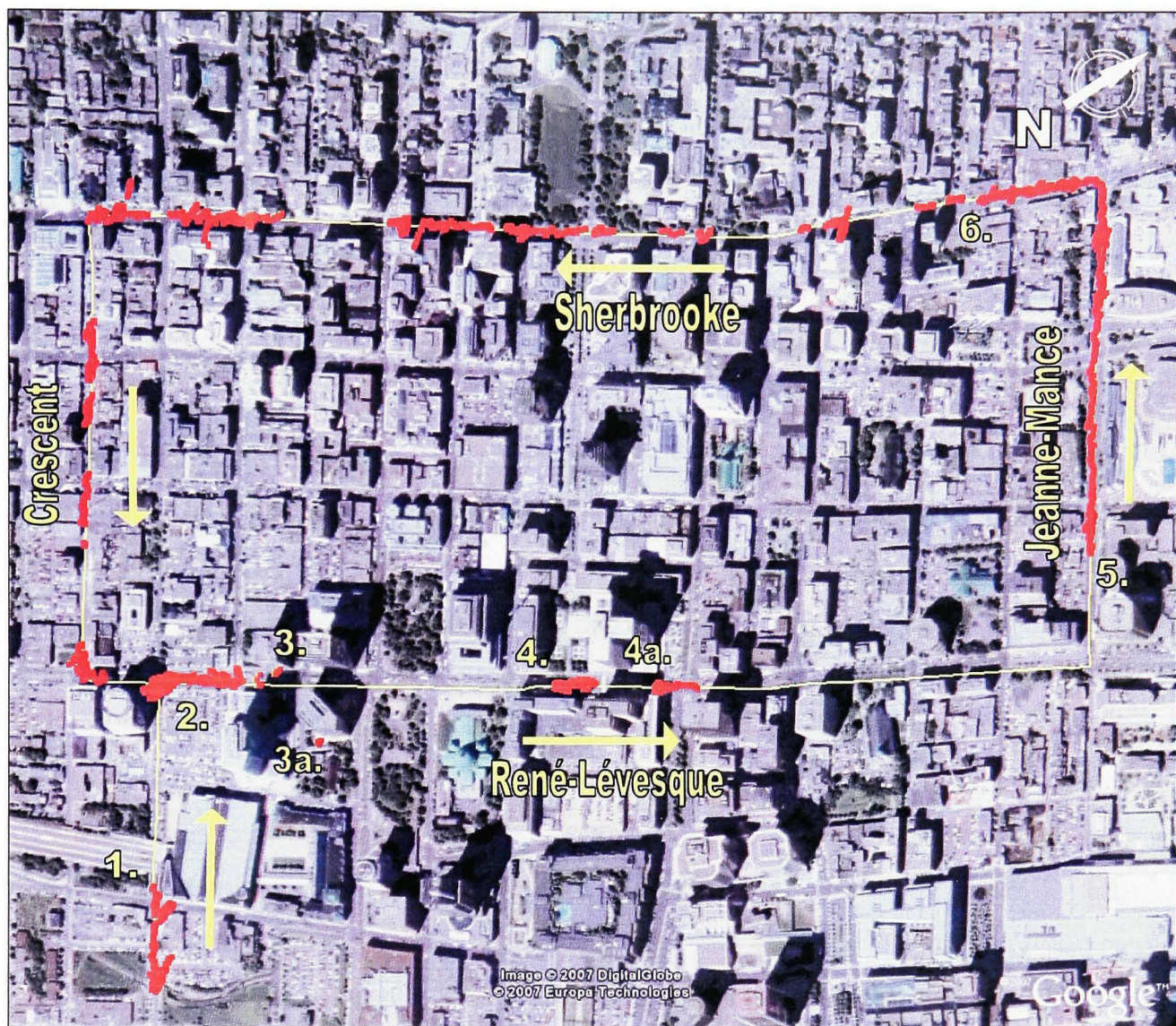


**Figure 6.26** *Satellites visibles à Montréal, le 3 octobre 2007, entre 14h30 et 14h50.*



**Figure 6.27** *Nombre de satellites verrouillés en fonction du temps écoulé.*





**Figure 6.28** *Trajectoire de test et résultats de positionnement en canyon urbain à Montréal.*

Photographie satellite tirée du logiciel Google Earth®

Tel qu'illustré à la Figure 6.28, le test a démarré sur la rue de la Montagne en bordure d'un parc de stationnement. Huit satellites ont pu y être acquis, pour être aussitôt perdus lors du passage sous un large viaduc ferroviaire (1). À partir de ce moment, il a fallu attendre près de 2 min 30 s avant d'obtenir une nouvelle solution de navigation, ce qui a été fait en patientant au coin de René-Lévesque et de la Montagne (2). Une fois engagés sur René-Lévesque, nous nous sommes rapidement engouffrés dans un canyon urbain pour perdre à nouveau la solution de navigation (3). Il faut noter la présence de données aberrantes à ce point (3a), alors que la solution de navigation s'est écartée d'environ 80 m de la trajectoire de référence.

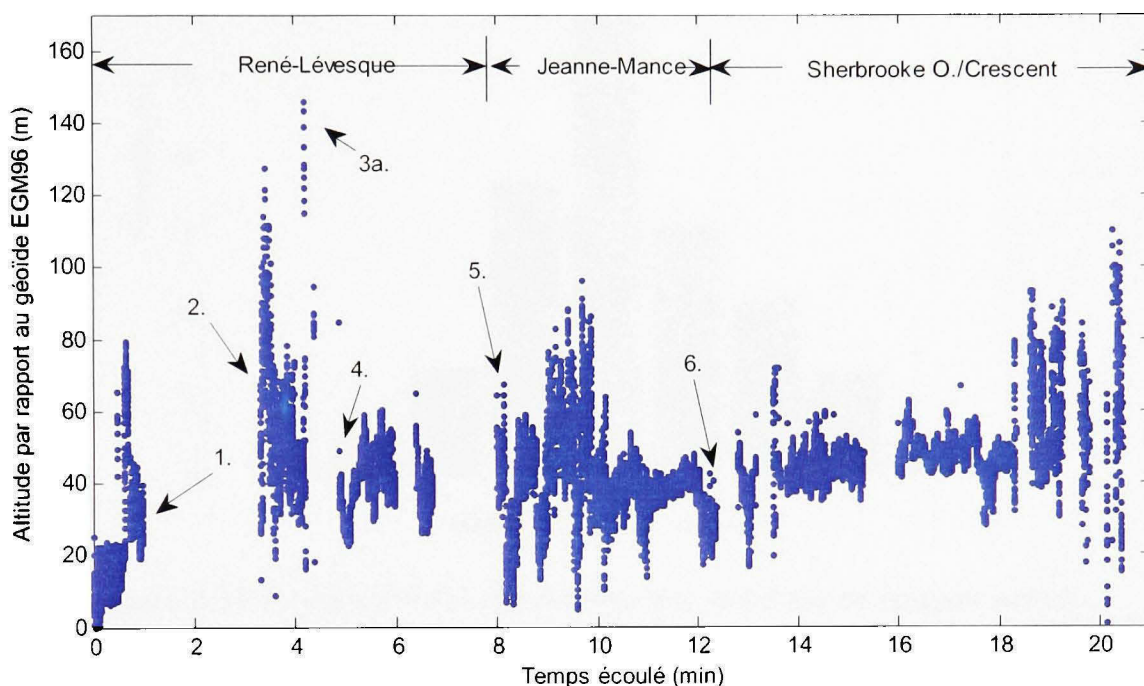


Le temps de déplacement entre les points (3) et (4) s'est avéré très court (environ 20 s), ce qui explique l'absence de solution à cet endroit. Nous avons cependant été coincés dans un bouchon de circulation au point (4), ce qui a laissé le temps au récepteur de réacquérir les satellites perdus. Le passage à proximité de la Place Ville-Marie (4a), un des édifices les plus hauts de Montréal, a toutefois fait retomber le nombre de satellites visibles à trois (soit les satellites GPS PRN #19, #25 et #27, les seuls situés au-dessus de 60° d'élévation - voir Figure 6.26), faisant perdre du coup la solution de navigation.

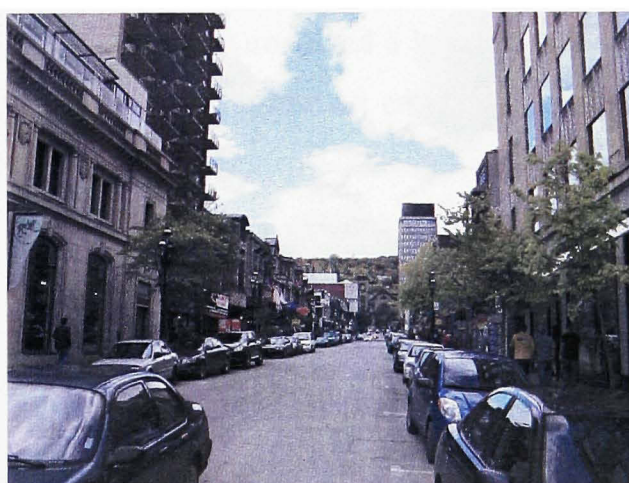
La suite du trajet sur René-Lévesque s'est sans doute avérée la pire partie de tout le test. En plus des édifices à proximité, des arbres étaient présents au milieu du boulevard, causant la perte de tous les satellites sauf deux. La situation s'est toutefois améliorée sur Jeanne-Mance (5), alors qu'une solution de navigation ininterrompue a pu être obtenue jusqu'à Sherbrooke (6). Le nombre de satellites visibles depuis Jeanne-Mance se situait alors entre quatre et sept et s'explique par la forte distribution des satellites dans l'axe nord-ouest/sud-est (ou azimut 315°/135° à la Figure 6.26), soit l'axe sur lequel la rue Jeanne-Mance est située. On observe finalement une solution de navigation par intermittence au niveau des rues Sherbrooke et Crescent, alors que le nombre de satellites oscillait entre trois et sept. Malgré le fait que Crescent soit sur le même axe que Jeanne-Mance, on remarque que la solution de navigation y est beaucoup moins bonne. Ceci est principalement dû à la proximité et à la hauteur des édifices sur Crescent, résultant en une plus grande obstruction du ciel (voir Figure 6.30).

Hormis l'erreur détectée au point (3a), la solution de navigation est demeurée relativement précise tout au long du trajet, avec des erreurs de l'ordre de 5 à 10 m horizontalement. On ne peut toutefois pas en dire autant au sujet du positionnement vertical. En examinant la Figure 6.29, on constate que la précision varie entre 20 et 100 m verticalement tout au long du trajet. Or, ceci est normal puisque les satellites près de l'horizon étaient généralement absents lors du test, résultant en une très mauvaise géométrie (représentée par une VDOP moyenne de 54, soit 20 fois celle mesurée en ciel dégagé à la section 6.6 - Test statique en environnement réel). Il y aurait toutefois possibilité d'améliorer cette précision en ayant recours au WAAS

par exemple. L'utilisation du WAAS demeure néanmoins difficile en hautes latitudes, tel qu'à Montréal par exemple, où l'élévation des satellites ne dépasse pas  $30^\circ$ .

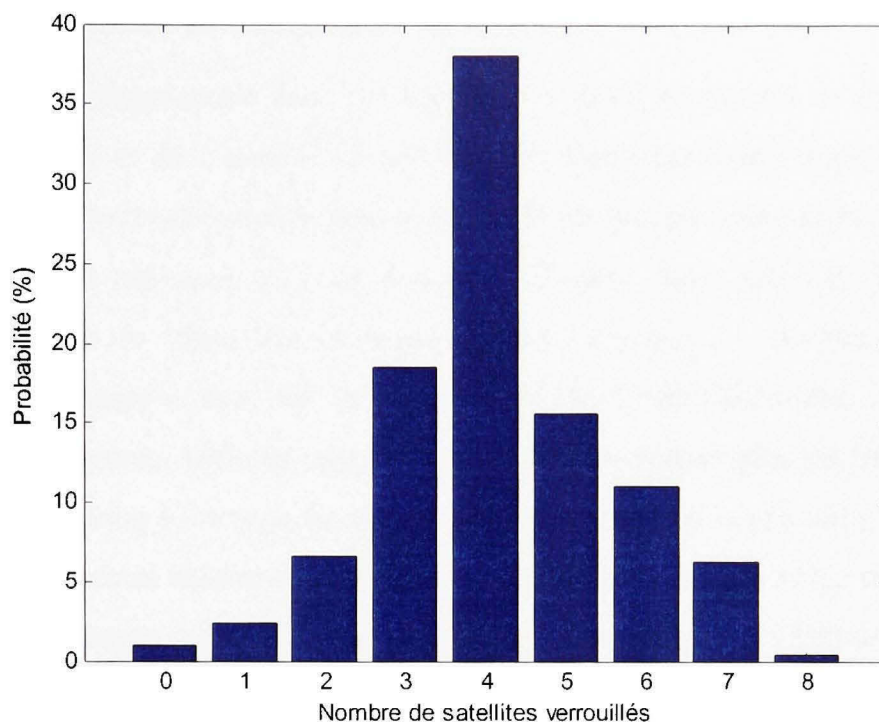


**Figure 6.29** Résultats de positionnement vertical en canyon urbain à Montréal.



**Figure 6.30** Photographies en direction nord des rues Jeanne-Mance (gauche) et Crescent (droite) à la hauteur de la rue Ste-Catherine.





**Figure 6.31** *Probabilité de verrouillage des satellites en canyon urbain.*

Pour terminer, examinons la Figure 6.31 qui fait une synthèse du nombre de satellites visibles lors de ce test. Tout d'abord, on y remarque que le récepteur a le plus souvent verrouillé sur quatre satellites à la fois, avec une probabilité de près de 40%. Par ailleurs, on obtient 71% de probabilités de verrouiller sur quatre satellites ou plus, ce qui correspond à la proportion du temps où nous avons obtenu une solution de navigation (cette valeur est en réalité de 61%, l'écart s'expliquant par le délai de convergence de la solution de navigation, qui est d'environ dix itérations, soit 0,5 s à une fréquence de 20 Hz). On obtient toutefois 90% en incluant les probabilités de verrouiller sur trois satellites, puis 97% en allant jusqu'à deux satellites. Il est donc clair qu'en développant des algorithmes de positionnement à deux et trois satellites, tels que ceux décrits au chapitre 2 (et en réduisant le temps de convergence de la solution de navigation), on améliorerait davantage les performances du récepteur en milieu difficile. Une augmentation de la sensibilité du récepteur serait aussi un atout et permettrait de capter un plus grand nombre de satellites en tout temps.



## 6.10 Conclusion et améliorations

Plusieurs tests ont été présentés dans ce chapitre. Ces tests ont permis de bien caractériser le récepteur développé et de s'assurer de son bon fonctionnement en temps réel. Nous avons entre autres vu que le mécanisme de mesure du SNR développé opère de façon très linéaire et garantie une bonne précision ( $< 1$  dB d'erreur). D'autres tests réalisés en simulation ont quant à eux permis de démontrer le fonctionnement adéquat du récepteur, que ce soit en statique ou en dynamique, avec une précision en 3D de l'ordre du mètre. Une série de tests menés en environnement réel ont ensuite permis de caractériser plus en détail le récepteur, que ce soit en mesurant les temps de démarrage ou en vérifiant la précision de la solution de navigation dans diverses situations et environnements. De façon générale, les résultats se sont avérés très satisfaisants pour un premier prototype, avec des performances similaires à d'autres récepteurs GPS L1 disponibles sur le marché.

Les tests menés dans ce chapitre ont aussi permis de détecter certains problèmes au niveau du récepteur. Tout d'abord, les tests en dynamique ont permis de constater que le mécanisme d'aide aux DLL ne fonctionne pas comme prévu. Des corrections devraient donc être apportées à ce niveau afin de garantir un positionnement robuste et précis. Ensuite, des tests de longue durée ont permis de détecter des sauts abrupts lors de l'ajout de nouveaux satellites dans la solution de navigation. Une révision adéquate de la solution de navigation s'impose donc afin d'éviter que ce problème ne se reproduise.

Parmi les autres améliorations pouvant être apportées au récepteur, nous avons vu que l'ajout d'une solution de navigation à deux et trois satellites serait fort avantageux. Or, une telle approche permettrait d'augmenter la probabilité d'obtenir un positionnement en environnement difficile, tel qu'en ville ou en forêt par exemple. Bien que le sujet n'ait pas été abordé ici, une miniaturisation du récepteur serait aussi très appréciable. Vu la taille très encombrante et l'énorme consommation électrique de ce dernier (environ 200 W selon les estimations du chapitre 3), ceci permettrait entre autres de faciliter les tests sur le terrain. Enfin, l'ajout de nouveaux canaux GNSS, que ce soit des canaux WAAS, GPS L2C ou L5,

ou Galileo E5 par exemple, devra être envisagé dans un futur rapproché. Ceci permettrait non seulement d'augmenter la précision du positionnement, mais aussi sa disponibilité, peu importe les environnements dans lesquels on est situé : en ville, en forêt, à l'intérieur des bâtiments, etc.

Pour terminer, nous avons réussi dans ce chapitre à démontrer le fonctionnement des canaux Galileo en temps réel. Vu l'absence de données valides au niveau du satellite GIOVE-A, nous n'avons toutefois pas été en mesure de réaliser une solution de navigation hybride GPS et Galileo. Avec le lancement prochain de nouveaux satellites Galileo, ceci pourra toutefois devenir réalité. Il serait alors intéressant de reprendre les tests afin de mesurer les gains d'une telle solution de navigation hybride.

## CONCLUSION

La mise en œuvre en temps réel d'un récepteur hybride GPS-Galileo s'est avérée être un projet beaucoup plus complexe que nous le pensions. Un simple regard sur l'histoire et sur les enjeux actuels de la radionavigation, sujets abordés au chapitre 1, donne un bon indice sur l'ampleur et les implications d'un tel projet. Toute la complexité n'est cependant comprise qu'une fois le cadre théorique abordé en détail, ce que nous avons fait au chapitre 2. Nous avons alors été surpris de constater que la radionavigation par satellite est une science qui dépend d'une multitude de domaines de recherche, incluant entre autres les télécommunications, la géomatique et la météorologie.

La première étape pour l'élaboration du récepteur hybride en temps réel a été de concevoir un récepteur GPS L1 à 12 canaux. Cette étape, décrite en détail au chapitre 3, a permis tout d'abord de mettre en place une plateforme de développement stable. Vu la disponibilité immédiate du système de positionnement GPS, nous avons ainsi pu valider cette plateforme avant de procéder au développement de nouveaux canaux GNSS. Il faut mentionner notre approche novatrice basée sur un FPGA, qui est flexible et peu coûteuse, contrairement aux méthodes actuellement employées en industrie. Notre approche permet ainsi de concevoir des récepteurs de navigation compacts et performants tout en limitant le temps et les coûts de recherche et de développement au minimum.

La seconde étape de notre projet a consisté en la mise en place de canaux de démodulation Galileo E1, présentée en détail au chapitre 4. Il faut noter que grâce à la plateforme de développement, cette étape a pu être réalisée en un temps record d'environ deux semaines, comparativement à un an pour le récepteur GPS L1. Toutefois, force a été de constater l'absence de satellites Galileo, dont un seul a été lancé jusqu'à présent. Les canaux de démodulation Galileo ont tout de même pu être validés grâce à ce satellite, en faisant preuve de patience entre chacune des périodes de passage de ce dernier.



L'élaboration d'une solution de navigation en temps réel a été un autre élément important de ce projet. Cette solution, calculée de manière très efficace sur un ordinateur de type PC, a été abordée au chapitre 5. Une série de bancs de tests, présentés au chapitre 6, ont par la suite été élaborés afin de valider cette solution de façon détaillée. Ces tests ont permis du même coup de déterminer la précision ainsi que la performance du récepteur que nous avons développé. Nous avons été heureux de constater la grande précision de notre récepteur ainsi que son temps de démarrage rapide, comparables à d'autres récepteurs disponibles sur le marché. Nous avons aussi été surpris de constater le fonctionnement relativement fiable du récepteur en milieu difficile lors de tests effectués au centre-ville de Montréal.

Pour conclure, rappelons que notre projet fait partie d'une démarche entreprise par le GRN afin d'explorer les avenues de la navigation par satellite GNSS. Notre projet, précédé de la réalisation d'une chaîne de simulation GPS et Galileo, est donc le premier de son genre à opérer en temps réel. Or, le récepteur que nous avons développé s'avère déjà un outil de recherche formidable, pouvant opérer en environnement réel et être configuré à souhait. La suite du projet s'annonce donc très intéressante avec l'ajout de nouveaux canaux GNSS, tels que WAAS, GPS L2C, GPS L5 et Galileo E5. Une précision, une fiabilité et une robustesse sans précédent sont alors à prévoir, et on ne sait toujours pas jusqu'où tout cela nous mènera. Le futur s'annonce ainsi très prometteur en ce qui concerne la radionavigation par satellite et notre récepteur constitue une base solide afin d'en explorer les avenues.

