

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
M.Ing.

PAR
Ahmed BEN SALAH

CONCEPTION ET MISE EN ŒUVRE D'UNE ARCHITECTURE LOGICIELLE POUR
L'AMÉLIORATION DES PERFORMANCES D'UN SIMULATEUR GNSS ET
L'INTÉGRATION DE MODÈLES MULTITRAJETS

MONTREAL, LE 3 MAI 2010

© Ahmed Ben Salah, 2010

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Ammar Kouki, directeur de mémoire
Département de Génie électrique à l'École de technologie supérieure

M. René Jr. Landry, président du jury
Département de Génie électrique à l'École de technologie supérieure

M. Jocelyn Dore, examinateur externe
AGENCE SPATIALE CANADIENNE

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

LE 9 AVRIL 2010

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je remercie tout d'abord mon directeur, Prof. Ammar Kouki, pour m'avoir accueilli dans son laboratoire, de m'avoir procuré toutes les conditions favorables à l'accomplissement de mon projet et de m'avoir donné cette opportunité de découvrir les systèmes de positionnement par satellites.

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 Rami Loukil et Yan Ding, les membres du projet Simulateur GNSS, avec qui la collaboration a été très intéressante que ce soit d'un côté professionnel ou humain. Je remercie aussi tous mes amis du Lacime pour leur bonne humeur habituelle et leur soutien.

Finalement, je tiens spécialement à remercier mes parents et mon frère pour leur support inconditionnel tout au long de mes études.

MISE EN ŒUVRE D'UNE NOUVELLE ARCHITECTURE LOGICIELLE POUR L'AMÉLIORATION DE PERFORMANCES D'UN SIMULATEUR GNSS ET L'INTÉGRATION DE MODÈLES MULTI-TRAJETS

BEN SALAH, Ahmed

RÉSUMÉ

Le positionnement par satellite connaît l'une des périodes les plus importantes depuis son avènement à la fin du 20^{ème} siècle. En effet, outre le système GPS, de nouvelles constellations sont en train de voir le jour, tel que Galileo le système de positionnement européen, ou encore le chinois Beidou (Système de navigation chinois), et le projet russe GLONASS. Tous ces systèmes regroupés sous le nom de GNSS (*Global Navigation Satellite System*) permettront de développer des applications de plus en plus précises et fiables. Ceci ne peut se concrétiser sans l'intégration des différents signaux GNSS.

Outre le défi de développer des récepteurs et de les intégrer dans différents autres appareils sans fil, comme les téléphones par exemple, il faut souligner l'importance de développer des simulateurs capables aussi de générer les signaux satellites permettant d'effectuer les tests de fiabilité, de précision et de robustesse nécessaires au niveau des récepteurs.

Ce mémoire présente donc la mise en œuvre d'une nouvelle architecture logicielle pour le simulateur GNSS, qui permet de produire en temps réel des scénarios de tests pour des récepteurs hybrides GPS/Galileo. Le simulateur se compose de trois parties : le logiciel, la partie de traitement en temps réel des signaux numériques et la partie analogique.

Notre intérêt dans ce mémoire va donc se porter sur la partie logicielle du simulateur. En effet, nous avons travaillé d'abord sur la validation des signaux générés, pour ensuite obtenir des positions correctes au niveau du récepteur de test. Ensuite, nous avons réussi à mettre en place une nouvelle architecture logicielle qui remédie aux nombreuses limitations de la première version. Cette première qui a été développée par une ancienne équipe du LACIME présentait en effet des limitations logicielles qui ne permettaient pas une évolution qualitative du simulateur.

Les avantages d'une telle manœuvre consistent essentiellement au développement d'une architecture logicielle de haute qualité, très flexible au niveau de la conception, réutilisable, facile à tester et concurrentiel au niveau des coûts. L'accomplissement du travail au niveau du logiciel nous permet donc d'aborder un autre objectif du projet qui consiste à étudier et modéliser l'effet des multi-trajets dans les signaux satellites. Cette modélisation a été effectuée et testée en mettant en place des scénarios, et en vérifier les effets au niveau des récepteurs de test utilisés.

Mots clés : GNSS, architecture, logiciel, multi-trajets.

DEVELOPMENT OF A GNSS SIMULATOR NEW SOFTWARE ARCHITECTURE FOR PERFORMANCES IMPROVEMENT AND MULTIPATH MODELS INTEGRATION

BEN SALAH, Ahmed

ABSTRACT

Satellite positioning technology is getting more attention than any other time in its history that begun at the end of the 20th century. In addition to GPS, new constellations are appearing like Galileo the European satellite system, the Chinese Beidou and GLONASS the Russian constellation. All of these positioning systems, called GNSS (*Global Navigation Satellite System*) will help to develop new applications which are more and more precise and reliable. The integration of different signals represents an important issue that can lead to those achievements.

Developing new types of receivers and integrating them in wireless applications is one of the biggest issues nowadays, but we have to consider the importance of developing GNSS simulators that allow us to generate real satellite signals, so we can test the reliability, the accuracy and the robustness of new developed receivers.

This thesis focuses on the development of new software architecture for a GNSS simulator. This GNSS simulator is used for the generation of real time scenarios in testing hybrid GPS/Galileo Receivers. The simulator project is divided into three parts: the software part, the real time signal processing part and the RF part.

The prime focus of this thesis is on the software part. Firstly, we validated the computations and the signals generated, so as to obtain accurate positions from the receiver. After that, we developed a new software architecture that is much better than the first software version.

Our purpose is to have high quality software, which has a flexible architecture, easy to test and at a competitive cost. The last part of the project focuses on modeling multipath with our GNSS simulator.

Key words: GNSS, architecture, software, multipath.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 Étude des systèmes GPS et Galileo	4
1.1 Principe de radionavigation	4
1.2 Le système GPS	6
1.2.1 Structure du système GPS.....	7
1.2.2 Signaux GPS et leur modernisation	7
1.3 Le système Galileo.....	10
1.3.1 Présentation du projet Galileo.....	10
1.3.2 Particularité des signaux Galileo	11
1.4 Le Simulateur de constellation GPS/Galileo : Premier aperçu.....	12
1.4.1 Partie logicielle	13
1.4.2 Partie traitement en temps réel des signaux numériques	15
1.4.3 Partie analogique.....	17
1.5 Conclusion	18
CHAPITRE 2 La simulation d'une constellation de satellites GPS/Galileo.....	19
2.1 Scénario de simulation d'une constellation	19
2.1.1 Choix de la constellation et configuration des paramètres orbitaux	19
2.1.2 Date et durée de la simulation.....	25
2.1.3 Configuration des paramètres atmosphériques	27
2.1.4 Choix de la trajectoire du récepteur	29
2.2 Algorithmes de calculs au niveau du simulateur	31
2.2.1 Description du processus de calcul effectués dans la DLL Matlab	31
2.2.2 Calcul des positions des satellites	33
2.2.3 Calcul de la position du récepteur.....	34
2.2.4 Calcul des distances géométriques.....	35
2.2.5 Calcul de l'élévation et de l'azimut des satellites.....	35
2.2.6 Calcul des biais d'horloges	38
2.2.7 Calcul des délais ionosphériques et troposphériques.....	39
2.2.8 Calcul de la pseudo-distance.....	42
2.2.9 Calcul de l'effet Doppler	43
2.2.10 Mise à jour des paramètres de navigation :.....	44
2.3 Processus de génération des signaux	45
2.4 Validation des résultats pour GPS L1	50
2.5 Limitations du simulateur	61
2.6 Conclusion	63
CHAPITRE 3 Nouvelle architecture logicielle du simulateur	64
3.1 Analyse des besoins du nouveau logiciel.....	64
3.2 Conception du logiciel	67

3.2.1	Architecture logicielle proposée	67
3.2.2	Simulation de la constellation GPS L1	71
3.3	Implémentation	74
3.4	Validation des résultats pour la nouvelle architecture	78
3.5	Positionnement du simulateur dans la chaîne de l'innovation.....	84
3.6	Conclusion	85
CHAPITRE 4 Étude d'un modèle de multi-trajets pour les signaux GNSS		86
4.1	Introduction sur les multi-trajets	86
4.1.1	Le modèle de Rayleigh	88
4.1.2	Le modèle de Rice	89
4.2	Le positionnement GPS dans un contexte de multi-trajets	91
4.2.1	Sensibilité des récepteurs GPS aux multi-trajets	91
4.2.2	Modèles de multi-trajets utilisés dans Spirent	93
4.3	Modèle de génération de multi-trajets utilisé.....	96
4.3.1	Architecture générale du modèle	97
4.3.2	Implémentation	98
	Générateur de multi-trajets manuel.....	98
	Générateur d'évanouissement Rayleigh et Rice	100
4.3.3	Tests et résultats.....	102
4.3.4	Modélisation optimisée des multi-trajets.....	106
4.3.5	Conclusion	109
CONCLUSION.....		111
ANNEXE I Calcul des positions des satellites		113
ANNEXE II Paramètres générés par la DLL MATLAB.....		115
ANNEXE III Message de navigation Galileo.....		118
ANNEXE IV Schéma détaillé de la partie IF		120
ANNEXE V Résultats de la DLL MATLAB		122
ANNEXE VI Diagramme des classes de la nouvelle architecture		127
ANNEXE VII Interface graphique de la nouvelle architecture logicielle		131
BIBLIOGRAPHIE.....		141

LISTE DES TABLEAUX

		Page
Tableau 2.1	Format YUMA.....	22
Tableau 2.2	Modèles de calcul des délais ionosphériques.....	40
Tableau 2.3	Modèles de calcul des délais troposphériques	41
Tableau 2.4	Types de messages de navigation Galileo	48
Tableau 3.1	Indices des priorités des nouvelles fonctionnalités.....	66
Tableau 3.2	Implémentation des nouvelles fonctionnalités.....	75
Tableau 4.1	Configuration manuelle	95
Tableau 4.2	Utilisation des modèles statistiques	96

LISTE DES FIGURES

	Page
Figure 1.1	Principe de mesure de la distance.4
Figure 1.2	Principe de trilatération.....5
Figure 1.3	Le signal GPS L1.....8
Figure 1.4	Évolution des signaux GPS.....9
Figure 1.5	Le signal GPS L5.....9
Figure 1.6	Spectres utilisés par le système Galileo.....12
Figure 1.7	Interface graphique de la partie Matlab.....14
Figure 1.8	Interface graphique de la partie C++.net.....15
Figure 1.9	Description haut niveau de la partie IF. Tirée de Sauriol (2007, p. 3).....17
Figure 2.1	Configuration des paramètres orbitaux.....20
Figure 2.2	Configuration des paramètres de correction d'orbites.....23
Figure 2.3	Paramètres de correction des horloges des satellites.....25
Figure 2.4	Configuration du temps de simulation.....26
Figure 2.5	Configuration des paramètres ionosphériques.....28
Figure 2.6	Fenêtre de configuration de la trajectoire du récepteur.....29
Figure 2.7	Diagramme des opérations effectuées au niveau de la DLL Matlab.....32
Figure 2.8	Affichage de la constellation(Skyplot) pour le ' <i>Scénario_Test</i> '.....38
Figure 2.9	Évolution temporelle de la pseudo-distance.....43
Figure 2.10	Processus de génération des signaux dans la partie logicielle.....46
Figure 2.11	Structure des sous-trames du message de navigation GPS.....47
Figure 2.12	Banc de test.....50
Figure 2.13	Comparaison des pseudo-distances (Satellite 2) pour le ' <i>Scénario_Test</i> '.52

Figure 2.14	Erreur de la pseudo-distance du satellite 2.	53
Figure 2.15	Valeurs des pseudo-distances pour le satellite 2.....	58
Figure 2.16	Erreur de calcul de la pseudo-distance.....	58
Figure 2.17	Position dans le récepteur de CMC.....	59
Figure 2.18	Niveau d'erreur de la position (Scénario 2).....	61
Figure 3.1	Étapes de développement d'un logiciel.	67
Figure 3.2	Schéma bloc de l'architecture logicielle.	68
Figure 3.3	Description fonctionnelle de l'architecture logicielle générale.	69
Figure 3.4	Architecture logicielle globale utilisant les extensions.....	70
Figure 3.5	Composantes utilisées dans la simulation de la constellation GPS L1.....	71
Figure 3.6	Composantes utilisées dans la simulation de la constellation Galileo E1.	72
Figure 3.7	Diagramme des classes de la librairie de calcul GPS L1.....	74
Figure 3.8	Affichage en temps réel.	76
Figure 3.9	Trajectoire du récepteur avec l'application Google Earth.	77
Figure 3.10	Fenêtre d'affichage du 'SkyPlot'.	77
Figure 3.11	Banc de test du simulateur avec la nouvelle architecture logicielle.	79
Figure 3.12	Pourcentage d'erreur dans la pseudo-distance entre l'ancienne et la nouvelle version du simulateur.	81
Figure 3.13	Position au niveau du récepteur CMC (6 minutes de simulation).	82
Figure 3.14	Chaine de l'innovation d'un produit technologique.	85
Figure 4.1	Distribution de Rayleigh.....	89
Figure 4.2	Distribution de Rice pour $\sigma = 1$	90
Figure 4.3	Fonction de corrélation pour un multi-trajet.....	92
Figure 4.4	Architecture générale du modèle multi-trajets.....	97
Figure 4.5	Modélisation du générateur de multi-trajets.	99

Figure 4.6 Interface graphique du générateur de modèles de multi-trajets.100

Figure 4.7 Générateur d'évanouissement Rayleigh/Rice.101

Figure 4.8 Architecture optimisée pour le générateur d'évanouissement.102

Figure 4.9 Variation de la position après 6 minutes de simulation (Délai=150m). ..103

Figure 4.10 Pic de corrélation pour le satellite 20.....104

Figure 4.11 Pic de corrélation pour le satellite 17.....104

Figure 4.12 Variation de la position après 6 minutes de simulation (Délai = 300m). 105

Figure 4.13 Pic de corrélation du satellite 17.....106

Figure 4.14 Pic de corrélation du satellite 4.....106

Figure 4.15 Optimisation de l'utilisation des canaux pour les multi-trajets.109

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

2D	Deux dimensions
3D	Trois Dimensions
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
CMC	Compagnie Marconi Canada (Actuellement Esterline)
CRC	Cyclic redundant code
CS	Commercial Service
C/A	Coarse acquisition
DLL	Dynamic Link Library
ECEF	Earth Centered Earth Fixed
EGNOS	European Geostationary Navigation Overlay System
ESA	European Space Agency
ÉTS	École de Technologie Supérieure
FEC	Forward Error Coorection
FPGA	Field Programmable Gate Array
FSB	Fast Synchronous Bus
Giove	Galileo In-Orbit Validation Element
GLONASS	GLObal'Naya Navigatsonnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GNSSU	Global Navigation System Sensor Unit
GPS	Global Positioning System

GTR	Galileo Test Receiver
GUI	Graphical User Interface
HOW	Hand Over Word
IF	Intermediate Frequency
LACIME	Laboratoire des communications et d'intégration de la microélectronique
LOS	Line Of Sight
MHz	Mega Hertz
NCO	Numerically Controlled Oscillator
NLOS	Non Line Of Sight
OS	Open Service
PVT	Position Velocity Time
PCI	Peripheral Component Interconnect
PRN	Pseudo Random Noise
PRS	Public Reglemented Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RMS	Root Mean Square
SAR	Search And Rescue
SBAS	Satellite Based Augmentation System
SOL	Safety Of Life
STANAG	Standardization Agreement
TLM	Telemetry Word
TOW	Time Of Week

UIT	Union Internationale des Télécommunications
UTC	Universal Time Coordinate
VHDL	VHSIC Hardware Design Level
VHSIC	Very High Speed Integrated Circuit
WN	Week Number
XMEM	eXternal MEMory
WGN	White Gaussian Noise

INTRODUCTION

De nos jours, la connaissance de la position devient une nécessité. En effet, les nouveaux produits en télécommunications sans fil tendent à intégrer un récepteur GPS embarqué qui permet à l'utilisateur d'avoir accès à différentes informations qui dépendent de sa position. D'un autre côté, plusieurs domaines, comme celui de l'aviation, nécessitent des systèmes de positionnement de plus en plus précis et robustes. Cette évolution dans les besoins nous mène à penser que les systèmes de navigation par satellite auront de plus en plus d'importance dans les prochaines années.

Le monde du positionnement par satellite étant en pleine effervescence avec la modernisation de GPS ainsi que l'avènement de Galileo, les défis actuels auxquels font face les concepteurs de récepteurs concernent principalement l'intégration de tous ces nouveaux signaux au sein d'une même plateforme technologique. Cette intégration des signaux GNSS permettra dans un futur proche de commercialiser des récepteurs hybrides GPS, Galileo ainsi que d'autres constellations comme GLONASS pour aboutir à des récepteurs plus précis et plus robustes que les récepteurs actuels.

Cependant, un des défis majeurs auxquels sont confrontées les industries des récepteurs dans le domaine de la radionavigation est le test et la validation de leurs équipements. En effet, pour tester le fonctionnement d'un récepteur GPS par exemple, il faut être en mesure de recevoir les signaux émis par les satellites. Ceci est cependant difficile lorsque le récepteur en cours de développement est dans un laboratoire.

Des méthodes de test restent cependant possibles, tel que l'utilisation d'une « *Roof Antenna* » pour acquérir une position fixe sur le toit d'un bâtiment, ou encore emmener le dispositif de test dans un véhicule et effectuer concrètement des trajectoires qui pourront valider le bon fonctionnement du récepteur en question. Ces méthodes sont efficaces puisqu'elles permettent de recevoir les vrais signaux des satellites, mais elles sont coûteuses et peu flexibles. De plus,

certaines signaux comme Galileo, ne sont même encore disponible dans un contexte, ce qui limite les scénarios de tests possibles.

En effet, si on veut par exemple tester le fonctionnement du récepteur avec des dynamiques importantes comme celles des avions, ceci devient beaucoup plus compliqué qu'un simple test dans un véhicule. De plus, un des domaines les plus concernés par la précision et l'intégrité des signaux est le domaine de l'avionique. En effet, un récepteur au bord d'un avion devrait être capable de fournir des positions très précises et sans interruption, quel que soit l'endroit où se trouve l'avion, ce qui nécessite des niveaux de robustesse très élevés.

C'est dans ce contexte que l'Agence spatiale canadienne veut mettre en place un projet de démonstrateur canadien pour la sauvegarde de la vie. L'ÉTS et la compagnie CMC sont des partenaires de l'agence spatiale canadienne pour mettre en œuvre ce projet d'envergure. En effet, la compagnie CMC a comme défi de développer un récepteur de sauvegarde de la vie « Safety Of Life » qui sera utilisé dans le domaine de l'avionique, alors que l'ÉTS aura comme projet le développement d'un simulateur de constellations hybride GPS et Galileo. La participation de l'agence spatiale canadienne est dans le cadre d'une contribution canadienne au programme Galileo.

Le développement de ce simulateur de constellations s'effectue au niveau du LACIME dans lequel j'effectue mon projet de maîtrise qui consiste à valider les solutions obtenues avec le simulateur, la mise en place d'une nouvelle architecture logicielle, ainsi que de la modélisation des effets des multi-trajets dans les signaux satellites générés.

Dans le premier chapitre de ce mémoire, nous ferons une étude théorique sur les systèmes GPS et Galileo, ainsi que sur le positionnement par satellite en général. Ensuite, dans le second chapitre, nous parlerons des étapes de simulation d'une constellation GPS ou Galileo, ainsi que les différents algorithmes de calculs en temps réel utilisés. Dans ce même chapitre, nous effectuons une correction et une validation de ces algorithmes pour obtenir des solutions PVT en GPS L1. Malgré les résultats satisfaisants obtenus avec le simulateur, nous avons relevé

plusieurs limitations qui ne peuvent pas être éliminés avec l'ancienne architecture logicielle du simulateur. En effet, il s'agit d'une architecture logicielle hybride (Matlab/C++) qui manque de flexibilité, et qui limite le simulateur à un certain nombre limité de fonctionnalités.

Le troisième chapitre sera consacré d'abord à la mise en évidence de la nécessité du développement d'une nouvelle architecture logicielle du simulateur. Une étude des besoins fonctionnels sera donc décrite. Nous présenterons donc la conception d'une nouvelle architecture logicielle modulaire qui répond à ces besoins. Ensuite nous décrirons les étapes du développement et les résultats retrouvés avec la nouvelle architecture logicielle après son intégration avec la partie matérielle du simulateur.

Enfin, dans le quatrième chapitre, nous aborderons la modélisation des multi-trajets dans les signaux satellites au niveau du simulateur. Nous effectuerons donc une description théorique des effets des multi-trajets dans le domaine de positionnement par satellites, ensuite nous présenterons les étapes de conception du modèle implémenté, son implémentation ainsi que les tests qui ont été effectués au niveau de notre simulateur.

CHAPITRE 1

Étude des systèmes GPS et Galileo

Dans ce premier chapitre, nous allons présenter les notions théoriques relatives aux deux systèmes de navigation par satellites GPS et Galileo. Nous commençons par la description des principes de radionavigation. La deuxième partie sera consacrée à une étude plus approfondie de la constellation GPS, alors que la troisième montrera les nouveautés qui seront apportées par la constellation Galileo. Enfin dans la dernière partie du chapitre, nous présenterons le simulateur de constellations GPS/Galileo sur lequel ce projet de maîtrise est basé.

1.1 Principe de radionavigation

L'avènement des systèmes de navigation par satellites est essentiellement dû au besoin incessant d'avoir des positions de plus en plus précises. Le positionnement dans un système de radionavigation est obtenu en calculant la distance séparant le récepteur des satellites visibles qui émettent les signaux radio. Cette mesure de distance, appelée pseudo-distance dans le domaine de la radionavigation à cause des imprécisions causés par plusieurs sources d'erreurs, est réalisée à partir de la mesure du délai de propagation des signaux entre le satellite et le récepteur comme le montre la Figure 1.1.

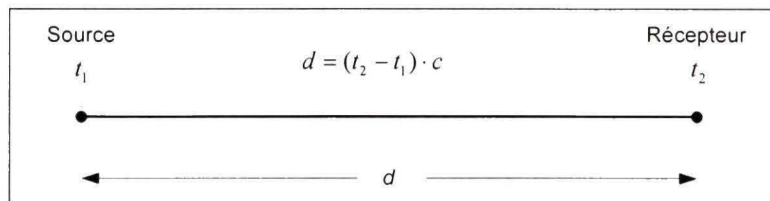


Figure 1.1 Principe de mesure de la distance.

où :

- c représente la vitesse de propagation d'une onde électromagnétique.
- $(t_2 - t_1)$ est le délai de propagation du signal radio.

Un récepteur GPS utilise donc cette distance qui le sépare des différents satellites dont il connaît les coordonnées pour calculer sa position en utilisant le principe de trilatération comme le montre la Figure 1.2 (Sauriol, 2007).

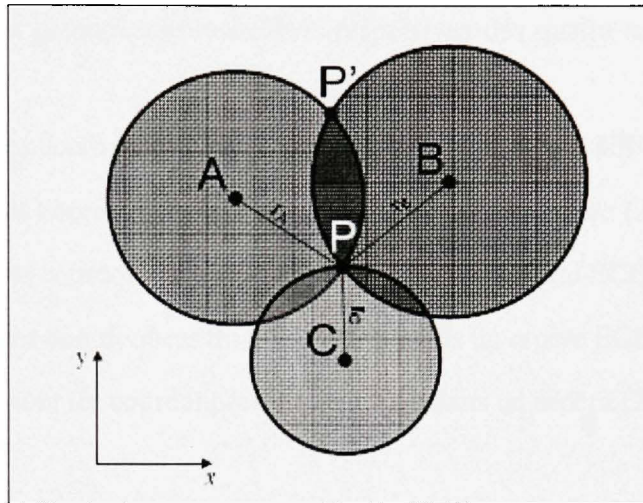


Figure 1.2 Principe de trilatération.
Tirée de Sauriol (2007, page. 26)

Dans cette figure, nous avons :

- A, B et C sont les positions des satellites.
- r_A , r_B , r_C sont les pseudo-distances calculées.
- P est la position du récepteur (en 2D).

En effet, en connaissant les coordonnées de trois satellites ainsi que les pseudo-distances, un récepteur GPS peut trouver sa position en 2D. Cette technique est étendue pour trouver la position 3D du récepteur en utilisant 4 satellites au lieu de 3. Cette approche nécessite la résolution des équations non linéaires suivantes :

$$r_A = \sqrt{(x_A - x_{REC})^2 + (y_A - y_{REC})^2 + (z_A - z_{REC})^2} \quad (1.1)$$

$$r_B = \sqrt{(x_B - x_{REC})^2 + (y_B - y_{REC})^2 + (z_B - z_{REC})^2} \quad (1.2)$$

$$r_C = \sqrt{(x_C - x_{REC})^2 + (y_C - y_{REC})^2 + (z_C - z_{REC})^2} \quad (1.3)$$

$$r_D = \sqrt{(x_D - x_{REC})^2 + (y_D - y_{REC})^2 + (z_D - z_{REC})^2} \quad (1.4)$$

où :

- r_A, r_B, r_C, r_D sont les distances géométriques respectives des quatre satellites par rapport au récepteur.
- (x_A, y_A, z_A) sont les coordonnées du 1^{er} satellite dans un repère ECEF.
- (x_B, y_B, z_B) sont les coordonnées du second satellite dans un repère ECEF.
- (x_C, y_C, z_C) sont les coordonnées du 3^{ème} satellite dans un repère ECEF.
- (x_D, y_D, z_D) sont les coordonnées du 4^{ème} satellite dans un repère ECEF.
- $(x_{REC}, y_{REC}, z_{REC})$ sont les coordonnées du récepteur dans un repère ECEF.

Il faut cependant noter qu'en réalité, la résolution de ces 4 équations ne permet pas de déterminer un seul point précis mais une région. Cette imprécision est essentiellement due aux erreurs de la mesure du délai de transmission des signaux. Ceci provient des erreurs d'horloges des satellites et du récepteur, ainsi qu'aux délais de propagations dans l'atmosphère.

1.2 Le système GPS

Le système GPS a été conçu dans les années soixante dix par le département de la défense américaine dans le but de l'utiliser dans des applications militaires. La déclaration officielle de sa mise en service a été effectuée en 1995. La constellation de satellites qui compose le système GPS transmet les signaux simultanément pour permettre aux récepteurs de trouver leurs positions en 3D. Cette transmission se fait sur une même bande de fréquence L1 en utilisation la technique d'accès CDMA (Code Division Multiple Access). Le système GPS est utilisé comme système de positionnement standard par les civils avec les codes d'étalement C/A qui permet d'obtenir une précision horizontale RMS de 5 m et une précision verticale de 7,5 m (Sauriol, 2007). Le service de précision qui utilise les codes P(Y). Ces codes P(Y), comme leur nom l'indique (*Precision*) sont des codes qui permettent d'avoir des solutions de navigation

beaucoup plus précises qu'avec les codes C/A. Cependant ils sont réservés à une utilisation exclusive de l'armée américaine, ainsi que ses alliés tels que le Canada.

1.2.1 Structure du système GPS

Le système GPS est composé de trois segments :

- Le segment utilisateur est représenté par les récepteurs utilisés dans l'acquisition des signaux et l'obtention de la position de l'utilisateur.
- Le segment de contrôle est composé de stations dont le rôle est de maintenir et de surveiller le bon fonctionnement du système GPS en termes d'orbites des satellites, de paramètres d'horloges ainsi que des éphémérides. Le nombre de stations de contrôle est 17 qui sont répartis dans plusieurs régions du monde.
- Le segment spatial est formé par une constellation de 24 satellites, 21 satellites actifs et 3 satellites de secours, répartis sur 6 orbites inclinées de 55 degrés par rapport au plan équatorial. Cependant, le nombre effectif de satellites sur orbites varie entre 29 et 31, et leurs signaux sont utilisés pour augmenter le degré de précision aux usagers.

1.2.2 Signaux GPS et leur modernisation

Le signal GPS est transmis sur deux ondes porteuse de la bande « L » appelées L1 et L2. La fréquence L1 qui est égale à 1575,42 MHz (soit une longueur d'onde de 19 cm) modulée par des codes d'étalement pseudo-aléatoires C/A et P(Y), et utilise une modulation BPSK. Les signaux émis par le satellite sur cette fréquence donnent la référence, l'heure ainsi que d'autres informations relatives à la position du satellite. La Figure 1.3 montre la construction du signal L1 :

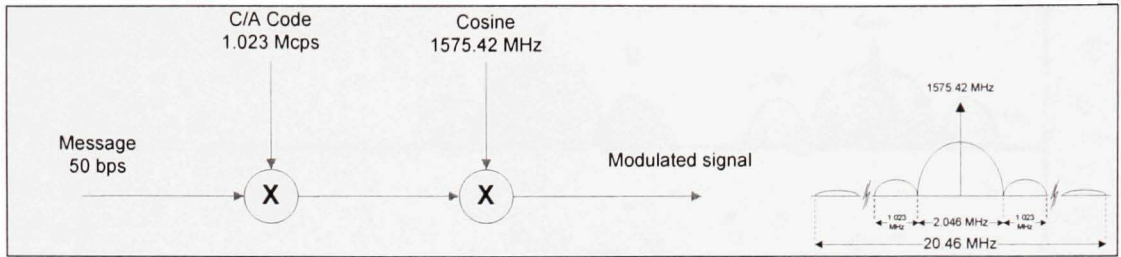


Figure 1.3 Le signal GPS L1.
Tiré de Sauriol (2007, p.1)

La fréquence L2 étant égale à 1227,60 MHz (soit une longueur d'onde de 24 cm) modulée par les codes P(Y). Ces codes sont générés par un mécanisme de chiffrement qui permet aux américains de garder l'exclusivité de son utilisation. Des pays comme le Canada ont eux aussi accès à l'utilisation des ces fréquences par l'entremise de récepteurs équipés à cet effet, et sous les conditions d'être munis de la clef de chiffrement, fournie sélectivement par les américains.