## RECOMMANDATIONS

Plusieurs recommandations ont été formulées tout au long de ce document. On résume ici les plus importantes.

Débutons par les recommandations formulées au chapitre 3. Nous y avons entre autres mentionné qu'un échantillonnage à plus haute fréquence serait souhaitable en IF. Ceci permettrait de démoduler les nouveaux signaux GNSS à large bande, tels que Galileo E5 par exemple. Vu la longueur des codes d'étalement de ces nouveaux signaux, l'ajout d'un mécanisme d'acquisition par FFT serait aussi très important. Sans ce mécanisme d'acquisition, il serait alors impossible d'acquérir ces nouveaux signaux dans des délais raisonnables (on parle de plus de dix minutes par acquisition séquentielle). Un autre point abordé concerne la miniaturisation du récepteur, ce qui permettrait de faciliter grandement la manipulation de ce dernier, en plus d'en augmenter l'autonomie lors de tests à longue durée à l'extérieur.

Parmi les autres recommandations formulées, on suggère au chapitre 4 une augmentation du temps d'intégration lors de l'acquisition et de la poursuite des signaux GNSS. Ceci permettrait entre autres d'abaisser le ratio  $C/N_0$  minimal auquel le récepteur peut opérer, rendant du coup possible le fonctionnement à l'intérieur des bâtiments, par exemple. On propose ensuite au chapitre 5 d'employer une matrice de covariance de bruit adaptative au niveau de la solution de navigation, ceci dans le but d'améliorer la précision du récepteur. Cette proposition a été appuyée de résultats au chapitre 6, démontrant qu'il est préférable d'exclure les satellites près de l'horizon afin d'augmenter la précision du positionnement.

Pour conclure, plusieurs recommandations importantes ont été formulées au chapitre 6. Une de ces recommandations touche directement la solution de navigation. On y souligne le fait que le filtre de Kalman a été développé rapidement et qu'une révision majeure accompagnée de tests rigoureux s'impose. Ensuite, on mentionne qu'aucune solution de navigation hybride n'a pu être calculée jusqu'à présent, ceci étant dû à l'absence de données valides au niveau

du message de navigation du satellite GIOVE-A. Une recommandation est donc faite en ce sens afin de poursuivre les travaux lors du lancement des prochains satellites Galileo. Finalement, la dernière recommandation et non la moindre consiste à poursuivre le développement du récepteur hybride GPS et Galileo en ajoutant de nouveaux canaux de démodulation GNSS. Nous suggérons fortement de débiter par l'ajout de canaux WAAS puisqu'il s'agit d'un système actuellement fonctionnel. Certains modules qui devront être élaborés pour ces canaux, tels qu'un décodeur de Viterbi par exemple, pourront ensuite être réutilisés au niveau des autres canaux GNSS. Enfin, tout dépendamment de l'état d'avancement du projet Galileo, nous recommandons de poursuivre avec l'élaboration de canaux GPS L2C et GPS L5, signaux dès maintenant disponibles. Des canaux Galileo E5 pourraient éventuellement être greffés au récepteur et des systèmes, comme GLONASS et Beidou, pourraient être considérés dans un futur rapproché.



## ANNEXE I

### RÉSUMÉ DES CARACTÉRISTIQUES DU RÉCEPTEUR HYBRIDE

#### Nombre de canaux

GPS L1 C/A	12
Galileo E1	12

#### Caractéristiques des canaux

##### PLL GPS L1 C/A

Période d'intégration	1 ms
Ordre de la boucle	3
Bande de bruit	10 Hz
Discriminateur	atan(Q/I)

##### DLL GPS L1 C/A (poursuite initiale)

Période d'intégration	10 ms
Ordre de la boucle	2
Bande de bruit	2 Hz
Discriminateur	Envpe. AMRN
Espcement. corrélateurs	$\pm 1/2$ bribe

##### DLL GPS L1 C/A (poursuite avancée)

Période d'intégration	10 ms
Ordre de la boucle	2
Bande de bruit	0,1 Hz
Discriminateur	Envpe. AMRN
Espcement. corrélateurs	$\pm 1/20$ bribe

##### PLL Galileo E1

Période d'intégration	1 ms
Ordre de la boucle	3
Bande de bruit	10 Hz
Discriminateur	atan(Q/I)

##### DLL Galileo E1 (poursuite initiale)

Période d'intégration	10 ms
Ordre de la boucle	2
Bande de bruit	2 Hz
Discriminateur	Envpe. AMRN
Espcement. corrélateurs	$\pm 1/4$ bribe

#### DLL Galileo E1 (poursuite avancée)

Période d'intégration	10 ms
Ordre de la boucle	2
Bande de bruit	0,1 Hz
Discriminateur	Envpe. AMRN
Espcement. corrélateurs	$\pm 1/20$ bribe

#### NCO

Accumulateur (PLL)	33 bits
Accumulateur (DLL)	47 bits
Tables sin/cos	$16 \times 3$ bits

#### Signal IF

Résolution	4 bits
Fréq. échantillonnage	60 MHz
Fréq. nominale	15 MHz
Puissance nominale	-39,5 dBm
Largeur de bande	22,3 MHz

#### Solution de navigation

##### Fréquence

Mesures et mise à jour de la position	1 kHz, 100 Hz, 20 Hz, 10 Hz, 5 Hz, 1 Hz, 1/2 Hz, 1/3 Hz
---------------------------------------	---

##### Temps de démarrage<sup>1</sup>

Programmation FPGA	12 s
À froid	124 s
À tiède	62 s
À chaud	16 s

##### Précision<sup>1,2</sup>

Position horizontale	2,2 m RMS
Position verticale	4,5 m RMS
Vitesse	0,03 m/s RMS

## Caractéristiques physiques et électriques

**Dimensions** 56 × 40 × 32 cm

**Poids** 20 kg

### Alimentation

Tension 120 VAC 60 Hz

Puissance typique 200 W

### Alimentation d'une antenne active

Tension de sortie 5 VDC

Courant maximal 100 mA

<sup>1</sup> Valeurs typiques basées sur les tests menés au chapitre 6.

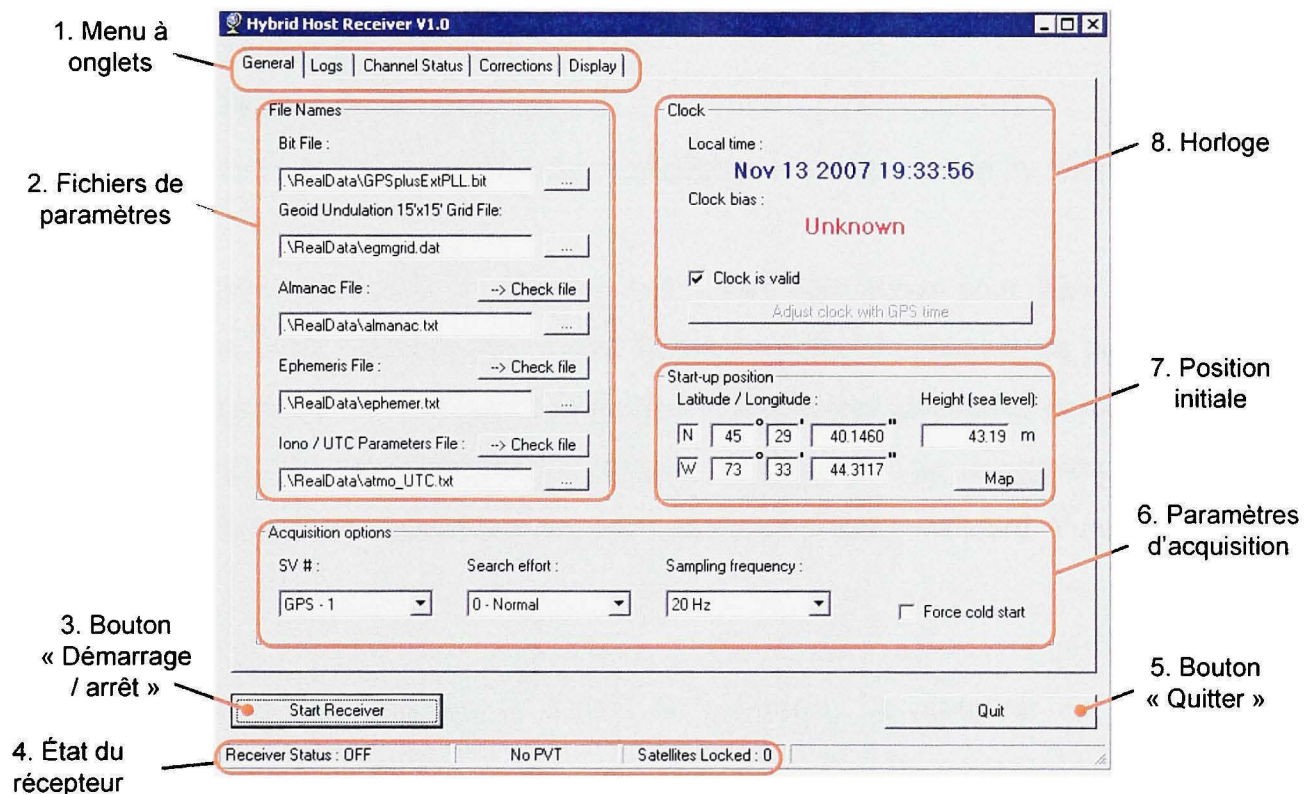
<sup>2</sup> HDOP = 0,9 et VDOP = 2,7

## ANNEXE II

### SURVOL DE L'INTERFACE GRAPHIQUE UTILISATEUR (GUI)

#### A2.1 Fenêtre de contrôle principale

Nous effectuons ici un survol de l'interface graphique utilisateur (GUI) que nous avons développée pour le contrôle du récepteur hybride. Commençons tout d'abord par la fenêtre de contrôle principale, chargée dès le lancement de la GUI (voir Figure A2.1).



**Figure A2.1 Fenêtre de contrôle principale de la GUI.**

Un examen de la Figure A2.1 permet de constater qu'il y a huit éléments principaux au niveau de la fenêtre de contrôle principale. Ces éléments sont les suivants :

1. Un menu à onglet afin de naviguer au travers des fenêtres du récepteur. Ces fenêtres sont :
  - a. Fenêtre de contrôle principale (*General*)



- b. Fenêtre de sauvegarde des données (*Logs*)
- c. Fenêtre d'état des canaux (*Channel Status*)
- d. Fenêtre d'affichage des corrections (*Corrections*)
- e. Fenêtre d'affichage étendu (*Display*)

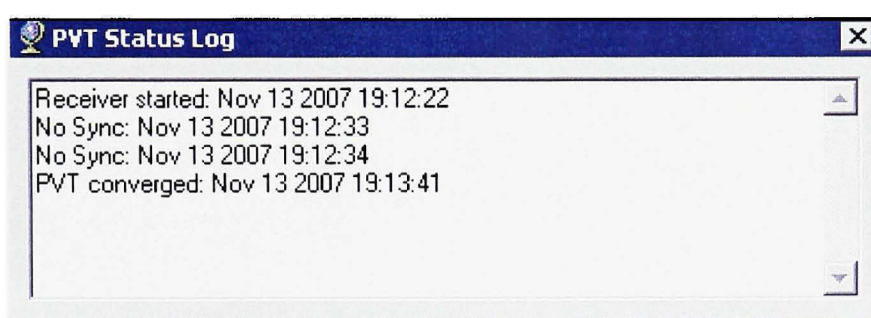
Les fenêtres b à e sont abordées dans les sections qui suivent.

2. Des champs de texte servant à spécifier les cinq fichiers requis par le récepteur, soit :
  - a. Fichier de programmation pour le FPGA (*bit file*)
  - b. Grille d'ondulation du géoïde
  - c. Fichier de l'almanach
  - d. Fichier des éphémérides
  - e. Fichier des paramètres atmosphériques et de conversion du temps GPS à UTC

Il faut noter que seuls les fichiers a et b sont nécessaires pour faire fonctionner le récepteur. Une omission des autres fichiers résulte simplement en un démarrage à tiède ou à froid du récepteur. Les fichiers c, d et e sont ainsi créés et mis à jour par le récepteur grâce aux données téléchargées des satellites de navigation. Mentionnons que le fichier de l'almanach est sauvegardé en format standard Yuma (NAVCEN, 2007).

3. Un bouton de démarrage et d'arrêt du récepteur. Le récepteur est initialement à l'arrêt.
4. Une barre d'information servant à indiquer l'état du récepteur. Les informations retournées par ce champ sont :
  - a. L'état du récepteur : arrêt, acquisition, poursuite ou navigation
  - b. L'état de la solution de navigation (solution valide ou non)
  - c. Le nombre de satellites verrouillés

Un clic sur cette barre permet d'ouvrir un journal très utile qui garde en mémoire les événements critiques qui se sont produits, tels que la convergence ou la divergence de la solution de navigation, ou encore les pertes de synchronisation entre le PC et le FPGA (voir Figure A2.2).

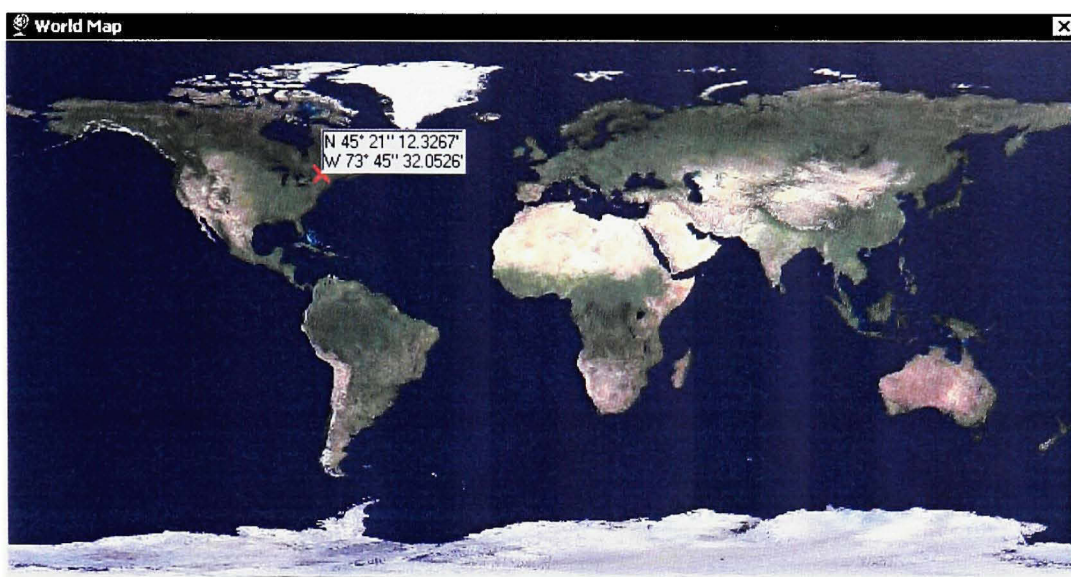


**Figure A2.2 Fenêtre journal de bord.**

5. Un bouton « Quitter » pour fermer la GUI.
6. Une série de listes déroulantes servant à spécifier les paramètres d'acquisition, soit :
  - a. L'effort de recherche des satellites (normal, modéré, haut et maximal, ce dernier niveau servant à chercher un satellite en particulier au détriment des autres)
  - b. La fréquence d'échantillonnage de la solution de navigation (les valeurs possibles sont : 1 kHz, 100 Hz, 20 Hz, 10 Hz, 5 Hz, 1 Hz, 1/2 Hz et 1/3 Hz)

Une case à cocher est aussi présente afin de forcer le démarrage à froid du récepteur (*Force cold start*). Une fois cette case cochée, le récepteur ignorera la position initiale entrée par l'utilisateur, de même que les fichiers almanach et éphémérides.

7. Des champs de texte afin de permettre à l'utilisateur d'entrer la position initiale du récepteur. Un bouton « carte » (*map*) permet aussi à l'utilisateur d'ouvrir une mappemonde afin de positionner facilement le récepteur sur le globe à l'aide de la souris (voir Figure A2.3). Nous conseillons toutefois de cocher la case « *Force cold start* » si la position n'est pas connue de l'utilisateur.



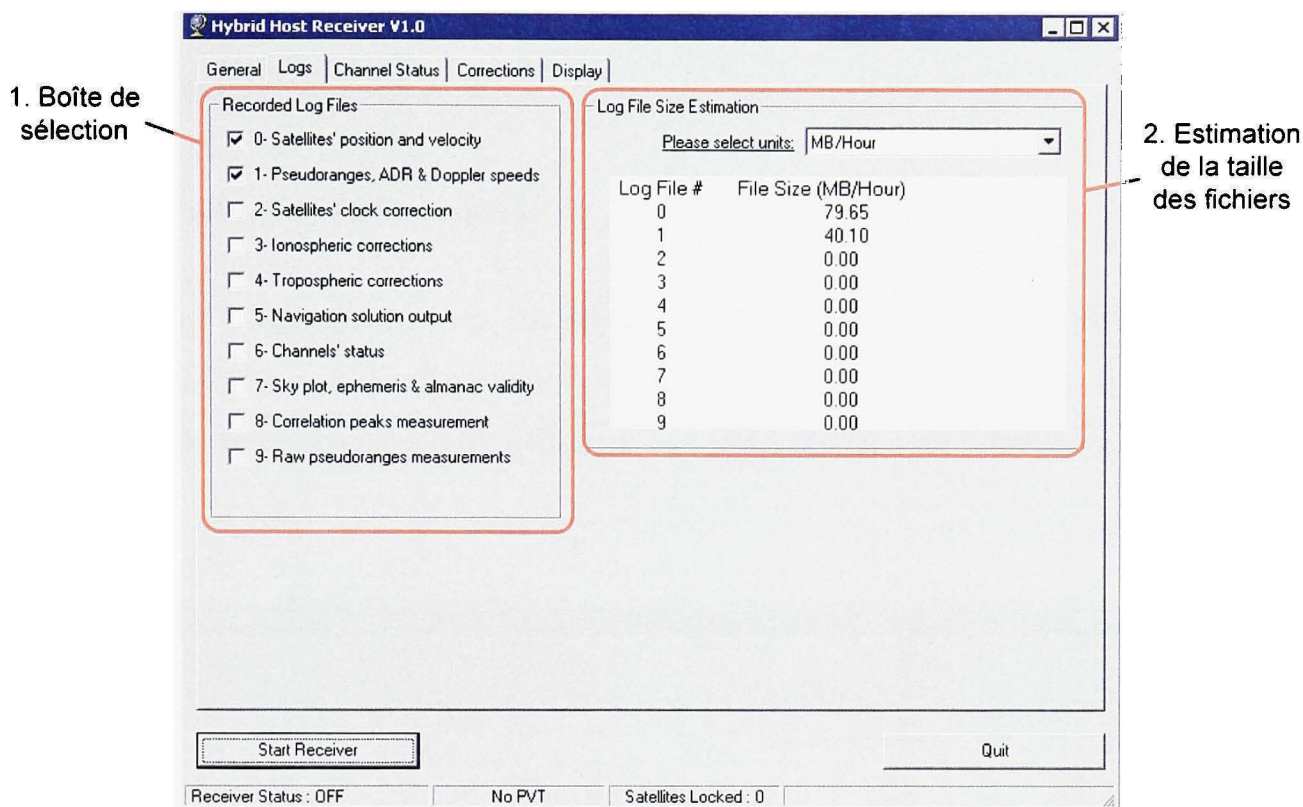
**Figure A2.3** *Fenêtre mappemonde.*

8. Une horloge en temps réel. Le biais entre cette horloge (l'horloge du PC) et le temps UTC y est aussi affiché une fois que la solution de navigation a convergé. Un bouton de synchronisation (*Adjust clock with GPS time*) est d'ailleurs mis à la disposition de l'utilisateur si ce dernier souhaite synchroniser l'horloge du PC au temps UTC. Enfin, une case à cocher située tout près permet d'indiquer au récepteur si l'horloge du PC est valide (*Clock is valid*). Si cette case n'est pas cochée, le récepteur sera forcé d'exécuter un démarrage à froid afin de déterminer le temps GPS local, nécessaire à ses calculs.

## **A2.2 Fenêtre de sauvegarde des données**

La seconde fenêtre, appelée « fenêtre de sauvegarde des données » (*Logs*), permet à l'utilisateur de sélectionner les données à sauvegarder lors de l'exécution du récepteur (voir Figure A2.4). Ceci est effectué par l'entremise d'une boîte de sélection (1). Une estimation de la taille des fichiers produits est immédiatement donnée à la droite de cette boîte (2). Cette estimation est calculée en fonction de la fréquence d'échantillonnage choisie pour la solution de navigation.