La modernisation des signaux GPS consiste au lancement de nouveaux satellites qui transmettent les signaux L2C utilisant la bande de fréquence L2 pour les applications civiles, ainsi que les signaux M pour utilisation militaire. Ces signaux L2C ont fait leur apparition en 2005. L'autre aspect de la modernisation du système GPS consiste à la transmission de signaux dans la bande L5, dont la fréquence est 1176.45 MHz, et qui sont disponibles depuis le 24 Mars 2009 (WORLD, 2009). Ce signal L5 a été lancé à titre expérimental et pour sauvegarder le droit d'utiliser cette fréquence approuvée par l'ITU. Ces signaux L5 possèdent aussi des codes d'étalement plus long que les codes C/A. Dans la Figure 1.4, nous trouvons une représentation de la modernisation des signaux GPS.

Le signal satellite L5 est composé de deux composantes, l'une en quadrature et l'autre en phase. Ce signal est modulé sur une porteuse de 1176.45 MHz, comme mentionné précédemment, et utilise la modulation QPSK comme le montre la Figure 1.5 (Ilie, 2003).

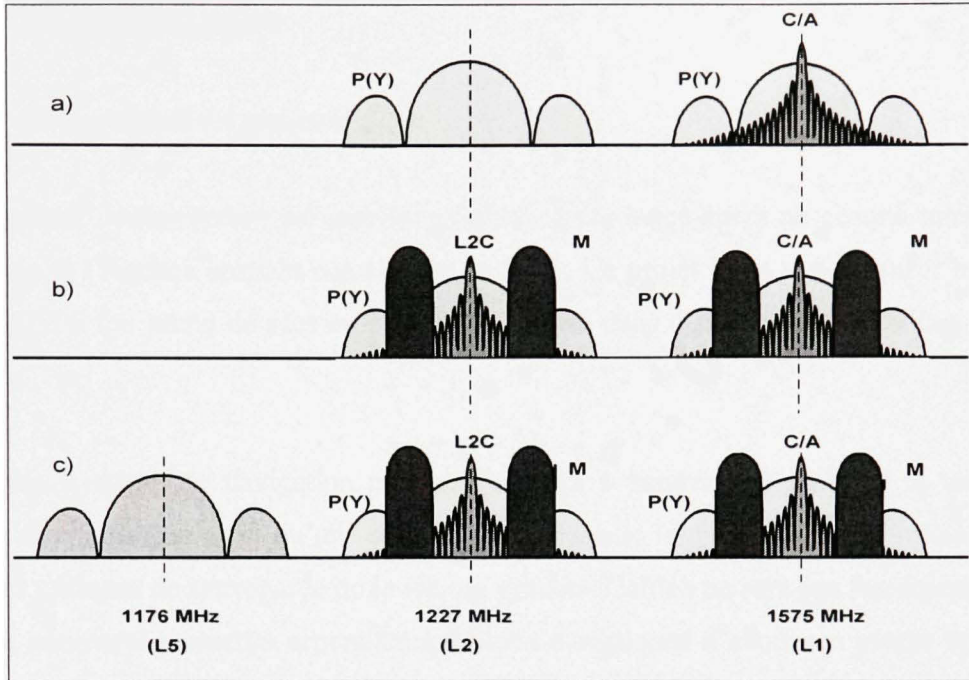


Figure 1.4 Évolution des signaux GPS.
Tiré de Ilie (2003, p. 45)

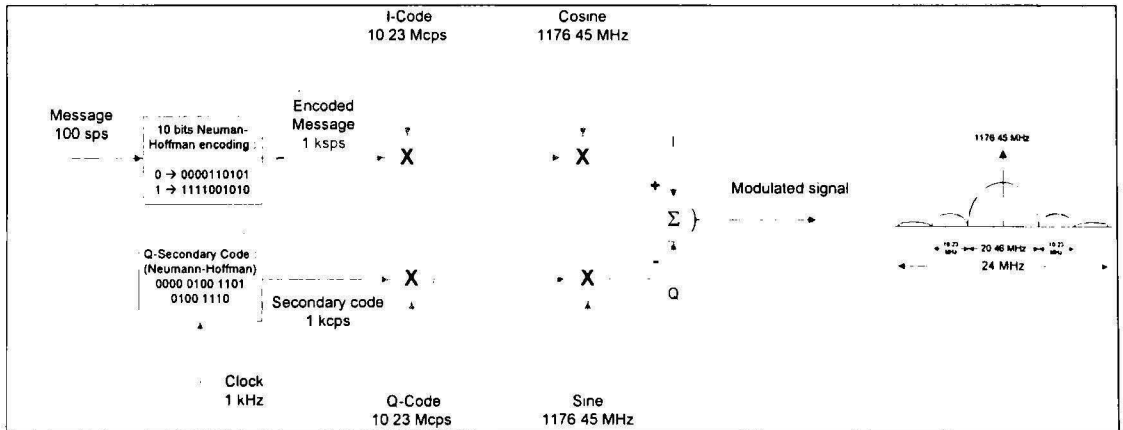


Figure 1.5 Le signal GPS L5.
Tiré de Sauriol (2007, p.1)

1.3 Le système Galileo

1.3.1 Présentation du projet Galileo

Le programme de navigation par satellites Galileo a été lancé après un accord entre l'Union européenne et l'Agence spatiale européenne en 2001. Ce projet vient concurrencer le système américain GPS qui prend de plus en plus d'importance dans différentes applications civiles et militaires.

Ce nouveau système de navigation par satellite aura à assurer une intégrité et une grande précision des messages ainsi qu'une couverture globale de la terre, pour être utilisé dans des opérations critiques de sauvegarde de la vie. Le système Galileo ne sera pas forcément meilleur que GPS, mais ceci permettra cependant à l'Union européenne d'avoir son propre système de positionnement. D'un autre côté, l'utilisation simultanée des deux systèmes GPS et Galileo permettra d'avoir une plus grande disponibilité des satellites en vue directe, et par conséquent des solutions de navigations plus robustes au niveau des récepteurs.

L'échéancier initial du projet prévoyait le début des opérations de commercialisation des récepteurs Galileo en 2008, mais une multitude de problèmes ont retardé cet échéancier (Sauriol, 2007). En effet, jusqu'à présent seulement deux satellites (GIOVE-A et GIOVE-B) ont été lancés. Ces deux satellites ne sont pas opérationnels au vrai sens du terme puisqu'il faut avoir au moins 4 satellites pour obtenir une position au niveau d'un récepteur. Cependant, leur lancement permet à l'Agence spatiale européenne de garder les fréquences allouées au projet Galileo.

Un des avantages du projet Galileo consiste à l'utilisation de la même fréquence que GPS L1 pour les signaux Galileo E1. Ceci va donc représenter un avantage important pour les concepteurs de récepteurs hybrides GPS/Galileo.

1.3.2 Particularité des signaux Galileo

La constellation Galileo devrait être formée de 27 satellites répartis en trois plans orbitaux avec 9 satellites chacun. Le rayon des orbites est de 29.994 km, l'inclinaison est de 56 degrés (ESA, 2006). Et comme pour GPS, il est prévu qu'un satellite de secours soit utilisé dans chacun des plans orbitaux pour arriver à une constellation de 30 satellites au total (ESA, 2006).

Les signaux du système Galileo vont être modulés autour des fréquences E1, E5a, E5b et E6 comme le montre la Figure 1.6 (ESA, 2006). Nous remarquerons que certaines fréquences sont les mêmes que pour GPS tel que E1 et E5a.

Grâce aux bandes de fréquences qui vont être déployées, plusieurs services vont être disponibles avec le système Galileo (ESA, 2009) :

- Le service ouvert (*OS ou Open Service*): Un service gratuit et ouvert à tous, similaire au service public de GPS, mais qui aura probablement plus de précision.
- Le service commercial (*CS ou Commercial Service*) : Un service payant qui pourra assurer de meilleures performances à ses usagers.
- Le service de sûreté de vie (*SOL ou Safety Of Life*) : Un service à très haute intégrité pour la sauvegarde de la vie.
- Le service public réglementé (*PRS ou Public Reglemented Service*) : Un service qui offre une grande robustesse face aux interférences et dédié principalement aux instances gouvernementales tels que l'armée.
- Le service de recherche et secours (*SAR ou Search and Rescue Service*) : Un service dédié aux opérations de recherche et de sauvetage.

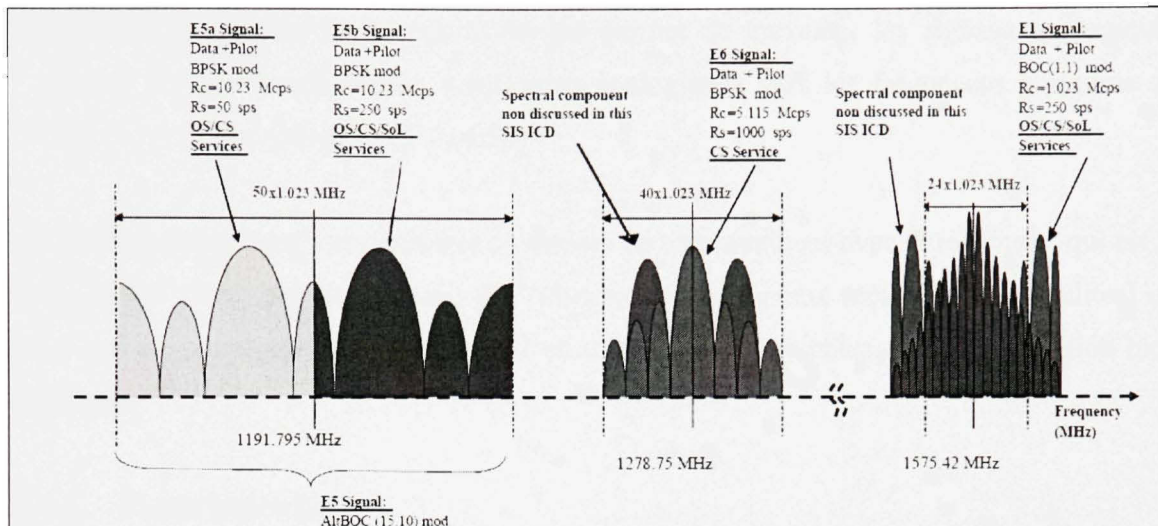


Figure 1.6 Spectres utilisés par le système Galileo.
Tirée de ESA (2006, p. 21)

1.4 Le Simulateur de constellation GPS/Galileo : Premier aperçu

Le développement d'un simulateur hybride de constellations GPS/Galileo fait partie d'un projet initié par l'agence spatiale canadienne et la compagnie CMC dans le cadre du démonstrateur canadien de sauvegarde de la vie (*Safety Of Life*).

Ce simulateur de constellations GPS/Galileo est constitué de trois parties:

- La partie Logicielle du simulateur (*The Software Core*) dans laquelle sont générées les données qui seront envoyées vers les récepteurs.
- La partie IF (*Intermediate Frequency Core*) dans laquelle sont effectués en temps réel les traitements numériques des signaux sur FPGA puis convertis en signaux analogiques centrés autour de la fréquence 70 MHz.

- La partie RF (Radio Frequency Core) qui permet de translater les signaux analogiques générés par les convertisseurs numériques analogiques vers les fréquences porteuses de GPS (L1, L5) et Galileo (E1, E5a, E5b).

Il est à noter que le simulateur doit être en mesure de communiquer avec le récepteur, qui est le '*Safety of life Unit*', par des signaux RF (identiques aux signaux reçus par les satellites) ou encore par communication en série RS232 en utilisant des protocoles de communication bien spécifiques.

1.4.1 Partie logicielle

La partie logicielle du simulateur est composée de deux parties distinctes. Une première partie qui a été développée en Matlab et une deuxième partie en C++ .net. Dans la partie Matlab, nous avons une interface graphique (Figure 1.7) permettant à l'utilisateur de choisir les paramètres de la simulation dont il a besoin. Les principaux paramètres sont :

- Le choix de la constellation.
- Le choix de la date de début et la durée de la simulation.
- La configuration des paramètres orbitaux et des corrections des horloges.
- La configuration des paramètres relatifs aux perturbations atmosphériques.
- Le choix de la trajectoire du récepteur.

Les différentes étapes de simulation d'un scénario seront détaillées dans le second chapitre.

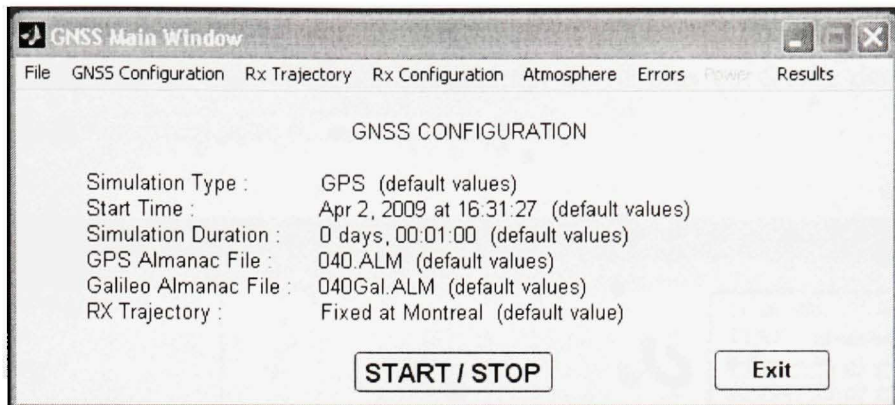


Figure 1.7 Interface graphique de la partie Matlab.

Dans la partie Matlab, nous effectuons tous les calculs qui permettent de générer les paramètres nécessaires à la constitution des messages de navigation GPS ou Galileo, ainsi que ceux nécessaires à la modulation des signaux. Ces paramètres sont ensuite envoyés vers la partie C++.net de façon continue pendant toute la simulation.

Dans la partie C++.net, différents processus sont programmés pour répondre aux spécifications exigées par le projet :

- Génération des messages de navigations pour GPS L1, L5, Galileo E1, E5a, E5b.
- Communication par port série RS232 avec le récepteur '*Safety Of Life*' de CMC.
- Effectuer les calculs nécessaires des données envoyées au FPGA.
- Communication avec le FPGA à travers le bus PCI.
- Gestion de l'interface graphique usager.

La Figure 1.8 montre l'interface graphique permettant de choisir le moyen de communication entre le simulateur et le récepteur (RF ou RS232), ainsi que les satellites inclus ou exclus de la simulation.

À noter que la partie Matlab du projet est compilée dans une DLL à laquelle le code C++.net fait appel au cours de la simulation. Ce procédé est décrit plus en détails dans le manuel d'utilisation de la partie logicielle (Constantinescu, 2006).

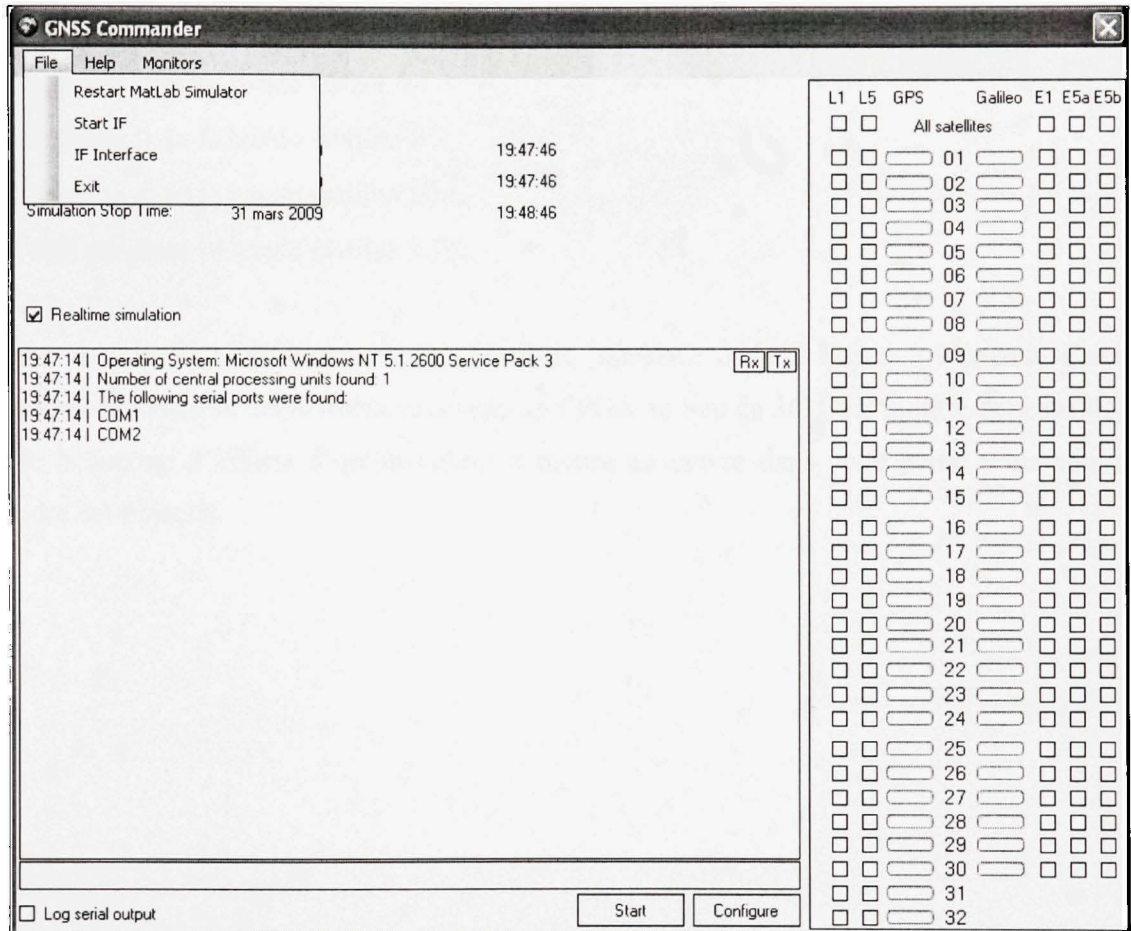


Figure 1.8 Interface graphique de la partie C++.net.

1.4.2 Partie traitement en temps réel des signaux numériques

Dans cette partie qu'on appelle plus communément la partie IF, les signaux sont modulés ensuite convertis en signaux analogiques. Au niveau matériel, cette partie est constituée par une carte de Lyrtech basée sur un FPGA Virtex4 et suivi de 8 convertisseurs numériques analogiques dont 3 sont utilisés. Les signaux analogiques que nous récupérons à la sortie des

convertisseurs sont autour de la fréquence 70 MHz. Pour l'instant, l'architecture en place permet de simuler 8 canaux simultanément alors que l'objectif initial était que le signal analogique à la sortie contienne 36 canaux répartis comme suit :

- 12 signaux dans la bande GPS L1.
- 6 signaux dans la bande GPS L5.
- 6 signaux dans la bande Galileo E1.
- 6 signaux dans la bande Galileo E5a.
- 6 signaux dans la bande Galileo E5b.

Comme mentionné précédemment, l'architecture présentée dans la Figure 1.9 permet d'avoir actuellement 8 canaux disponibles au niveau du FPGA au lieu de 36. Ceci montre donc qu'il y a encore beaucoup d'efforts d'optimisations à mettre en œuvre dans cette partie pour pouvoir atteindre cet objectif.

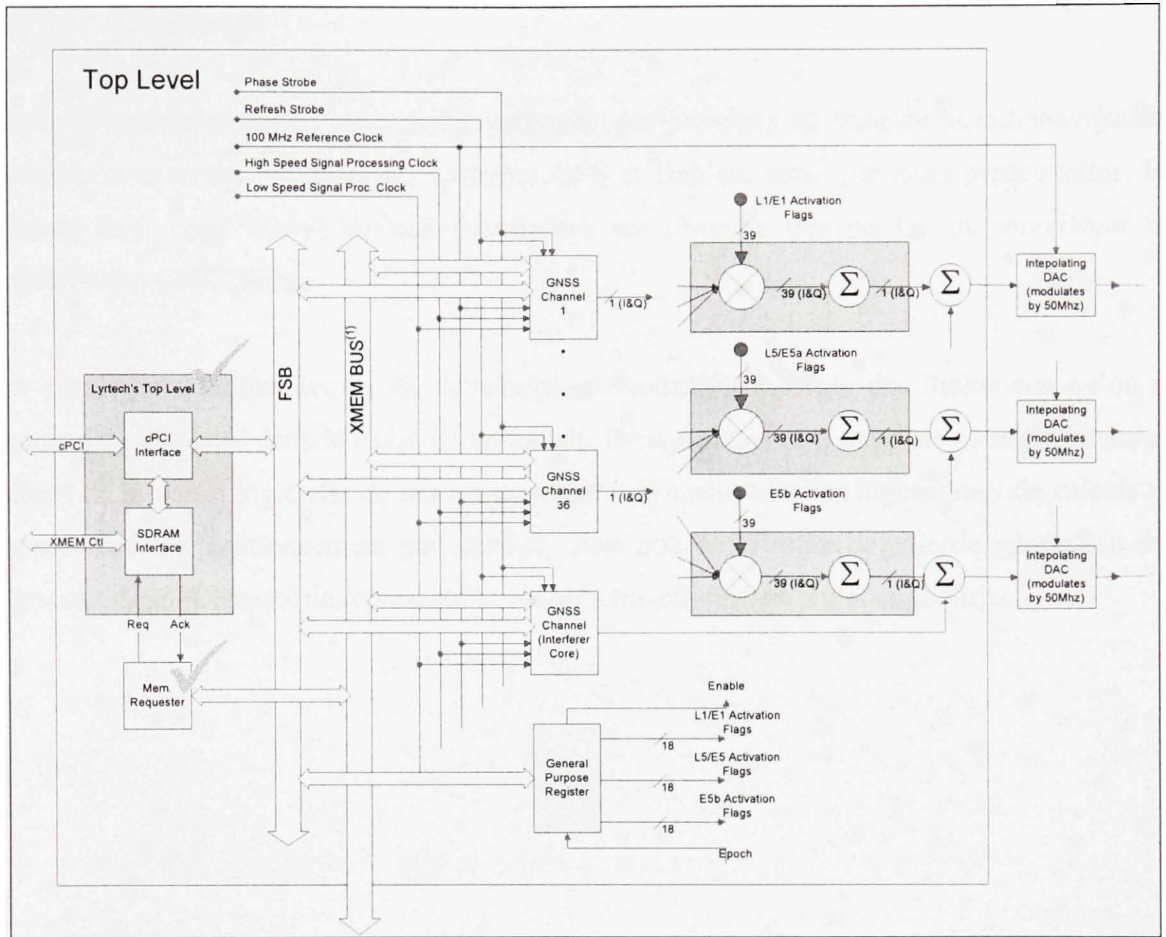


Figure 1.9 Description haut niveau de la partie IF.
Tirée de Sauriol (2007, p. 3)

1.4.3 Partie analogique

La partie analogique ou partie RF du simulateur permet de récupérer les signaux des convertisseurs à la sortie de la partie IF, et les translater aux fréquences radio fréquences des bandes évoquées antérieurement GPS L1, L5 et Galileo E1, E5a, E5b.

Cette partie a connu une évolution intéressante pour aboutir vers un circuit qui se branche directement sur la plateforme utilisée et ayant un niveau d'intégration qui permet au simulateur de garder des dimensions très concurrentielles par rapport aux autres produits sur le marché des simulateurs de constellations (LOUKIL, 2010).

1.5 Conclusion

Dans ce premier chapitre, nous avons parcourus les principes de base de la radionavigation. Ensuite, nous avons présentés les systèmes GPS et Galileo, ainsi que leurs particularités. En dernier lieu, nous avons vu une description sur chacune des parties du simulateur de constellation GPS/Galileo.

Un prototype du simulateur a été développé, cependant il présente des limitations qu'on se propose de résoudre dans le cadre de ce travail. Dans les chapitres suivants nous parlerons en détails de la partie logicielle de notre simulateur, qui inclut tous les algorithmes de calculs en rapport avec le positionnement par satellite, ainsi que de la méthodologie de génération des signaux GNSS. Cette partie représente le cœur du travail de mon projet de maîtrise.

CHAPITRE 2

La simulation d'une constellation de satellites GPS/Galileo

Dans ce deuxième chapitre, nous présenterons la partie logicielle du simulateur de constellations GPS/Galileo de l'ÉTS. Cette partie nécessite une connaissance approfondie des systèmes de navigation pour la mise en œuvre de tous les algorithmes de calculs nécessaires à la simulation des constellations en question. D'autre part, elle requiert aussi la connaissance de tout ce qui se passe au niveau du traitement numérique des données dans le FPGA pour pouvoir mettre en place les protocoles de communication avec la partie matérielle du simulateur tout en respectant les aspects de synchronisations qui sont critiques pour la précision au niveau des résultats.

La première partie de ce chapitre nous donnera une description détaillée des étapes de simulation d'une constellation. La deuxième partie sera consacrée aux algorithmes de calculs nécessaires dans un scénario en temps réel défini par l'utilisateur. Dans la troisième partie, une vue globale sur la génération des signaux satellites effectuée dans la partie logicielle du simulateur sera présentée. Enfin, dans la dernière partie, nous mettrons en évidence les difficultés ainsi que les différentes corrections apportées au niveau logiciel pour pouvoir valider le bon fonctionnement de notre simulateur.

2.1 Scénario de simulation d'une constellation

Cette première partie permettra de montrer les étapes d'une simulation, ainsi que les différents paramètres que nous devons configurer pour pouvoir générer les signaux des satellites que nous voulons simuler.

2.1.1 Choix de la constellation et configuration des paramètres orbitaux

La première étape de la configuration de la simulation consiste à choisir la constellation GPS, Galileo ou les deux en même temps. Pour chacune des constellations utilisées, les valeurs de

plusieurs paramètres orbitaux doivent être fixées pour pouvoir calculer les positions des satellites. Les différentes catégories de ces paramètres ainsi que leur impact sur la simulation seront étudiées dans les sections suivantes. Il faut cependant préciser qu'avec la version actuelle nous ne pouvons simuler que la constellation GPS.

Paramètres relatifs au mouvement des satellites :

Au cours de la simulation, plusieurs algorithmes de calculs, présentés dans les sections suivantes, sont exécutés pour pouvoir générer les signaux GPS avec précision. Dans cette section, nous décrivons les paramètres utilisés dans le calcul des positions des satellites sur leurs orbites à chaque instant de la simulation comme le montre la Figure 2.1.

The screenshot shows the 'Signal_sources_GPS' window with the 'Motion' section selected. The 'Orbits' sub-section is active, displaying a table of orbital parameters for satellite 1. The 'Almanac file 204.alm information' box shows 29 satellites present and a reference time of 8-Dec-1984 19:50:24.000. The 'Motion' section includes a 'Present' checkbox (checked) and a list of parameters with input fields and 'All' buttons.

Parameter	Value	Unit	Action
Reference time (Week number and TOW(s))	256	589824	All ✓
Root semi major axis ($A^{*1/2}$)	5152.6	metres ^{*1/2}	All ✓
Eccentricity (e)	0.00529718		All ✓
Argument of perigee (w)	-1.65339	semicircles	All ✓
Mean anomaly (Mo)	2.30013	semicircles	All ✓
Inclination (Io)	0.979893	semicircles	All ✓
Longitude of ascending node (Weekly Epoch) (W0)	2.01431	semicircles	All ✓
Mean motion difference (Delta N)	0	semicircles/s	All ✓
Inclination rate (Idot)	0	semicircles/s	All ✓
Right ascension rate (Wdot)	-7.73747e-009	semicircles/s	All ✓
Mean motion difference rate (Delta Ndot)	0	semicircles/s ²	All ✓

Figure 2.1 Configuration des paramètres orbitaux.

Les paramètres orbitaux que nous utilisons pour notre simulateur sont décrits tel que dans (ARINC, 2001):

- **Satellite**: Le numéro du satellite.
- **WN (Reference Week)** : La semaine référence pour laquelle les paramètres orbitaux sont valides.
- **TOW (Time of Week)** : Le temps de référence en secondes pour lequel les paramètres ont été générés.
- $A^{1/2}$: La racine carrée du demi-grand axe de l'orbite ($m^{1/2}$).
- e : L'excentricité de l'orbite (sans dimension).
- ω : L'argument du périhélie (radian).
- M_0 : L'anomalie moyenne au temps de référence (radian).
- I_0 : L'angle d'inclinaison au temps de référence (radian).
- ω_0 : L'ascension droite au nœud ascendant (radian).
- Δn : La différence entre le mouvement moyen et la valeur calculée (rad/s).
- \dot{I} : Le taux de variation de l'angle d'inclinaison (rad/s).
- $\dot{\omega}$: La vitesse d'ascension droite (rad/s).