**Figure A2.4 Fenêtre de sauvegarde des données.**

Les fichiers de sauvegarde pouvant être sélectionnés par l'utilisateur sont les suivants :

0. Position et vitesse des satellites : valeurs calculées à partir des éphémérides
1. Pseudo-distances, ADR et vitesses relatives : valeurs mesurées par le récepteur
2. Corrections d'horloge des satellites : valeurs calculées à partir de éphémérides
3. Corrections ionosphériques : valeurs calculées à partir des paramètres ionosphériques transmis par les satellites GPS
4. Corrections troposphériques : valeurs calculées à partir du modèle troposphérique d'EGNOS (voir chapitre 2)
5. Solution de navigation : position et vitesse du récepteur sous forme de coordonnées ECEF WGS-84 ainsi que le biais et la dérive d'horloge du récepteur
6. État des canaux et du récepteur en général (voir l'annexe 9 pour plus de détails)
7. Élévation et azimut des satellites GPS
8. Mesures des fonctions de cross-corrélation des canaux
9. Mesures de pseudo-distances brutes (à un taux indépendant de 1 kHz)

Ces dix fichiers sont sauvegardés sous format binaire, ceci dans le but d'en minimiser la taille. Une description détaillée de ces fichiers peut être retrouvée à l'annexe 9.

### A2.3 Fenêtre d'état des canaux

Sous l'onglet « *Channel Status* », on retrouve la fenêtre d'état des canaux. Cette fenêtre permet de visualiser en un clin d'œil l'état de chacun des 24 canaux GPS et Galileo (voir Figure A2.5). Les données y sont affichées en temps réel et sont rafraîchies deux fois par secondes.

Channel #	SV #	Level	Failure	Week	Time of Week	Bad SV	PLL Locked	Apparent Doppler (Hz)	Bandwidth (MHz)	C/N0 (dB Hz)
GPS - 0	01	4	0	406	158816.335313		X	248	24	44.7
GPS - 1	02	4	0	406	158816.338329		X	-763	24	45.9
GPS - 2	20	0	0	---	---			922	24	---
GPS - 3	22	0	0	---	---			-2578	24	---
GPS - 4	23	4	0	---	---		X	-1214	24	47.5
GPS - 5	21	0	0	---	---			922	24	---
GPS - 6	24	0	0	---	---			-2578	24	---
GPS - 7	08	4	0	406	158816.336143		X	721	24	44.7
GPS - 8	25	0	0	---	---			-2578	24	---
GPS - 9	26	0	0	---	---			-2578	24	---
GPS - 10	11	4	0	406	158816.336947		X	-1803	24	44.9
GPS - 11	12	4	0	406	158816.329860		X	-3379	24	43.1
GAL - 0	01	0	0	---	---			-4000	24	---
GAL - 1	01	0	0	---	---			-3000	24	---
GAL - 2	01	0	0	---	---			-2000	24	---
GAL - 3	01	0	0	---	---			-1000	24	---
GAL - 4	01	0	0	---	---			0	24	---
GAL - 5	01	0	0	---	---			1000	24	---
GAL - 6	01	0	0	---	---			2000	24	---
GAL - 7	01	0	0	---	---			3000	24	---
GAL - 8	01	0	0	---	---			1500	24	---
GAL - 9	01	0	0	---	---			5000	24	---
GAL - 10	01	0	0	---	---			6000	24	---
GAL - 11	01	0	0	---	---			7000	24	---

Stop Receiver      Quit

Receiver Status : Nav. Ready      PVT Stable      Satellites Locked : 6

**Figure A2.5 Fenêtre d'état des canaux.**

Les données affichées dans la fenêtre d'état des canaux sont les suivantes :

1. Numéro de canal (*Channel #*)
2. Numéro de PRN (*SV #*)
3. Niveau d'acquisition des canaux (*Level* - voir section A2.2, point 6f)
4. Nombre d'échecs à l'acquisition (*Failure*)
5. Semaine GPS (*Week*)
6. Temps GPS (*Time of Week* - en secondes)
7. Drapeaux « satellite défectueux » (*Bad SV*)
8. Drapeaux « PLL verrouillée » (*PLL Locked*)
9. Doppler mesuré (*Apparent Doppler* – en Hz)
10. Largeur de bande (*Bandwidth* - champs non utilisé, fixé à 24 MHz)
11.  $C/N_0$  (en dB Hz)

Un clic sur une des données affichées permet d'ouvrir une fenêtre d'information détaillée, telle qu'illustrée à la Figure A2.6. Dans ce cas-ci, nous avons cliqué sur une des données du canal #2. Une panoplie d'information est alors affichée au sujet de ce canal, à savoir le numéro PRN du satellite poursuivi, son élévation et son azimut (*Elevation* et *Azimuth*) ainsi que les décalages Doppler réel et mesuré (*Doppler* et *Apparent Doppler*). On y indique aussi la validité des éphémérides et de l'almanach pour le satellite en question (*Ephemeris valid* et *Almanac valid*) ainsi que le type de démarrage du canal (*Start Type*). Enfin, un premier bouton, identifié « *Correlation Peak* », permet de visualiser en temps réel la fonction de cross-corrélation du satellite poursuivi (voir Figure A2.7). Un second bouton, identifié « *Log Nav. Message* », permet de sauvegarder le message de navigation du satellite dans un fichier de données brutes. Le témoin « RX » sert quant à lui à indiquer les instants où le message de navigation est enregistré dans ce fichier.



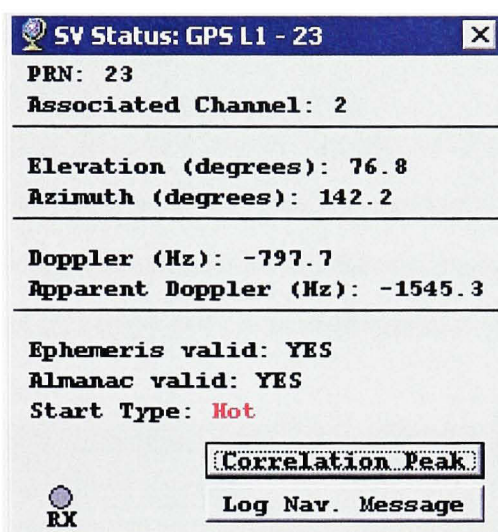


Figure A2.6 Fenêtre d'information détaillée.

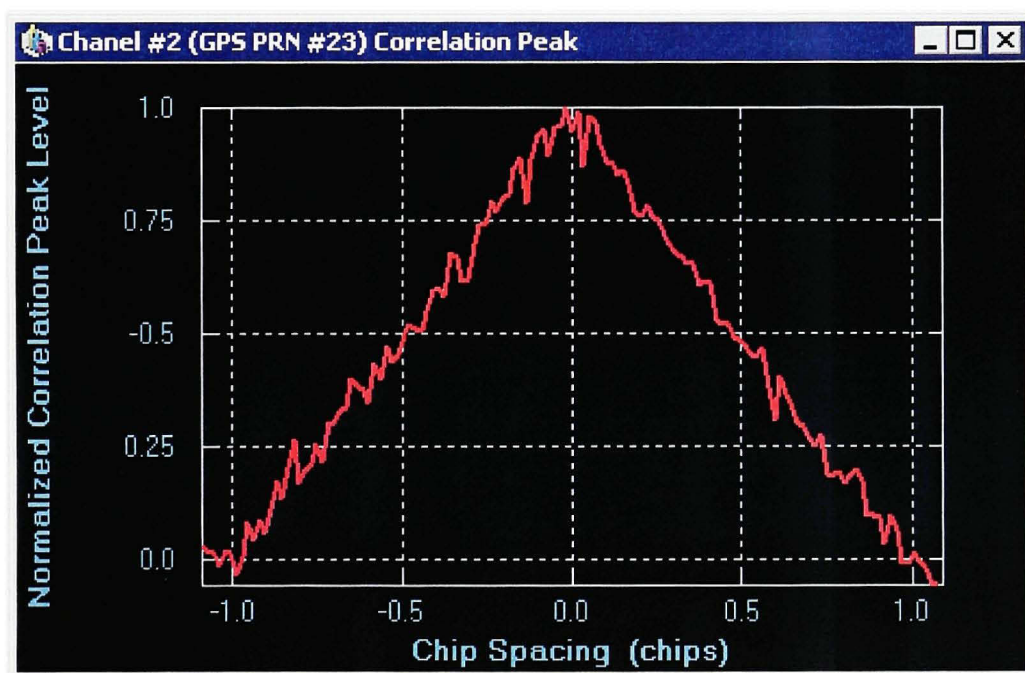


Figure A2.7 Affichage en temps réel de la fonction de cross-corrélation.

Notons qu'un clic droit sur la fenêtre ci-dessus (Figure A2.7) permet d'accéder à un menu supplémentaire. Ce menu permet entre autres de changer le niveau de zoom de la figure, de figer le graphique dans le temps ou encore d'en modifier les couleurs.

## A2.4 Fenêtre d'affichage des corrections

Sous l'onglet « *Corrections* », on retrouve la fenêtre d'affichage des corrections. Tout comme la fenêtre d'état des canaux, cette fenêtre permet de visualiser d'un seul coup plusieurs mesures et corrections destinées à la solution de navigation (voir Figure A2.8). Les données y sont aussi affichées en temps réel et sont rafraîchies deux fois par secondes.

Channel #	SV #	Pseudo range (m)	Clk Cor (m)	Iono Cor (m)	Tropo Cor (m)	Nav. Solution Residual	
						Position (m)	Velocity (m/s)
GPS - 0	01	20889997.15	0.00	3.38	4.49	-0.420	0.0026
GPS - 1	02	19992551.92	0.00	2.83	3.48	0.263	-0.0027
GPS - 2	27	...	...	...	...	...	...
GPS - 3	22	...	...	...	...	...	...
GPS - 4	23	...	...	...	...	...	...
GPS - 5	28	...	...	...	...	...	...
GPS - 6	24	...	...	...	...	...	...
GPS - 7	08	20638066.32	0.00	3.13	4.15	-0.016	-0.0010
GPS - 8	25	...	...	...	...	...	...
GPS - 9	26	...	...	...	...	...	...
GPS - 10	11	20414022.23	0.00	3.24	3.89	-0.136	0.0022
GPS - 11	12	22549256.14	0.00	5.48	9.06	-0.344	-0.0011
GAL - 0	01	...	...	...	...	...	...
GAL - 1	01	...	...	...	...	...	...
GAL - 2	01	...	...	...	...	...	...
GAL - 3	01	...	...	...	...	...	...
GAL - 4	01	...	...	...	...	...	...
GAL - 5	01	...	...	...	...	...	...
GAL - 6	01	...	...	...	...	...	...
GAL - 7	01	...	...	...	...	...	...
GAL - 8	01	...	...	...	...	...	...
GAL - 9	01	...	...	...	...	...	...
GAL - 10	01	...	...	...	...	...	...
GAL - 11	01	...	...	...	...	...	...

Stop Receiver      Quit

Receiver Status : Nav. Ready      PVT Stable      Satellites Locked : 7

**Figure A2.8 Fenêtre d'affichage des corrections.**

Les données affichées dans cette fenêtre sont les suivantes :

1. Numéros de canal (*Channel #*)
2. Numéros de PRN (*SV #*)
3. Mesures de pseudo-distances (*Pseudo range* – en m)
4. Corrections d'horloge (*Clk Cor* – en m)
5. Corrections ionosphériques (*Iono Cor* – en m)
6. Corrections troposphériques (*Tropo Cor* – en m)

7. Erreurs résiduelles au niveau de la solution de navigation, comprenant :

- a. Résiduels du calcul de position (*Position* – en m)
- b. Résiduels du calcul de vitesse (*Velocity* – en m)

De la même manière que pour la fenêtre d'état des canaux, un clic sur une des données affichées permet d'ouvrir une fenêtre d'information détaillée. Se référer à la section précédente pour plus de détails à ce sujet.

## A2.5 Fenêtre d'affichage étendu

Le dernier onglet, identifié « Display », renferme la fenêtre d'affichage étendu. Cette fenêtre permet à l'utilisateur d'accéder à une panoplie d'informations utiles concernant le récepteur de navigation (voir Figure A2.9).

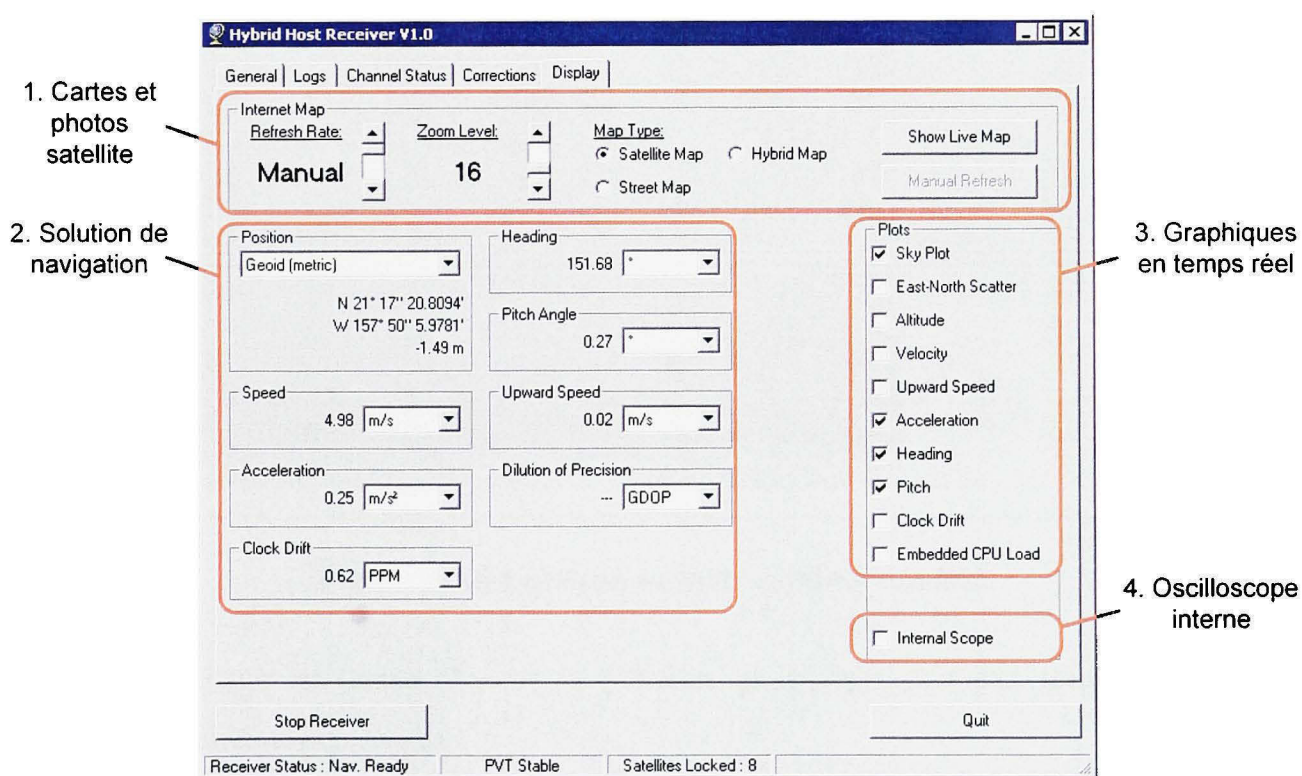
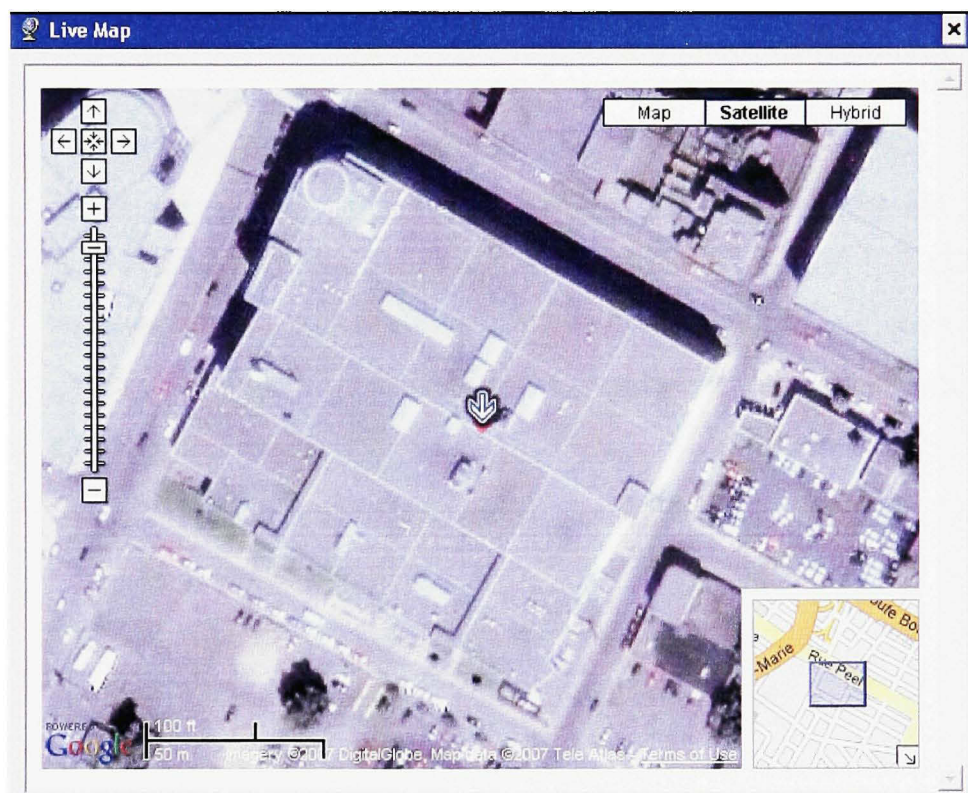


Figure A2.8 Fenêtre d'affichage étendu.



Les fonctionnalités de la fenêtre d'affichage étendu sont :

1. L'affichage de cartes et photos satellites. On emploie des cartes et des photos du site Internet Google Maps (<http://www.google.com/maps>), auxquelles on superpose la trajectoire calculée par le récepteur de navigation (voir Figure A2.9). On ne conserve toutefois que les 30 dernières minutes calculées afin de limiter le temps d'attente lors des requêtes auprès du serveur Google. Le niveau de zoom, la fréquence rafraîchissement et le type de carte affichée sont des paramètres qui peuvent être ajustés au préalable par l'utilisateur.



**Figure A2.8** Exemple de carte et photo satellite.






2. L'affichage en temps réel de la solution de navigation (sous format texte). Les paramètres affichés sont :
  - a. La position (*Position*)
  - b. La vitesse (*Speed*)
  - c. L'accélération (*Acceleration*)
  - d. La dérive d'horloge (*Clock Drift*)

- e. Le cap (ou lacet - *Heading*)
- f. L'angle de pente (ou tangage - *Pitch Angle*)
- g. La vitesse ascensionnelle (*Upward Speed*)
- h. La dilution de précision (*Dilution of Precision*)

L'utilisateur a le choix de plusieurs unités pour l'affichage de ces paramètres, telles que le m/s, le km/h, le ft/s (pied par seconde) ou le mph (mile à l'heure) dans le cas de la vitesse. Il faut noter que la dilution de précision n'est pas disponible pour le moment, cette valeur n'étant pas calculée en temps réel.

3. L'affichage de graphiques en temps réel. Une grande variété de graphiques est présentement disponible, ceux-ci étant :

- a. Affichage de la constellation (*Sky Plot* - voir Figure A2.9). Un graphique de type élévation et azimuth permet de visualiser la constellation GPS et Galileo présentement visible ainsi que le  $C/N_0$  (en dB Hz) des satellites poursuivis. Un système de symboles permet aussi de connaître l'état de chacun des canaux. Ces symboles sont :

-  Phase de poursuite initiale des signaux
-  Poursuite
-  Synchronisation au message de navigation
-  Téléchargement du message de navigation
-  Satellite défectueux

Il faut noter qu'un clic sur un des satellites affichés permet d'ouvrir une fenêtre d'information détaillée. Se référer à la section A2.3 (Fenêtre d'état des canaux) pour plus de détails sur cette fenêtre d'information.

- b. Affichage de la position horizontale (*East-North Scatter* – voir Figure A2.10). Ce graphique est particulièrement utile lors de scénarios en statique car il permet d'estimer la précision de la solution de navigation.

- c. Affichages en fonction du temps. Permet d'afficher un graphique en temps réel de :
- i. la position verticale (*Altitude*)
  - ii. la vitesse (*Velocity*)
  - iii. la vitesse ascensionnelle (*Upward Speed*)
  - iv. l'accélération (*Acceleration*)
  - v. la dérive d'horloge (*Clock Drift*)
  - vi. la charge du CPU embarqué (*Embedded CPU Load* – i.e. le microcontrôleur Microblaze).

Se référer à la Figure A2.11 pour un exemple à ce sujet. Il faut noter que ces graphiques peuvent tous être paramétrés via un clic droit de la souris, tel que décrit à la section A2.3 (Fenêtre d'état des canaux) pour l'affichage de la fonction de cross-corrélation.