Dans cette partie, nous pouvons configurer manuellement les paramètres pour chacun des satellites de la constellation ou encore charger les données à partir des fichiers almanachs. Les almanachs utilisés sont écrits selon le format YUMA, un standard défini par le gouvernement américain (US Department of Homeland Security, 2009) comme le montre le Tableau 2.1 :

Tableau 2.1 Format YUMA.

Satellite ID
Eccentricity (e)
Time of Week (TOW)
Orbital Inclination (I_0)
Rate of right ascension($\omega\dot{\ } $)
SQRT(A)
Right ascension at TOA(ω_0)
Argument of Perigee(ω)
Mean Anomaly(M_0)
Af(0)
Af(1)
Week

Paramètres de correction d'orbites :

Comme pour les paramètres énumérés dans la section précédente, ceux des corrections d'orbites sont définis pour chacun des satellites de la constellation. Ils sont utilisés dans le calcul des termes de correction d'orbites, appelés aussi perturbations harmoniques secondes (Sauriol, 2007).

Les paramètres (Figure 2.2) que l'utilisateur doit configurer pour les différents satellites présents dans la simulation sont :

- C_{uc} : Amplitude du terme de correction harmonique cosinusoidal de l'argument de latitude (μrad).
- C_{us} : Amplitude du terme de correction harmonique sinusoidal de l'argument de latitude (μrad).

- C_{ic} : Amplitude du terme de correction harmonique cosinusoidal de l'angle d'inclinaison (μrad).
- C_{is} : Amplitude du terme de correction harmonique sinusoidal de l'angle d'inclinaison (μrad).
- 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).

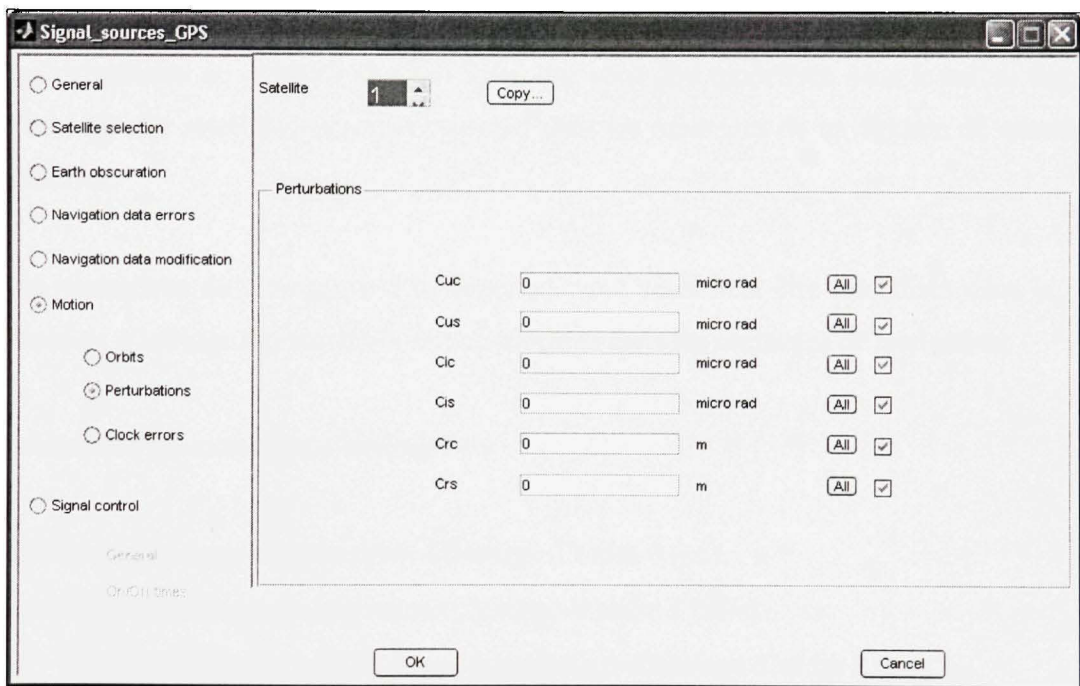


Figure 2.2 Configuration des paramètres de correction d'orbites.

Les termes de correction d'orbites sont définis dans (ARINC, 2001) selon les équations utilisées dans le calcul des positions des satellites dans ANNEXE I.

Paramètres de correction des horloges des satellites :

Dans le cas idéal, tous les satellites d'une même constellation sont supposés être synchronisés entre eux, ce qui n'est pas vrai dans le cas réel. En effet, il existe un biais et une dérive d'horloge propre à chacun des satellites.

Pour que les signaux générés par le simulateur soient donc le plus fidèles possible à la réalité, cette partie permet à l'utilisateur de définir, pour les différents satellites de la constellation, deux types de paramètres comme le montre la Figure 2.3 :

- Les paramètres de corrections d'horloges, qui vont être modélisés dans le calcul des biais d'horloge des satellites, et seront transmis dans les messages de navigation de chacun des satellites.
- Les paramètres de divergence d'horloge, qui vont seulement être modélisés dans le calcul des biais d'horloge des satellites, et non transmis dans les messages de navigation.

Les paramètres de correction d'horloge sont :

- a_{f0} : Biais d'horloge ou correction d'horloge d'ordre 0 (μs).
- a_{f1} : Dérive d'horloge ou correction d'horloge d'ordre 1 (ns/s).
- a_{f2} : Variation de la dérive d'horloge ou correction d'horloge d'ordre 2 (ps/s^2).

Les paramètres de divergence d'horloge sont :

- Δa_{f0} : Divergence d'ordre 0 (ms).
- Δa_{f1} : Divergence d'ordre 1 ($\mu\text{s/s}$).
- Δa_{f2} : Divergence d'ordre 2 (ps/s^2).

Les équations de calculs des biais d'horloges utilisant ces paramètres vont être discutées dans les sections suivantes.

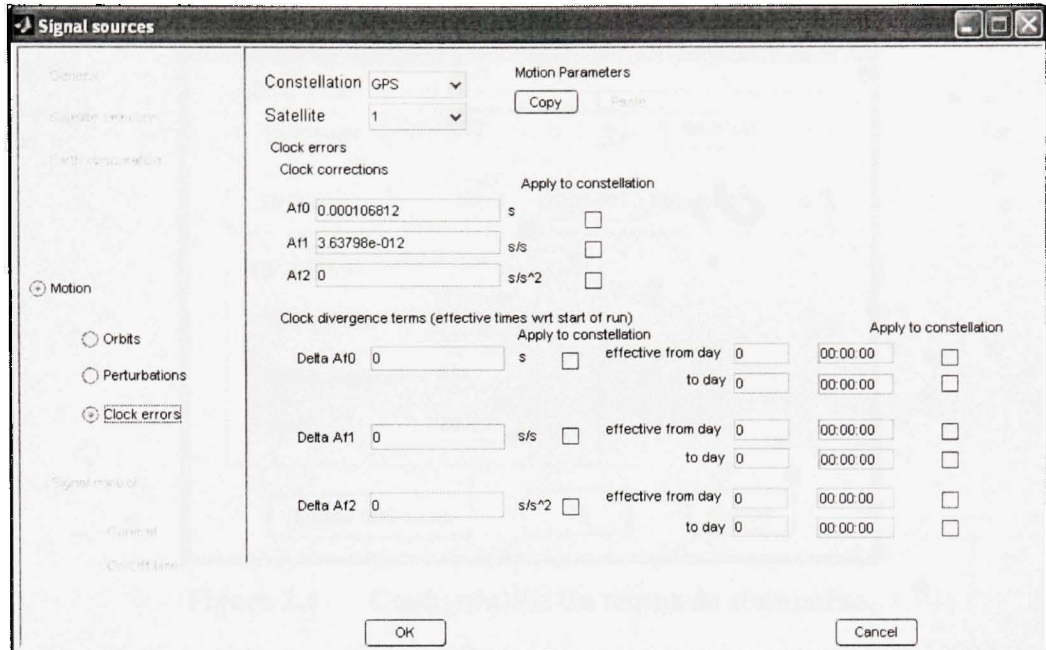


Figure 2.3 Paramètres de correction des horloges des satellites.

2.1.2 Date et durée de la simulation

Après avoir défini les paramètres orbitaux, nous devons choisir les paramètres relatifs au temps ainsi qu'à la durée de la simulation. Comme nous le voyons sur la Figure 2.4, l'utilisateur doit configurer les paramètres suivants :

- Le temps de début de la simulation : [aaaa-mm-jj] et [hh :mm :ss].
- La durée de la simulation : [nombre de jours] et [hh :mm :ss].

Ces paramètres sont configurés selon le temps UTC (Temps Universel Coordonné) qui est mesuré par l'utilisation d'une horloge atomique.

Après avoir configuré le temps de début de simulation (UTC), l'utilisateur peut voir, sur l'interface, le temps GPS équivalent à cette date.

The screenshot shows a dialog box titled "Start Time and Duration". It is divided into two main sections: "UTC Time" and "GPS Time".

- UTC Time section:**
 - Start time:** A dropdown menu showing "2009-04-02" and a text box showing "16:31:27".
 - Duration:** A text box showing "0" followed by "days", and another text box showing "00:01:00" followed by "hh:mm:ss".
- GPS Time section:**
 - WN rollover:** A text box containing the value "1".
 - Week number:** A text box containing the value "501".
 - TOW:** A text box containing the value "405087" followed by a small "s" for seconds.

At the bottom of the dialog, there are three buttons: "Update GPS Time", "OK", and "Cancel".

Figure 2.4 Configuration du temps de simulation.

Le temps GPS est défini par le nombre de semaine (WNa) et le nombre de secondes (TOW). La référence du temps GPS est le 6 Janvier 1980.

- Week Number (WNa) : Ce paramètre varie de 0 jusqu'à 1023 semaines.
- TOW : Le TOW est remis à zéro tous les dimanches à minuit et varie entre 0 et 604800 (secondes).
- WN Rollover : Ce paramètre a été défini pour indiquer le passage d'un cycle de temps GPS à un autre. En effet, puisque le 'WN' varie entre 0 et 1023, la semaine du 15 Aout au 21 Aout 1999 représente la semaine numéro 1023 (à partir du 6 Janvier 1980). Le paramètre 'WN' est remis donc à 0 pour la semaine suivante (du 22 Aout au 28 Aout 1999), alors que le 'WN Rollover' est incrémenté de 1 pour indiquer le passage à un nouveau cycle de temps GPS durant 1023 semaines (GPS Epoch).

Étant donné que le temps universel UTC est corrigé périodiquement de quelques secondes, il diffère donc du temps GPS. Pour remédier à cette légère différence, le message de navigation contient des paramètres qui vont permettre au récepteur de faire la conversion du temps GPS au temps UTC (ARINC, 2001).

2.1.3 Configuration des paramètres atmosphériques

Dans un contexte réel, la propagation des signaux GPS est retardée par la traversée des couches ionosphérique et troposphérique. Par conséquent, nous avons prévu d'inclure ces délais atmosphériques dans la modélisation des signaux RF du simulateur au niveau des pseudo-distances.

Paramètres ionosphériques :

Il existe plusieurs méthodes qui permettent de modéliser les délais ionosphériques (terrestre, spatial ou encore combiné). Celui qui est implémenté dans notre simulateur est le modèle ionosphérique terrestre ou encore modèle de Klobuchar (Youcef, 2009) comme le montre la Figure 2.5.

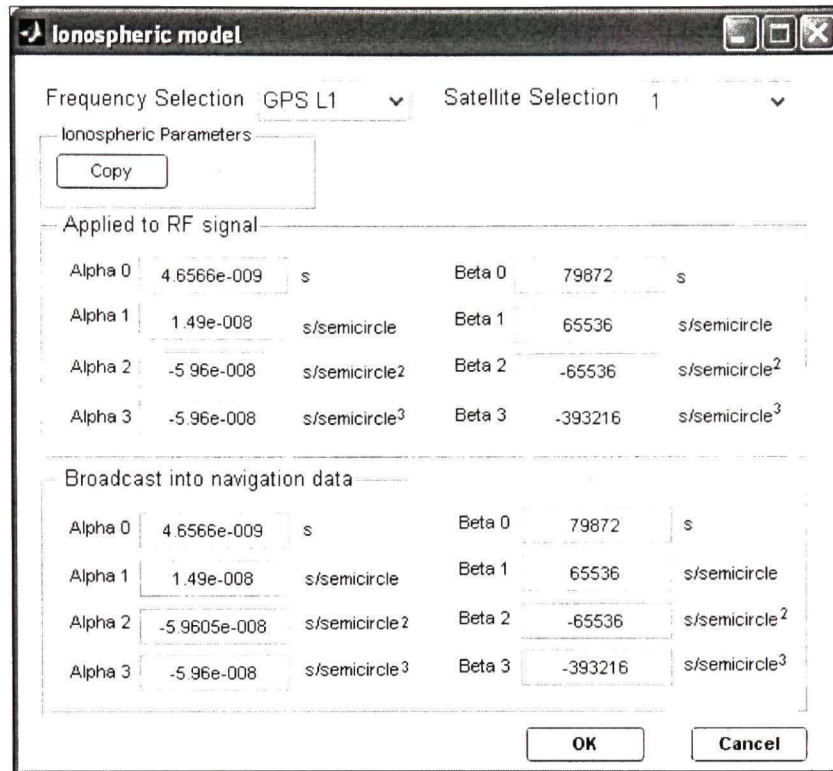


Figure 2.5 Configuration des paramètres ionosphériques.

Le modèle permet à l'utilisateur de configurer deux types de paramètres :

- Les paramètres appliqués au signal RF : Il s'agit des variables $Alpha_i$ et $Beta_i$, utilisés dans le calcul du délai ionosphérique à la fréquence indiquée, qui permettront d'appliquer le modèle choisi au signal généré pour chaque satellite et chacune des fréquences simulées.
- Les paramètres transmis dans le message de navigation : Ces variables $Alpha_i$ et $Beta_i$ seront transmises dans le message de navigation et permettront au récepteur de faire les calculs nécessaires à la compensation du délai ionosphérique contenu dans signal reçu.

Si les deux types de paramètres présentés ci-dessus sont différents, ceci peut engendrer des erreurs dans la position calculée par le récepteur et tester ainsi sa robustesse.

Paramètres troposphériques :

Comme pour les modèles ionosphériques, il existe plusieurs modèles troposphériques que nous pouvons utiliser dans le simulateur (STANAG, BD2, RTCA 96, RTCA 98). Dans notre simulateur, un modèle par défaut (STANAG) est implémenté. Le délai troposphérique est donc calculé en fonction de paramètres qui ne sont pas configurables par l'utilisateur.

2.1.4 Choix de la trajectoire du récepteur

Cette partie de la configuration du scénario permet à l'utilisateur de définir une trajectoire que le récepteur va parcourir le long de la simulation. Pour décrire une trajectoire complète, plusieurs manœuvres sont donc disponibles. La Figure 2.6 présente l'interface qui permet de séquencer les manœuvres de différents types pour une trajectoire donnée.

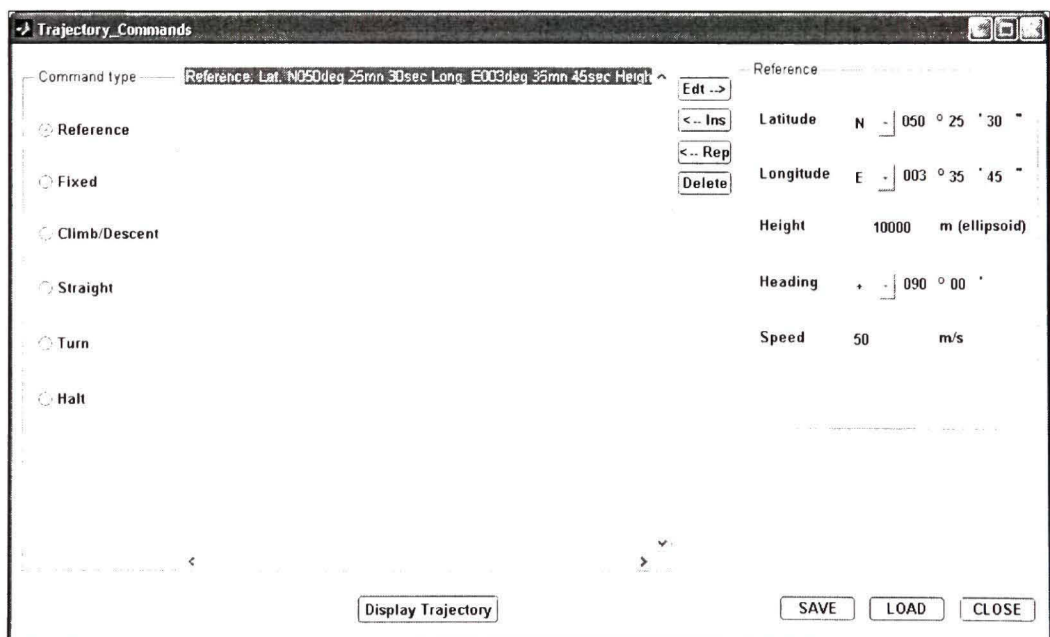


Figure 2.6 Fenêtre de configuration de la trajectoire du récepteur.

Les différentes commandes utilisées dans le simulateur de trajectoire sont les suivantes:

- La commande 'Référence' : Cette commande doit toujours être définie en premier, il s'agit du point de départ de la trajectoire. L'utilisateur doit définir à cette étape la position initiale du récepteur (Latitude, Longitude, Altitude) ainsi que la direction et la vitesse initiale.
- La commande 'Fixed' : Cette commande permet de garder le récepteur dans une position fixe pendant une durée déterminée (sec).
- La commande 'Climb/Descent' : Cette commande permet de configurer une montée ou une descente du récepteur en définissant le changement d'altitude (m), ainsi qu'une vitesse (m/s) et une accélération (m/s^2) de changement d'altitude.
- La commande 'Straight' : Elle permet d'effectuer une trajectoire en ligne droite en définissant une durée (s) ou une distance (m).
- Le commande 'Turn' : L'utilisateur doit définir pour cette commande le changement de direction du récepteur (en degrés), ainsi que l'accélération latérale pendant le virage.
- La commande 'Stop' : Cette commande permet d'arrêter définitivement le mouvement du récepteur.
- La commande 'Save/Load' : Une fois la trajectoire est définie, elle peut être sauvegardée à l'aide de la commande 'Save'. Une trajectoire peut aussi être chargée à partir d'un fichier déjà sauvegardé.
- La commande 'Display Trajectory' : Elle permet de visionner la trajectoire avant le début de la simulation.

2.2 Algorithmes de calculs au niveau du simulateur

Une fois le scénario commencé, le simulateur effectue les opérations nécessaires à la génération des signaux satellites selon ce scénario que le récepteur utilisera afin de retrouver les positions qui correspondraient à la trajectoire définie dans le simulateur de trajectoire avec une très haute précision.

Pour pouvoir générer des signaux, plusieurs calculs sont effectués au niveau logiciel pour envoyer des paramètres précis à la plateforme matérielle (FPGA). Nous décrivons dans cette partie les différents algorithmes que nous utilisons avant d'expliquer dans la troisième partie du chapitre le processus complet qui permet de générer les signaux satellites.

2.2.1 Description du processus de calcul effectués dans la DLL Matlab

Comme nous l'avons expliqué à la fin du premier chapitre, la partie logicielle du simulateur est divisée en deux parties distinctes :

- La DLL Matlab dans laquelle sont effectués tous les calculs par rapport à la constellation.
- La partie C++.net qui permet d'encoder les données et de communiquer avec les autres parties du simulateur (FPGA/Carte RF) ou de communiquer directement avec le récepteur en mode RS232.

La Figure 2.7 présente un synoptique du fonctionnement de la DLL Matlab. Le cœur de la simulation consiste à mettre à jour, chaque seconde, les paramètres relatifs à la constellation en mettant en œuvre plusieurs algorithmes de calculs que nous expliquerons en détails dans les sections qui suivent.

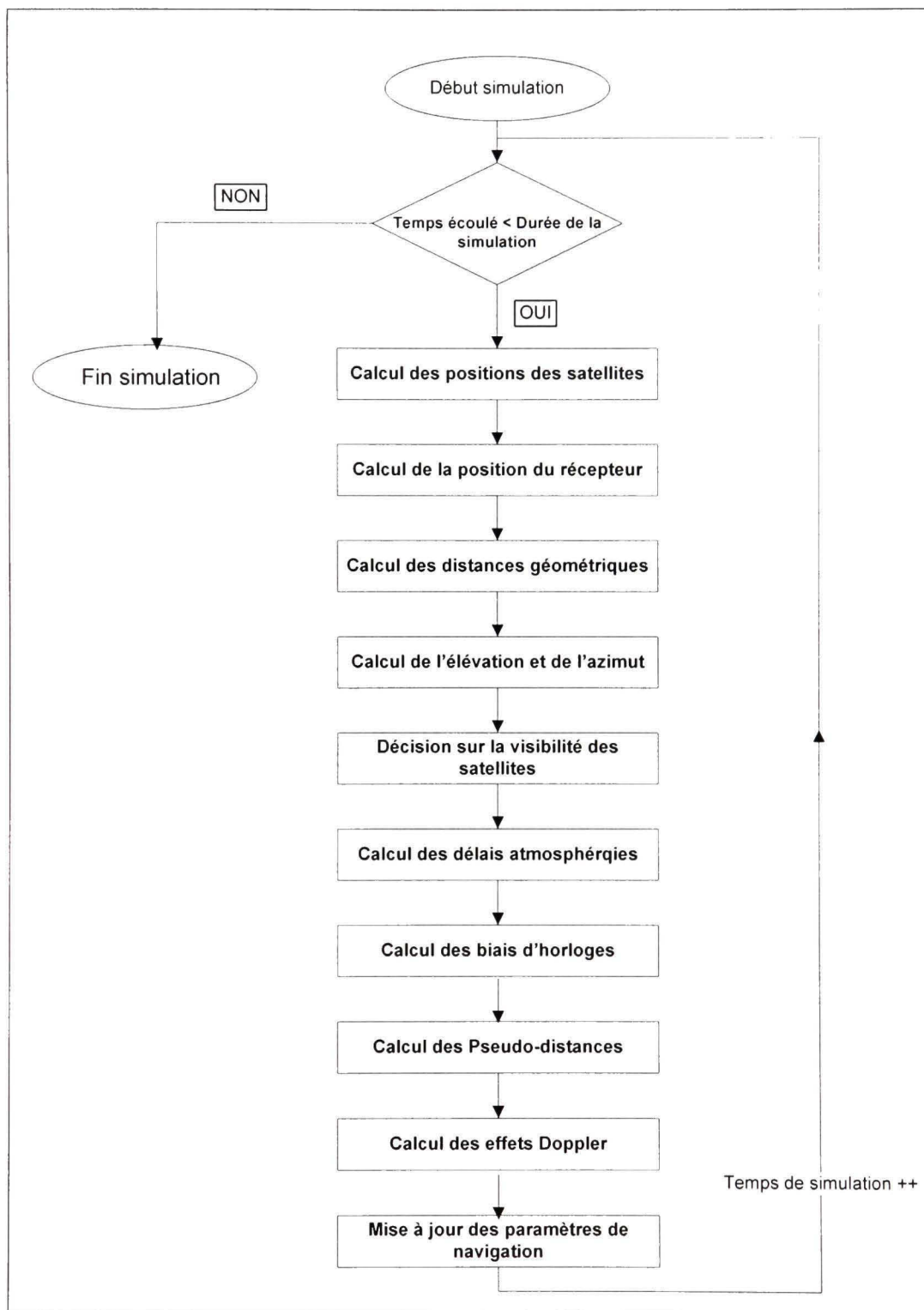


Figure 2.7 Diagramme des opérations effectuées au niveau de la DLL Matlab.

2.2.2 Calcul des positions des satellites

La première opération de ce processus implémenté dans Matlab consiste à calculer les positions de tous les satellites à partir des paramètres orbitaux (Figure 2.1). Cet algorithme est un algorithme très important au niveau de la simulation, car la moindre imprécision dans les positions calculées aura des conséquences néfastes sur les signaux qui seront générés.

La procédure permettant le calcul de la position d'un satellite en temps réel dans un référentiel cartésien ECEF (Earth Centered Earth Fixed) commence par la vérification de la validité des paramètres orbitaux que nous utilisons selon les équations (2.2) et (2.3).

Ces paramètres proviennent d'un fichier almanach dont le temps référence est utilisé dans les équations comme indiqué dans (ARINC, 2001) :

$$t_k = t - t_{oe} \quad (2.1)$$

Où :

- t est le temps de simulation pour lequel on veut calculer la position du satellite.
- t_k est la différence entre le temps de simulation et le temps référence (relatif au fichier almanach utilisé).
- t_{oe} est le temps de référence par rapport aux éphémérides.

Il faut vérifier donc que le t_k calculé soit dans son intervalle de validité, sinon on doit le recalculer selon les équations suivantes :

$$\text{Si } t_k > 302400 \text{ alors } t_k \leq t_k - 604800 \quad (2.2)$$

$$\text{Si } t_k < -302400 \text{ alors } t_k \leq t_k + 604800 \quad (2.3)$$

Où 604800 est le nombre de secondes dans une semaine. Cette correction relève le fait que les paramètres orbitaux des almanachs sont valides pendant une semaine.

Après avoir vérifié la validité des paramètres orbitaux utilisés, nous pouvons procéder au calcul des positions des satellites au temps t comme décrit dans la norme GPS (ARINC, 2001). Dans le simulateur, ce calcul est effectué par une fonction, ‘CompCoord_{Sat}’ qui retourne les coordonnées du satellite à l’instant t en fonction des paramètres orbitaux comme suit :

$$\left[X_{sat}(t), Y_{sat}(t), Z_{sat}(t) \right] = \text{CompCoord}_{SAT} \left(A^{\frac{1}{2}}, M_0, \omega, e, I_0, \omega_0, \omega_{dot}, toe, tk \right) \quad (2.4)$$

Où $X_{sat}, Y_{sat}, Z_{sat}$ sont des vecteurs contenant les valeurs calculés pour tous les satellites de la constellation.

L’algorithme utilisé pour ce calcul est détaillé dans l’ANNEXE I. Il faut cependant noter que les variables à l’entrée de la fonction sont des paramètres de l’almanach, et non des éphémérides, ce qui veut dire que les résultats ne sont pas aussi précis que dans le cas réel. En effet, les premiers tests effectués avec ces algorithmes ne nous ont pas permis de converger vers une position au niveau du récepteur. Nous reviendrons sur ce problème dans la dernière partie de ce chapitre afin de montrer les corrections qui ont été apportés au niveau des algorithmes de calcul.

2.2.3 Calcul de la position du récepteur

Il s’agit dans cette fonction de faire appel au simulateur de trajectoires permettant de calculer les positions successives du récepteur en temps réel selon le scénario choisi. Ces positions sont calculées dans le repère cartésien ECEF par différentes fonctions dépendamment de la trajectoire qui est en train d’être simulée. Ces fonctions sont du type :

$$\left[X_{RX}(t), Y_{RX}(t), Z_{RX}(t) \right] = \text{CompCoord}_{RX}(\text{LastPositionRx}(t), \Delta t, \text{CurrentTrajectory}) \quad (2.5)$$

où :

- $X_{RX}(t), Y_{RX}(t), Z_{RX}(t)$ sont les coordonnées du récepteur le long de sa trajectoire à l'instant t de la simulation.
- $LastPositionRx$ est la position du récepteur à la seconde précédente.
- Δt est le temps de simulation écoulé.
- $CurrentTrajectory$ est la trajectoire actuellement simulée.

La tâche de validation du simulateur de trajectoire a été attribuée à d'autres membres du groupe ayant travaillé dans le projet, ce qui ne nous permet pas de creuser en détail cette partie du projet (Chaabane, 2009).

2.2.4 Calcul des distances géométriques

À partir des positions des satellites et des positions instantanées du récepteur, nous pouvons donc calculer la distance géométrique entre chaque satellite et le récepteur, $GeometricRange(t)$, en utilisant l'équation 2.6. Il s'agit d'une distance théorique qui n'inclut aucun délai, ni correction :

$$Gr(t) = \sqrt{(X_{sat}(t) - Y_{Rx}(t))^2 + (Y_{sat}(t) - Y_{Rx}(t))^2 + (Z_{sat}(t) - Z_{Rx}(t))^2} \quad (2.6)$$

Cette distance géométrique nous permettra de calculer la pseudo-distance en y incluant les délais atmosphériques, ainsi que les biais d'horloges et les effets relativistes.

2.2.5 Calcul de l'élévation et de l'azimut des satellites

L'angle d'élévation est la mesure entre le satellite et l'horizon. Cet angle est mesuré en degrés. Le calcul de l'élévation de chacun des satellites nous permet de définir en temps réel les satellites visibles au récepteur et ceux qui ne le sont pas. Ces informations sont aussi utilisées dans l'affichage de la constellation (*Skyplot*).