- d. Affichage du cap (*Heading*) et de l'angle de pente (*Pitch Angle*). Des cadrans sont employés afin d'afficher ces éléments (voir Figure A2.12).



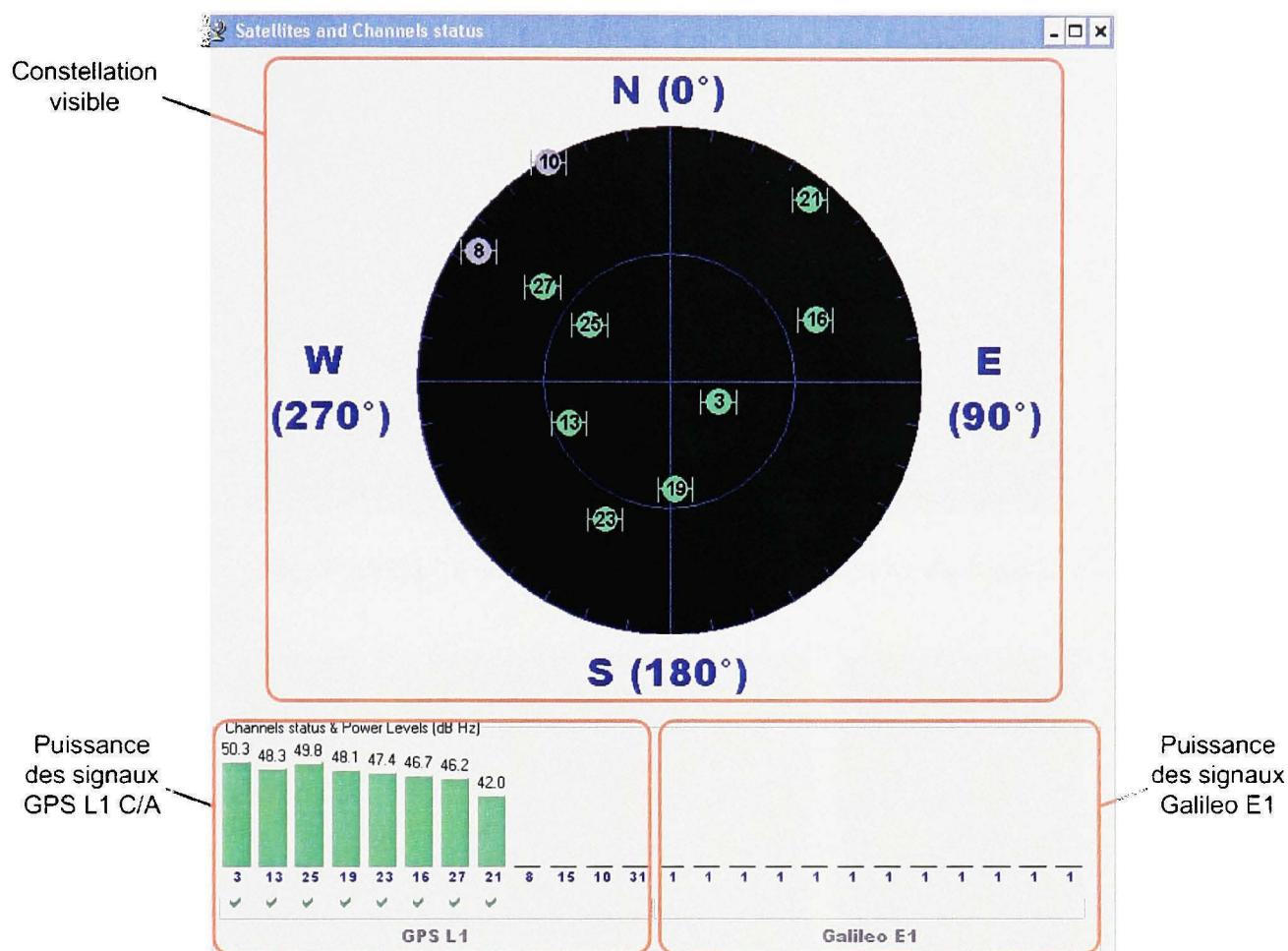


Figure A2.9 Affichage de la constellation.

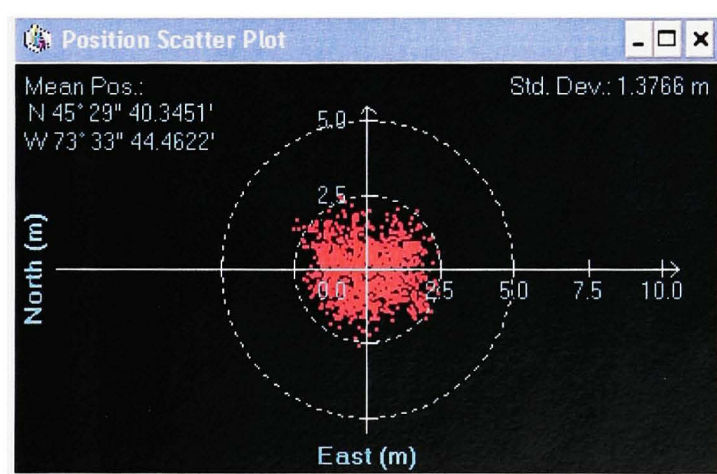


Figure A2.10 Affichage de la position horizontale.

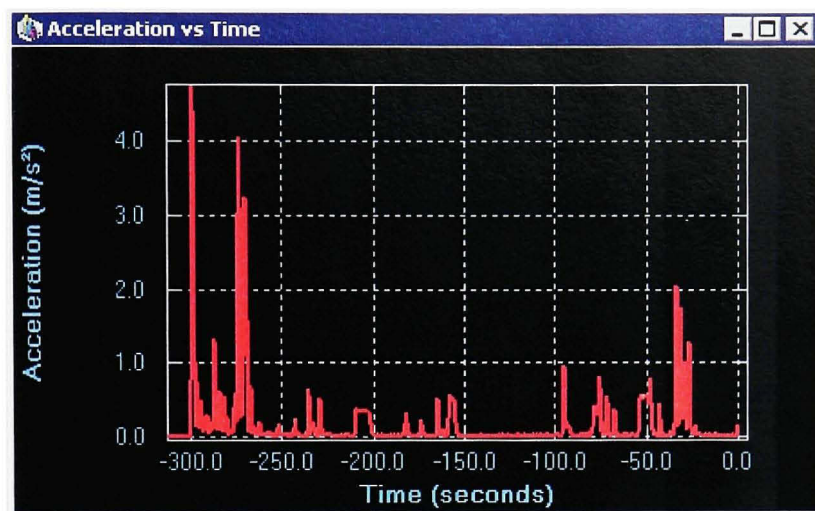


Figure A2.11 Exemple d'affichage en fonction du temps.

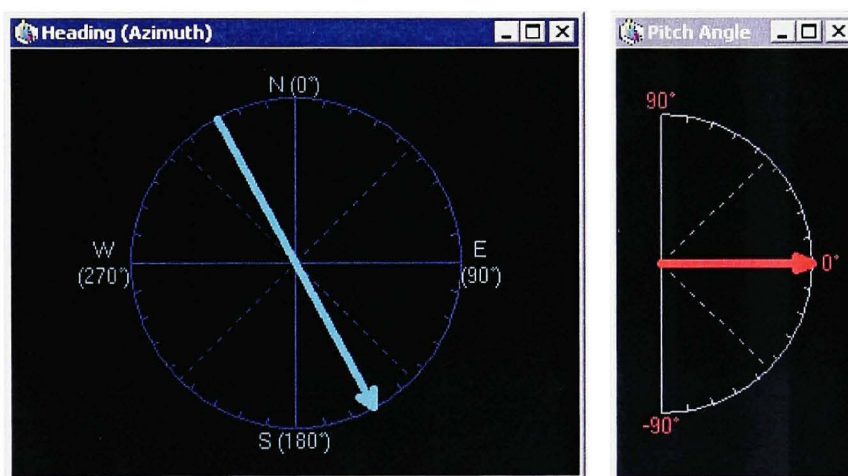
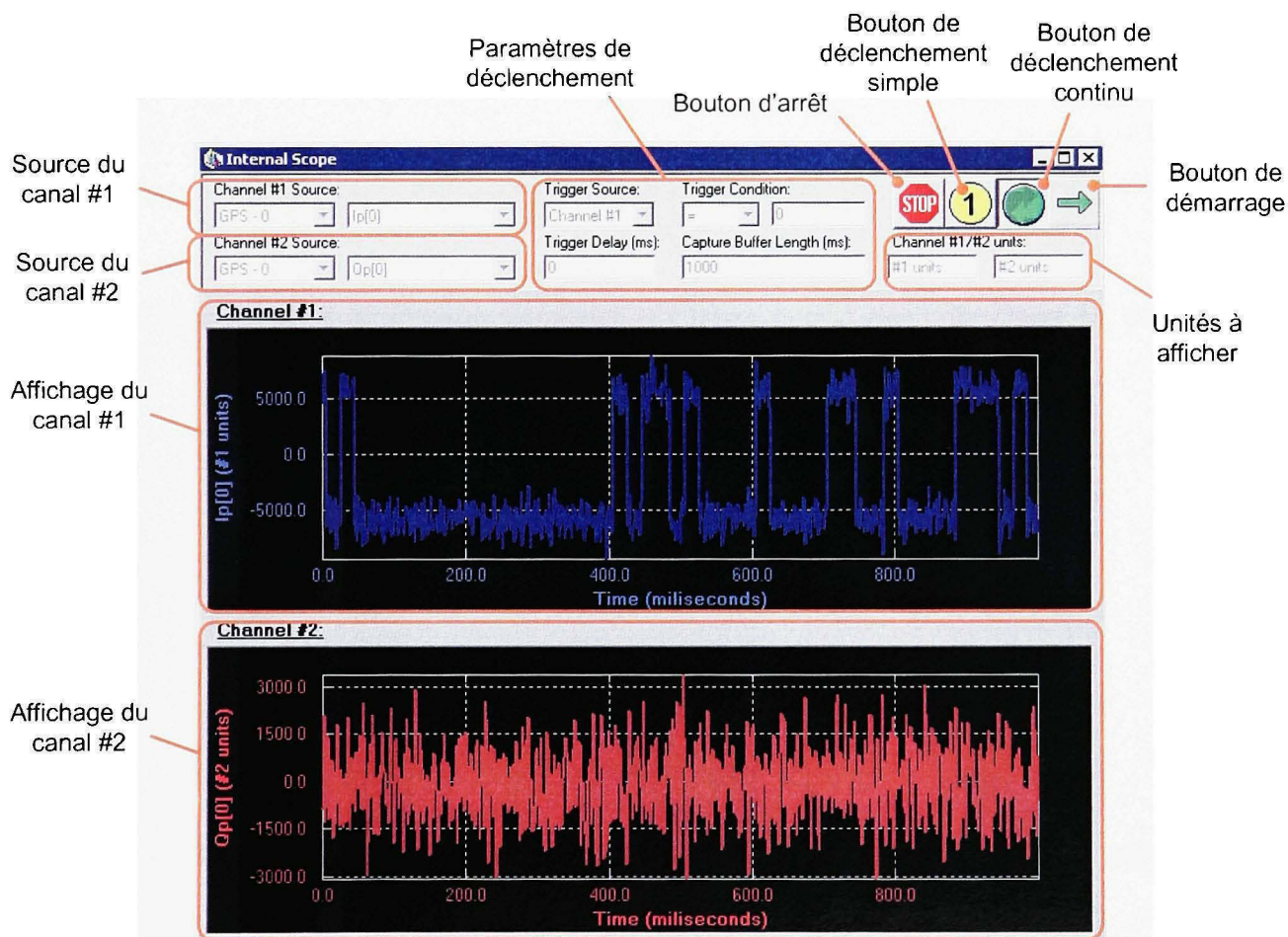


Figure A2.12 Affichage du cap (gauche) et de l'angle de pente (droite).

4. Oscilloscope interne. Ce module permet d'afficher en temps réel les paramètres internes au microcontrôleur embarqué, un peu à la manière d'un oscilloscope. Cette fonctionnalité s'avère extrêmement utile lorsque vient le temps d'apporter des modifications au récepteur et qu'on doit le déboguer (voir Figure A2.13 - on y affiche les signaux  $I_p$  et  $Q_p$  du canal #0 en temps réel).

L'oscilloscope possède deux canaux indépendants. On doit dans un premier temps sélectionner la source de ces deux canaux à l'aide de menus déroulants. Si nécessaire, on doit aussi ajuster les paramètres de déclenchement (*Trigger*). Cette étape est

effectuée de la même manière que pour oscilloscope conventionnel (sélection d'une source de déclenchement, du délai et des conditions). Il s'agit ensuite d'appuyer sur les boutons pour obtenir le comportement souhaité : déclenchement simple ou continu, ou encore déclenchement régulier. Pour un déclenchement simple, on doit tout d'abord appuyer sur le bouton correspondant, puis appuyer sur le bouton de démarrage. Pour un déclenchement continu, il suffit d'appuyer sur le bouton de déclenchement continu. Pour un déclenchement régulier, on doit simplement appuyer sur le bouton de démarrage. Pour arrêter l'oscilloscope, il suffit d'appuyer sur le bouton d'arrêt.



**Figure A2.13 Fenêtre de l'oscilloscope interne.**



## ANNEXE III

### MÉTHODE DE NEWTON

La méthode de Newton est une des méthodes les plus couramment utilisées pour la résolution des équations non linéaires. Ceci s'explique d'une part par la convergence très rapide de cette méthode, que l'on qualifie de quadratique (Fortin, 2001). D'autre part, la méthode de Newton est très facile à implémenter, ce qui la rend très intéressante.

Plusieurs approches sont possibles afin de démontrer la méthode de Newton. L'approche que nous avons choisie ici est basée sur les séries de Taylor. Ainsi, posons une équation  $f(x)$  à résoudre, telle que :

$$f(x) = 0 \quad (\text{A3.1})$$

Afin d'évaluer  $x$  à partir d'un point de départ arbitraire  $x_0$  et d'une correction  $\Delta x$ , posons :

$$f(x) = f(x_0 + \Delta x) = 0 \quad (\text{A3.2})$$

À partir d'ici, réalisons un développement de Taylor autour du point  $x_0$ , tel que :

$$f(x_0) + f'(x_0)\Delta x + \frac{f''(x_0)\Delta x^2}{2!} + \frac{f'''(x_0)\Delta x^3}{3!} + \dots = 0 \quad (\text{A3.3})$$

En tronquant ce développement au deuxième terme, on obtient l'approximation suivante :

$$f(x_0) + f'(x_0)\Delta x \approx 0 \quad (\text{A3.4})$$

En réécrivant l'équation (A3.4), on trouve :

$$\Delta x = -\frac{f(x_0)}{f'(x_0)} \quad (\text{A3.5})$$

Il est alors possible d'évaluer la solution à l'équation (A3.1) par itération, tel que :

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \quad (\text{A3.6})$$

où  $x_n$  est une estimation de  $x$  lors d'une  $n^{\text{ième}}$  itération.

L'algorithme suivant, tiré de (Fortin, 2001), permet de mettre en œuvre facilement la méthode de Newton :

1. Poser  $\varepsilon$  comme critère d'arrêt
2. Fixer  $N$ , le nombre maximal d'itérations
3. Poser  $x_0$ , une estimation de la solution
4. Calculer  $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$
5. Si  $\frac{|x_n - x_{n-1}|}{x_n} < \varepsilon$  :
  - convergence atteinte
  - noter la solution  $x_n$
  - arrêt
6. Si le nombre maximal d'itérations  $N$  est atteint :
  - convergence non atteinte en  $N$  itérations
  - arrêt
7. Retourner à l'étape 4

En guise d'exemple, tentons de résoudre l'équation suivante par la méthode de Newton :

$$E_k - e \cdot \sin E_k - M_k = 0 \quad (\text{A3.7})$$

Ceci est en fait l'équation de Kepler pour l'anomalie excentrique ( $E_k$ ). Les paramètres  $e$  et  $M_k$  représentent de façon respective l'excentricité et l'anomalie moyenne d'une orbite, des paramètres *a priori* connus. Fixons-les ici à  $e = 7,048 \times 10^{-3}$  et  $M_k = 1,145$  rad. Posons ensuite  $\varepsilon = 10^{-9}$ ,  $N = 10$  et  $x_0 = 0$ . Il s'agit alors de calculer :

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} = x_{n-1} - \frac{x_{n-1} - e \cdot \sin x_{n-1} - M_k}{1 - e \cdot \cos x_{n-1}} \quad (\text{A3.8})$$

Pour la première itération, on obtient :

$$x_1 = x_0 - \frac{x_0 - e \cdot \sin x_0 - M_k}{1 - e \cdot \cos x_0} = \frac{1,145}{1 - 7,048 \times 10^{-3}} = 1,153127240793109992 \quad (\text{A3.9})$$

On calcule ensuite :

$$\frac{|x_n - x_{n-1}|}{x_n} = \frac{|1,15312724079311 - 0|}{1,15312724079311} = 1 > \varepsilon \quad (\text{A3.10})$$

Puisque le critère d'arrêt n'est pas satisfait, on doit donc réaliser une nouvelle itération. Les résultats obtenus pour les itérations suivantes sont présentés au Tableau A3.1.

Tableau A3.1  
Solution de l'équation de Kepler pour l'anomalie  
excentrique via la méthode de Newton

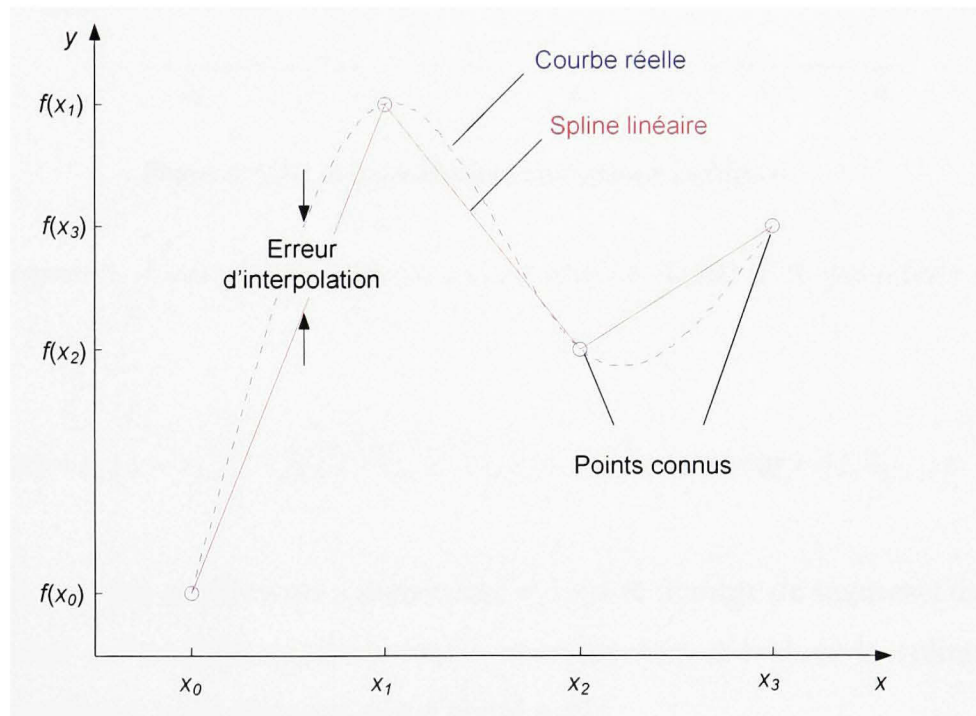
$n$	$x_n$	$\frac{ x_n - x_{n-1} }{x_n}$
1	1,153127240793109992	1
2	1,151437300818404219	$14,7 \times 10^{-3}$
3	1,151437291595204441	$8,01 \times 10^{-9}$
4	1,151437291595204219	$1,92 \times 10^{-16}$



## ANNEXE IV

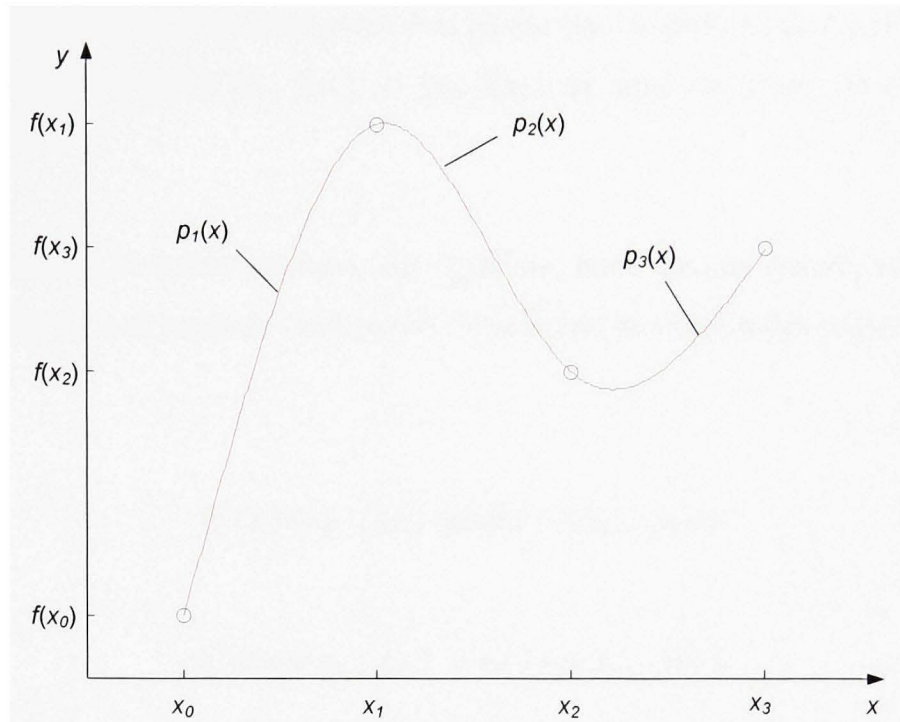
### SPLINES CUBIQUES

Plusieurs techniques existent afin de réaliser des interpolations. La plus simple demeure sans doute l'interpolation linéaire, ou *spline linéaire*, où il s'agit de faire passer un segment de droite par chacune des paires de points connus d'une courbe. Toutefois, cette approche peut, dans certains cas, produire de très mauvaises approximations, sans compter la présence de discontinuités sévères tout au long de la courbe interpolée (voir Figure A4.1).