Le calcul de l'élévation d'un satellite à chaque instant de la simulation se fait selon l'équation suivante (Monnerat, 2007) :

$$\acute{E}lévation = \frac{\pi}{2} - a \cos\left(\frac{(X_{RX} \times (X_{SAT} - X_{RX}) + Y_{RX} \times (Y_{SAT} - Y_{RX}) + Z_{RX} \times (Z_{SAT} - Z_{RX}))}{(Gometric_Range \times Dist_{RX-Earth})}\right) \quad (2.7)$$

où :

- (X_{RX}, Y_{RX}, Z_{RX}) : sont les coordonnées du récepteur dans un référentiel cartésien ECEF.
- $(X_{SAT}, Y_{SAT}, Z_{SAT})$: sont les coordonnées du satellite dans un référentiel cartésien ECEF.
- *Gometric_Range* : est la distance géométrique entre le satellite et le récepteur.
- $Dist_{RX-Earth}$: est la distance du récepteur par rapport au centre de la terre.

Selon la valeur de l'élévation d'un satellite donné, nous pouvons décider si ce satellite est visible par le récepteur. En effet :

Si Élévation(deg) > 0 \Rightarrow *le Satellite est visible géométriquement.*

Si Élévation(deg) < 0 \Rightarrow *le Satellite est invisible géométriquement.*

La visibilité géométrique ne tient pas compte du patron de rayonnement de l'antenne du récepteur. Pour tenir compte de cette dernière, nous avons défini au niveau du simulateur le masque d'élévation, qui est utilisé comme suit :

Si Élévation(deg) > Masque d'élévation(deg) \Rightarrow *le Satellite est visible .*

Si Élévation(deg) < Masque d'élévation(deg) \Rightarrow *le Satellite est invisible .*

L'angle d'azimut représente la mesure entre la direction du satellite et la direction du Nord. Cet angle est mesuré en degrés. Au niveau du simulateur, l'angle d'azimut de chaque satellite est

obtenu à partir d'une fonction mathématique qui utilise les coordonnées du satellite ainsi que celle du récepteur. Cette fonction est du type :

$$Azimuth = Fct_Azimut(X_{Sat}, Y_{Sat}, Z_{Sat}, Latitude_{Rx}, Longitude_{Rx}) \quad (2.8)$$

où :

- $Latitude_{Rx}$ représente la latitude du récepteur en temps réel.
- $Longitude_{Rx}$ représente la longitude du récepteur en temps réel.

En ayant obtenu toutes les données (visibilité, élévation, azimuth) de chacun des satellites, nous pouvons afficher la constellation (*Skyplot*). La Figure 2.8 montre un exemple de visibilité pour le scénario suivant :

Constellation : *GPS (Fichier Almanach 040.alm)*
 Date : 12/02/2007 à 15 :00 :30
 Position du récepteur : *Latitude N 45deg 29' 00''*
 Longitude W 73deg 41' 00''
 Masque d'élévation = 5 degrés.

Ce scénario sera désormais appelé '*Scénario_Test*', et sera utilisé dans les parties de test et de validation des résultats obtenus avec le simulateur.

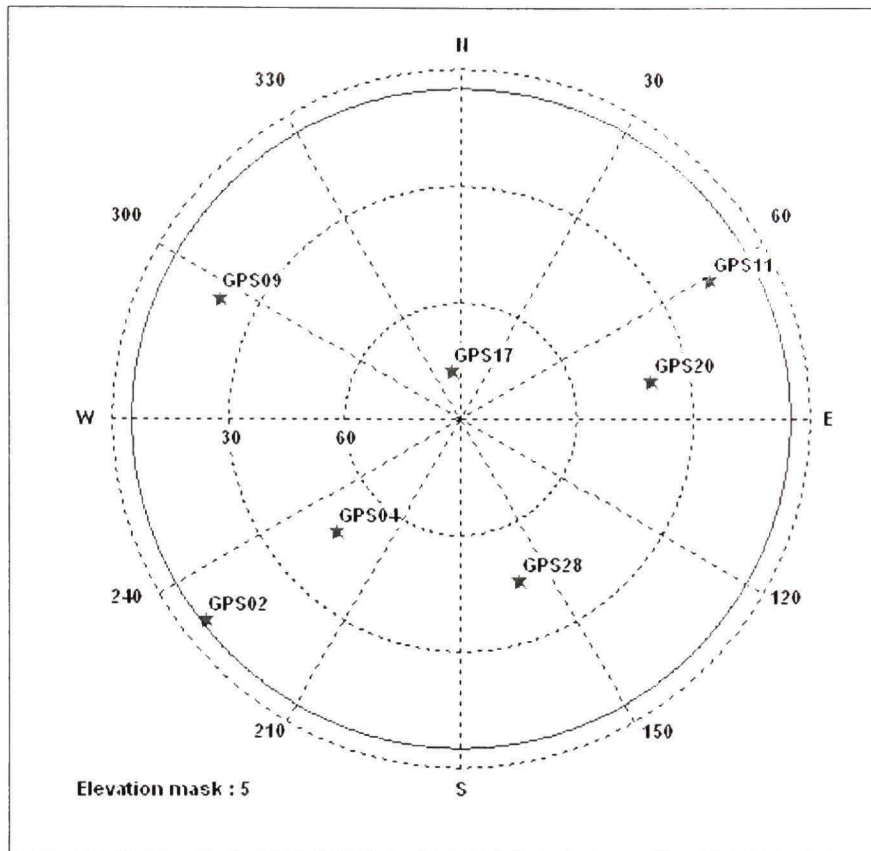


Figure 2.8 Affichage de la constellation(Skyplot) pour le 'Scénario_Test'.

2.2.6 Calcul des biais d'horloges

Le biais d'horloge est caractérisé par un ensemble de paramètres (voir section 2.1.1) qui sont envoyés dans le message de navigation, mais aussi qui peut être modélisé dans le signal RF généré. En effet, au niveau d'un récepteur GPS (Sauriol, 2007), les paramètres temporels reçus des satellites sont corrigés pour qu'ils soient les mêmes qu'au temps de leur émission. Au niveau du simulateur, nous faisons donc l'effet inverse en injectant cette erreur dans le signal. Le biais d'horloge, $B_{CLK}(t)$, se calcule selon l'équation 2.9 (ARINC, 2001):

$$B_{CLK}(t) = af_0 + af_1 \times (t - t_{oc}) + af_2 (t - t_{oc})^2 + \Delta t_r \quad (2.9)$$

où :

- af_0 , af_1 et af_2 sont les paramètres de corrections d'horloges définis dans la section 2.1.1.
- t est le temps GPS actuel en secondes.
- t_{oc} est le temps de référence des corrections d'horloges.
- Δt_r est un terme de correction relativiste.

D'ailleurs, selon la même source (ARINC, 2001), ce paramètre Δt_r se calcule selon l'équation 2.10 :

$$\Delta t_r = \frac{-2\mu^{1/2}}{c^2} \times eA^{1/2} \sin(E_k) \quad (2.10)$$

où :

- μ est la constante gravitationnelle universelle de la terre ($3,986005 \times 10^{14} m^3 s^{-2}$).
- c est la célérité de la lumière dans le vide ($2,99792458 \times 10^8 ms^{-1}$).
- e représente l'excentricité de l'orbite du satellite.
- $A^{1/2}$ est la racine carrée du grand axe.
- E_k est l'anomalie excentrique.

Nous verrons dans la section 2.2.8 comment ce biais d'horloge sera pris en compte dans le signal généré.

2.2.7 Calcul des délais ionosphériques et troposphériques

Plusieurs modèles existent pour calculer les délais ionosphériques comme nous pouvons le voir sur le Tableau 2.2:

Tableau 2.2 Modèles de calcul des délais ionosphériques.

Modèle	Implémentation dans le simulateur	Observations spécifiques
Fixed	Oui	Un délai fixe est spécifié lors de la configuration.
Off (Cas trivial)	Non	Le délai est fixé à zéro.
Terrestrial (Modèle de Klobuchar) (ARINC, 2001)	Oui	Il s'agit du modèle standard pour les utilisateurs du GPS à une seule fréquence (50% de correction des erreurs introduites par l'ionosphère) qui limite l'erreur à 5 mètres. (Sauriol, 2007)
Spacecraft (Klobuchar et Kunches, 2001)	Non	Modèle qui se base sur le calcul du TEC (Total Electron Content) 75% de correction.
Switched (Klobuchar et Kunches, 2001)	Non	Les modèles Terrestrial et Spacecraft sont utilisés en alternance suivant certains critères (altitude...).
Dual-frequency Observation (Sauriol, 2007)	Non	L'utilisation d'observations basées sur deux fréquences différentes permet de limiter l'erreur à 2,3 mètres.(Sauriol, 2007)

Autre que le modèle 'Fixed', le modèle ionosphérique que nous utilisons au niveau du simulateur, soit le modèle de Klobuchar (ARINC, 2001), permet de calculer le délai

ionosphérique en fonction des paramètres (α_i et β_i) configurés par l'utilisateur comme indiqué dans la section 2.1.3 :

$$D_{IONO}(t) = Fct_Ionosphere(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3) \quad (2.11)$$

Plusieurs modèles existent aussi pour calculer les délais troposphériques comme le montre le Tableau 2.3:

Tableau 2.3 Modèles de calcul des délais troposphériques.

Modèle	Implémentation dans le simulateur	Observations spécifiques
Modèle Hopfield	Non	Modèle étudié, mais non recommandé, car il représente moins de performances (Youcef, 2009)
Off	Non	Le délai est fixé à zéro
Modèle EGNOS	Oui	Dépend de la température, la pression et l'humidité.
Modèle STANAG	Non	Dépend des paramètres météorologiques.

Le modèle troposphérique que nous utilisons au niveau du simulateur, soit le modèle EGNOS, permet de calculer le délai troposphérique selon une fonction de ce type :

$$D_{TROP}(t) = Fct_TroposphericEffect(Elevation(t), Lat_Rx(t), D) \quad (2.12)$$

où :

- $Elevation(t)$: représente l'élévation du satellite en radian à l'instant t .
- $Lat_Rx(t)$: est la latitude du récepteur à l'instant t .

- D : représente le jour dans l'année. Le premier jour étant le 1^{er} Janvier.

2.2.8 Calcul de la pseudo-distance

La pseudo-distance est un paramètre qui nous permet d'ajouter tous les types de modélisations que nous voulons inclure dans notre signal RF. En effet, comme nous allons le voir, la pseudo-distance est un paramètre qui sera utilisé dans le calcul de la phase de la porteuse et du code d'étalement au niveau du FPGA. Ceci implique que tout changement au niveau de ces deux paramètres aura une conséquence directe sur la modulation des signaux RF que nous générons.

Nous effectuons donc le calcul de la pseudo-distance à partir des paramètres précédemment calculés suivants selon l'équation 2.13:

$$Pd(t) = Gr(t) - (B_{CLK}(t) \times c) + D_{Tropo}(t) + (D_{iono}(t) \times c) \quad (2.13)$$

Où :

- $Gr(t)$ est la distance géométrique entre le satellite et le récepteur au temps de simulation (en mètres).
- $D_{Tropo}(t)$ est le délai troposphérique au temps de simulation (en mètres).
- $D_{iono}(t)$ est le délai Ionosphérique au temps de simulation (en secondes).
- $B_{CLK}(t)$ est le biais d'horloge satellite au temps de simulation (en secondes).

La Figure 2.9 montre les valeurs de pseudo-distances calculées pour quelques satellites visibles pour le 'Scénario_Test', défini dans le paragraphe 2.2.5 sur une durée de simulation d'une heure (rouge : satellite 11, bleu : satellite 9, vert : satellite 2, rose : satellite 4). Étant donné que le calcul des pseudo-distances nécessite l'utilisation de tous les autres calculs effectués dans la DLL Matlab, la validation de la précision du calcul des pseudo-distances valide par conséquent tous les calculs effectués des autres algorithmes.

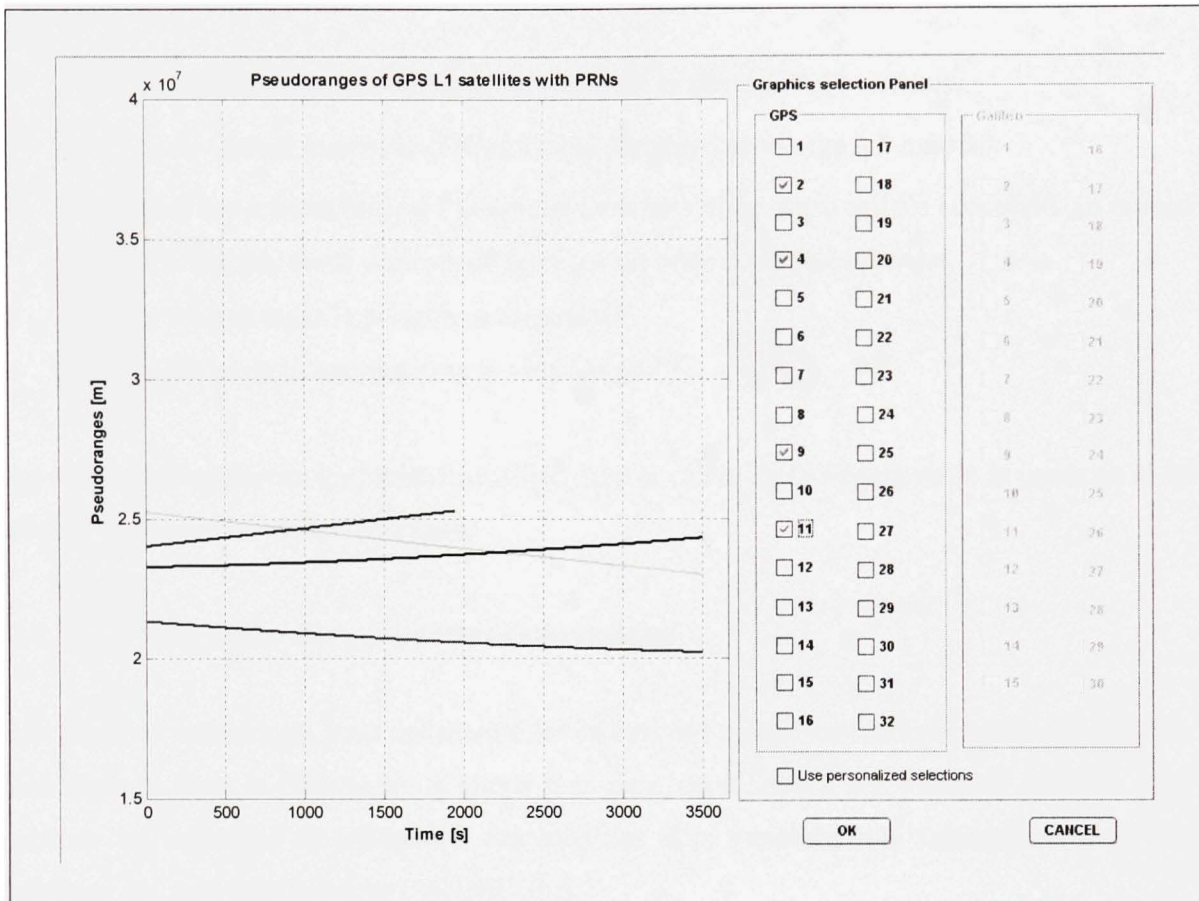


Figure 2.9 Évolution temporelle de la pseudo-distance.

2.2.9 Calcul du décalage Doppler

L'effet Doppler est le décalage fréquentiel entre la mesure à l'émission et celle à la réception d'une onde électromagnétique. Cette variation se produit en conséquence du mouvement relatif du satellite par rapport au récepteur, ou encore lorsque le récepteur se déplace au cours du temps.

Le décalage Doppler de chaque signal satellite est calculé selon l'équation 2.14:

$$D(t) = \left(\frac{Pd(t) - Pd(t-1)}{\Delta t} \right) \times \left(\frac{F}{c} \right) \quad (2.14)$$

où :

- $Pd(t)$ représente la pseudo-distance au temps de la simulation (en mètres).
- $Pd(t-1)$ représente la pseudo-distance de la seconde précédente (en mètres).
- Δt Durée d'un échantillon est l'intervalle de temps entre deux calculs successifs au niveau de la DLL Matlab. Cette durée a été fixée à 1 seconde au début du projet.
- F est la fréquence de la porteuse du signal RF.
- c est la célérité de la lumière dans le vide (en ms^{-1}).

Le décalage Doppler est un paramètre utilisé dans le calcul de la fréquence de la porteuse et du code d'étalement au niveau du FPGA.

2.2.10 Mise à jour des paramètres de navigation

Les sections précédentes nous ont montré les calculs des paramètres utilisés dans la modulation des signaux dans le FPGA. Nous allons voir dans cette section les variables utilisées pour générer les messages de navigation des satellites. Ces variables sont rassemblées dans des tableaux qui sont détaillés dans l'ANNEXE II.

Ces tableaux sont arrangés pour contenir successivement les paramètres de 32 satellites GPS et 30 satellites Galileo. Nous verrons d'ailleurs les limites de cette architecture dans la dernière section de ce chapitre. Voici donc la structure des tableaux utilisés :

- *ConstParamNavMessage* : contient les variables dont les valeurs ne changent pas au cours de la simulation.
- *VarParamNavMessage* : contient les variables dont les valeurs changent au cours de la simulation.
- *RangeParam* : contient les variables calculées relatives aux distances entre le satellite et le récepteur.
- *VisibilityParam* : un tableau de valeur booléennes qui indique pour chaque satellite s'il est visible ou pas.

- *TimeParam* : une structure de données qui contient les paramètres relatifs au temps de la simulation ainsi qu'à sa durée.

Les valeurs des tableaux sont mises à jour toutes les secondes par la DLL Matlab, et sont ensuite récupérés pour encoder les messages de navigation dans la partie C++.net de la partie logicielle.

2.3 Processus de génération des signaux

Ayant parcouru toutes les étapes de calculs effectuées dans la DLL Matlab, nous pouvons maintenant voir le processus par lequel les signaux satellites sont générés. 'WinSimu' est la partie logicielle développée en C++.net qui permet l'encodage des messages de navigation, d'effectuer les calculs nécessaires à la modulation des signaux, et de définir le moyen de communication avec le récepteur. Cette communication s'effectue de deux manières comme le montre la Figure 2.10 :

- La communication par le port série (RS232) dans laquelle le simulateur envoie au récepteur toutes les données calculées suivant un protocole de communication bien déterminé, sans avoir à générer des signaux satellite analogique.
- La communication IF dans laquelle le simulateur doit générer de vrais signaux satellites et les convertir de signaux numériques en signaux analogiques.

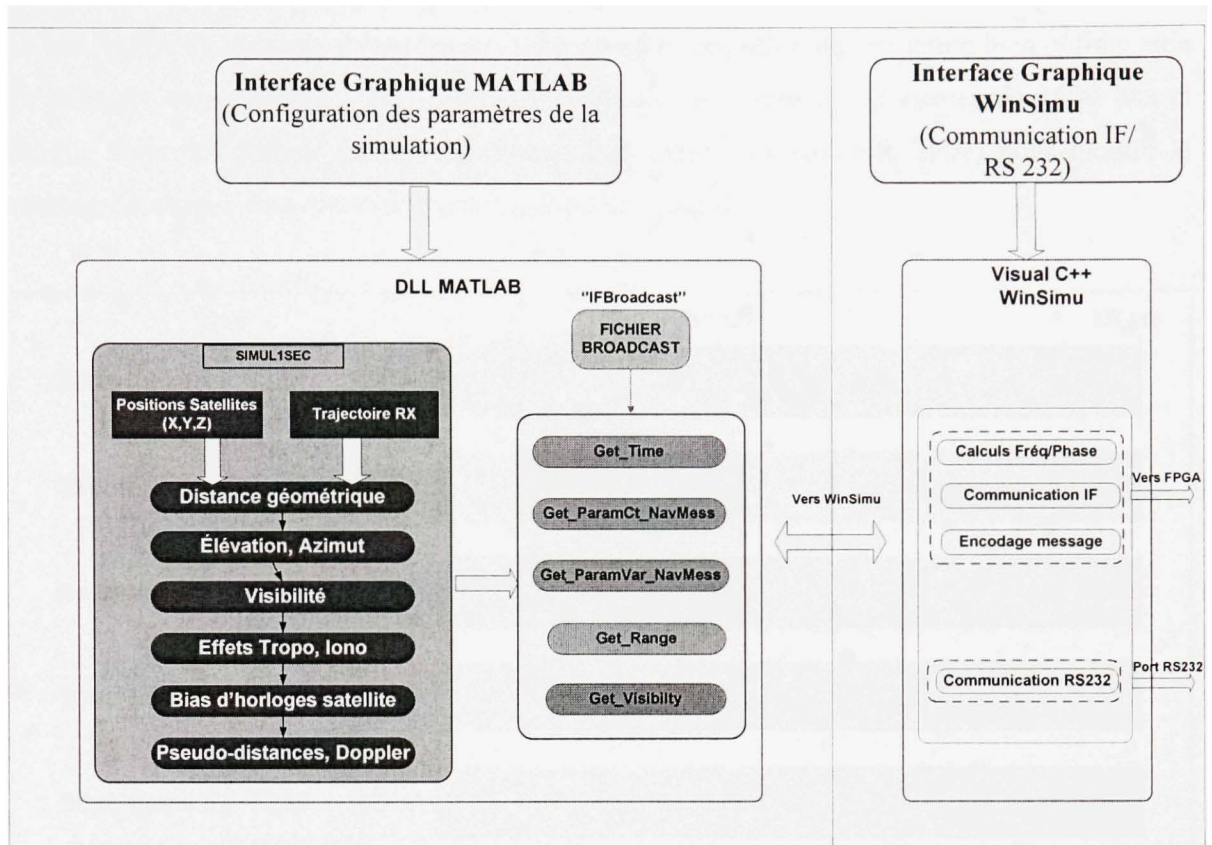


Figure 2.10 Processus de génération des signaux dans la partie logicielle.

Les données calculées au niveau Matlab nous permettent de générer ces signaux en quatre étapes :

1. Récupérer et structurer les variables à partir de Matlab à travers les fonctions 'Get_Time', 'Get_ParamCt_NavMess', 'Get_ParamVar_NavMess', 'Get_Range' et 'Get_Visibility'.
2. Générer les messages de navigation dans la partie 'WinSimu'.
3. Effectuer les calculs des fréquences et phases de la porteuse et du code dans la partie 'WinSimu'.
4. Transmettre les messages de navigation ainsi que les fréquences et les phases vers le FPGA dans la partie 'WinSimu'.

La génération du message de navigation GPS est effectuée selon une structure bien définie telle qu'indiquée dans (ARINC, 2001). Chaque message est formé de 25 trames de 1500 bits et chaque trame est formée par 5 sous-trames. La Figure 2.11 (Sauriol, 2007) nous montre le contenu de chaque sous-trame d'un message de navigation.

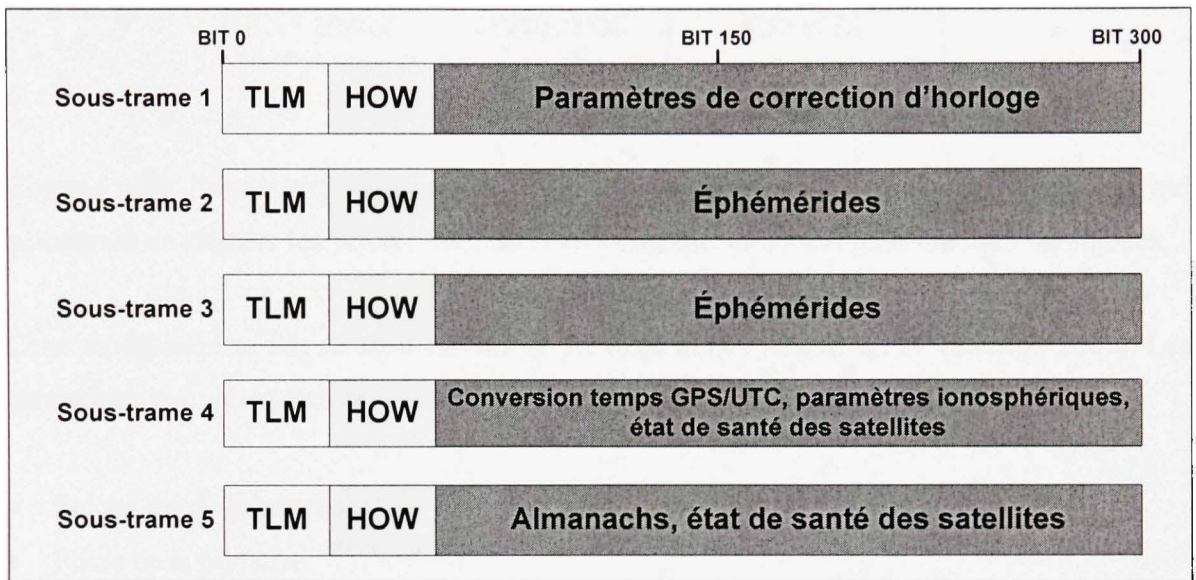


Figure 2.11 Structure des sous-trames du message de navigation GPS.

D'autres codes ont été développés et intégrés pour la génération du message de navigation pour la constellation Galileo en se basant sur les spécifications de l'Agence spatiale européenne (ESA, 2006). Cependant, aucun test n'a encore été effectué pour les valider (voir ANNEXE III). Le système Galileo utilise deux types de messages de navigation selon les services et les fréquences. Le Tableau 2.4 résume donc ces services en fonction de la bande de fréquence utilisée :

Tableau 2.4 Types de messages de navigation Galileo.

Type du message	Services	Bandes de fréquence
F/NAV	OS	E5a
I/NAV	OS/CL/SOL	E5b et E1

Comme nous l'avons déjà indiqué, la partie C++ du simulateur effectue les opérations qui permettent de calculer les valeurs des paramètres transmis au FPGA pour moduler les signaux.

Cette modulation se fait selon le schéma IF présenté dans l'ANNEXE IV (Sauriol, 2007). Les paramètres calculés et transmis sont :

- Fréquence de la porteuse.
- Phase de la porteuse.
- Fréquence du code.
- Phase du code.

Nous pouvons en effet calculer ces variables en fonction des valeurs des pseudo-distances et des Doppler (pour GPS L1) récupérés en temps réel à partir de la Matlab DLL suivant les équations :

$$Fp(t) = [F_{L1} + D(t)] \times \frac{2^{NB_Bits_Porteuse}}{100 \times 10^6} \quad (2.15)$$

$$Pp(t) = \left[\text{mod}(Pd(t), \lambda_{L1}) \times \frac{2\pi}{\lambda_{L1}} \right] \times \frac{2^{NB_Bits_Porteuse}}{2\pi} \quad (2.16)$$

où :

- F_{L1} est la fréquence L1 (1575,42 MHz).
- $Pd(t)$ représente la pseudo-distance au temps de la simulation (en mètres).
- $D(t)$ représente le décalage Doppler au temps de la simulation (en mètres).
- $NB_Bits_Porteuse$ représente la taille en bits des accumulateurs des NCO pour la porteuse (ANNEXE IV).
- Mod représente la fonction modulo.
- λ_{L1} est la longueur d'onde pour la fréquence L1.

Le code d'étalement est caractérisé par sa fréquence, ' $F_c(T)$ ', et sa phase ' $P_c(t)$ '. Les calculs de ces paramètres sont effectués selon les équations suivantes :

$$F_c(t) = \left[F_c + \left(\frac{F_c}{F_{L1}} \right) \times D(t) \right] \times \frac{2^{NB_Bits_code}}{100 \times 10^6} \quad (2.17)$$

$$P_c(t) = \left[\text{mod}(Pd(t), \lambda_c) \times \frac{2\pi}{\lambda_c} \right] \times \frac{2^{NB_Bits_code}}{2\pi} \quad (2.18)$$

où :

- F_c représente la fréquence du code d'étalement pour GPS L1 (1.023 MHz).
- $Pd(t)$ représente la pseudo-distance au temps de la simulation (en mètres).
- $D(t)$ représente le décalage Doppler au temps de la simulation (en mètres).
- NB_Bits_code : représente la taille en bits des accumulateurs des NCO pour le code d'étalement (Voir ANNEXE IV).
- λ_c représente longueur d'onde pour la fréquence du code d'étalement de GPS L1.

Nous pouvons donc conclure cette partie en confirmant que le générateur de signaux est essentiellement un '*convertisseur*' des données de pseudo-distances en signaux RF.

2.4 Validation des résultats pour GPS L1

Étant donné que la première partie de mon projet de maîtrise était de valider le bon fonctionnement du simulateur, il fallait prospector les états préexistants pour délimiter les parties où il faut intervenir. Un dispositif de test a donc été mis en place pour la validation des résultats des simulations comme le montre la Figure 2.12.

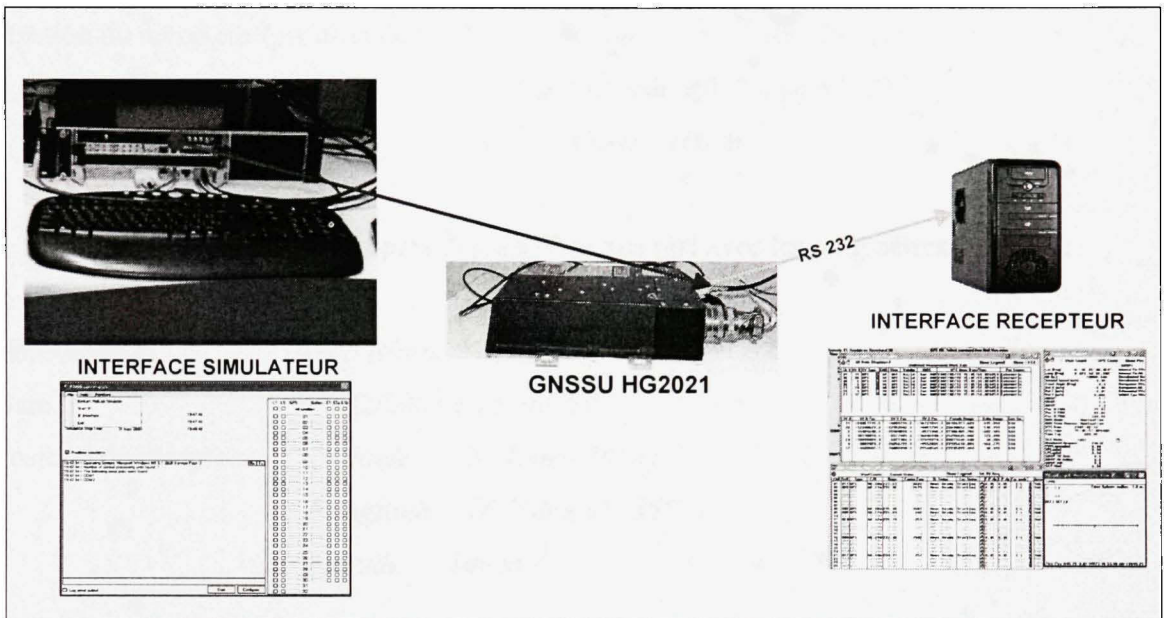


Figure 2.12 Banc de test.

Le banc de test est donc composé du matériel suivant :

- Le simulateur de constellation GPS/Galileo avec son interface usager '*WinSimu*'.
- Un récepteur commercial hybride GPS/Galileo de CMC '*GNSSU HG2021*' avec son interface usager '*PCMon*'.
- Un ordinateur de bureau relié au récepteur CMC par port série RS 232 dans lequel est installée l'application qui permet d'utiliser l'interface graphique '*PCMON*' du récepteur.

Deux scénarios ont alors été testés :

1. Le simulateur transmet les données préenregistrées sans effectuer de calculs. Ces données proviennent d'un fichier qui a été créé avec les informations enregistrées par un récepteur GPS. La date et la position du récepteur sont :

Date : 12/02/2007 à 15 :00 :30

Position du récepteur (position de CMC): *Latitude N 45deg 29' 41''*

Longitude W 73deg 41' 35''

Altitude 100 m

2. Le simulateur calcule ses propres données en temps réel avec les paramètres suivants :

Constellation : *GPS (Fichier Almanach 040.alm)*

Date : 12/02/2007 à 15 :00 :30

Position du récepteur : *Latitude N 45deg 29' 41''*

Longitude W 73deg 41' 35''

Altitude 100 m

Nous avons mis en place ces deux scénarios identiques pour pouvoir comparer rigoureusement les résultats des algorithmes utilisés. L'encodage des données et le matériel utilisé sont les mêmes pour les deux scénarios. Le constat était donc le suivant :

- Obtention de la bonne solution de navigation (PVT) au niveau du récepteur en utilisant le scénario 1.
- Pas de solution de navigation (PVT) au niveau du récepteur (ni celui CMC, ni celui de l'ÉTS) en utilisant le scénario 2.

Ces tests nous permettent de déduire que tous les calculs IF, ainsi que tous les schémas de modulation et la carte RF du simulateur fonctionnent correctement. La source d'erreurs, qui

nous empêche d'avoir une position au niveau du récepteur, se situe donc au niveau de la partie des calculs effectués dans la DLL Matlab.

La Figure 2.13 montre bien la différence entre les valeurs des pseudo-distances obtenus lors d'une simulation de 20 minutes pour le satellite 2 à titre d'exemple. La différence est de l'ordre de $1,5 \times 10^5$ mètres ce qui représente une source d'erreur qui engendre l'impossibilité d'avoir une position comme nous l'avons indiqué dans les résultats des tests.

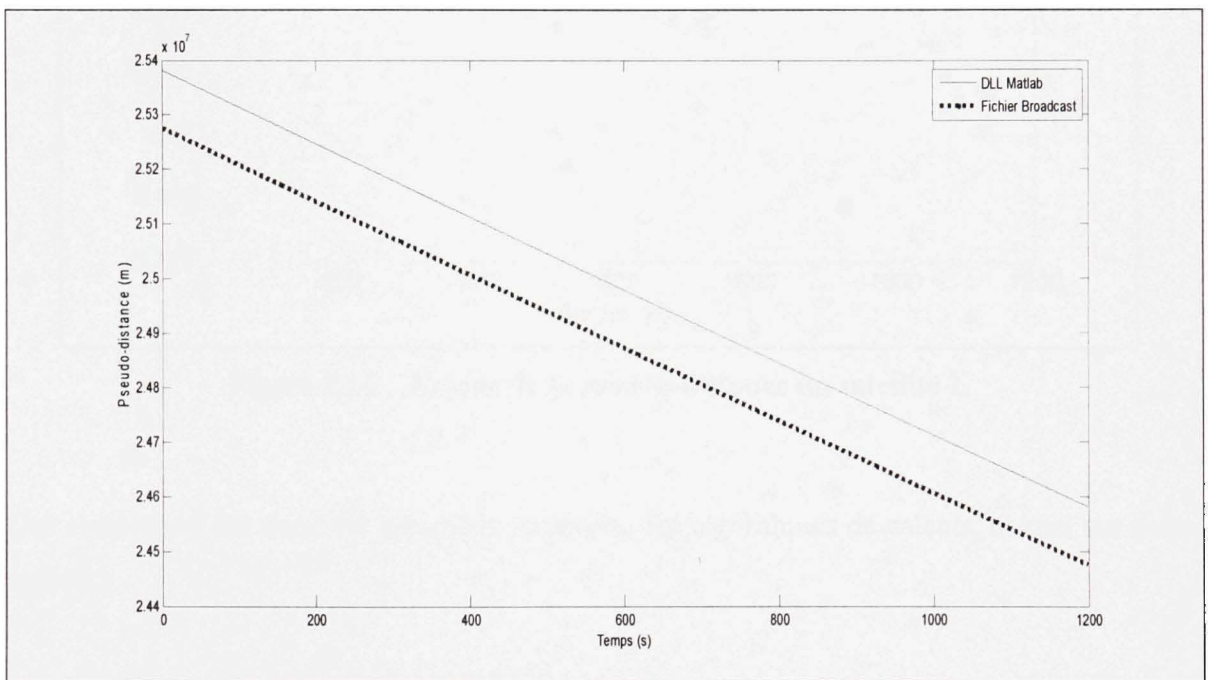


Figure 2.13 Comparaison des pseudo-distances (Satellite 2) pour le '*Scénario_Test*'.

Nous pouvons aussi quantifier l'erreur du calcul de la pseudo-distance par rapport aux valeurs préenregistrées sur la Figure 2.14, où nous remarquons un pourcentage d'erreur qui se situe dans les alentours de 0,42%.

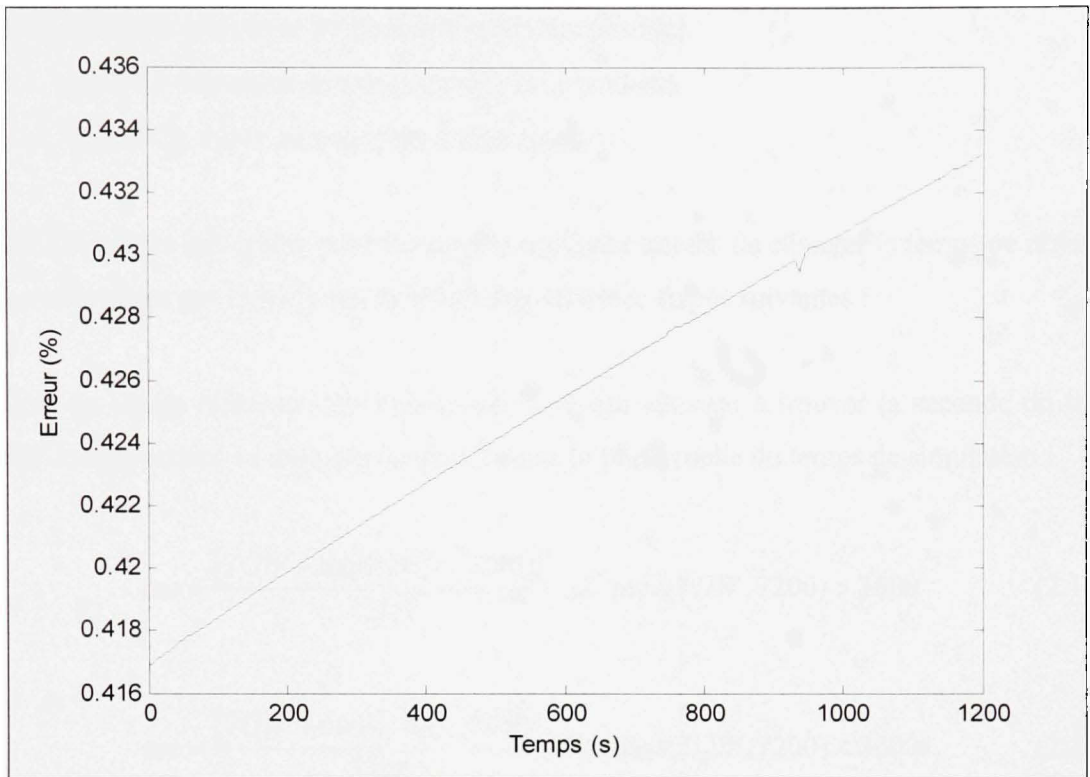


Figure 2.14 Erreur de la pseudo-distance du satellite 2.

Des corrections ont donc été apportées au niveau des algorithmes de calculs, et ceci sur deux aspects :

- Les paramètres du message de navigation.
- Le calcul des positions des satellites.

Corrections des paramètres du message de navigation

Comme nous l'avons mentionné précédemment, notre simulateur prend les variables d'un fichier almanach, alors que les données enregistrées proviennent d'un récepteur qui utilise les éphémérides. Pour corriger cette erreur, nous avons changé les paramètres orbitaux qui avaient un temps de référence par rapport à l'almanach. Ces paramètres sont :

- M_0 : L'anomalie moyenne au temps de référence (radian).
- I_0 : L'angle d'inclinaison au temps de référence (radian).
- ω_0 : L'ascension droite au nœud ascendant (radian).

Les modifications effectuées pour l'anomalie moyenne ont été de changer le temps de référence de ces paramètres par rapport aux éphémérides selon les étapes suivantes :

1. calcul du temps référence des éphémérides, ce qui consiste à trouver la seconde du temps GPS correspondant au multiple de deux heures le plus proche du temps de simulation :

$$toe = \frac{[TOW - \text{mod}(TOW, 7200)]}{7200} \quad \text{si } \text{mod}(TOW, 7200) > 3600 \quad (2.19)$$

$$toe = \frac{[TOW + \text{mod}(TOW, 7200)]}{7200} \quad \text{si } \text{mod}(TOW, 7200) < 3600 \quad (2.20)$$

où :

- TOW est la seconde de simulation en temps GPS.

2. Calcul de la nouvelle valeur de l'anomalie moyenne :

$$M_0_Eph = M_0 + n \times (toe - toa) \quad (2.21)$$

où :

- M_0_Eph est l'anomalie moyenne au temps de référence des éphémérides.
- M_0 est l'anomalie moyenne au temps de référence des almanachs.
- n est le mouvement moyen corrigé (Voir ANNEXE I).
- toe représente le temps référence des éphémérides.

- toa représente le temps référence des almanachs.

De même, les modifications effectuées pour l'angle d'inclinaison a été suivant l'équation suivante :

$$I_0_Eph = I_o + Idot \times (Toe - Toa) \quad (2.22)$$

où :

- I_0_Eph est l'angle d'inclinaison au temps de référence des éphémérides.
- I_o est l'angle d'inclinaison au temps de référence des almanachs.
- $Idot$ est le taux de variation de l'angle d'inclinaison.

Finalement, les modifications effectuées pour l'ascension droite au nœud ascendant a été la suivante :

$$w_0_Eph = w_0 + wdot \times (toe - toa) \quad (2.23)$$

où :

- w_0_Eph est l'ascension droite au nœud ascendant au temps de référence des éphémérides.
- w_0 est l'ascension droite au nœud ascendant au temps de référence des almanachs.
- $wdot$ représente la vitesse d'ascension droite.

Correction du calcul des positions des satellites

Des comparaisons avec d'autres produits existants sur le marché comme Spirent (SPIRENT GSS6700) ou encore la bibliothèque Matlab de SatNav (SatNav, 2007), nous ont montré que la source d'erreurs la plus importante dans notre DLL Matlab est la position des satellites, ce qui expliquerait bien une erreur de l'ordre de $1,5 \times 10^5$ mètres au niveau des pseudo-distances.

Plusieurs corrections ont été donc apportées au niveau des algorithmes de calcul des positions des satellites. Ces corrections consistent à :

- L'inclusion du temps de propagation des signaux dans le calcul de la position.
- Inclure l'effet de la rotation de la terre sur les coordonnées des satellites.

En effet, la position du satellite n'est pas la même lors de l'émission des signaux et leur réception. Il faut donc s'assurer de retrancher le délai de propagation des signaux lors de la génération des signaux.

Ce délai se calcule selon l'équation suivante :

$$\Delta t = \frac{\sqrt{(X_{RX} - X_{SAT})^2 + (Y_{RX} - Y_{SAT})^2 + (Z_{RX} - Z_{SAT})^2}}{c} \quad (2.24)$$

Où :

- (X_{RX}, Y_{RX}, Z_{RX}) sont les coordonnées du récepteur à l'instant du calcul.
- $(X_{SAT}, Y_{SAT}, Z_{SAT})$ sont les coordonnées du satellite à l'instant du calcul.
- c représente la célérité de la lumière.

D'un autre côté, le phénomène de la rotation de la terre autour d'elle-même est un phénomène dont les récepteurs prennent en compte (Sauriol, 2007). C'est pour cela qu'au niveau du simulateur, nous devons aussi modéliser cet effet dans le calcul des positions des satellites pour ne pas biaiser les résultats retrouvés lors des simulations.

En effet, le calcul des positions des satellites est à la base de la modélisation des signaux générés par le simulateur à chaque instant de la simulation. Si la terre tourne lors de la transmission des signaux, il est donc nécessaire d'utiliser les positions des satellites lors de l'émission des signaux et non pas lors de leur réception. Pour cela, il suffit de faire une rotation

inverse du référentiel dans lequel se trouvent les satellites avant de modéliser les signaux. Cette rotation est de $(-\Delta\theta)$ qui se calcule selon l'équation suivante :

$$\Delta\theta = \omega\dot{\times}\Delta t \quad (2.25)$$

Où :

- $\omega\dot{\times}$ représente la vitesse de rotation de la terre autour d'elle-même.
- Δt représente le temps de propagation du signal entre le satellite et le récepteur.

Ces modifications ont été apportées au niveau de la DLL Matlab, et le '*Scénario_Test*' a été de nouveau testé. Avec ces corrections, nous pouvons constater une correspondance quasi parfaite entre les nouvelles valeurs des pseudo-distances comparées à celles du fichier préenregistré comme le montre la Figure 2.15. Pour pouvoir quantifier la précision des calculs nous pouvons aussi voir sur la Figure 2.16 le pourcentage d'erreur entre les valeurs calculées dans la DLL Matlab et celles du fichier préenregistrée pour un même scénario (*Scénario_Test*). Dans la même figure, nous avons ajouté le pourcentage d'erreur obtenu en utilisant les données calculées par le simulateur SPIRENT GSS 7600 (SPIRENT, 2006). Les deux courbes sont superposables, ce qui montre la que la précision de notre simulateur est comparable à celle du produit commercial.

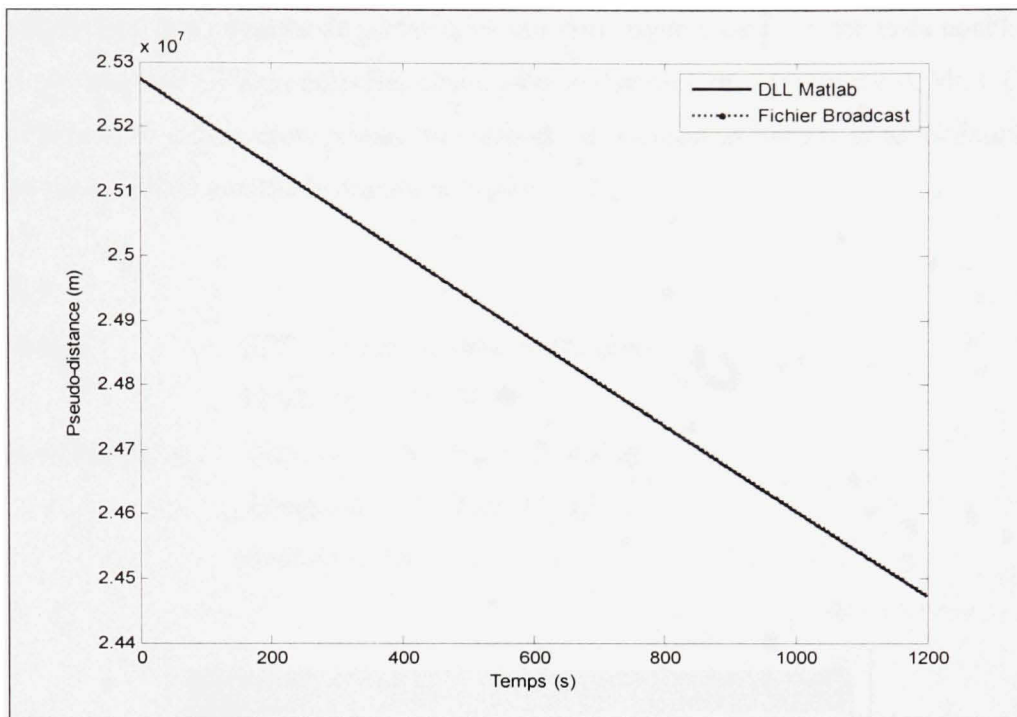


Figure 2.15 Valeurs des pseudo-distances pour le satellite 2.

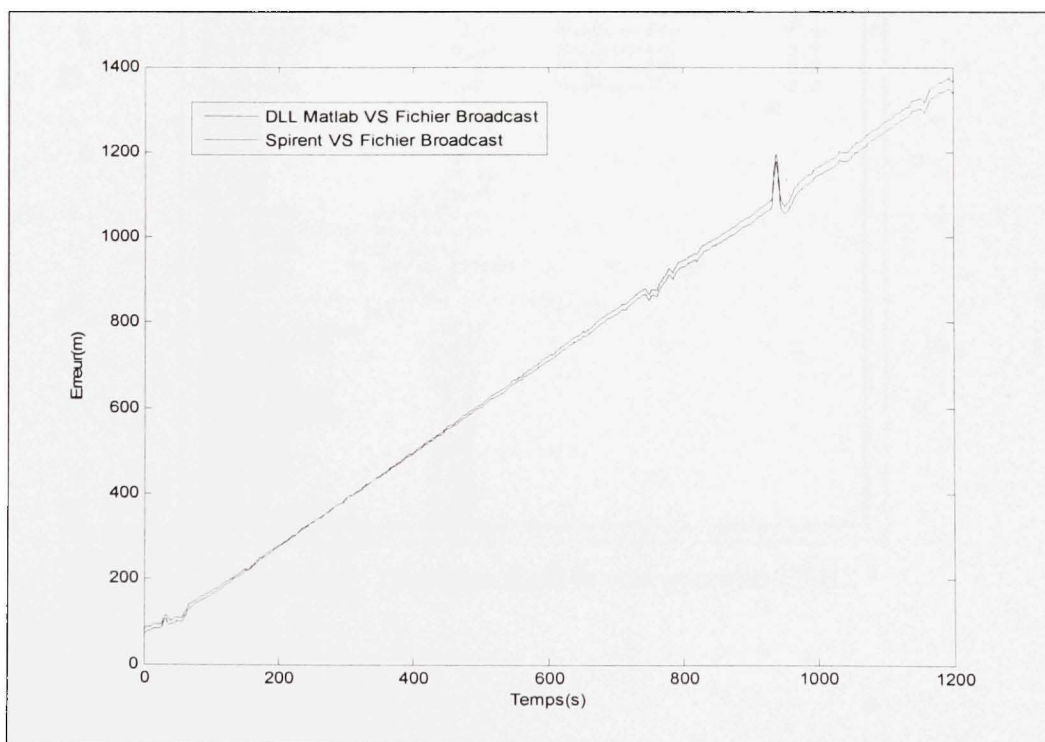


Figure 2.16 Erreur de calcul de la pseudo-distance.

Les résultats dont nous venons de parler nous ont donc permis de faire des tests concluants en utilisant un récepteur GPS commercial conçu pour le domaine de l'avionique (CMC). Ces tests nous ont permis de retrouver au niveau du récepteur la position simulée dans le 'Scénario_Test' que nous avons utilisé comme le montre la Figure 2.17 :

Scénario

Constellation : *GPS (Fichier Almanach 040.alm)*

Date : 12/02/2007 à 15 :00 :30

Position du récepteur : *Latitude N 45deg 29' 42''*

Longitude W 73deg 41' 35''

Altitude 100 m

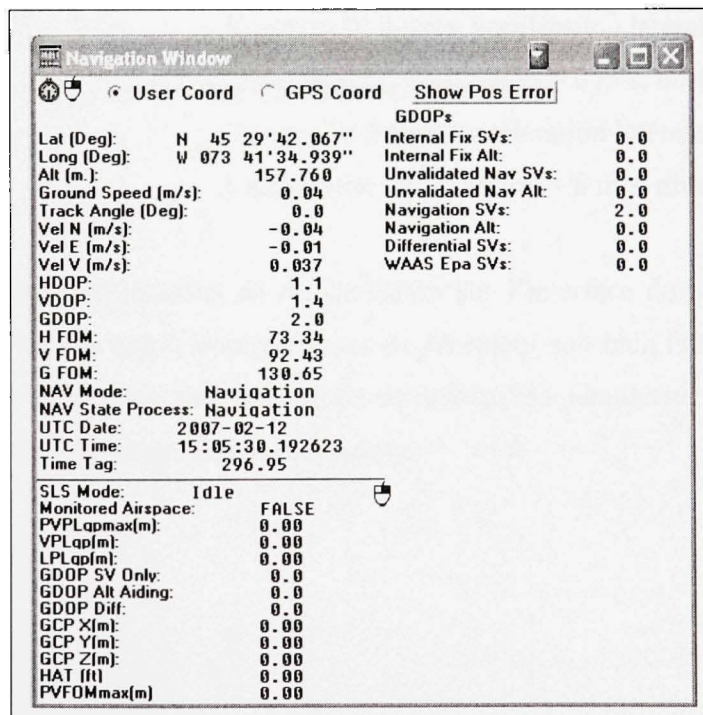


Figure 2.17 Position dans le récepteur de CMC.

D'autres tests ont été effectués avec ce même récepteur ainsi qu'avec le récepteur hybride GPS/Galileo développé à l'ETS. Ces tests nous ont permis de valider finalement les calculs effectués au niveau de notre simulateur pour des positions statiques ainsi que pour des trajectoires dynamiques.

En effet, nous avons défini un scénario au niveau du simulateur comme suit :

Constellation :	<i>GPS (Fichier Almanach 040.alm)</i>
Date :	12/02/2007 à 15 :00 :30
Position initiale du récepteur :	<i>Latitude N 45deg 29' 00''</i> <i>Longitude W 73deg 41' 00''</i>
Trajectoire du récepteur :	Ligne droite (accélération = 5 m/s, durée = 80 secondes) Virage (180 degrés, accélération latérale = 0.5 m/s) Ligne droite (accélération = 0 m/s, durée=240 secondes) Virage (90 degrés, accélération latérale = 0.5 m/s) Ligne droite (accélération = 0 m/s, durée=240 secondes).

La Figure 2.18 montre les résultats de la simulation sur l'interface du récepteur de CMC à altitude fixe. Nous voyons que le comportement du récepteur suit bien la trajectoire défini. Le simulateur présente cependant des limitations au niveau du simulateur de trajectoire. Ces limitations seront détaillées dans la section suivante.

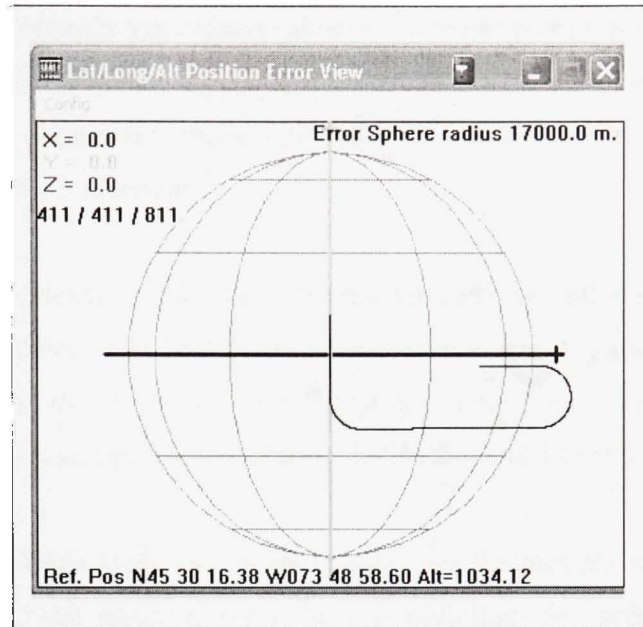


Figure 2.18 Niveau d'erreur de la position (Scénario 2).

2.5 Limitations du simulateur

Malgré les résultats très satisfaisants obtenus avec les corrections effectués au niveau des algorithmes de calculs, le simulateur présentait encore de grandes limitations qui ne peuvent pas être éliminées avec sa structure logicielle hybride (Matlab/C++). En effet, nous avons noté, tout au long du processus de validation effectué lors de ce projet, plusieurs désavantages associés à l'utilisation de Matlab :

- Difficulté à effectuer des changements au niveau des algorithmes de calculs. En effet, à chaque modification effectuée, une nouvelle DLL doit être générée en utilisant une version bien déterminée de Matlab, puis l'intégrer dans le projet pour pouvoir effectuer de nouveaux tests.

- Impossibilité d'avoir accès aux valeurs calculées en temps réel : Ceci est dû au fait qu'on ne peut accéder aux algorithmes qui sont inclus dans la DLL Matlab, car elle renvoie juste des tableaux contenant des données mises à jour toutes les secondes. Ceci rend le débogage, lors des tests, très difficile à effectuer.
- Une architecture logicielle rigide. En effet, des tableaux de tailles fixes sont utilisés pour transmettre les données de la DLL Matlab vers les autres parties du simulateur. Ceci rend impossible l'ajout de nouvelles fonctionnalités telles que l'ajout d'une nouvelle constellation, ou la mise en œuvre de canaux dédiés aux multi-trajets.
- Une interface graphique ayant des fonctionnalités très limitées par rapport aux simulateurs commerciaux. Ceci est principalement dû à la définition des scénarios qui manque de flexibilité, ainsi qu'à l'impossibilité d'afficher les données en temps réel.
- Absence ou manque de commentaires dans les codes développés.

Outre ces limitations imposées par la structure logicielle, d'autres limitations au niveau de la simulation de la trajectoire ont été relevées. Ces limitations se traduisent principalement par :

- Les valeurs maximales d'accélération qui ne peuvent dépasser 10 m/s^2 pour la commande '*Straight*', et 0.5 m/s^2 comme accélération latérale pour la commande '*Turn*'.
- Des commandes qui ne fonctionnent pas correctement pour certaines très grandes distances, comme pour le cas de la commande '*Straight*'.
- Les commandes '*Climb/Descent*' ne fonctionnent pas pour certaines valeurs d'altitudes (inférieures à 1000 mètres).

Ces limitations devraient faire l'objet d'un travail dédié.

Bien que cette version logicielle du simulateur puisse répondre aux exigences initiales du projet, qui consiste à concevoir un simulateur capable de générer des signaux satellites avec certaines performances, elle ne peut pas répondre aux exigences d'un simulateur qui peut offrir une bonne alternative aux solutions disponibles sur le marché.

2.6 Conclusion

Dans ce chapitre, nous avons abordé les différentes étapes de la simulation d'une constellation ainsi que les différents types de paramètres utilisés lors d'un scénario (paramètres orbitaux, corrections des horloges des satellites, perturbations atmosphériques, paramètres relatifs au temps de simulation, trajectoire du récepteur).

Nous avons ensuite fait une description des algorithmes de calculs qui permettent de générer les signaux satellites souhaités, commençant par les positions des satellites, le calcul des délais relatifs à l'atmosphère ainsi qu'aux biais d'horloge, et finissant par le calcul des pseudo-distances et leur utilisation dans la modulation des signaux IF.

En dernier lieu, nous avons effectué des corrections sur les algorithmes de calculs des pseudo-distances qui ont été testés et validés en utilisant le récepteur de CMC.

Cependant, nous avons relevé des limitations du simulateur dont plusieurs sont dues à son architecture logicielle hybride (Matlab/C++). Dans le chapitre suivant, nous abordons ces limitations et proposons les solutions logicielles que nous avons adoptées.