**Figure A4.1** *Interpolation par spline linéaire.*

Une approche populaire afin de régler ces problèmes consiste à remplacer les segments de droites par des polynômes d'ordre 3, d'où l'appellation « spline cubique ». Il devient alors possible d'obtenir des courbes d'interpolation lisses et régulières en fixant les dérivées premières et secondes de manière à ce qu'elles soient continues d'un polynôme à l'autre (voir Figure A4.2). Le texte qui suit démontre le fonctionnement de la spline cubique.



**Figure A4.2** *Interpolation par spline cubique.*

Chacun des segments d'une spline cubique s'écrit sous la forme d'un polynôme d'ordre 3,  $p_i(x)$ , tel que :

$$p_i(x) = a_i (x - x_{i-1})^3 + b_i (x - x_{i-1})^2 + c_i (x - x_{i-1}) + d_i \quad \text{pour } i = 1, 2, \dots, n \quad (\text{A4.1})$$

où  $a_i$ ,  $b_i$ ,  $c_i$  et  $d_i$  sont des coefficients à déterminer et  $n$  est le nombre de segments de la spline. Nous avons ainsi un total de  $4n$  inconnues à résoudre afin d'évaluer la spline. Afin de résoudre ces inconnues, nous pouvons tout d'abord poser :

$$p_i(x_{i-1}) = f(x_{i-1}) \quad \text{pour } i = 1, 2, \dots, n \quad (\text{A4.2})$$

et

$$p_i(x_i) = f(x_i) \quad \text{pour } i = 1, 2, \dots, n \quad (\text{A4.3})$$

Ceci revient à dire que le polynôme  $p_1(x)$  doit passer par les points  $[x_0, f(x_0)]$  et  $[x_1, f(x_1)]$ , le polynôme  $p_2(x)$  par le point  $[x_1, f(x_1)]$  et  $[x_2, f(x_2)]$  et ainsi de suite. On obtient ainsi  $2n$  équations.

De manière à ce que la spline cubique soit régulière, nous devons ensuite nous assurer que les dérivées premières et secondes aux points de jonction de chacun des polynômes soient les mêmes :

$$p_i'(x_i) = p_{i+1}'(x_i) \text{ pour } i = 1, 2, \dots, n-1 \quad (\text{A4.4})$$

et

$$p_i''(x_i) = p_{i+1}''(x_i) \text{ pour } i = 1, 2, \dots, n-1 \quad (\text{A4.5})$$

De cette façon, nous obtenons  $(2n-2)$  équations supplémentaires, pour un total de  $(4n-2)$  équations. Il nous manque alors deux équations afin de résoudre le système. Afin d'y parvenir, nous posons généralement :

$$p_1''(x_0) = p_n''(x_n) = 0 \quad (\text{A4.6})$$

On force ainsi à zéro la dérivée seconde aux deux extrémités de la spline cubique. On qualifie la courbe résultante de *spline naturelle* (Fortin, 2001).

À partir d'ici, il est possible de manipuler les équations développées afin d'obtenir un système à  $4n$  inconnues et  $4n$  équations linéaires. Le lecteur est invité à consulter (Fortin, 2001) pour plus de détails à ce sujet. Suite à ce développement, on obtient le système matriciel suivant, composé des matrices **F**, **H** et **S**, où :

$$\mathbf{H} \cdot \mathbf{S} = \mathbf{6F} \quad (\text{A4.7})$$

Chacune des matrices se compose donc comme suit :



$$\mathbf{H} = \begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix} \quad (\text{A4.8})$$

où :

$$h_i = x_{i+1} - x_i \quad (\text{A4.9})$$

$$\mathbf{F} = \begin{bmatrix} \frac{f_1}{h_1} - \frac{f_0}{h_0} \\ \frac{f_2}{h_2} - \frac{f_1}{h_1} \\ \vdots \\ \frac{f_n - f_{n-1}}{h_{n-1}} - \frac{f_{n-1} - f_{n-2}}{h_{n-2}} \end{bmatrix}^T \quad (\text{A4.10})$$

où :

$$f_i = f(x_{i+1}) - f(x_i) \quad (\text{A4.9})$$

$$\mathbf{S} = [s_0 \quad s_1 \quad s_2 \quad \cdots \quad s_n]^T \quad (\text{A4.11})$$

On remarque que  $\mathbf{H}$  est de dimension  $(n-1) \times (n-1)$  et que  $\mathbf{F}$  est de dimension  $(n-1) \times 1$ . Toutefois, la matrice  $\mathbf{S}$  complique les choses avec une taille de  $(n+1) \times 1$  éléments. Puisque nous recherchons les paramètres d'une spline naturelle, nous pouvons néanmoins poser  $s_0 = s_n = 0$  pour résoudre le système ainsi :

$$[s_1 \quad s_2 \quad \cdots \quad s_{n-1}]^T = 6\mathbf{H}^{-1}\mathbf{F} \quad (\text{A4.12})$$

Finalement, on calcule les coefficients de chacun des polynômes de la façon suivante :

$$a_{i+1} = \frac{s_{i+1} - s_i}{6h_i}, \quad b_{i+1} = \frac{s_i}{2}, \quad c_{i+1} = \frac{f_i}{h_i} - \frac{2h_i s_i + h_i s_{i+1}}{6} \text{ et } d_{i+1} = f(x_i) \quad (\text{A4.13})$$

où  $i = 0, 1, 2, \dots, n-1$ .

Tentons par exemple de faire passer une spline cubique par les points  $[x_i, f(x_i)]$  affichés au Tableau A4.1.

Tableau A4.1  
Données pour le calcul d'une spline cubique

$i$	0	1	2	3
$x_i$	1	3	4	5
$f(x_i)$	2	3	10	4

La première étape consiste à calculer les éléments  $h$  et  $f$ , tels que :

- $h_0 = x_1 - x_0 = 3 - 1 = 2$
- $h_1 = x_2 - x_1 = 4 - 3 = 1$
- $h_2 = x_3 - x_2 = 5 - 4 = 1$
- $f_0 = f(x_1) - f(x_0) = 3 - 2 = 1$
- $f_1 = f(x_2) - f(x_1) = 10 - 3 = 7$
- $f_2 = f(x_3) - f(x_2) = 4 - 10 = -6$

On peut ensuite construire la matrice  $\mathbf{H}$  :

$$\mathbf{H} = \begin{bmatrix} 2(2+1) & 1 \\ 1 & 2(1+1) \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ 1 & 4 \end{bmatrix} \quad (\text{A4.14})$$

puis la matrice  $\mathbf{F}$  :

$$\mathbf{F} = \begin{bmatrix} \frac{7}{1} - \frac{1}{2} \\ \frac{-6}{1} - \frac{7}{1} \end{bmatrix} = \begin{bmatrix} 6,5 \\ -13 \end{bmatrix} \quad (\text{A4.15})$$

On doit ensuite inverser la matrice  $\mathbf{H}$ , ce qui peut être fait ici par la méthode de Kramer :

$$\mathbf{H}^{-1} = \frac{\text{cof } \mathbf{H}}{\det \mathbf{H}} = \frac{1}{6 \cdot 4 - 1 \cdot 1} \begin{bmatrix} 4 & -1 \\ -1 & 6 \end{bmatrix} = \begin{bmatrix} 4/23 & -1/23 \\ -1/23 & 6/23 \end{bmatrix} \quad (\text{A4.16})$$

On peut enfin résoudre le système pour obtenir :

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = 6 \begin{bmatrix} 4/23 & -1/23 \\ -1/23 & 6/23 \end{bmatrix} \cdot \begin{bmatrix} 6,5 \\ -13 \end{bmatrix} = \begin{bmatrix} 10,1739 \\ -22,0435 \end{bmatrix} \quad (\text{A4.12})$$

On obtient alors la matrice  $\mathbf{S}$  suivante :

$$\mathbf{S} = \begin{bmatrix} 0 & 10,1739 & -22,0435 & 0 \end{bmatrix}^T \quad (\text{A4.12})$$

Finalement, on calcule les coefficients  $a_i$ ,  $b_i$ ,  $c_i$  et  $d_i$  pour chacun des polynômes :

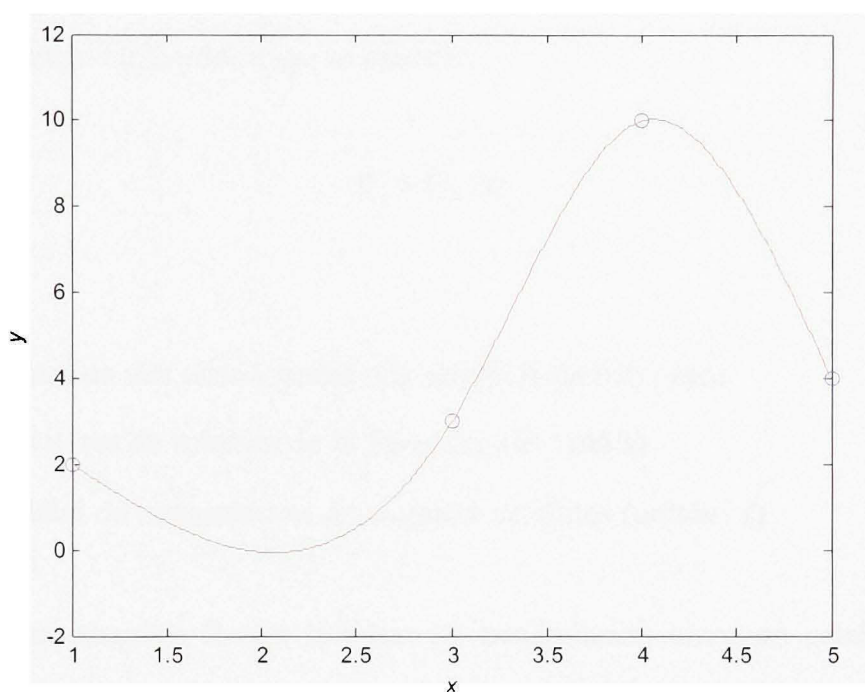
- $a_1 = \frac{s_1 - s_0}{6h_0} = \frac{10,1739 - 0}{6 \cdot 2} = 0,8478$
- $b_1 = \frac{s_0}{2} = \frac{0}{2} = 0$
- $c_1 = \frac{f_0}{h_0} - \frac{2h_0 s_0 + h_0 s_1}{6} = \frac{1}{2} - \frac{2 \cdot 2 \cdot 0 + 2 \cdot 10,1739}{6} = -2,8913$
- $d_1 = f(x_0) = 2$

Les autres coefficients sont présentés au Tableau A4.2. Le résultat final est affiché à la Figure A4.3.



Tableau A4.2  
Coefficients d'une spline cubique

$i$	$a_i$	$b_i$	$c_i$	$d_i$
1	0,8478	0	-2,8913	2
2	-5,3696	5,0870	7,2826	3
3	3,6739	-11,0217	1,3478	10



**Figure A4.3** *Exemple de spline cubique.*

## ANNEXE V

### ERREUR RÉSIDUELLE LIÉE À LA CORRECTION DE LA ROTATION DE LA TERRE

Tel que vu au chapitre 2, il est nécessaire, avant de procéder à la résolution des équations de navigation, de corriger le décalage introduit par la rotation de la Terre au niveau de la position des satellites. Or, ce décalage se calcule :

$$\theta_e = \dot{\Omega}_e \Delta t_t \quad (\text{A5.1})$$

où :

- $\theta_e$  est la rotation des coordonnées des satellites (unités : rad)
- $\dot{\Omega}_e$  est la vitesse de rotation de la Terre (unités : rad/s)
- $\Delta t_t$  est le délai de transmission des signaux satellites (unités : s)

Nous avons vu au chapitre 2 que le délai de transmission entre un satellite GPS et un utilisateur au sol est de l'ordre de  $\Delta t_t = 67,3$  ms, ce qui résulte en un décalage de  $\theta_e = 4,9$   $\mu$ rad (on considère ici une vitesse de rotation  $\dot{\Omega}_e = 72,9$   $\mu$ rad/s) . Nous avons aussi vu que si ce décalage n'était pas corrigé, une erreur maximale de 130,5 m pouvait être induite au niveau de la solution de navigation, telle que :

$$\Delta s = r_{sat} \sin \theta_e = 26600 \times 10^3 \text{ m} \sin(4,906 \times 10^{-6} \text{ rad}) \approx 130,5 \text{ m} \quad (\text{A5.2})$$

où :

- $\Delta s$  est le déplacement du satellite associé à la rotation du référentiel ECEF
- $r_{sat}$  est le rayon de l'orbite du satellite

Nous avons toutefois vu que cette erreur peut être corrigée en quatre étapes pour chacun des satellites de navigation. Ces étapes sont :

- 1) Estimation de la distance satellite-utilisateur ( $d$ ) :

$$d = \sqrt{(x_u - x_{sat})^2 + (y_u - y_{sat})^2 + (z_u - z_{sat})^2} \quad (\text{A5.3})$$

où :

- $x_u, y_u$  et  $z_u$  représentent la position ECEF de l'utilisateur au temps de réception
- $x_{sat}, y_{sat}$  et  $z_{sat}$  représentent la position ECEF du satellite au temps d'émission

- 2) Calcul du délai de propagation des signaux ( $\Delta t_t$ ) :

$$\Delta t_t = \frac{d}{c} \quad (\text{A5.4})$$

où  $c$  représente la vitesse de la lumière.

- 3) Calcul de l'angle de rotation associé au référentiel ECEF ( $\theta$ ) :

$$\theta = -\dot{\Omega}_e \Delta t_t \quad (\text{A5.5})$$

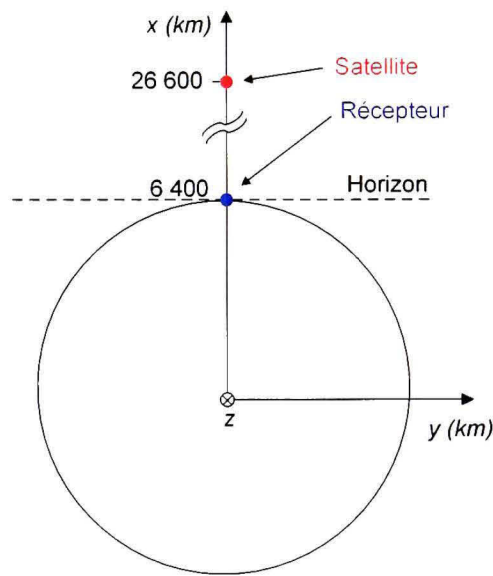
- 4) Correction de la position du satellite :

$$\begin{aligned} x'_{sat} &= x_{sat} \cos\theta - y_{sat} \sin\theta \\ y'_{sat} &= x_{sat} \sin\theta + y_{sat} \cos\theta \\ z'_{sat} &= z_{sat} \end{aligned} \quad (\text{A5.6})$$

Malgré cela, une faible erreur résiduelle peut subsister après avoir appliqué les corrections. Une estimation de cette erreur résiduelle est faite dans le texte qui suit.



Tel que mentionné au chapitre 2, on prévoit un décalage maximal des coordonnées pour un satellite situé à l'équateur. Supposons donc qu'un satellite GPS soit situé au-dessus de l'équateur aux coordonnées  $(x_{sat}, y_{sat}, z_{sat}) = (26,6 \times 10^6, 0, 0)$  m (où  $26,6 \times 10^6$  m est le rayon d'orbite moyen d'un satellite GPS – voir chapitre 2). Supposons aussi que le satellite soit situé au zénith par rapport à un utilisateur, en fixant les coordonnées de ce dernier à  $(x_u, y_u, z_u) = (6,4 \times 10^6, 0, 0)$  m (voir Figure A5.1 -  $6,4 \times 10^6$  m est une approximation du rayon de la Terre).



**Figure A5.1** *Satellite GPS situé au zénith.*

En appliquant les équations A5.3 à A5.6, on obtient alors :

$$\begin{aligned}
 d &= \sqrt{(x_u - x_{sat})^2 + (y_u - y_{sat})^2 + (z_u - z_{sat})^2} \\
 &= \sqrt{(26,6 \times 10^6 - 6,4 \times 10^6)^2 - 0^2 - 0^2} = 20,2 \times 10^6 \text{ m}
 \end{aligned}
 \tag{A5.7}$$

$$\Delta t_t = \frac{d}{c} = \frac{20,2 \times 10^6 \text{ m}}{2,99792458 \times 10^8 \text{ m/s}} = 67,38 \text{ ms}
 \tag{A5.8}$$

$$\theta = -\dot{\Omega}_e \Delta t_t = -7,2921151467 \times 10^{-5} \text{ rad/s} \cdot 67,38 \times 10^{-3} \text{ s} = -4,9134 \text{ } \mu\text{rad}
 \tag{A5.9}$$

$$\begin{aligned}
x_{sat}' &= x_{sat} \cos\theta - y_{sat} \sin\theta \\
&= 26,6 \times 10^6 \cos(-4,9134 \times 10^{-6}) - 0 \sin(-4,9134 \times 10^{-6}) \\
&= 26,6 \times 10^6 \text{ m} \\
y_{sat}' &= x_{sat} \sin\theta + y_{sat} \cos\theta \\
&= 26,6 \times 10^6 \sin(-4,9134 \times 10^{-6}) + 0 \cos(-4,9134 \times 10^{-6}) \\
&= -130,697 \text{ m} \\
z_{sat}' &= z_{sat} = 0 \text{ m}
\end{aligned} \tag{A5.10}$$

On n'obtient donc aucune correction significative en  $x$  ( $x_{sat}' - x_{sat} < 1 \text{ mm}$ ) et une correction de  $-130,697 \text{ m}$  en  $y$ . Afin de déterminer l'erreur résiduelle sur ces corrections, il suffit tout simplement d'appliquer à nouveau les équations A5.3 à A5.6, tel que :

$$\begin{aligned}
d' &= \sqrt{(x_u - x_{sat}')^2 + (y_u - y_{sat}')^2 + (z_u - z_{sat}')^2} \\
&= \sqrt{(26,6 \times 10^6 - 6,4 \times 10^6)^2 + (-130,697 - 0)^2} = 20,20000000010173 \times 10^6 \text{ m}
\end{aligned} \tag{A5.11}$$

$$\Delta t_t' = \frac{d'}{c} = \frac{20,20000000010173 \times 10^6 \text{ m}}{2,99792458 \times 10^8 \text{ m/s}} = 67,37994723036606 \text{ ms} \tag{A5.12}$$

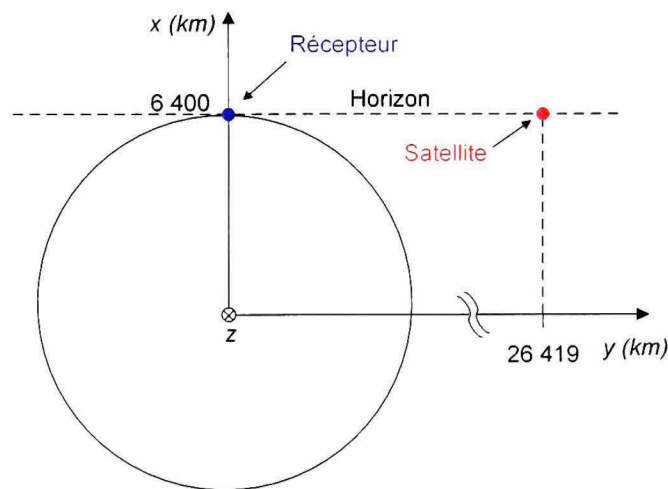
$$\begin{aligned}
\theta' &= -\dot{\Omega}_\varphi \Delta t_t' \\
&= -7,2921151467 \times 10^{-5} \cdot 67,37994723036606 \times 10^{-3} \\
&= -4,91342333782399 \text{ } \mu\text{rad}
\end{aligned} \tag{A5.13}$$

$$\begin{aligned}
x_{sat}'' &= x_{sat} \cos\theta' - y_{sat} \sin\theta' \\
&= 26,6 \times 10^6 \cos(-4,91342333782399 \times 10^{-6}) \\
&= 26,6 \times 10^6 \text{ m} \\
y_{sat}'' &= x_{sat} \sin\theta' + y_{sat} \cos\theta' \\
&= 26,6 \times 10^6 \sin(-4,91342333782399 \times 10^{-6}) \\
&= -130,697 \\
z_{sat}'' &= z_{sat} = 0 \text{ m}
\end{aligned} \tag{A5.14}$$

On obtient ainsi des résultats tout à fait similaires à la première itération, ce qui témoigne d'une erreur résiduelle très faible. Une comparaison plus poussée permet de détecter une erreur résiduelle maximale de l'ordre de  $10^{-7}$  m sur l'axe des  $y$  (obtenue en calculant  $y_{sat}' - y_{sat}''$ ), ce qui peut être considéré comme négligeable dans la majorité des applications.

Bien que l'erreur résiduelle soit quasiment nulle pour un satellite situé au zénith, vérifions à nouveau pour un satellite situé à l'horizon. Supposons que l'utilisateur et le satellite soient une fois de plus localisés le long de l'équateur aux coordonnées suivantes (voir Figure A5.2) :

- $(x_u, y_u, z_u) = (6,4 \times 10^6, 0, 0)$  m
- $(x_{sta}, y_{sta}, z_{sta}) = (6,4 \times 10^6, 26,419 \times 10^6, 0)$  m



**Figure A5.2** *Satellite GPS situé à l'horizon.*

En appliquant à nouveau les équations A5.3 à A5.6, on obtient alors une correction de 169,772 m en  $x$  et de -41,128 m en  $y$ . Un dernier calcul permet enfin de déterminer une erreur résiduelle de -0,3 mm et 0,06 mm sur les axes  $x$  et  $y$  respectivement, ce qui est raisonnablement petit pour la majorité des applications. Une telle erreur peut toutefois s'avérer gênante dans certaines applications où un positionnement de très haute précision est recherché.



En résumé, le décalage introduit par la rotation de la Terre sur les coordonnées des satellites GNSS peut être corrigé par une seule itération des équations A5.3 à A5.6. Une erreur résiduelle de l'ordre du dixième de millimètre est toutefois à prévoir dans ce cas-ci. Si cette erreur ne convient pas, on peut alors procéder à une seconde itération, telle que vue aux équations A5.11 à A5.14, dans lequel cas l'erreur sera pratiquement supprimée.

## ANNEXE VI

### CALCUL DE L'INTÉGRALE SOUS LA FORME $\sin^2(ax)/x^2$

Cette annexe détaille le calcul d'une intégrale de la forme  $G(x) = \sin^2(ax)/x^2$ . Ceci est utile, par exemple, afin de déterminer la puissance d'un signal de type NRZ ou modulé DS-CDMA à partir de sa densité spectrale de puissance. Le développement est présenté ci-dessous :

$$P(x) = \int_{\alpha}^{\beta} G(x) dx = \int_{\alpha}^{\beta} \frac{\sin^2(ax)}{x^2} dx \quad (A6.1)$$

Cette intégrale peut se résoudre par parties, tel que :

$$\int_p^q u dv = uv \Big|_p^q - \int_p^q v du \quad (A6.2)$$

En posant  $u = \sin^2(ax)$  et  $dv = dx/x^2$ , on calcule :

$$du = \frac{d}{dx} \sin^2(ax) = 2a \sin(ax) \cos(ax) dx = a \sin(2ax) dx \quad (A6.3)$$

$$v = \int dv = \int \frac{dx}{x^2} = -\frac{1}{x} \quad (A6.4)$$

On obtient donc l'intégrale par parties suivante :

$$P(x) = -\frac{\sin^2(ax)}{x} \Big|_{\alpha}^{\beta} + a \int_{\alpha}^{\beta} \frac{\sin(2ax)}{x} dx \quad (A6.5)$$

D'après les tables de mathématiques (Spiegel et Liu, 2000), la primitive de  $\sin(kx)/x$  est une série infinie, telle que :

$$\int \frac{\sin(kx)}{x} dx = kx - \frac{(kx)^3}{3 \cdot 3!} + \frac{(kx)^5}{5 \cdot 5!} - \dots \quad (\text{A6.6})$$

Par substitution, l'équation (A6.5) devient alors :

$$P(x) = -\frac{\sin^2(ax)}{x} \Big|_a^\beta + a \left( 2ax - \frac{(2ax)^3}{3 \cdot 3!} + \frac{(2ax)^5}{5 \cdot 5!} - \dots \right) \Big|_a^\beta \quad (\text{A6.7})$$

Appliquons par exemple cette solution à une fonction  $G(f)$  qui représente la densité spectrale de puissance en fonction de la fréquence ( $f$ ) d'un signal binaire NRZ aléatoire, encodé entre -1 et +1 et de période égale à  $T$  (Gerakoulis et Geraniotis, 2001) :

$$G(f) = T \left( \frac{\sin(\pi f T)}{\pi f T} \right)^2 \quad (\text{A6.8})$$

La puissance du signal ( $C$ ) sur une bande de fréquence de largeur  $B$  et symétrique par rapport à la fréquence centrale du spectre  $G(f)$  s'exprime alors :

$$C = \int_{-B/2}^{B/2} G(f) df = \frac{1}{\pi^2 T} \int_{-B/2}^{B/2} \frac{\sin^2(\pi f T)}{f^2} df \quad (\text{A6.9})$$

En substituant l'équation (A6.7) à l'équation (A6.9), on obtient :

$$C = \frac{1}{\pi^2 T} \left[ -\frac{\sin^2(\pi f T)}{f} + \pi T \left( 2\pi f T - \frac{(2\pi f T)^3}{3 \cdot 3!} + \frac{(2\pi f T)^5}{5 \cdot 5!} - \dots \right) \right]_{-B/2}^{B/2} \quad (\text{A6.10})$$

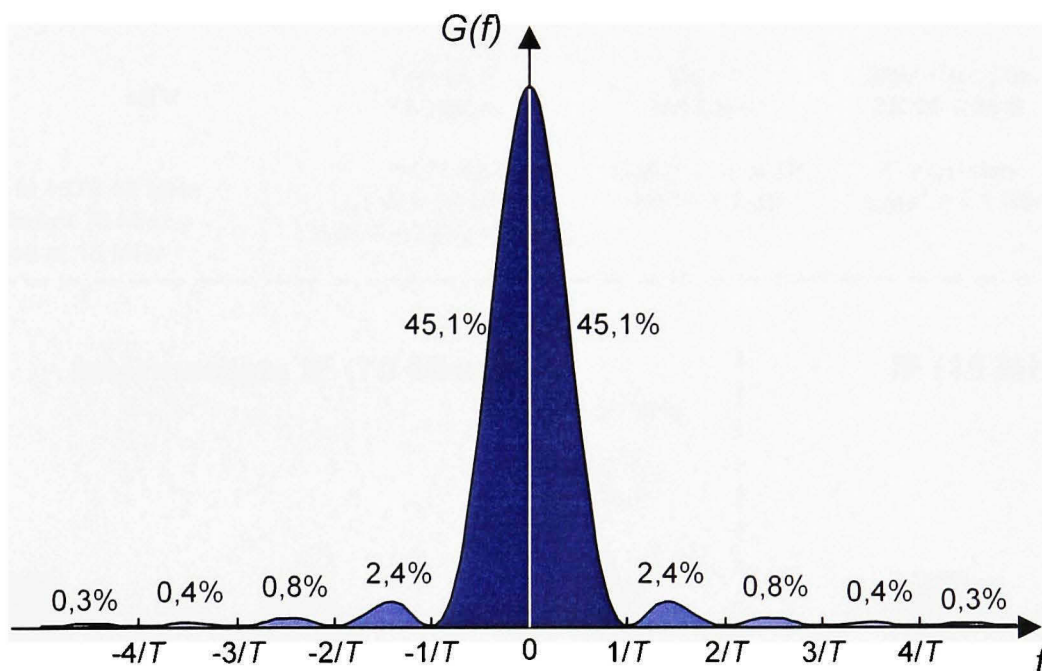
Si on s'en tient au lobe principal de  $G(f)$ , c'est-à-dire que  $B/2 = 1/T$ , on calcule alors :



$$C = \frac{2}{\pi^2} \left[ -\sin^2(\pi) + \pi \left( 2\pi - \frac{(2\pi)^3}{3 \cdot 3!} + \frac{(2\pi)^5}{5 \cdot 5!} - \dots \right) \right] \quad (\text{A6.11})$$

$$C = \frac{2}{\pi} \left( 2\pi - \frac{(2\pi)^3}{3 \cdot 3!} + \frac{(2\pi)^5}{5 \cdot 5!} - \dots \right) \cong 0,9028 \quad (\text{A6.12})$$

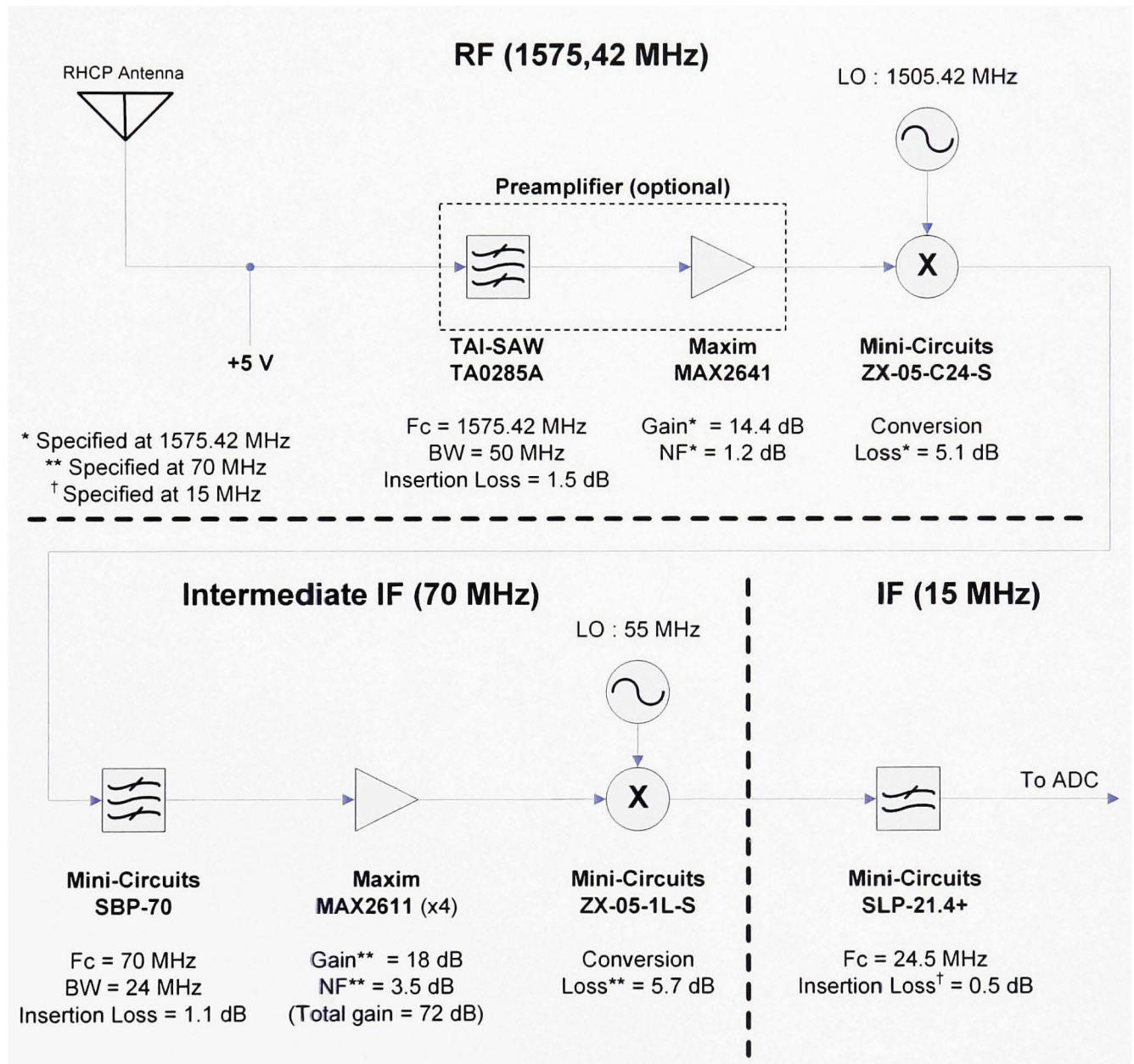
Ainsi, près de 90 % de la puissance est comprise à l'intérieur du lobe principal d'un signal NRZ, ce peu importe la fréquence du signal. La distribution de la puissance sur les principaux lobes d'un signal NRZ est illustrée à la Figure A6.1.



**Figure A6.1** *Distribution de la puissance sur les principaux lobes d'un signal NRZ.*

## ANNEXE VII

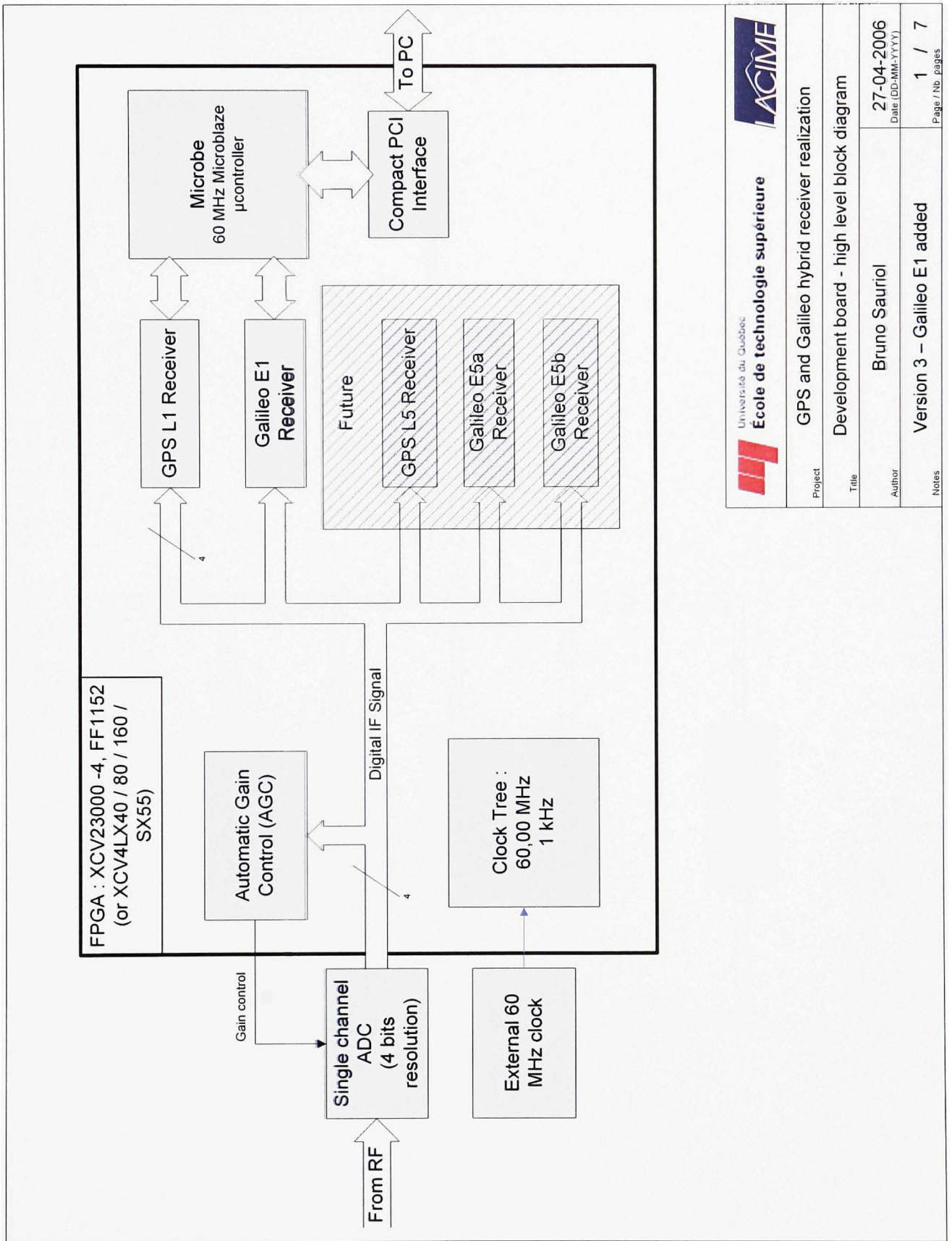
### SCHÉMA DÉTAILLÉ DE LA TÊTE RF DÉVELOPPÉE