CHAPITRE 3

Nouvelle architecture logicielle du simulateur

Dans le chapitre précédent, plusieurs limitations ont été mises en évidence dans la partie logicielle du simulateur. Ceci est surtout dû au manque de flexibilité de l'architecture utilisée, ainsi qu'au manque de fonctionnalités disponibles dans l'interface usager. Pour remédier à ces lacunes, nous commençons par une analyse des besoins d'une version améliorée du simulateur et nous prouverons la nécessité du développement d'une nouvelle architecture logicielle. Ensuite nous définirons les étapes de développement de cette nouvelle version que nous sommes en cours de mettre en place. Finalement, la validation des résultats du nouveau logiciel nous permettra positionner notre simulateur dans la chaîne d'innovation industrielle.

3.1 Analyse des besoins du nouveau logiciel

Dans cette section, nous menons une étude des besoins fonctionnels souhaités pour la nouvelle version logicielle. Nous commençons par détailler une liste de fonctionnalités inexistantes dans la première version logicielle du simulateur. Cette liste nous permettra de prendre des décisions sur l'évolution du projet en vue d'établir une nouvelle architecture logicielle mieux adaptée aux besoins identifiés. Les principaux éléments de cette liste sont énumérés et détaillés dans ce qui suit :

- L'affichage en temps réel des données de navigation. Ceci implique l'accès en tout temps, lors de la simulation, à des données tels que les positions, vitesses, élévation et azimut des satellites, la position et trajectoire du récepteur, les délais atmosphériques, les pseudo-distances et Doppler des différents satellites par rapport au récepteur.
- La modélisation des multi-trajets. C'est-à-dire que l'utilisateur pourra créer différents modèles de propagations et les appliquer sur les signaux générés en temps réel. Cette fonctionnalité sera revue plus en détail dans le dernier chapitre de ce mémoire.

- La mise en place d'une interface logicielle pour le contrôle de puissance des signaux RF à la sortie du simulateur.
- L'ajout d'autres constellations autres que GPS/Galileo. En effet, d'autres constellations sont en train de voir le jour tel que GLONASS et BEIDOO, ce qui rendra leur intégration une nécessité dans un futur proche. Ceci implique un travail au niveau logiciel, mais aussi au niveau matériel pour mettre en place des canaux additionnels et simuler, par conséquent, le plus de constellations possible.
- L'accès à distance au simulateur : c'est-à-dire qu'un utilisateur pourra à travers un logiciel installé sur son propre ordinateur accéder au simulateur et lancer des simulations à distance.
- L'intégration de Google Earth pour la visualisation en temps réel de la trajectoire du récepteur lors d'une simulation.
- La possibilité de configurer plusieurs scénarios dans l'espace de travail. Cette fonctionnalité permettra à l'utilisateur de pouvoir comparer différents scénarios lors de simulations successives, en ayant un accès facile à tous les paramètres modifiées.

Ces améliorations que nous désirons intégrer dans la nouvelle version du simulateur ont été classées dans le Tableau 3.1 selon un indice de priorité qui varie de 1 jusqu'à 5 (1 étant le moins important et 5 étant le plus important). Les indices de priorités ont été établis en fonction des paramètres suivants :

- Les besoins exprimés par des utilisateurs potentiels du simulateur.
- Faciliter les travaux de recherche futurs visant à améliorer la performance du simulateur.
- Les technologies utilisées dans les produits concurrents.
- La veille technologique dans le domaine des systèmes de positionnement par satellite.

Tableau 3.1 Indices des priorités des nouvelles fonctionnalités.

Fonctionnalité	Ordre de priorité
Affichage en temps réel des données de navigation	5
Utilisation de Google Earth	4
Modélisation des multi-trajets	4
Ajout d'autres constellations	4
Possibilité de configuration de plusieurs scénarios	3
Contrôle de puissance pour les signaux RF	2
Accès à distance	1

Étant donné que l'architecture originale du simulateur qui repose sur une DLL Matlab ne peut répondre à la majorité des ces nouvelles fonctionnalités, nous devons revoir l'architecture même du logiciel. Pour ce faire, nous suivrons donc dans notre démarche une méthodologie de conception logicielle (Futrell, Shafer et Shafer, 2002) qui met en œuvre les étapes montrées dans la Figure 3.1. Dans ce cadre, nous pouvons dire que les besoins à identifier dans la première étape sont les fonctionnalités existantes du simulateur augmentées de la liste du Tableau 3.1 ainsi que les critères de qualité suivants :

- L'exactitude : En effet, pour le positionnement par satellites, la précision est un critère extrêmement important, donc les algorithmes développés doivent effectuer des calculs exacts et très précis.
- L'extensibilité : C'est la possibilité d'adapter le logiciel à de nouveaux besoins. En effet, le simulateur doit être toujours extensible à de nouvelles fonctionnalités tels que l'ajout de différents types de véhicules par exemple, ou encore la simulation de nouvelles constellations de satellites.

- La réutilisabilité : c'est la capacité à récupérer des parties développées dans le projet pour les réutiliser dans un autre projet ou une autre solution.
- La vérifiabilité : Le logiciel doit être facile à tester.
- La portabilité : C'est la possibilité d'utiliser le logiciel développé sur différentes plateformes.

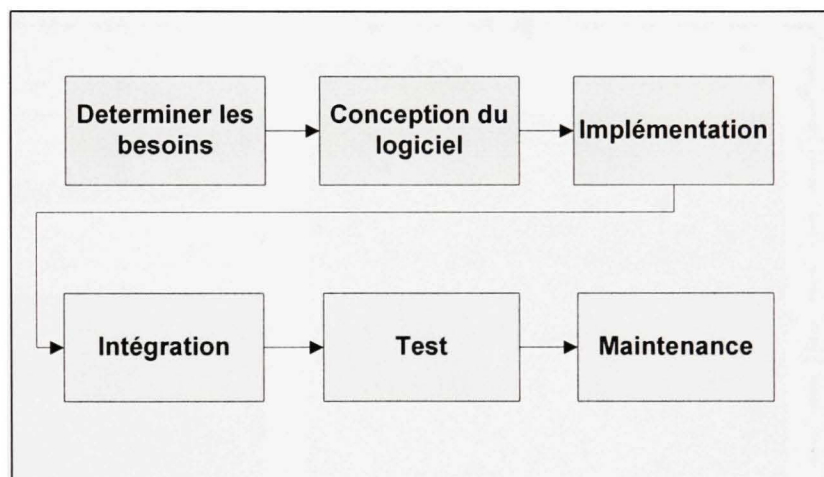


Figure 3.1 Étapes de développement d'un logiciel.

3.2 Conception du logiciel

Ayant déterminé les besoins de nouvelle version du simulateur, nous procédons dans cette section à la conception du logiciel.

3.2.1 Architecture logicielle proposée

Afin de mettre en œuvre un logiciel qui répond aux spécifications fixées, nous proposons une architecture logicielle modulaire articulée autour d'une application hôte à laquelle nous

pouvons greffer plusieurs extensions (Plugins) pour rajouter de nouvelles fonctionnalités comme le montre la Figure 3.2. Chaque extension est implémentée par une DLL dédiée.

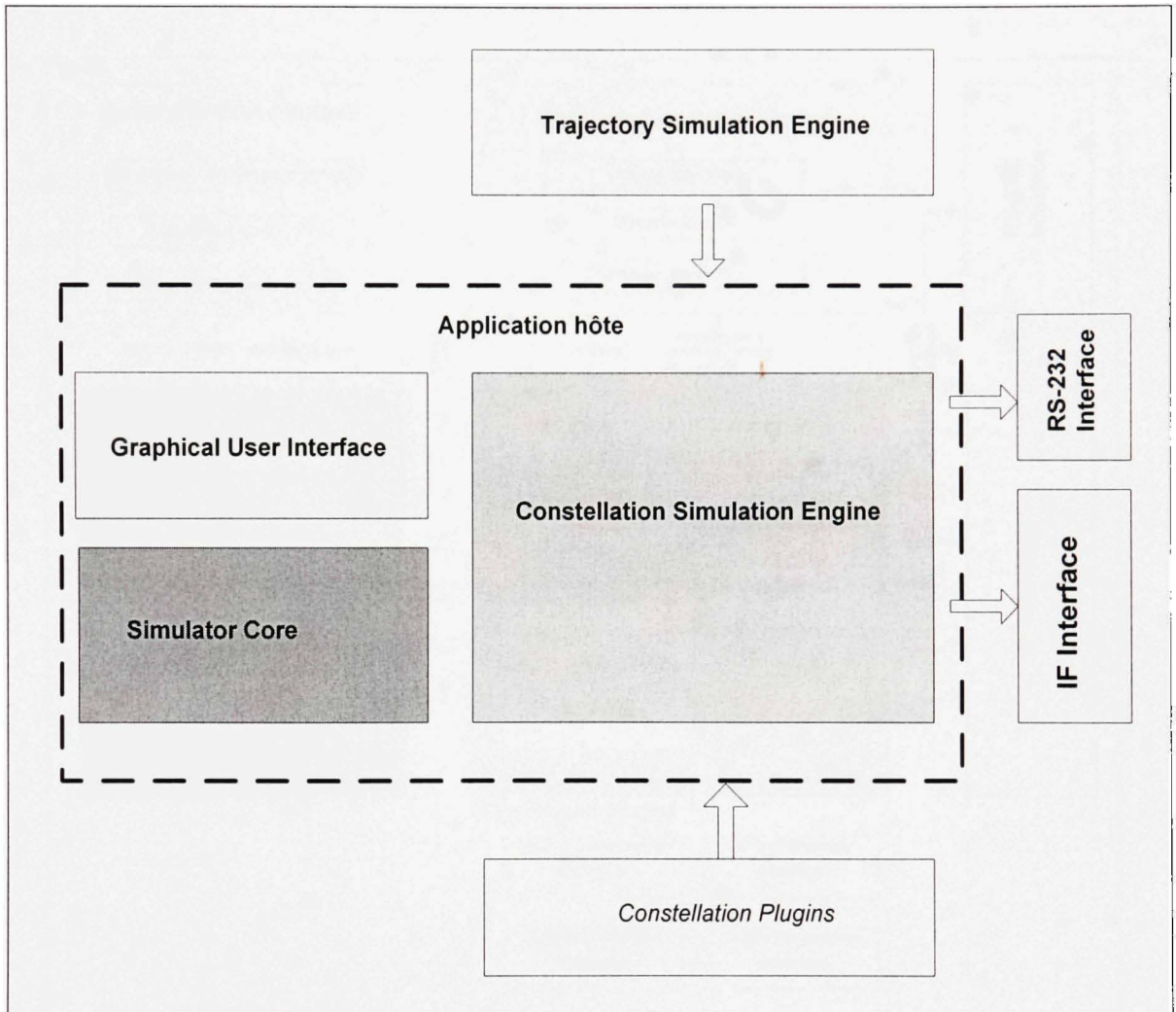


Figure 3.2 Schéma bloc de l'architecture logicielle.

Pour comprendre le principe de fonctionnement de cette nouvelle architecture logicielle, nous en faisons maintenant une description plus détaillée que celle effectuée dans le schéma en blocs. La Figure 3.3 montre, en effet, les interactions entre les différents modules. Cette architecture est basée sur la composante de l'application hôte '*Constellation Simulation Core*',

qui permet de la communication entre l'interface graphique et les autres modules, comme le simulateur de trajectoire. La composante '*Constellation Simulation Engine*' représente une interface entre l'application hôte et les extensions (Plugins).

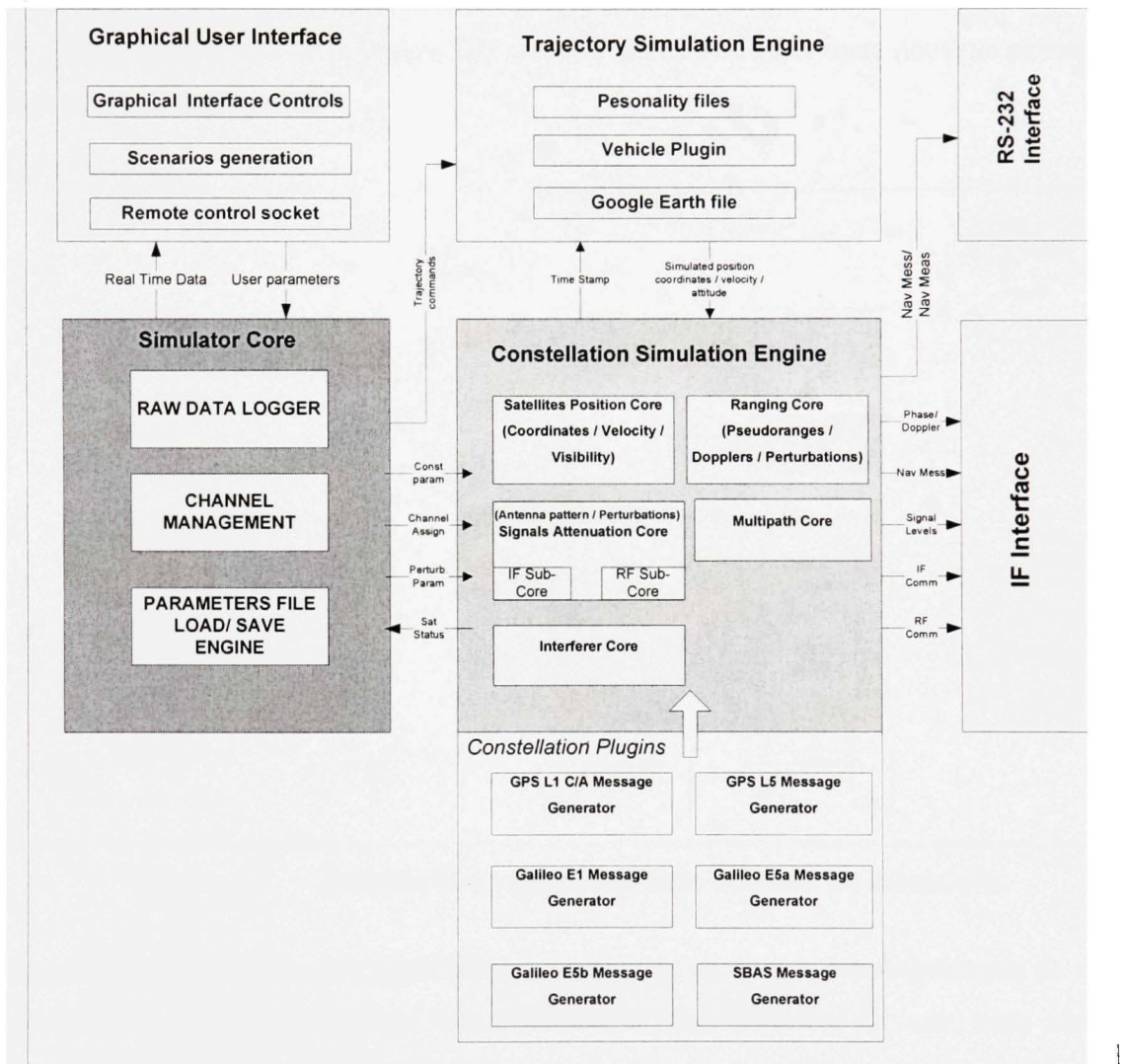


Figure 3.3 Description fonctionnelle de l'architecture logicielle générale.

L'architecture logicielle, comme la montre la Figure 3.4, présente une marge de manœuvre très intéressante tant au niveau de l'extensibilité du logiciel qu'au niveau de sa réutilisabilité. En effet, si nous voulons dans le futur améliorer notre simulateur en y rajoutant de nouvelles

constellations ou de nouveaux véhicules par exemple, il suffit de développer les plugins nécessaires et de les compiler avec l'application déjà en place. Ceci peut s'avérer très bénéfique dans le cas où de nouveaux intervenants (Étudiant, stagiaire...) veulent implémenter de nouvelles fonctionnalités. Ils pourront le faire sans prendre le risque de compromettre le bon fonctionnement de l'architecture déjà validée. L'application hôte, dont les composantes sont en vert dans ce diagramme de la Figure 3.4, est donc capable de gérer toute nouvelle extension et l'inclure dans l'application.

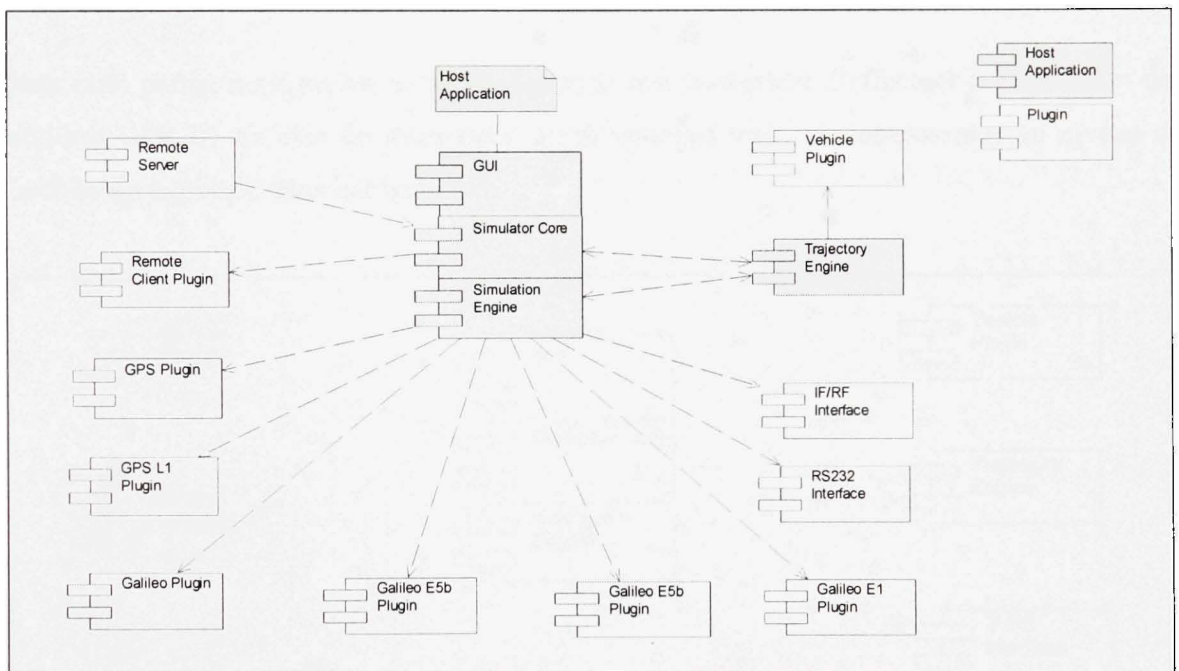


Figure 3.4 Architecture logicielle globale utilisant les extensions.

Cependant, cela nécessite une modélisation intelligente au niveau des diagrammes de classes qui répondent aux exigences d'une telle architecture. Dans la section suivante, nous décrivons les composantes utilisées dans la simulation d'une constellation GPS L1. Les autres extensions 'GPS L5', 'Galileo E1', 'Galileo E5a', 'Galileo E5b' n'ont pas encore été développés, mais leur structure est déjà en place au sein du simulateur.

D'autres composantes ne sont pas décrites dans la section suivante. Ceci est dû au fait qu'elles ont été développées par d'autres membres de l'équipe. Il s'agit des composantes suivantes :

- Le module de connexion à distance '*Remote Client*'.
- Le module de simulation de la trajectoire pour chacun des véhicules.
- Les extensions des véhicules '*Vehicle Plugin*' qui permet de définir les caractéristiques de chacun des véhicules simulés.
- Le module de la communication IF/RF qui permet de transmettre les données vers le FPGA.
- Le module RS 232 qui permet de transmettre les données directement au récepteur.

3.2.2 Simulation de la constellation GPS L1

Dans cette partie, nous parlerons des composants qui permettent d'effectuer la simulation des satellites GPS L1 au sein du simulateur. Nous pouvons voir ces composantes au niveau de l'architecture globale dans la Figure 3.5.

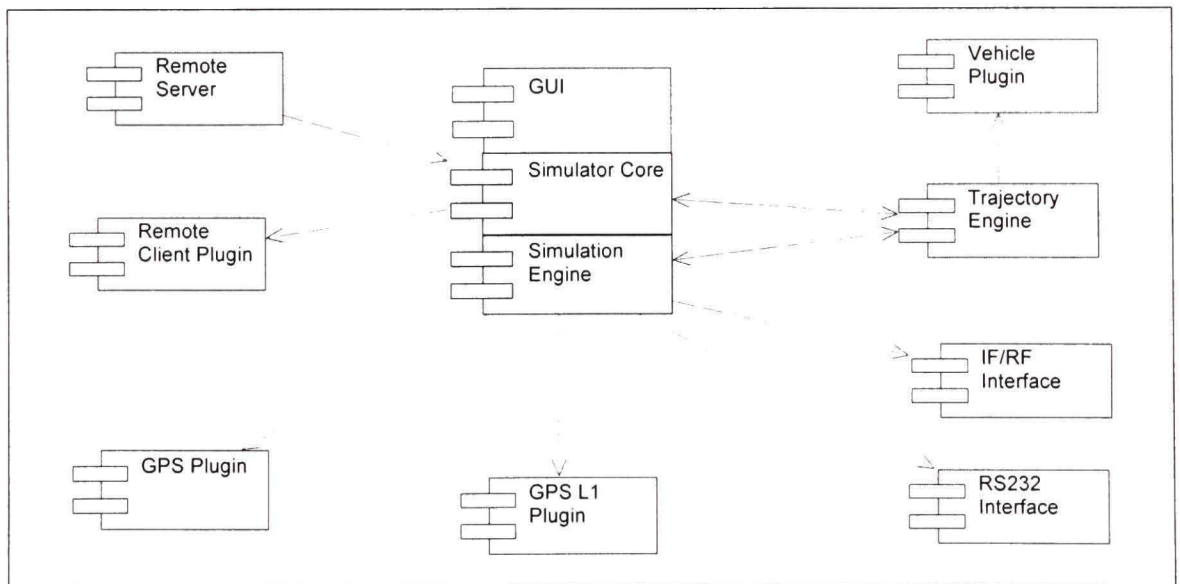


Figure 3.5 Composantes utilisées dans la simulation de la constellation GPS L1.

Dans le cas d'une simulation de la constellation Galileo, il suffit donc de remplacer les '*Plugins*' relatifs à GPS L1 et les remplacer dans le projet par ceux relatifs à Galileo E1 par exemple, comme le montre la Figure 3.6. Ceci met en évidence les critères de réutilisabilité et d'extensibilité dont bénéficie cette architecture logicielle.

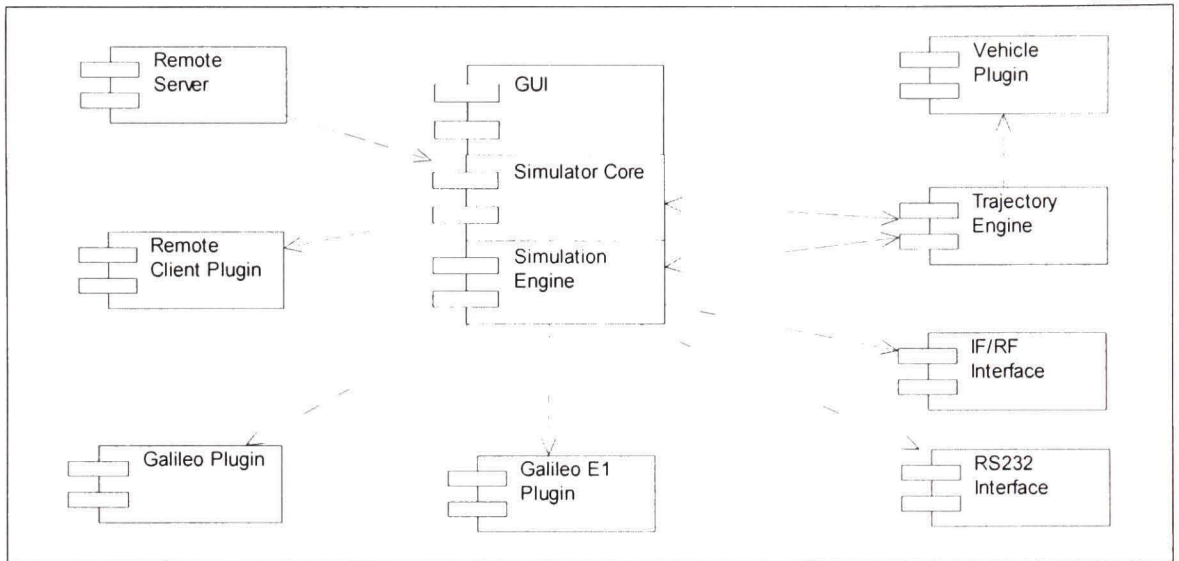


Figure 3.6 Composantes utilisées dans la simulation de la constellation Galileo E1.

Une telle architecture nécessite une modélisation intelligente au niveau des diagrammes de classes. La Figure 3.7 permet d’avoir une idée sur la manière avec laquelle les classes ont été modélisées, ainsi que les différentes relations d’héritage utilisées. Les deux classes ‘Satellite’ et ‘Frequency’ sont utilisées comme des ‘classes mères’ à partir desquelles héritent les autres classes. En effet, la classe ‘GPS’ est une ‘classe fille’ de la classe ‘Satellite’. La classe ‘GPS L1’ est une ‘classe fille’ de la classe ‘Frequency’.

Les classes mères sont des classes abstraites qui contiennent les méthodes qui vont être utilisées dans toutes leurs classes filles. Les méthodes qui sont implémentés dans les classes mères sont utilisés dans toutes les ‘classes filles’ de la même manière. Nous avons donc implémenté les méthodes de calculs qui vont être les mêmes pour toutes les constellations dans la classe ‘Satellite’. La même chose a été effectuée avec les méthodes de calcul implémentées dans la classe ‘Frequency’ et qui vont être utilisées dans toutes les classes ‘GPS L1’, ‘GPS L5’, ‘Galileo E1’, ‘Galileo E5a’, ‘Galileo E5b’.

D’autres classes sont utilisées dans la modélisation des composants utilisés dans la simulation de la constellation GPS L1. En effet, comme le montre le diagramme de classes de la Figure 3.7, pour GPS L1, nous avons 3 types de classes :

- Classes de Type 1 : ce sont les classes dans lesquelles sont implémentées des méthodes qui permettent d'effectuer les calculs nécessaires ('*Satellite*', '*GPS*', '*Frequency*', '*GPS LI*') ou encore de gérer l'interfaçage avec l'interface graphique ('*Simulation Engine*' et '*Simulation Core*').
- Classes de Type 2 : ce sont les classes qui permettent de gérer les données provenant de l'interface graphique lors de la configuration des scénarios ('*Time parameters*', '*Orbital parameters*', '*Sat Clock Errors*', '*Atmospheric Model*').
- Classes de Type 3 : ce sont les classes qui permettent de gérer les données qui résultent des calculs effectués dans les autres classes ('*Receiver Position*', '*Almanach Data*', '*Ephemeris Data*', '*Const_Result Data*').

Nous pouvons aussi noter les relations qui existent entre ces différentes classes. En effet nous pouvons noter par exemple :

- La relation d'agrégation : indique qu'une classe contient l'autre classe et interagit avec elle, comme c'est le cas pour la classe satellite avec les différentes classes de données avec lesquelles elle est en relation.
- La relation d'association : indique qu'une classe interagit avec une autre classe d'une manière générale, comme dans le cas entre la classe '*Simulation Engine*' et la classe '*Time Parameters*'.
- La relation d'héritage : indique qu'une classe est une spécialisation d'une autre et qu'elle en hérite toutes les propriétés, comme dans le cas de la classe '*GPS LI*' qui est une *classe* fille de la classe '*Frequency*'.

Toutes les propriétés et les méthodes implémentées dans ces classes sont clairement détaillées dans l'ANNEXE VI.