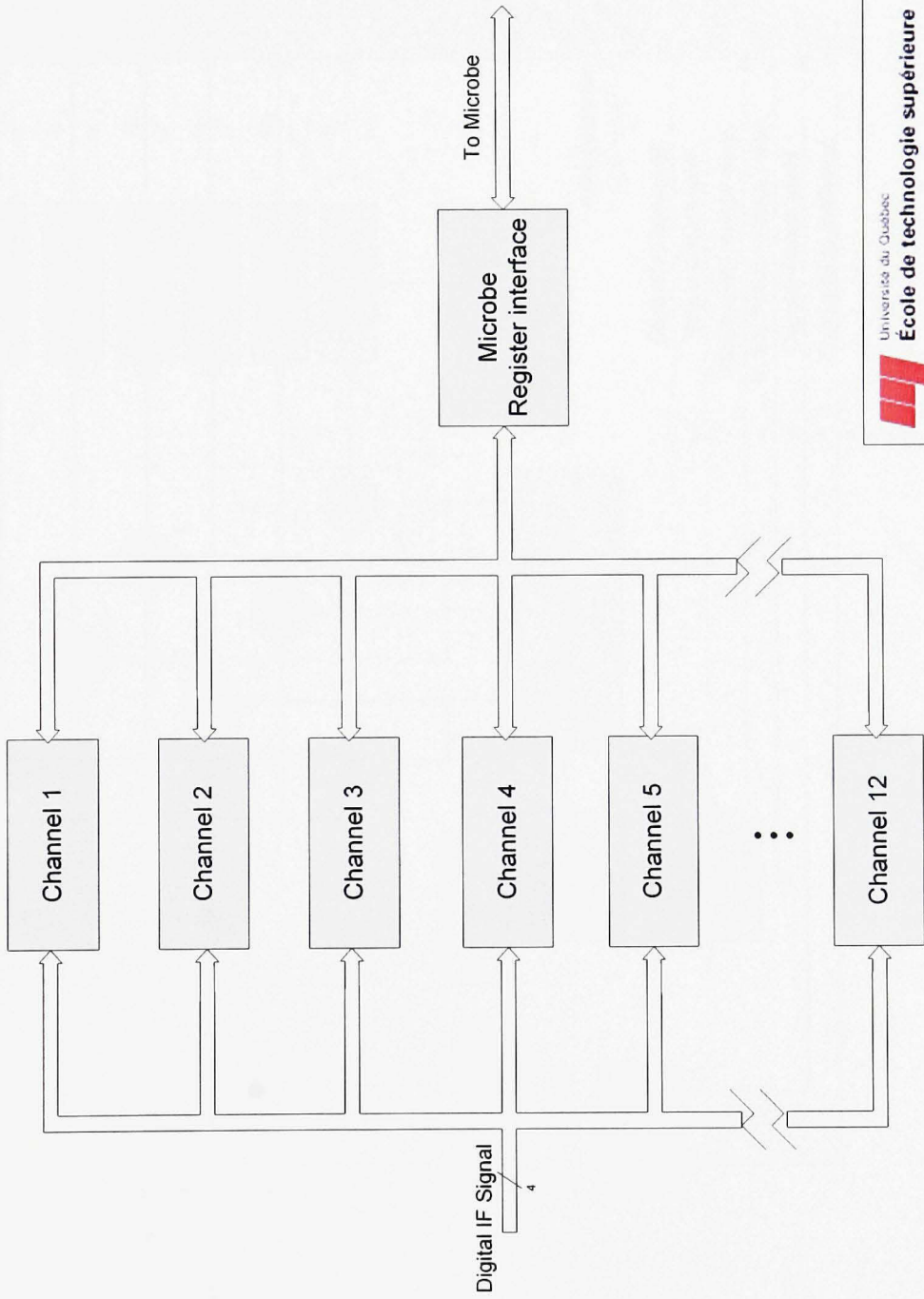
## **ANNEXE VIII**

### **SCHÉMAS DÉTAILLÉS DE L'ARCHITECTURE IF**





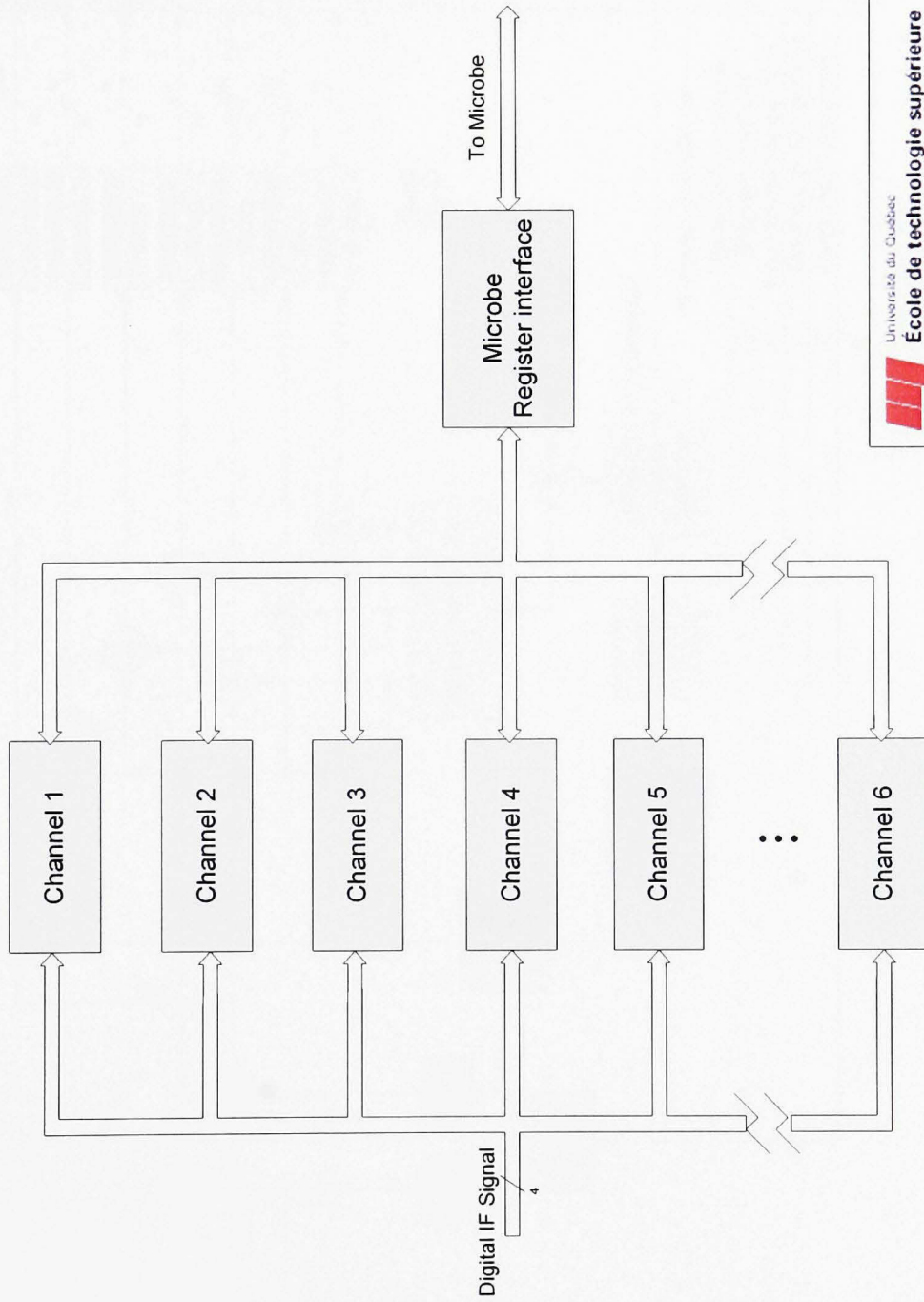
 <b>École de technologie supérieure</b> 	
Project	GPS and Galileo hybrid receiver realization
Title	Development board - high level block diagram
Author	Bruno Sauriol
Notes	Version 3 – Galileo E1 added
	27-04-2006 Date (DD-MM-YYYY)
	1 / 7 Page / Nb. pages



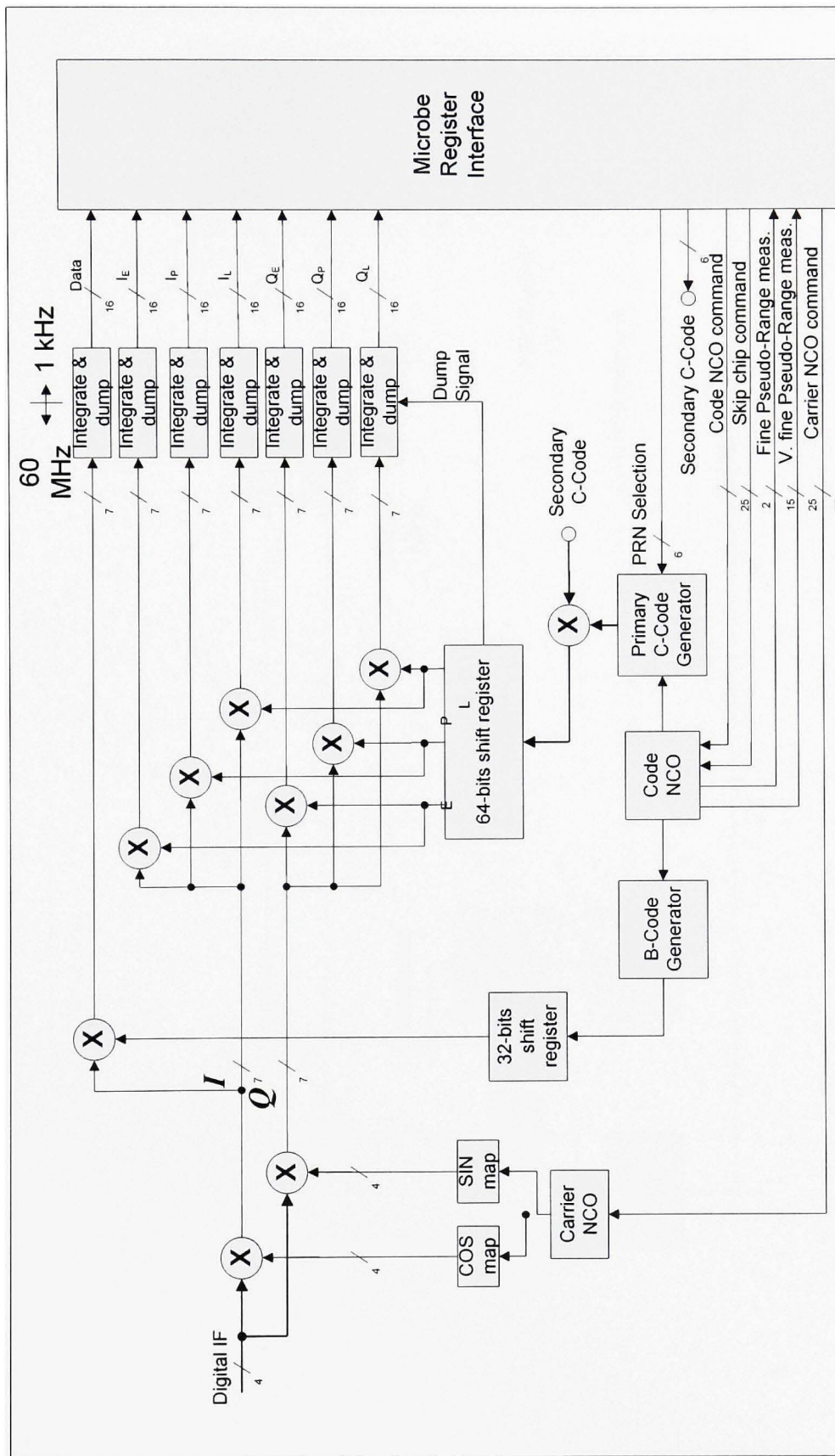
Project	GPS and Galileo hybrid receiver realization		
Title	GPS L1 Receiver Architecture		
Author	Bruno Sauriol	24-04-2006 Date (DD-MM-YYYY)	
Notes	Version 2 – Microbe added	2 / 7	Page / Nb. pages

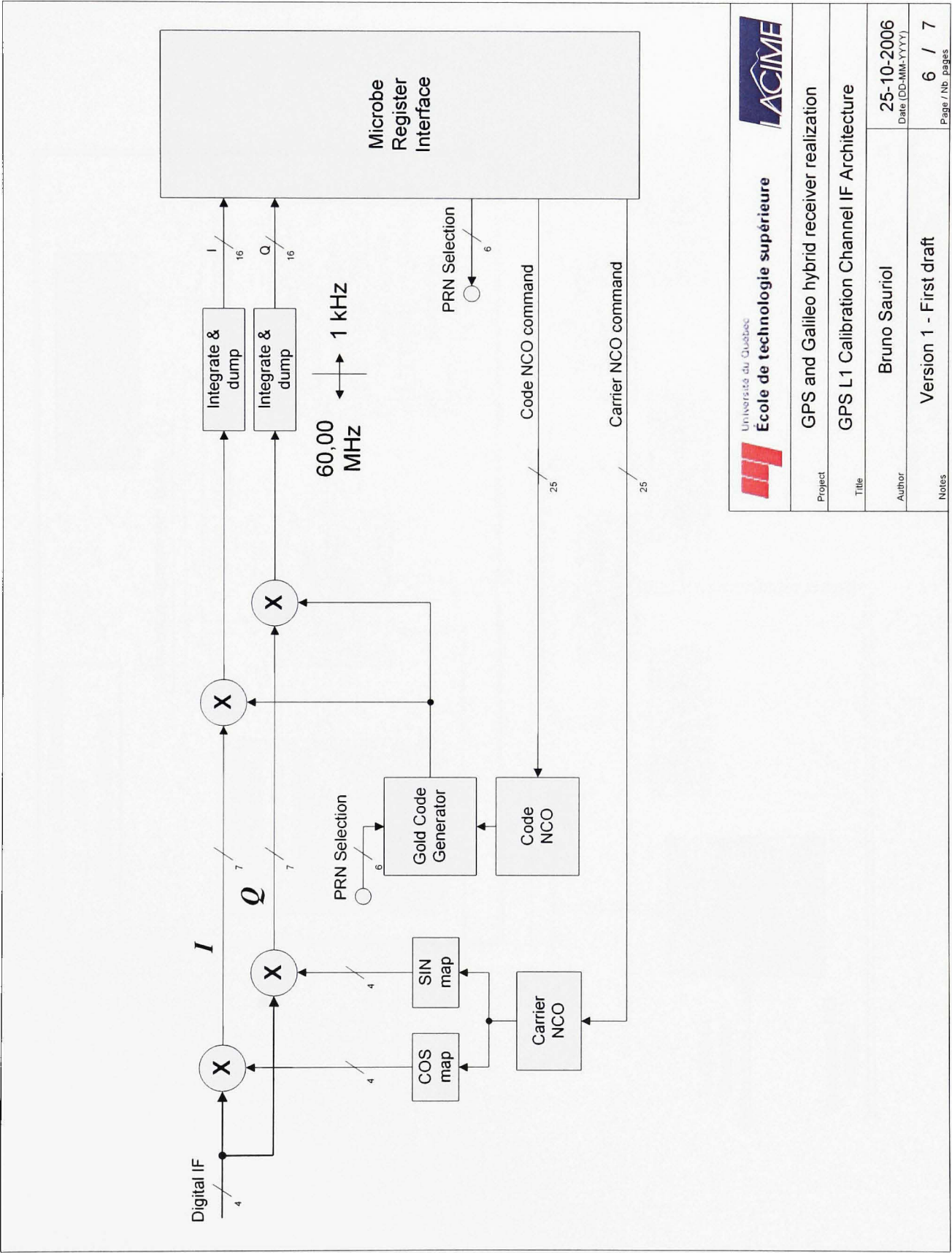






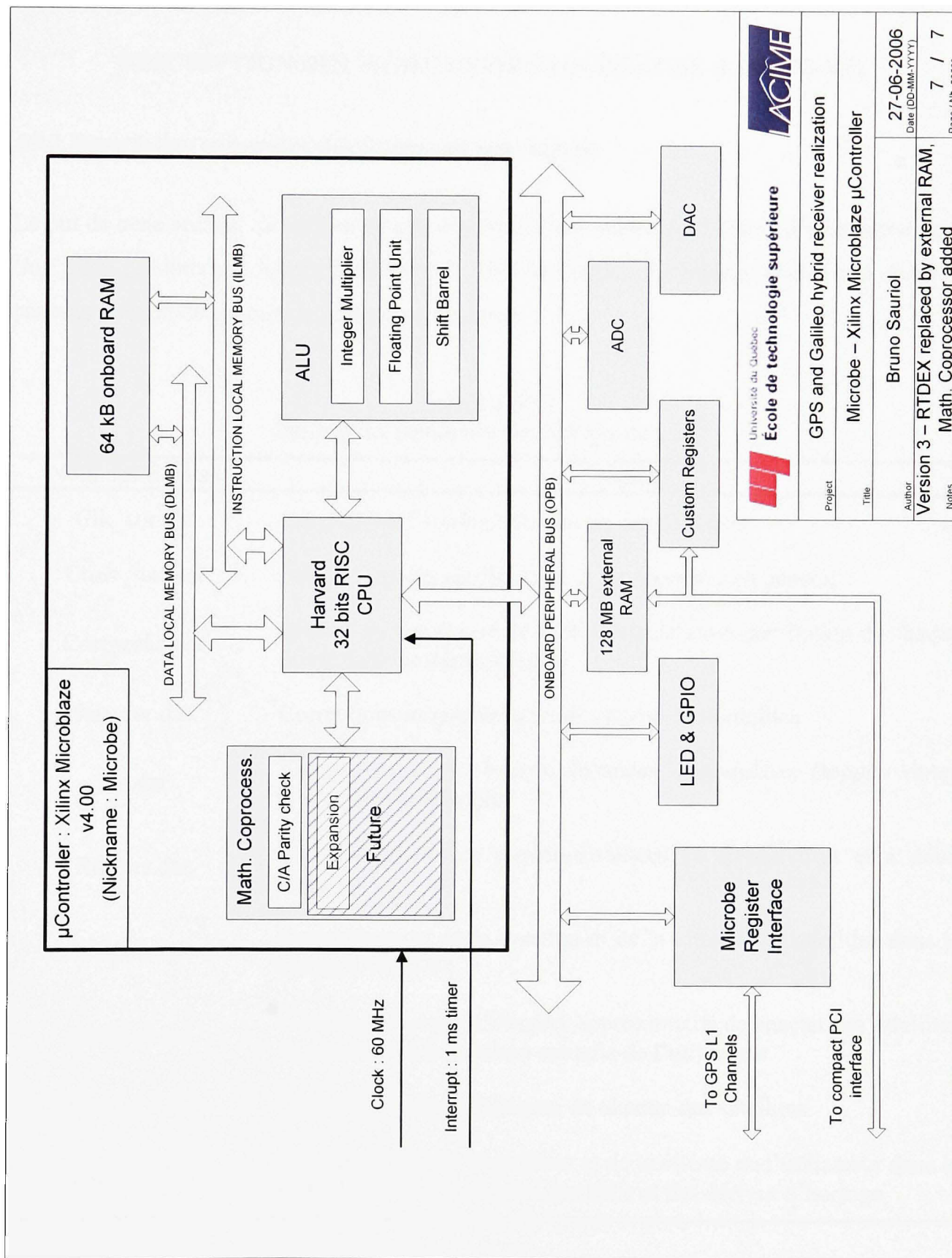
Project	GPS and Galileo hybrid receiver realization		
Title	Galileo E1 Receiver Architecture		
Author	Bruno Sauriol	Date (DD-MM-YYYY)	13-03-2007
Notes	6 Channels Galileo E1 Receiver Added	Page / Nb. pages	4 / 7





Université du Québec  
**Ecole de technologie supérieure**





## ANNEXE IX

### DESCRIPTION DES FICHIERS DE SAUVEGARDE (LOG FILES)

#### A9.1 Description sommaire des fichiers de sauvegarde

Le but de cette section est de fournir une description sommaire des fichiers d'enregistrements (*log files*) produits par le récepteur hybride GPS et Galileo. Le tableau ci-dessous décrit le contenu général de chacun de fichiers en question.

Tableau A9.1  
Description sommaire des fichiers de LOG

Nom des fichiers	Contenu
Clk_cor.dat	Corrections d'horloge de chacun des satellites
Chan_stat.dat	État des canaux de réception et du récepteur en général
Corrpeaks.dat	Mesure en temps réel de la fonction de cross-corrélation de chacun des canaux de démodulation
Ion_cor.dat	Corrections ionosphériques de chacun des satellites
Prs.dat	Enregistrement des pseudo-distances, <i>accumulated Doppler range</i> et des vitesses Doppler
Rawprs.dat	Enregistrement des pseudo-distances en format brut et à débit maximal
Satpos.dat	Enregistrement de la position et de la vitesse des satellites dans le système WGS-84
Sattrack.dat	Élévation, azimuth et Doppler approximatifs de chacun des satellites en référence à la position actuelle de l'utilisateur
Trop_cor.dat	Corrections troposphériques de chacun des satellites
Userpos.dat	Enregistrement de la position et de la vitesse de l'utilisateur dans le système WGS-84, en plus des biais et des dérives d'horloge

## A9.2 Description détaillée des fichiers de sauvegarde

Les fichiers de sauvegarde présentés ici ont été enregistrés en format binaire, ceci dans le but de minimiser leur taille et de maximiser leur vitesse de traitement. Les formats employés sont présentés au tableau A9.2.

Tableau A9.2  
Formats binaires employés

Formats de données	Acronymes employés	Taille (octets)
Entier non signé 8 bits	UINT8	1
Entier signé 32 bits	INT32	4
Entier non signé 32 bits	UINT32	4
Nombre à point flottant double précision 64 bits IEEE-754	DOUBLE	8

### A9.2.1 Description du fichier Chan\_stat.dat

- Quatre premiers octets : Nombre de canaux GNSS (N) sous format UINT32
- Octets suivants (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (secondes) – DOUBLE
  - Drapeau indiquant si l'algorithme de navigation a convergé – UINT8
  - Drapeau indiquant la synchronisation entre le PC et le FPGA – UINT8
  - Charge du CPU embarqué (%) – DOUBLE
  - N enregistrements consécutifs de 30 octets composés tels que suit et indiquant le statut des N canaux GNSS du récepteur :
    - #PRN (le #64 indique un canal désactivé) – UINT32
    - $C/N_0$  (dB Hz) - DOUBLE
    - Niveau K d'acquisition du canal (UINT8), où K représente :
      - 0) Recherche
      - 1) Phase initiale de poursuite
      - 2) Poursuite / synchronisation aux bits de données
      - 3) Poursuite / synchronisation au message de navigation
      - 4) Poursuite / téléchargement du message de navigation



- Nombre d'échecs à l'acquisition – UINT8
- Drapeau « *PLL Locked* » – UINT8
- Drapeau « *SV bad* » – UINT8
- Drapeau « *SV Unhealthy* » – UINT8
- Drapeau « *SV time valid* » – UINT8
- Semaine propre au satellite – UINT32
- Temps propre au satellite (s) – DOUBLE

#### **A9.2.2 Description des fichiers Clk\_cor, Ion\_cor et Trop\_cor.dat**

- Quatre premiers octets : Nombre de canaux GNSS (N) sous format UINT32
- Octets suivants (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (secondes) – DOUBLE
  - N enregistrements consécutifs de 8 octets composés tels que suit et contenant la correction appropriée (horloge, ionosphérique ou troposphérique) pour N canaux GNSS du récepteur :
    - Correction (m) – DOUBLE

#### **A9.2.3 Description du fichier Corrpeaks.dat**

- Le fichier est composé d'une série d'enregistrements consécutifs de 521 octets de longueur. Chacun des enregistrements se compose comme suit :
  - Temps de référence (secondes) – DOUBLE
  - Numéro de canal GNSS – UINT8
  - Fonction de corrélation – 128 x INT32

#### **A9.2.4 Description du fichier Prs.dat**

- Quatre premiers octets : Nombre de canaux GNSS (N) sous format UINT32
- Octets suivants (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (secondes) – DOUBLE

- N enregistrements consécutifs de 24 octets composés tels que suit :
  - Pseudo-distance pour chaque canal (m) - DOUBLE
  - Vitesses relatives pour chaque canal (m/s) - DOUBLE
  - ADR pour chaque canal (m) - DOUBLE

#### **A9.2.5 Description du fichier Rawprs.dat**

- Quatre premiers octets : Nombre de canaux GNSS (N) sous format UINT32
- Octets suivants (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (millisecondes) – UINT32
  - N enregistrements consécutifs de 12 octets composés tels que suit :
    - Compteur d'époque (ms) – UINT32
    - Numéro de bribe – UINT32
    - Phase du NCO de code (valeur machine) – UINT32

#### **A9.2.6 Description du fichier Satpos.dat**

- Quatre premiers octets : Nombre de canaux GNSS (N) sous format UINT32
- Octets suivants (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (secondes) – DOUBLE
  - N enregistrements consécutifs de 48 octets composés tels que suit et contenant la position ainsi que la vitesse des satellites verrouillés par les N canaux :
    - Position en x (WGS-84, en m) - DOUBLE
    - Position en y (WGS-84, en m) - DOUBLE
    - Position en z (WGS-84, en m) - DOUBLE
    - Vitesse en x (WGS-84, en m/s) - DOUBLE
    - Vitesse en y (WGS-84, en m/s) - DOUBLE
    - Vitesse en z (WGS-84, en m/s) - DOUBLE

### **A9.2.7 Description du fichier Sattrack.dat**

- Tous les octets (se répètent pour chaque échantillon consécutif) :
  - Temps de référence (secondes) – DOUBLE
  - 32 enregistrements consécutifs de 26 octets composés tels que suit et contenant les informations suivantes au sujet des satellites affectés aux 32 numéros de PRN GPS L1 valides :
    - Élévation du satellite (degrés) – DOUBLE
    - Azimut du satellite (degrés) – DOUBLE
    - Doppler du satellite (Hz) – DOUBLE
    - Drapeau almanach valide – UINT8
    - Drapeau éphéméride valide – UINT8

### **A9.2.8 Description du fichier Userpos.dat**

- Tous les octets (se répètent pour chaque échantillon consécutif) :
  - Temps GPS local (s) – DOUBLE
  - Position de l'utilisateur en x (WGS-84, en m) – DOUBLE
  - Position de l'utilisateur en y (WGS-84, en m) – DOUBLE
  - Position de l'utilisateur en z (WGS-84, en m) – DOUBLE
  - Vitesse de l'utilisateur en x (WGS-84, en m/s) – DOUBLE
  - Vitesse de l'utilisateur en y (WGS-84, en m/s) – DOUBLE
  - Vitesse de l'utilisateur en z (WGS-84, en m/s) – DOUBLE
  - Biais d'horloge (m) – DOUBLE
  - Dérive d'horloge (m/s) – DOUBLE
  - Accélération lissée ( $\text{m/s}^2$ ) – DOUBLE
  - Ondulation du géoïde (EGM-96, en m) – DOUBLE



## BIBLIOGRAPHIE

- Antcom Corporation. 2001. « Active or Passive L1/L2 GPS/Omnistar Antennas, P/N: 5GO1215A-XT-X ». Torrance, Californie: [http://www.canalgeomatics.com/product\\_files/5GO1215A\\_94.pdf](http://www.canalgeomatics.com/product_files/5GO1215A_94.pdf). Consulté le 18 octobre 2007.
- ARINC Incorporated. 2001. *NAVSTAR GPS Space Segment / User segment L5 Interfaces, Interface Control Document (ICD)*, ICD-GPS-705. El Segundo, California: ARINC Research Corporation, 70 p.
- ARINC Research Corporation. 2000. *NAVSTAR GPS Space Segment/Navigation User Interface, Interface Control Document (ICD)*, ICD-GPS-200, Rev. C. El Segundo, California: ARINC Research Corporation, 138 p.
- Best, Roland E. 2003. *Phase-Locked Loops: Design, Simulation and Application*, 5th Edition. McGraw-Hill, 421 p.
- Betz, John W., Mario A. Blanco, Charles R. Cahn, Philip A. Dafesh, Christopher J. Hegarty, Kenneth W. Hudnut, Vidapa Kasemri, Richard Keegan, Karl Kovach, Lawrence S. Lenahan, Howard H. Ma, Joseph J. Rushanan, Dean Sklar, Thomas A. Stansell, Charles C. Wang et Soon K. Yi. 2007. « Enhancing the Future of Civil GPS, Overview of L1C Signal ». Inside GNSS, vol. 2, n° 3 (printemps), p. 42-49.
- Borre, K. 2006. *A software-defined GPS and Galileo receiver : a single-frequency approach*. Coll. « Applied and numerical harmonic analysis ». Boston: Birkhauser, xxi, 176 p. p.
- Brown, Robert Grover, et Patrick Y. C. Hwang. 1997. *Introduction to random signals and applied Kalman filtering with Matlab exercises and solutions*, 3rd. New York, N.Y.: J. Wiley and Sons, xi, 484 p.
- Bullock, J. Blake, G. Jeffrey Geier, T. Michael King et Howard L. Kennedy. 1996. « Effectiveness of Two Satellite Tracking in Urban Canyon Environments ». In *Proceedings of the 9th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS 1996* (Sep 17-20, Kansas City, Mo). p. 1699-1707.
- Chavagnieux, Cédric. 2006. « Développement d'un récepteur hybride GPS/Galileo en environnement réel ». Montréal, École de technologie supérieure, 178 p.
- Cloutier, Mark, Theodore Varelas, Christian Cojocar et Florinel Balteanu. 1999. « 4-dB NF GPS receiver front-end with AGC and 2-b A/D ». *Proceedings of the Custom Integrated Circuits Conference*, p. 205-208.

- Constantinescu A., et R.Jr Landry. 2005. « GPS/GALILEO/GLONASS Hybrid Satellite Constellation Simulator – GPS Constellation Validation and Analysis ». In *Proceedings of the 61st Annual Meeting of the Institute of Navigation ION AM 2005* (Jun 27-29, Cambridge, Massachusetts ). p. 733-744.
- Constantinescu A., R.Jr Landry et I. Ilie. 2004. « Performance Evaluation and Analysis of a Hybrid Version of a Software Defined GPS/Galileo GNSS Receiver for Dynamic Scenarios ». In *Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2004* (Sep 21-24, Long Beach, California). p. 2697-2708
- Cook, Gerald L. 1997. « GLONASS Performance, 1995-1997, and GPS-GLONASS Interoperability Issues ». *Journal of the Institute of Navigation*, vol. 44, n° 3, p. 291-300.
- Coordination Scientific Information Center. 2002. *GLONASS Interface Control Document*, Ver. 5. Mosou: Coordination Scientific Information Center, 55 p.
- Coordination Scientific Information Center. 2007. « GLONASS Constellation Status ». In *The Information-Analytical Center*. <<http://www.glonass-ianc.rsa.ru>>. Consulté le 10 août 2007.
- Deergha Rao, K., M. N. S. Swamy et E. I. Plotkin. 2004. « GPS navigation with increased immunity to modeling errors ». *Aerospace and Electronic Systems*, IEEE Transactions on, vol. 40, n° 1, p. 2-11.
- DoD. Septembre 1996. *NAVSTAR GPS User Equipment Introduction (Public Release Version)*. GPS Joint Program Office, 215 p.
- Eastwood, Raymond A. . 1990. « An Integrated GPS/GLONASS Receiver ». In *Proceedings of the 1990 National Technical Meeting of the Institute of Navigation* (San Diego, CA, January 23-25 1990). p. 37-43.
- Emmott, Norman W. 2006. « Capt. P.V.H. Weems, The Grand Old Man of Navigation ». *ION Newsletter*, vol. 16, n° 2-3 (été-automne), p. 8-9;4-7.
- European Comission. 2007. *United States and the European Union announce final design for GPS-Galileo common civil signal*. Press Release IP/07/1180. Bruxelles, 1 p. In *Europa Rapid Press Releases*.
- European Comission, et European Space Agency. 2002. *Mission High Level Definition*. Ver. 3.0. 51 p.  
<[http://ec.europa.eu/dgs/energy\\_transport/galileo/documents/technical\\_en.htm](http://ec.europa.eu/dgs/energy_transport/galileo/documents/technical_en.htm)>. Consulté le 15 août 2007.



- European Space Agency. 2006. *Galileo Open Service Signal In Space Interface Control Document (OS SIS ICD)*, GAL OS SIS ICD/D.0. Bruxelles: Galileo Joint Undertaking, 192 p.
- European Space Agency. 2007. « GIOVE-A transmits first navigation message ». In *News*. <[http://www.esa.int/esaNA/SEM0QSU681F\\_galileo\\_0.html](http://www.esa.int/esaNA/SEM0QSU681F_galileo_0.html)>. Consulté le 15 août 2007.
- European Space Research and Technology Center. 2007. *GIOVE-A Navigation/Signal-In-Space Interface Control Document*, ESA-DEUI-NG-ICD/02703. Noordwijk, The Netherlands: European Space Research and Technology Center, 46 p.
- FAA William J. Hughes Technical Center. 2006. *Wide-Area Augmentation System Performance Analysis Report*. 17. Atlantic City, NJ: Federal Aviation Administration, 95 p.
- FAA William J. Hughes Technical Center. 2007. *Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report*. 57. Atlantic City, NJ: Federal Aviation Administration, 40 p.
- Federal Aviation Administration. 2007. « Satellite Based Augmentation System News ». In *Federal Aviation Administration*. <[http://www.faa.gov/about/office\\_org/headquarters\\_offices/ato/service\\_units/techops/navservices/gnss/waas/news/](http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/waas/news/)>. Consulté le 21 août 2007.
- Fortin, André. 2001. *Analyse numérique pour ingénieurs*. Montréal: Éditions de l'École Polytechnique de Montréal, x, 448 p.
- Gaisler Research. 2007. *Gaisler Research Website*. <<http://www.gaisler.com>>. Consulté le 17 septembre 2007.
- Gerakoulis, Diakoumis, et Evaggelos Geraniotis. 2001. *CDMA: Access and Switching for Terrestrial and Satellite Networks*. Chichester (UK): John Wiley & Sons, 270 p.
- Goldsmith, Andrea. 2005. *Wireless communications*. Cambridge ; New York: Cambridge University Press, xxviii, 644 p. p.
- GPS Networking. 2007. *GPS Antenna Splitters*. <<http://gpsnetworking.com/GPS-antenna-splitters.asp>>. Consulté le 12 octobre 2007.
- Grewal, Mohinder S., et Angus P. Andrews. 2001. *Kalman filtering theory and practice using MATLAB*, 2nd. New York, N.Y.: J. Wiley and Sons, xiii, 401 p.



- Grewal, Mohinder S., Lawrence R. Weill et Argus P. Andrews. 2001. *Global positioning systems, inertial navigation, and integration*. New York, N.Y.: J. Wiley and Sons, xix, 392 p.
- Hein, Günter W., Jose-Angel Avila-Rodríguez, Stefan Wallner, John W. Betz, Christopher J. Hegarty, Joseph J. Rushanan, Andrea L. Kraay, Anthony R. Pratt, Lawrence S. Lenahan, John Owen, Jean-Luc Issler et Thomas A. Stansell. 2006. « MBOC: The New Optimized Spreading Modulation Recommended for Galileo L1 OS and GPS L1C ». *Inside GNSS*, vol. 1, n° 4 (mai/juin), p. 57-66.
- Inside GNSS. 2006. « 360 Degrees, Compass: And China's GNSS Makes Four ». vol. 1, n° 8 (novembre/décembre), p. 14.
- Inside GNSS. 2007. « 360 Degrees, Galileo's Plan B (and C) ». vol. 2, n° 4 (mai/juin), p. 16.
- International GNSS Service. 2007. *IGS Products*. <http://igscb.jpl.nasa.gov/components/prods.html>. Consulté le 5 juillet 2007.
- Julien, Olivier, Christophe Macabiau, Jean-Luc Issler et Lionel Ries. 2007. « Two for One, Tracking Galileo BOC Signal with TMBOC ». *Inside GNSS*, (printemps), p. 50-57.
- Kaplan, Elliott D. 1996. *Understanding GPS principles and applications*. Coll. « Artech House mobile communications library ». Boston, Mass.: Artech House, xv, 554 p.
- Knight, Mark. 1998. *Ionospheric Effects on Global Positioning System Receivers*. Coll. « AD-a342 594. ». [United States]: defence science and technology organisation canberra (australia), 105 p.
- Kogure, S., M. Sawabe et M. Kishimoto. 2006. « Status of QZSS Navigation System in Japan ». In *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2006* (Sep 26-29). p. 2092-2102. Fort Worth, Texas.
- Kosenko, Victor, Vladimir Bartenev et Victor Chebotarev. 2007. « Replenishment of the GLONASS Constellation, Key Task of the Russian Space Industry ». *Inside GNSS*, vol. 2, n° 2 (mars/avril), p. 20-21;73.
- Leveson, Irving. 2006. « Benefits of the New GPS Civil Signal, The L2C Study ». *Inside GNSS*, vol. 1, n° 5 (juillet/août), p. 42-47;56.
- Lo, Sherman, Alan Chen, Per Enge, Grace Gao, Denis Akos, Jean-Luc Issler, Lionel Ries, Thomas Grelier et Joel Dantepal. 2006. « GNSS Album, Images and Spectral Signatures of the New GNSS Signals ». *Inside GNSS*, vol. 1, n° 4 (mai/juin), p. 46-56.

- Lyrtech. 2007. *Site web de Lyrtech*. <<http://www.lyrtech.com>>. Consulté le 28 août 2007.
- MacGougan, Glenn, Per Ludvig Normark et Christian Stahlberg. 2005. « GNSS Evolution: The Software Receiver ». *GPS World*, (Jan), p. 48-55.
- Maybeck, Peter S. 1979. *Chapter 1, "Introduction" from Stochastic models, estimation, and control*, 1. New York: Academic Press, 16 p.
- Misra, Pratap, et Per Enge. 2006. *Global positioning system : signals, measurements, and performance*, 2e édition. Lincoln, Mass.: Ganga-Jamuna Press, 570 p.
- NASA GSFC, et NIMA. 2004. *EGM96, The NASA GSFC and NIMA Joint Geopotential Model*. <<http://cddis.nasa.gov/926/egm96/egm96.html>>. Consulté le 6 juillet 2007.
- Novatel. 2007a. *Antennas GPS Devices, GPS Antennas and Receivers: NovAtel Inc. - Products*. <<http://www.novatel.com/products/antennas.htm>>. Consulté le 19 octobre 2007.
- Novatel. 2007b. *Engines GPS Devices, GPS Antennas and Receivers: NovAtel Inc. - Products*. <<http://www.novatel.com/products/engines.htm>>. Consulté le 17 juillet 2007.
- Onidi, Olivier 2002. *Galileo is launched*. Bruxelles: European Commission, 4 p. <[http://ec.europa.eu/dgs/energy\\_transport/galileo/doc/galileo\\_is\\_launched.pdf](http://ec.europa.eu/dgs/energy_transport/galileo/doc/galileo_is_launched.pdf)>. Consulté le 15 août 2007.
- Parkinson, Bradford W., et James J. Spilker. 1996. *The global positioning system : theory and applications*. Coll. « Progress in astronautics and aeronautics ; v. 163-164 ». Washington, DC: American Institute of Aeronautics and Astronautics.
- Penna, N., A. Dodson et Chen Wu. 2001. « Assessment of EGNOS tropospheric correction model ». *Journal of Navigation*, vol. 54, n° 1, p. 37-55.
- Proakis, John G. 2001. *Digital communications*, 4th. New York, N.Y.: McGraw-Hill, xxi, 1002 p.
- Sauriol, Bruno. 2006. *Optimisation d'un filtre de Kalman appliqué à la résolution de la position par satellite GPS*. Coll. « Rapport de projet synthèse présenté dans le cadre du cours SYS833, signaux et systèmes numériques ». Montréal, 30 p.
- Sauriol, Bruno, et R.Jr Landry. 2007. « FPGA-Based Architecture for High Throughput, Flexible and Compact Real-Time GNSS Software Defined Receiver ». In *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation* (Jan 22-24, San Diego, California). p. 708 - 717.



- Simsky, Andrew. 2006. « Three's the Charm, Triple-Frequency Combinations in Future GNSS ». Inside GNSS, vol. 1, n° 5 (juillet/août), p. 38-41.
- Sleewaegen, J.-M., W. De Wilde et Hollreiser M. 2006. « Galileo AltBOC Receiver ». In *ENC-GNSS 2004* (May 17, Rotterdam).
- Spiegel, Murray R., et John Liu. 2000. *Formules et tables de mathématiques*, deuxième édition. Coll. « Schaum's ». New York: McGraw-Hill, viii, 278 p.
- Spirent Communications. 2006. « Spirent GSS7700 Multi-Channel High Dynamic GPS/SBAS Simulation System ». Paignton, Angleterre: <<http://www.spirentcom.com>>. Consulté le 14 octobre 2007.
- Stanford Research Systems. 2003. *CG635 Synthesized Clock Generator*. <<http://www.thinksrs.com/products/CG635.htm>>. Consulté le 29 août 2007.
- Stansell, Tom. 2006. « Boc or MBOC ? The Common GPS/Galileo Civil Signal Design, A Manufacturers Dialog, Part 1 ». Inside GNSS, vol. 1, n° 5 (juillet/août), p. 30-37.
- The Institute of Navigation. 2004. *ION Virtual Navigation Museum*. <<http://www.ion.org/museum/>>. Consulté le 20 août 2007.
- The Institute of Navigation. 2007. « GNSS Program Updates ». The Institute of Navigation Newsletter, vol. 16, n° 4 (hiver 2006/2007), p. 14-16.
- Tsui, James Bao-Yen. 2000. *Fundamentals of global positioning system receivers a software approach*. Coll. « Wiley series in microwave and optical engineering ». New York, N.Y.: J. Wiley and Sons, xvii, 238 p.
- Turkowski, Ken. 1995. « Fixed-Point Square Root ». In *Graphics gems V*, sous la dir. de Paeth, Alan W. p. 22-24. Boston: AP Professional.
- U.S. Department of Homeland Security. 2007. « United States Coast Guard Navigation Center ». In *GPS Almanac Information*. <<http://www.navcen.uscg.gov/gps/almanacs.htm>>. Consulté le 19 octobre 2007.
- U.S. Naval Observatory. 2007. « Current GPS Constellation ». In *Time Service Department*. <<http://tycho.usno.navy.mil/gpscurr.html>>. Consulté le 30 novembre 2007.
- Ward, Phillip W. 2004. « A Design Technique to Remove the Correlation Ambiguity in Binary Offset Carrier (BOC) Spread Spectrum Signals ». In *Proceedings of the 2004 National Technical Meeting of the Institute of Navigation* (Jan 26-28). p. 886-896. San Diego, California.



- Wi-Sys Communications Inc. 2003. « RF Modules ». In *Wi-Sys: GPS antenna - Embedded GPS antenna - GPS tracking - Mobile Data*. <[http://www.wi-sys.com/products/accessories/rf\\_modules.php](http://www.wi-sys.com/products/accessories/rf_modules.php)>. Consulté le 15 octobre 2007.
- Wikipedia. 2007a. « Circular error probable ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Circular\\_error\\_probable](http://en.wikipedia.org/wiki/Circular_error_probable)>. Consulté le 10 novembre 2007.
- Wikipedia. 2007b. « Rayleigh distribution ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Rayleigh\\_distribution](http://en.wikipedia.org/wiki/Rayleigh_distribution)>. Consulté le 5 septembre 2007.
- Wikipedia. 2007c. « Reference ellipsoid ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Reference\\_ellipsoid](http://en.wikipedia.org/wiki/Reference_ellipsoid)>. Consulté le 18 juillet 2007.
- Wikipedia. 2007d. « Rice distribution ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Rice\\_distribution](http://en.wikipedia.org/wiki/Rice_distribution)>. Consulté le 5 septembre 2007.
- Wikipedia. 2007e. « Standard deviation ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Standard\\_deviation](http://en.wikipedia.org/wiki/Standard_deviation)>. Consulté le 18 juillet 2007.
- Wikipedia. 2007f. « Wide Area Augmentation System ». In *Wikipedia, the free encyclopedia*. <[http://en.wikipedia.org/wiki/Wide\\_Area\\_Augmentation\\_System](http://en.wikipedia.org/wiki/Wide_Area_Augmentation_System)>. Consulté le 14 août 2007.
- Williams, J. E. D. 1992. *From sails to satellites : the origin and development of navigational science*. Oxford ; New York: Oxford University Press, ix, 310 p. p.
- Won, Jong-Hoon, Thomas Pany et Günter W. Hein. 2006. « GNSS Software Defined Radio: Real Receiver or Just a Tool for Experts? ». Inside GNSS, vol. 1, n° 5 (juillet/août), p. 48-56.
- Xilinx. 2005a. « MicroBlaze Processor Reference Guide ». <<http://www.xilinx.com>>. Consulté le 17 septembre 2007.
- Xilinx. 2005b. « Virtex-4 User Guide ». <<http://www.xilinx.com>>. Consulté le 17 septembre 2007.