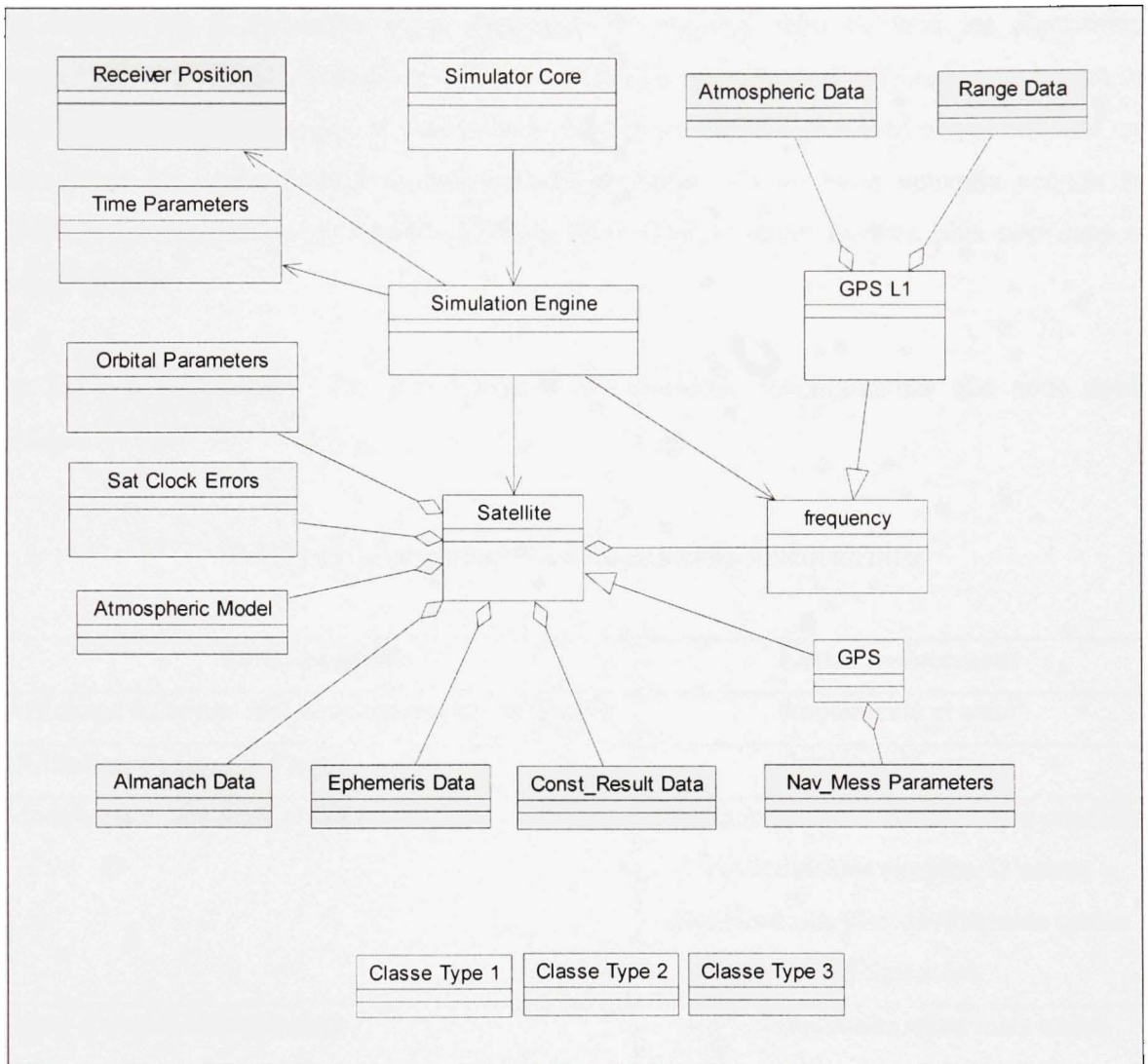


Figure 3.7 Diagramme des classes de la librairie de calcul GPS L1.

3.3 Implémentation

Cette section représente la troisième étape de la Figure 3.1. Tous les algorithmes de calculs ont été développés en C++.net alors que toutes les parties concernant l'interface graphique sont développées en C#.net. Le projet est compilé sous l'environnement Visual Studio 2005.

Au moment de la rédaction de ce mémoire, le développement de tous les algorithmes permettant la génération des signaux GPS L1 a été achevé. Ceci a été effectué en se basant sur les algorithmes qui ont déjà été validés sous Matlab, c'est-à-dire que nous avons effectué une conversion des codes Matlab en langage orienté objet C++.net et la nouvelle version est totalement indépendante de Matlab. Les résultats obtenus seront discutés plus tard dans ce même chapitre.

Le Tableau 4.2 montre l'état d'avancement des nouvelles fonctionnalités que nous avons définies au début de ce chapitre.

Tableau 3.2 Implémentation des nouvelles fonctionnalités

Fonctionnalité	État d'avancement
Affichage en temps réel des données de navigation	Implémenté et testé
Utilisation de Google Earth	Implémenté et testé
Modélisation des multi-trajets	Quelques tests ont été effectués pour des modélisations simples. D'autres modélisations plus développées seront implémentées.
Ajout d'autres constellations	Les extensions existent mais aucun développement n'a été effectué
Possibilité de configuration de plusieurs scénarios	Implémenté et testé
Contrôle de puissance pour les signaux RF	Implémenté et testé
Accès à distance	Implémenté et testé

Affichage en temps réel des données de navigation :

L'affichage est contrôlé par une barre de gestion des fenêtres d'affichage '*Display Manager*'. Nous pouvons donc choisir en tout moment d'afficher les valeurs des paramètres de la simulation en temps réel. La Figure 3.8 montre ces différents paramètres :

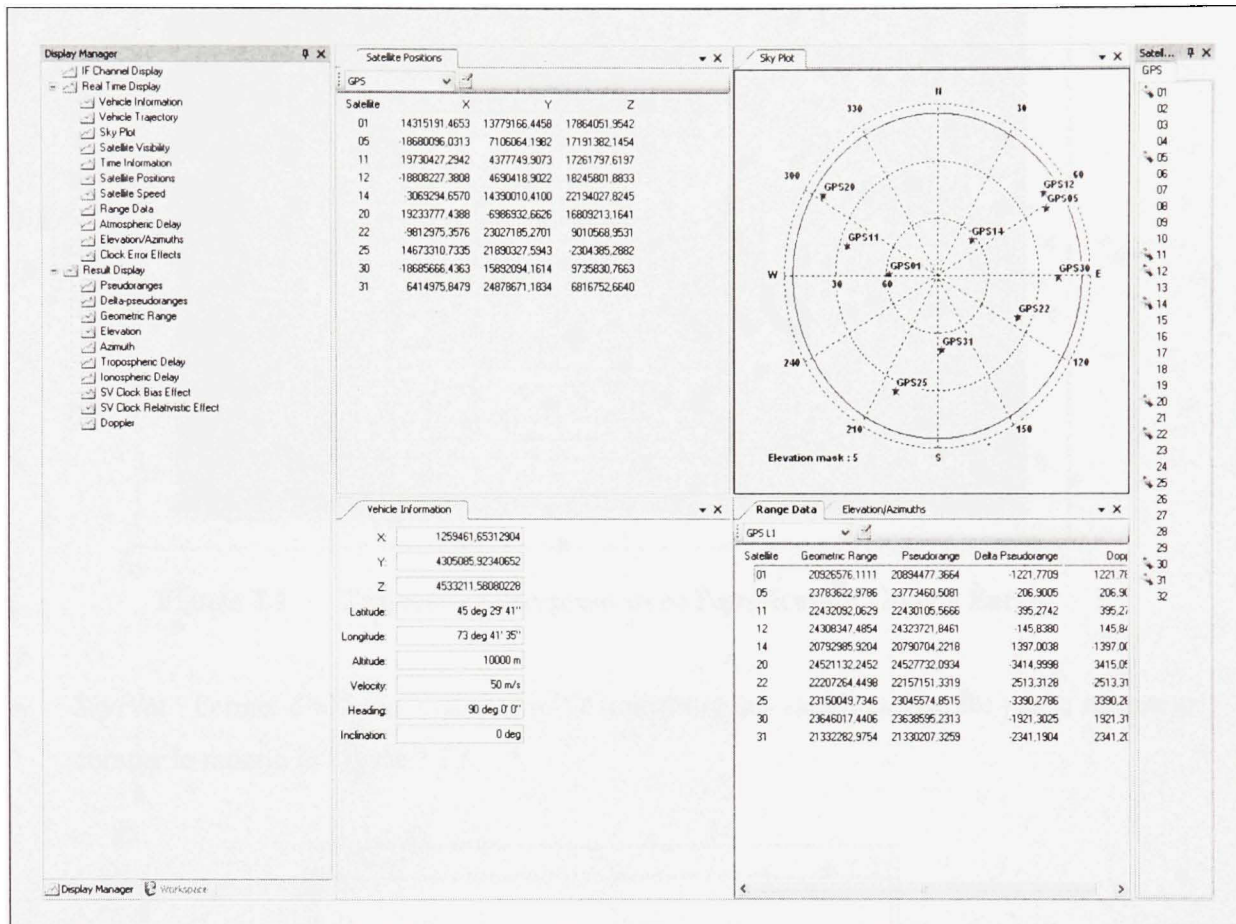


Figure 3.8 Affichage en temps réel.

- *Vehicle information* : Permet d'afficher en temps réel les coordonnées du récepteur dans un repère ECEF.
- *Vehicle trajectory* : Permet d'afficher la trajectoire du récepteur en utilisant l'application Google Earth comme le montre la Figure 3.9.

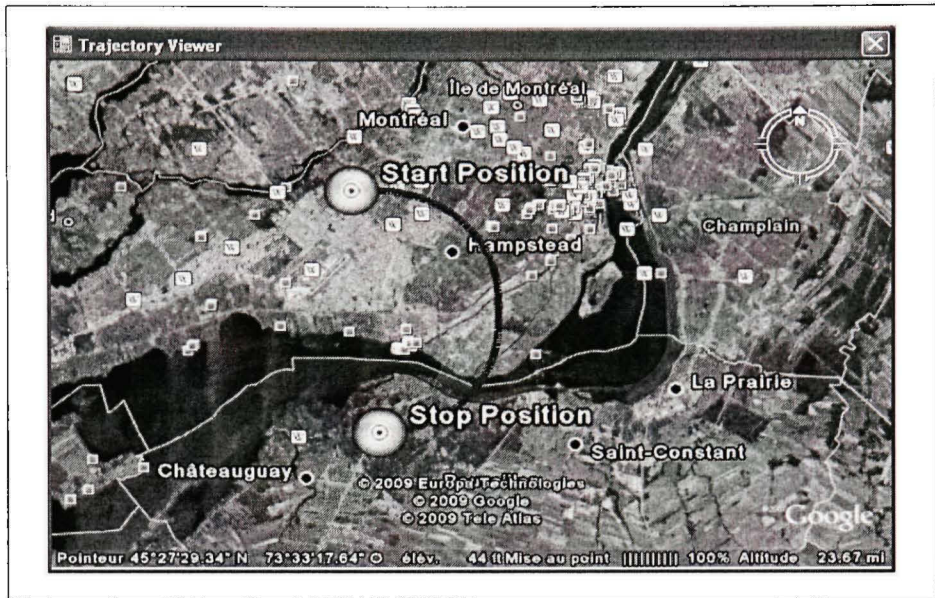


Figure 3.9 Trajectoire récepteur avec l'application Google Earth.

- *SkyPlot* : Permet d'afficher dans le ciel la trajectoire des satellites visibles par le récepteur comme le montre la Figure 3.10.

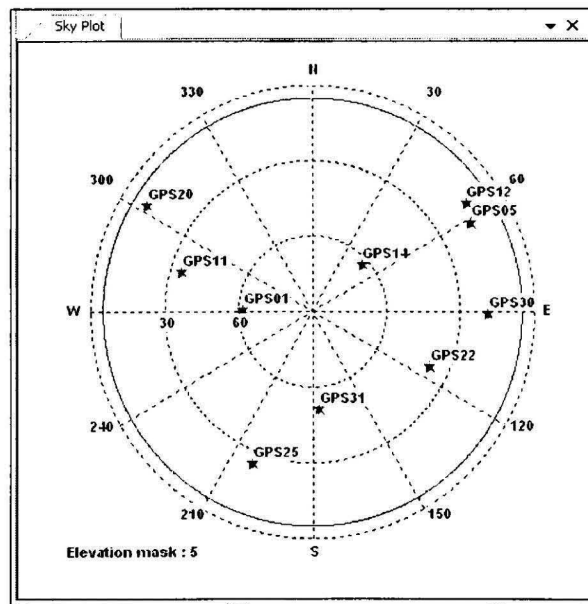


Figure 3.10 Fenêtre d'affichage du 'SkyPlot'.

- *Satellite Visibility* : permet d'afficher en temps réel les numéros des satellites visibles.
- *Time information* : permet d'afficher les données reliées au temps et à la durée de la simulation.
- *Satellite positions* : permet d'afficher en temps réel les coordonnées des satellites dans un repère ECEF. L'utilisateur peut choisir les satellites qu'il souhaite afficher.
- *Satellite Speed* : permet d'afficher en temps réel les vitesses des satellites dans un repère ECEF. L'utilisateur peut choisir les satellites qu'il souhaite afficher.
- *Range Data* : permet d'afficher en temps réel les données relatives aux distances entre les satellites et le récepteur, ainsi que les effets doppler.
- *Atmospheric Delay* : permet d'afficher en temps réel les données relatives aux délais ionosphériques et troposphériques.
- *Elevation/Azimuth*: permet d'afficher en temps réel les valeurs d'élévation et d'azimut des satellites.
- *Clock error effects*: permet d'afficher en temps réel les valeurs des paramètres de corrections d'horloges pour tous les satellites.

Modélisation des multi-trajets :

Une modélisation des multi-trajets a été implémentée et testée avec les signaux RF, mais cette partie n'est pas complétée. Nous verrons tous les détails concernant cette modélisation dans le dernier chapitre.

3.4 Validation des résultats pour la nouvelle architecture

Dans le second chapitre, nous avons validé tous les algorithmes implémentés au niveau de l'ancienne version logicielle. Dans cette section nous validons la nouvelle architecture logicielle ainsi que l'implémentation qui a été effectuée, incluant tous les calculs effectués pour générer les signaux GPS L1.

Ayant obtenu des résultats précis en utilisant la version utilisant la DLL Matlab (position fixe à l'intérieur de 5 mètres pour une durée de 20 minutes), nous nous sommes basés sur ces résultats pour effectuer des comparaisons des paramètres envoyés à la partie matérielle avec la nouvelle version du simulateur. Cette validation est menée en trois étapes en utilisant le scénario de test 'Scénario_Test':

- La comparaison des sous-frames du message de navigation générées.
- La comparaison des variables utilisées dans la modulation des signaux.
- L'utilisation d'un récepteur pour l'acquisition des signaux RF générés et la visualisation des résultats obtenues.

Nous allons utiliser dans cette démarche de validation le dispositif mis en place dans la Figure 3.11 avec le même 'Scénario_test' utilisé dans le chapitre précédent.

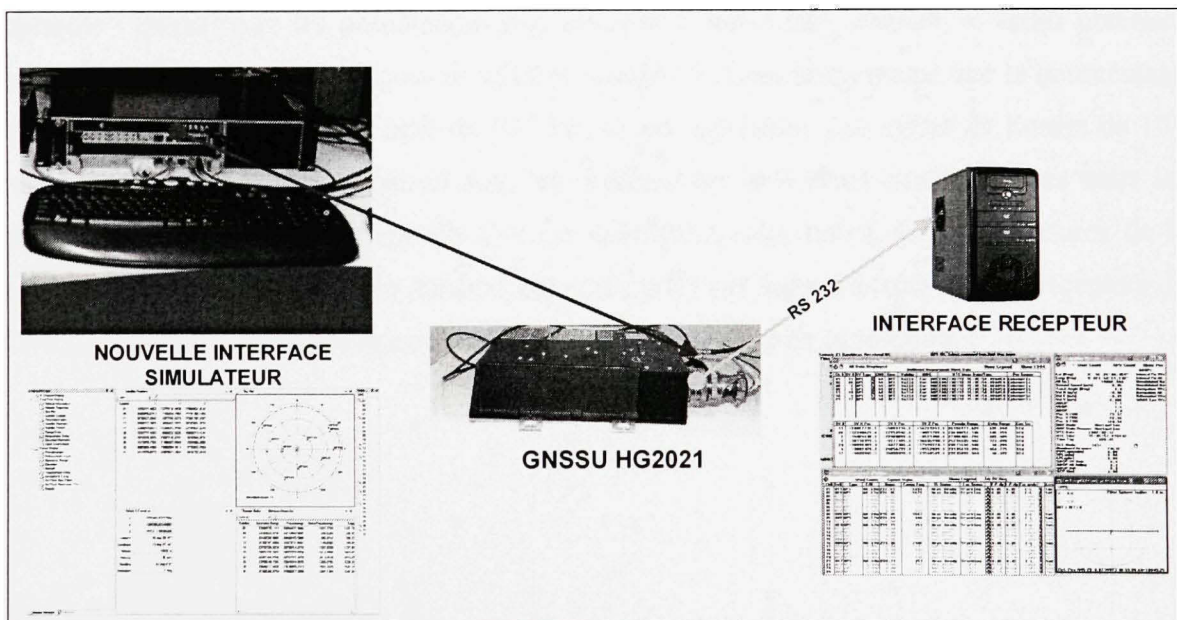


Figure 3.11 Banc de test du simulateur avec la nouvelle architecture logicielle.

Constellation : *GPS (Fichier Almanach 040.alm)*
Date : 12/02/2007 à 15 :00 :30
Position du récepteur : *Latitude N 45deg 29' 41''*
Longitude W 73deg 41' 35''
Altitude 100 mètres.

Dans la première étape nous effectuons une comparaison rigoureuse des toutes les sous-frames des messages de navigation des satellites simulés en les écrivant dans des fichiers binaires que nous générons lors de la phase de test. Nous avons trouvé que les messages de navigation étaient identiques. Ceci montre donc qu'en utilisant la nouvelle architecture logicielle, nous générons correctement les messages de navigation des satellites.

La deuxième étape nécessite, comme nous l'avons évoqué dans le second chapitre, une validation des valeurs des pseudo-distances entre les satellites et le récepteur. La Figure 3.12 présente l'erreur entre les pseudo-distances obtenues avec la DLL Matlab, et celles obtenues avec la nouvelle architecture, pour le satellite numéro 2. Nous remarquons que le pourcentage d'erreurs est très faible (de l'ordre de 10^{-6} %), ce qui représente une erreur de l'ordre de 10^{-1} mètre. Certaines petites différences dans les résultats des opérations mathématiques entre les bibliothèques de Matlab et celle du C++.net expliquent cette erreur dans les mesures de la pseudo-distance. Cependant, les résultats obtenus, suites aux tests effectués avec le récepteur de CMC, nous montrent que cette erreur n'affecte pas la précision de la solution.

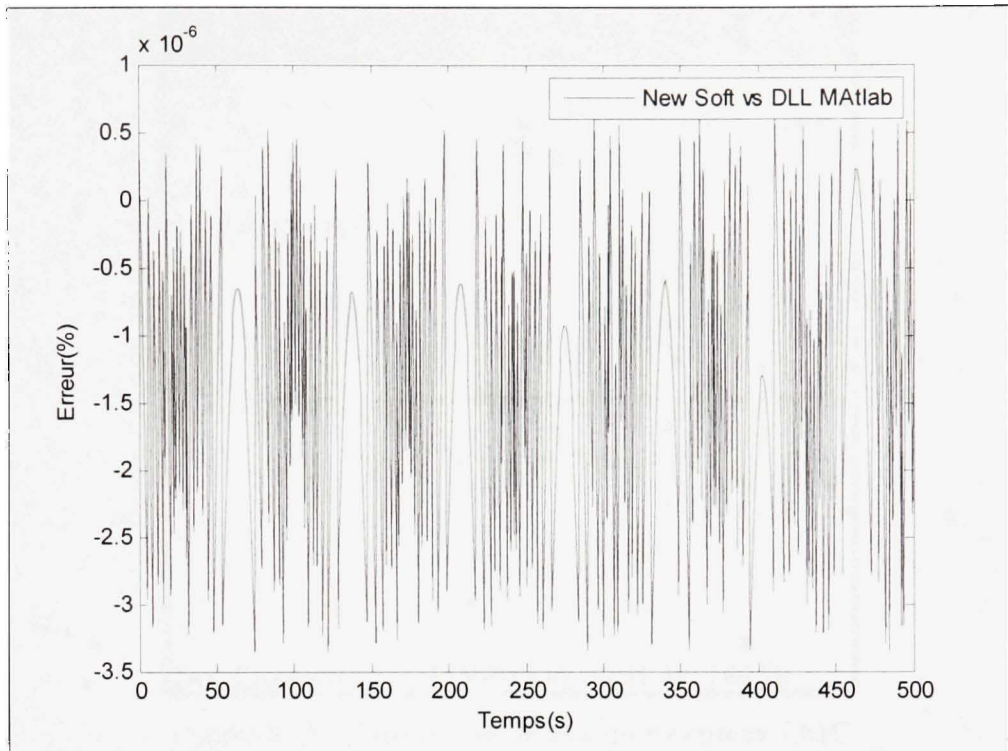


Figure 3.12 Pourcentage d'erreur dans la pseudo-distance entre l'ancienne et la nouvelle version du simulateur.

La troisième étape a été de valider la précision de la position obtenue en utilisant le récepteur de CMC dans un rayon de 1 mètre. Comme nous le montre la Figure 3.13, le récepteur converge vers une position PVT correcte et qui reste stable au bout de 6 minutes de simulation pour le même scénario :

Constellation : *GPS (Fichier Almanach 040.alm)*
 Date : *12/02/2007 à 15 :00 :30*
 Position du récepteur : *Latitude N 45deg 29' 42''*
Longitude W 73deg 41' 35''
Altitude 100 m

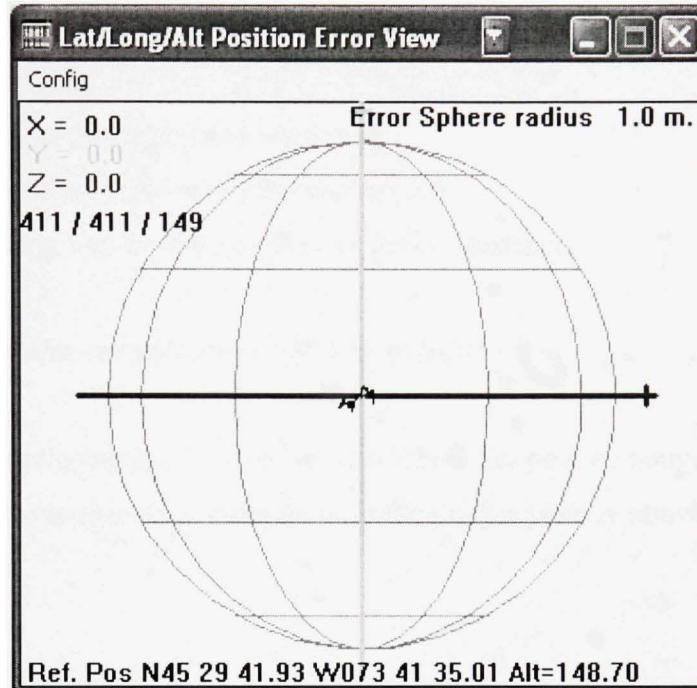


Figure 3.13 Position au niveau du récepteur CMC (6 minutes de simulation).

Outre la validation des résultats de calculs qui ont été présentés et l'obtention d'une position exacte au niveau du récepteur, ce qui représente d'ailleurs l'un des objectifs principaux du projet, nous avons également validés les nouvelles fonctionnalités logicielles qui ont été ajoutées dans la nouvelle version du logiciel du simulateur. L'ensemble des fonctionnalités suivantes ont été utilisées lors des tests effectués avec le récepteur CMC et toutes ont bien fonctionnées.

Les fonctionnalités intégrées et testées :

- Possibilité de définir une multitude de scénarios.
- Utilisation de l'extension Google Earth pour visualiser la trajectoire du récepteur lors de la simulation.
- Visualisation des valeurs de tous les paramètres de calculs en temps réel pendant la simulation.

- Visualisation des courbes représentant l'évolution de tous les paramètres de calculs au cours du temps.
- Modélisation des multi-trajets pour les signaux.
- Contrôle du bruit et de la puissance des signaux RF.
- Possibilité de lancer des simulations avec un accès à distance.

Les fonctionnalités non complètement validées et testées :

- Possibilité de développement d'extensions additionnelles pour de nouvelles constellations.
- Possibilité de développement d'extensions additionnelles pour de nouveaux véhicules.

Autre constats :

- Interface graphique de très haute qualité.
- La possibilité de définir des scénarios très rapidement en utilisant le '*Wizard*'.
- Une facilité d'utilisation et une clarté des outils de configuration.
- Une flexibilité au niveau de la gestion de la disposition des différentes fenêtres.
- Utilisation de seulement 37% des ressources du processeur alors que l'ancienne version utilise 30% des ressources avec beaucoup moins de fonctionnalités.

Comme mentionné précédemment, ce projet a été amorcé dans un contexte industriel, avec deux partenaires qui sont la compagnie CMC et l'agence spatiale canadienne. Les résultats atteints avec cette nouvelle architecture nous ont amené à l'idée d'effectuer un transfert technologique de ce projet. Dans ce contexte, il serait intéressant d'examiner le positionnement du simulateur dans la chaîne d'innovation.

3.5 Positionnement du simulateur dans la chaîne de l'innovation

Le transfert technologique, qui permet le transfert de découvertes résultantes de la recherche universitaire à l'industrie dans le but de les commercialiser, est un processus qui nécessite l'achèvement de plusieurs étapes. En effet, plusieurs étapes ont été franchies pour amener notre simulateur à ce niveau concurrentiel avec ce qui se trouve sur le marché. Il est alors envisageable à ce stade de considérer le simulateur comme un produit industriel qui peut être commercialisé.

Selon Porter (Porter, 1986), le développement de tout nouveau produit technologique se fait en plusieurs phases comme la montre la Figure 3.14. Actuellement, notre simulateur se positionnerait à l'étape 4 de cette chaîne d'innovation qui permet de montrer les résultats que nous avons atteints à nos clients potentiels. Plusieurs facteurs doivent être pris en considération pour que notre simulateur puisse franchir les étapes restantes et devienne un produit innovant commercialisable.

En effet, dans un contexte académique, l'indisponibilité des fonds et des financements représente souvent un obstacle de taille pour la continuité de projets de ce genre, qui demandent un travail continu à long terme. Les étapes 5 et 6 sont plus adaptées aux produits industriels qui nécessitent des procédés de fabrication. Dans notre cas, ces deux étapes représentent plus une étape d'intégration et d'assemblage.

La réalisation des étapes 5, 6, 7 et 8 nécessite la mise en place d'une équipe d'ingénieurs capables de continuer le développement de ce produit tant au niveau technique que commercial. Nous pouvons d'ailleurs émettre quelques suggestions qui peuvent s'avérer utiles :

- Prévoir une autre plateforme matérielle qui puisse intégrer plus de canaux dans le FPGA.
- Mettre les ressources matérielles et humaines nécessaires pour la réussite du projet.
- Rédiger un manuel d'utilisation qui permettra à l'utilisateur de comprendre toutes les fonctionnalités disponibles dans le simulateur.

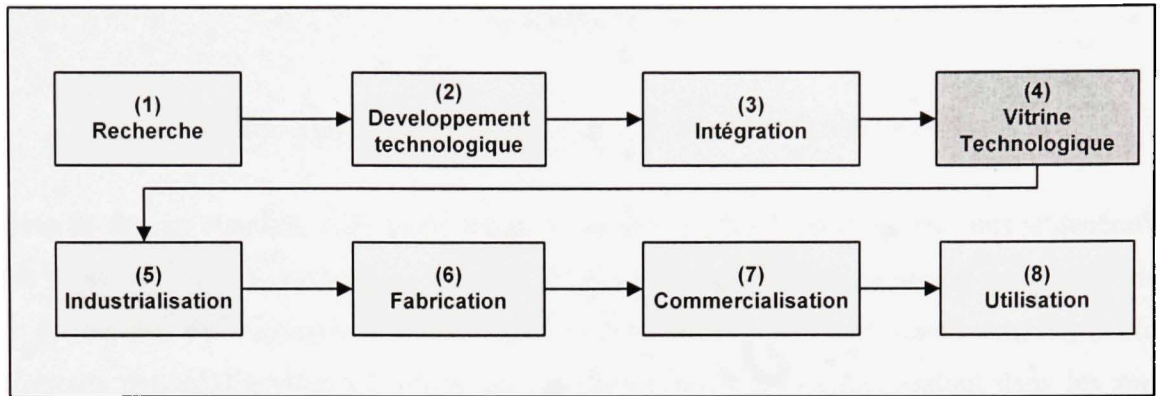


Figure 3.14 Chaîne de l'innovation d'un produit technologique.

3.6 Conclusion

Ce chapitre a été l'occasion de parler de la transition fondamentale qui a été effectuée au niveau de la partie logicielle du simulateur de constellations.

Nous avons d'abord eu une analyse des besoins fonctionnels auxquels ne pouvait pas répondre l'ancienne architecture du simulateur. Ensuite, nous avons décrit les composantes du nouveau logiciel utilisé ainsi que les diagrammes des classes associés. L'utilisation d'une architecture modulaire ainsi que l'utilisation de la programmation orientée objet nous ont permis d'améliorer certains critères tels que l'extensibilité, la réutilisabilité et la facilité de tester qui manquaient à notre ancienne version.

La dernière partie de ce chapitre a été consacré à la validation de la nouvelle architecture et son implémentation par l'examen des résultats obtenus avec cette nouvelle architecture. Le positionnement de notre simulateur dans la chaîne d'innovation de Porter a aussi été brièvement discuté.

CHAPITRE 4

Étude d'un modèle de multi-trajets pour les signaux GNSS

Dans ce dernier chapitre, nous parlerons de la modélisation des multi-trajets dans la génération des signaux GNSS. Cette fonctionnalité est devenue très importante au niveau des simulateurs, vu les besoins des concepteurs des récepteurs GPS. En effet, les multi-trajets sont des sources d'erreurs non négligeables au niveau du positionnement par satellite, surtout dans les zones urbaines, comme nous le verrons au cours de ce chapitre.

Nous parlerons d'abord des modèles de multi-trajets utilisés dans les communications sans fil, ensuite dans un contexte de positionnement par satellites. Par la suite, nous allons proposer les solutions possibles pour une modélisation efficace dans la nouvelle architecture du simulateur.

4.1 Introduction sur les multi-trajets

Nous pouvons définir les multi-trajets comme étant la propagation d'un signal radio suivant deux ou plusieurs trajets différents (Rappaport, 2002). Dans un environnement de multi-trajets, nous avons généralement la présence de deux types de signaux :

- Les signaux en vue directe (LOS) qui n'ont pas subi de réflexions durant leur propagation entre le transmetteur et le récepteur.
- Les signaux qui ne sont pas en ligne de vue directe (NLOS), c'est-à-dire les signaux qui ont subi une réflexion quelconque avant d'atteindre le récepteur.

La présence du signal en ligne de visibilité (LOS) dépend de l'environnement dans lequel se trouve le récepteur. Lors de la modélisation des multi-trajets dans un contexte de communications sans fil en milieu urbain, nous avons généralement la présence des composants réfléchis du signal (NLOS) alors que la présence du signal en ligne de visibilité

(LOS) n'est pas assurée. En milieu rural, le signal en vue directe (LOS) est généralement présent.

Un signal subissant l'effet des multi-trajets s'exprime sous la forme suivante :

$$s(t) = \alpha_0 e(t) + \sum_1^N \alpha_n e^{j\theta_n} e(t - \tau_n) \quad (4.1)$$

où :

- $e(t)$: Le signal original.
- $s(t)$: Le signal reçu par le récepteur.
- α_0 : L'atténuation associée au trajet direct.
- α_n : L'atténuation associée au trajet n.
- θ_n : Le déphasage du trajet n.
- τ_n : Le délai associé au trajet n.
- N : Le nombre de trajets réfléchis.

Les paramètres importants dans la modélisation des multi-trajets sont :

- Le délai : c'est le retard du signal réfléchi par rapport au signal direct. Pour le cas des signaux GPS, ce délai va engendrer une pseudo-distance plus grande pour le signal réfléchi. Ce paramètre va donc être exprimé en mètre dans notre cas.
- L'atténuation du signal : les signaux réfléchis subissent généralement une atténuation qui, dans le pire des cas, élimine complètement le signal réfléchi (*Signal Obscuration*). Ce paramètre va être exprimé en dB.
- L'effet Doppler : il s'agit de l'effet du mouvement relatif du récepteur par rapport au satellite ou encore de celui de l'environnement qui entoure le récepteur, ce qui engendre un

décalage en fréquence du signal reçu par le récepteur GPS qui s'exprime selon l'équation suivante :

$$\Delta f_D = \frac{v}{\lambda} \quad (4.2)$$

où :

Δf_D représente le décalage fréquentiel.

v représente la vitesse relative du récepteur par rapport au satellite.

λ représente la longueur d'onde du signal transmis.

Plusieurs modèles sont présentés dans la littérature pour modéliser les évanouissements des signaux dans un contexte de multi-trajets pour les communications sans fil (Rappaport, 2002). Les deux modèles les plus connus sont les modèles de Rayleigh et de Rice.

4.1.1 Le modèle de Rayleigh

Ce modèle caractérise l'évanouissement que subissent les signaux qui ne sont reçus directement par le récepteur (NLOS), c'est-à-dire ceux qui ont subi des réflexions au cours de leurs trajets. La distribution de Rayleigh est très utilisée dans les systèmes de communications sans fils mobiles pour modéliser la variation temporelle de l'enveloppe d'un signal ayant subi un évanouissement plat (Flat Fading).

La densité de probabilité de la fonction de Rayleigh, comme le montre la Figure 4.1, est exprimée selon l'équation 4.3 :

$$p(x) = \begin{cases} \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) & \text{pour } (0 \leq x \leq \infty) \\ 0 & \text{pour } (x < 0) \end{cases} \quad (4.3)$$

où :

- σ est la moyenne quadratique (RMS) du signal reçu avant la détection l'enveloppe.

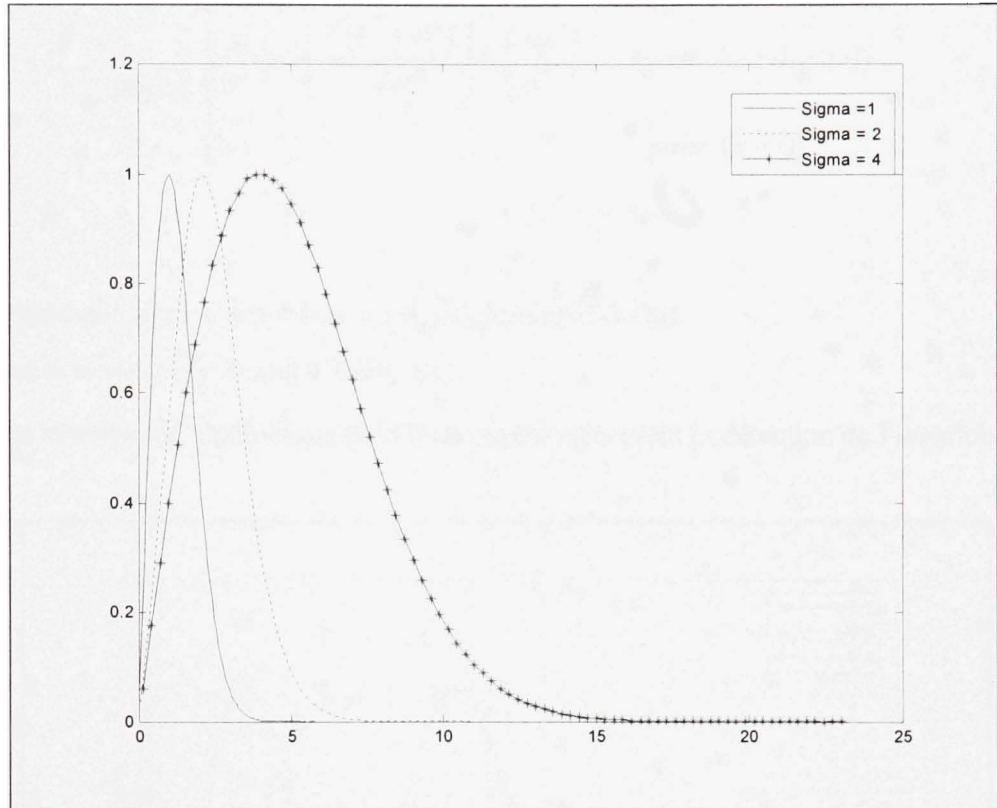


Figure 4.1 Distribution de Rayleigh.

4.1.2 Le modèle de Rice

Le modèle de Rice est utilisé lorsque le signal en ligne de vue (LOS) est plus fort que les autres signaux réfléchis. L'enveloppe du signal reçu est donc caractérisée par une distribution de Rice. Si le signal dominant devient de plus en plus faible, le signal reçu aura donc une enveloppe dont la distribution devient celle de Rayleigh. On peut donc dire que la distribution de Rayleigh est un cas particulier de la distribution de Rice ayant un signal dominant (LOS) qui est fortement évanoui.

La densité de probabilité de la fonction de Rice, comme le montre la Figure 4.2 (Rappaport, 2002) est exprimée selon l'équation 4.4 :

$$p(x) = \begin{cases} \frac{x}{\sigma^2} \exp\left(-\frac{(x^2 + \nu^2)}{2\sigma^2}\right) I_0\left(\frac{\nu x}{\sigma^2}\right) & \text{pour } (\nu \geq 0, x \geq 0) \\ 0 & \text{pour } (x < 0) \end{cases} \quad (4.4)$$

où :

- ν représente le pic d'amplitude du signal dominant (LOS)
- I_0 est la fonction de Bessel d'ordre zéro.
- σ est la moyenne quadratique (RMS) du signal reçu avant la détection de l'enveloppe.

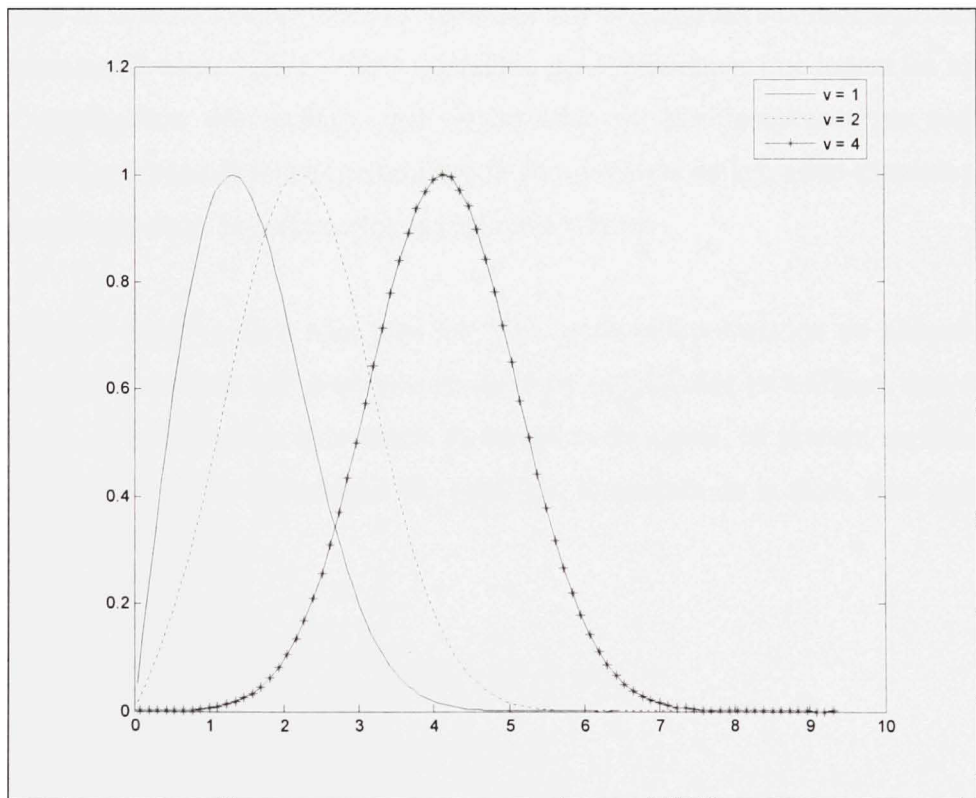


Figure 4.2 Distribution de Rice pour $\sigma = 1$.

La distribution de Rice est souvent décrite en fonction du paramètre K qui est défini comme le ratio entre la puissance du signal dominant et la puissance dissipée :

$$K = \frac{\nu^2}{2\sigma^2} \quad (4.5)$$

4.2 Le positionnement GPS dans un contexte de multi-trajets

Après cette brève description des modèles utilisés dans les communications sans fil, nous parlerons d'une façon plus ciblée de la modélisation des multi-trajets dans le domaine du positionnement par satellite.

4.2.1 Sensibilité des récepteurs GPS aux multi-trajets

Les systèmes de radionavigation sont plus sensibles aux erreurs dues aux réflexions des signaux que les autres applications sans fil. Ceci s'explique par l'importance que jouent les mesures de temps de propagation des signaux dans le domaine du positionnement par satellite. La fonctionnalité la plus sensible du récepteur réside dans le calcul de la pseudo-distance qui est un facteur déterminant dans la précision de la position retrouvée.

Dans les faits, si nous voulons aller plus loin dans notre compréhension du phénomène côté récepteur, nous trouverons que cette pseudo-distance est calculée en utilisant une estimation entre le temps de transmission et le temps de réception du signal, en prenant en considération d'autres aspects tels que la dynamique des satellites, la rotation de la terre, ainsi que d'autres paramètres (Tsui, 2005) :

$$\rho = c(t_r - t_e) + c\Delta t \quad (4.6)$$

où :

- ρ est la pseudo-distance.
- t_r est le temps de réception du signal.
- t_e est le temps d'émission du signal.

- Δt représente les autres délais introduits lors de la transmission du signal.

La précision de l'estimation de la différence entre le temps d'émission et le temps de réception représente donc un défi considérable, car à partir de la pseudo-distance sont calculées la position et la vitesse au niveau du récepteur.

En effet, un processus de poursuite au niveau de la boucle de verrouillage de délai du récepteur permet de maintenir une synchronisation entre le signal reçu et une réplique de ce signal (générée localement au niveau du récepteur) pour pouvoir estimer le temps de propagation (Kelly, Braasch et DiBenedetto, 2003). Toute imprécision dans l'estimation de la phase du code d'étalement cause une erreur dans la mesure de la pseudo-distance (Ilie, Landry et Constantinescu, 2005).

L'impact des multi-trajets dans la mesure de la pseudo-distance peut donc être déterminé en analysant la distorsion de la fonction d'auto-corrélation au niveau de la boucle de verrouillage de délai. En effet, une combinaison d'un trajet direct et d'un trajet réfléchi résulte donc à une déformation de la fonction de corrélation résultante comme la montre la Figure 4.3 (Weiss, 2007) :

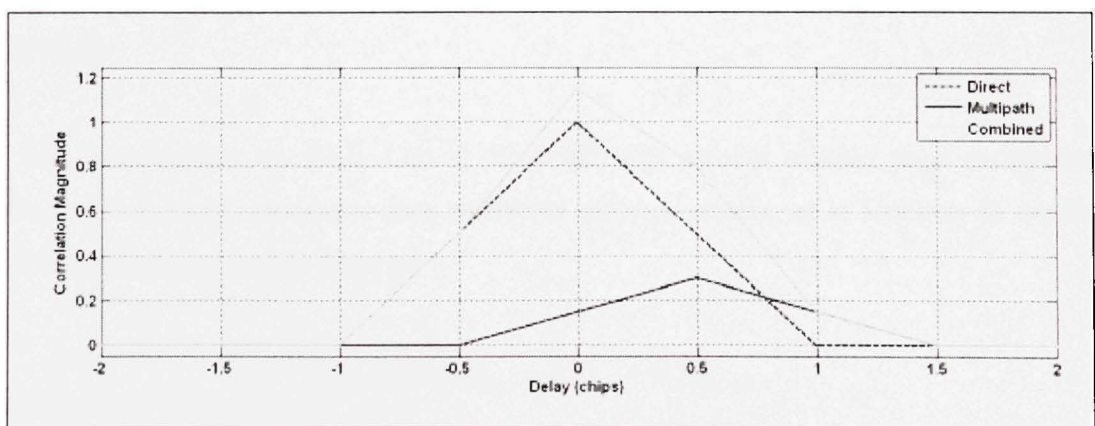


Figure 4.3 Fonction de corrélation pour un multi-trajet.
Tirée de Weiss (2007, p. 14)

4.2.2 Modélisation des évanouissements dans un milieu urbain

Après avoir traité les effets des délais dans les signaux multi-trajets, nous parlerons des évanouissements qui en résultent. En effet, les études que nous présentons ont été effectuées pour établir les modèles des évanouissements des signaux GPS dans un environnement de multi-trajets en milieu urbain. Le modèle suivant suggère la présence de trois états (Karasawa, Minamisono et Matsudo, 1995). Il assume donc que ces trois états, dans un canal de communication par satellite, sont constitués par :

- L'état 1 dans lequel nous avons le signal en ligne de vue directe, qui est dominant, avec la présence des multi-trajets. L'évanouissement dans cet état est modélisé par une distribution de Rice.
- L'état 2 dans lequel nous avons le signal en ligne de vue directe, qui est atténué, avec la présence des multi-trajets. Ce genre d'évanouissement a été modélisé par Loo selon une distribution f_{Loo} qu'il a développé avec des modèles statistiques (Loo, 1985).
- L'état 3 où il y a juste la présence des multi-trajets et le signal en ligne de vue de directe est complètement bloqué. L'évanouissement dans ce cas est modélisé par une distribution de Rayleigh.

Ces trois distributions de Rice, Loo et Rayleigh sont ensuite utilisée pour caractériser la distribution des évanouissements dans différents milieux urbains, et la fonction de densité de probabilité résultante sera de la forme :

$$f_{totale}(v) = C \cdot f_{Rice} + S \cdot f_{Loo} + B \cdot f_{Rayleigh} \quad (4.7)$$

où :

- C, S et D sont des valeurs quelconques.
- $C + S + B = 1$.

D'autres études que celle de Karasawa ont été effectuées pour améliorer la caractérisation des évanouissements dans les milieux urbains. Les modèles de Akturan et Vogel, par exemple, ont éliminé la distribution de Rayleigh pour la remplacer par une distribution de Loo modifiée dans leur caractérisation des différents états présentés précédemment (Akturan et Vogel, 1997).

L'implémentation de ces modèles au niveau du simulateur serait très utile dans le cas de la modélisation de signaux satellites dans un milieu urbain, ce qui n'est pas le cas actuellement.

4.2.3 Modèles de multi-trajets utilisés dans Spirent

Après avoir parlé de l'effet des multi-trajets sur le positionnement au niveau d'un récepteur, nous entamons une description détaillée des modèles de multi-trajets utilisées au niveau d'un simulateur commercial. Dans le simulateur SPIRENT (SPIRENT, 2006), qui est disponible au LACIME, deux configurations sont possibles comme indiqué dans les Tableau 4.1 et Tableau 4.2.

La première configuration consiste à ajouter manuellement des trajets pour un ou plusieurs signaux satellites simulés, et ceci en fonction des canaux disponibles. Dans le cas du simulateur SPIRENT GSS 7700, le nombre de canaux total est 16. Par exemple, si dans une simulation, nous avons 6 satellites visibles, l'utilisateur peut utiliser les 10 canaux restants pour modéliser les multi-trajets. Les paramètres configurés par l'utilisateur dépendent du modèle utilisé comme l'indique le Tableau 4.1.

Tableau 4.1 Configuration manuelle

Configuration manuelle des modèles de multi- trajets	Réflexion sur le sol	Nombre de trajets
		Atténuation (dB)
	Délai Fixe	Nombre de trajets
		Atténuation (dB)
		Décalage (m)
	Délai avec Doppler Fixe	Nombre de trajets
		Atténuation (dB)
		Décalage initial (m)
		Décalage Doppler (m/s)
	Plan vertical	Nombre de trajets
		Position et hauteur du plan vertical par rapport au véhicule

La seconde configuration consiste à définir la catégorie dans laquelle se trouve chacun des satellites simulés. Une modélisation des multi-trajets est effectuée en fonction de la catégorie à laquelle appartient le satellite concerné comme l'indique le Tableau 4.2. Le nombre maximal de canaux simulés reste 16 comme nous l'avons indiqué dans la configuration précédente.

Il faut cependant mentionner que dans le modèle le plus récent GSS8000 de la compagnie SPIRENT le nombre de canaux disponible est de 48, avec un nombre de canaux configurables pour les multi-trajets qui peut atteindre les 192 canaux (Spirent, 2006). Ceci veut dire que l'implémentation des multi-trajets a été optimisée pour qu'autant de canaux soient modélisés.

Tableau 4.2 Utilisation des modèles statistiques

Configuration des multi-trajets avec des modèles statistiques	Catégorie A	Les satellites qui sont de cette catégorie ne sont pas simulés.
	Catégorie B	Les satellites qui sont de cette catégorie sont en visibilité directe sans trajets réfléchis (Distribution de Rice).
	Catégorie C	Les satellites qui sont dans cette catégorie sont en visibilité directe avec des trajets réfléchis (Distribution de Rice + Distribution de Rayleigh).
	Catégorie D	Les satellites qui sont dans cette catégorie ne présentent que des trajets réfléchis (Distribution de Rayleigh).

4.3 Modèle de génération de multi-trajets utilisé

Nous proposons dans cette partie une modélisation des multi-trajets en utilisant la nouvelle architecture du simulateur, car l'ancienne version utilisant une DLL Matlab ne permet pas d'inclure une modélisation des multi-trajets dans les signaux satellites générés.

4.3.1 Architecture générale du modèle

La méthode la plus directe pour effectuer cette opération consiste à consacrer un nouveau canal physique pour chacun des trajets simulés en mettant en place des générateurs de délais dans la partie logicielle combinés avec des générateurs d'évanouissements dans la partie IF (Jeng-Kuang et al., 2006). À la Figure 4.4, nous proposons une méthode d'implémentation des multi-trajets au niveau de notre simulateur.

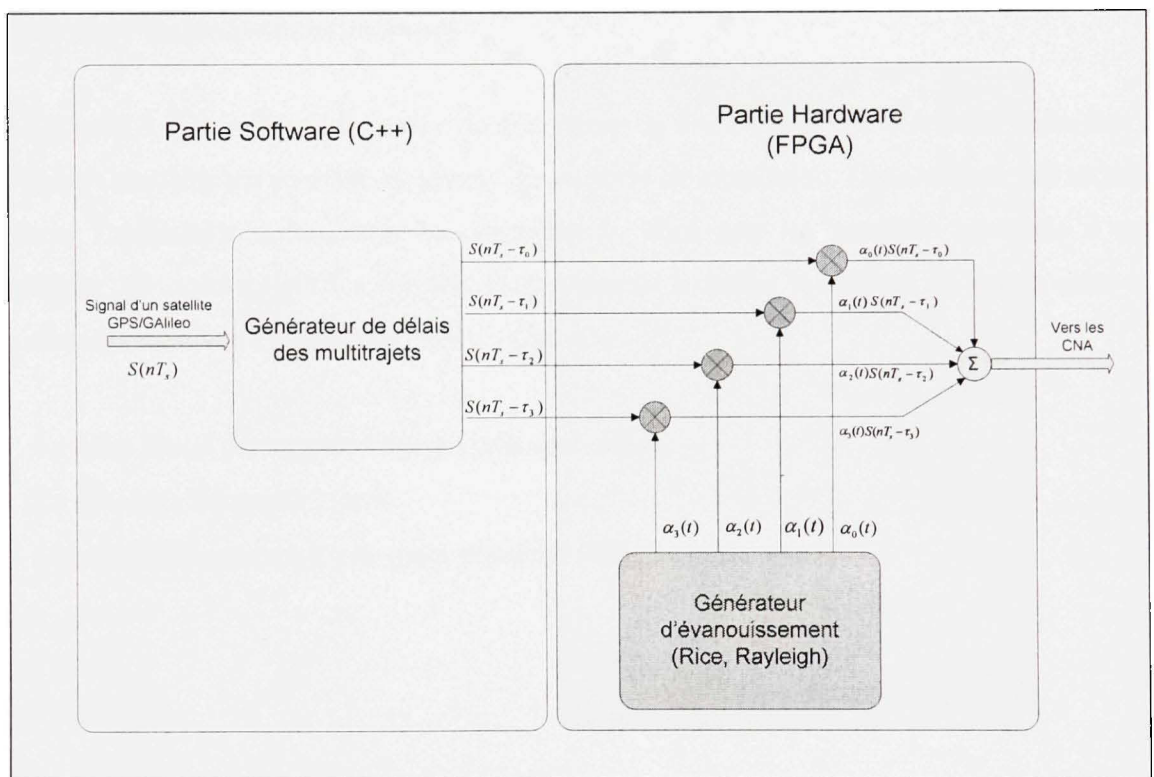


Figure 4.4 Architecture générale du modèle multi-trajets.

Cette architecture nécessite une allocation de ressources importante au niveau du FPGA à chaque fois qu'un nouveau trajet est modélisé (Jeng-Kuang et al., 2007). En effet, seulement 8 canaux peuvent être utilisés lors d'une simulation avec l'architecture IF actuelle. Ceci nous restreint dans le nombre de trajets simulés. Sachant que dans une simulation, nous devons avoir au moins 4 satellites visibles pour que le récepteur soit capable de converger vers une position PVT, nous pouvons donc modéliser au maximum 4 multi-trajets par simulation.

4.3.2 Implémentation

L'implémentation d'un générateur de multi-trajets peut être réalisée de deux manières différentes. La première solution consiste à la mise en œuvre d'un générateur de multi-trajet manuel qui permet à l'utilisateur de modéliser manuellement chacun des trajets qu'il veut ajouter. La deuxième solution consiste à développer en VHDL un générateur d'évanouissement qui permettra de modéliser les distributions de Rayleigh et Rice au niveau du FPGA.

Générateur de multi-trajets manuel

La Figure 4.5 montre la modélisation du générateur de multi-trajets. En effet, une définition de différents modèles est possible au niveau du scénario de simulation. Dans chacun des modèles définis, l'utilisateur doit choisir la constellation, ainsi que les satellites auxquels il veut appliquer des multi-trajets. La dernière étape consiste à choisir le nombre de trajets, ainsi que les données relatives à chacun des trajets telles que :

- Le délai initial par rapport au trajet principal (m).
- Le décalage fréquentiel (m/s).
- L'atténuation par rapport au trajet principal (dB).

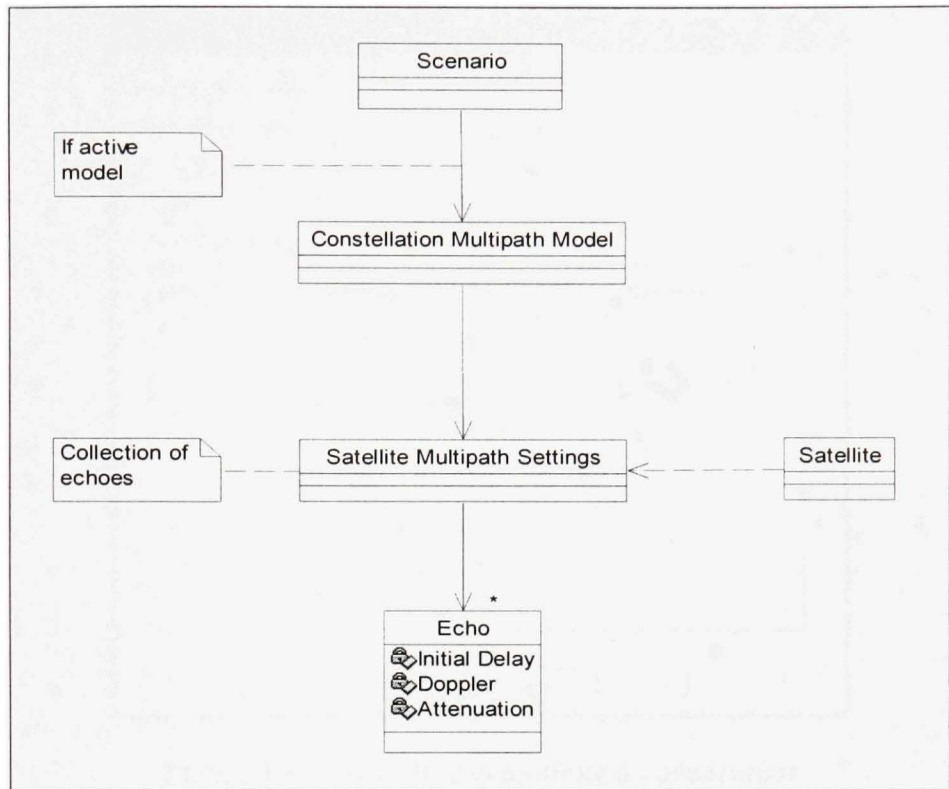


Figure 4.5 Modélisation du générateur de multi-trajets.

La Figure 4.6 nous montre donc l'interface graphique qui permet à l'utilisateur de définir les différents modèles de multi-trajets dans le scénario de simulation. L'utilisateur ne peut cependant activer qu'un seul modèle à la fois au cours d'un scénario comme le montre la figure A7.1 dans l'ANNEXE VII. D'autres options qui permettent de copier et coller les données d'un modèle de multi-trajet à un autre sont aussi disponibles.

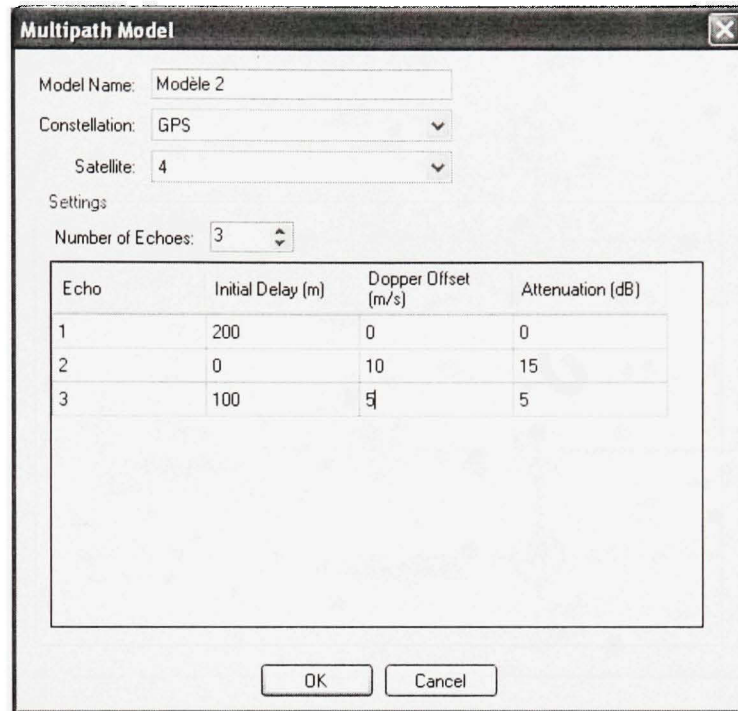


Figure 4.6 Interface graphique du générateur de modèles de multi-trajets.

Générateur d'évanouissement Rayleigh et Rice

Comme nous l'avons évoqué précédemment, le générateur d'évanouissement doit être implémenté au niveau du FPGA pour chacun des canaux, ce qui représente une allocation importante des ressources. L'implémentation d'un générateur d'évanouissements hybride Rayleigh et Rice nécessite, comme le montre la Figure 4.7, deux générateurs de bruits gaussiens, ainsi que deux filtres Doppler. Le module de ces deux branches en quadrature et en phase, nous donne une distribution de Rayleigh ou de Rice.

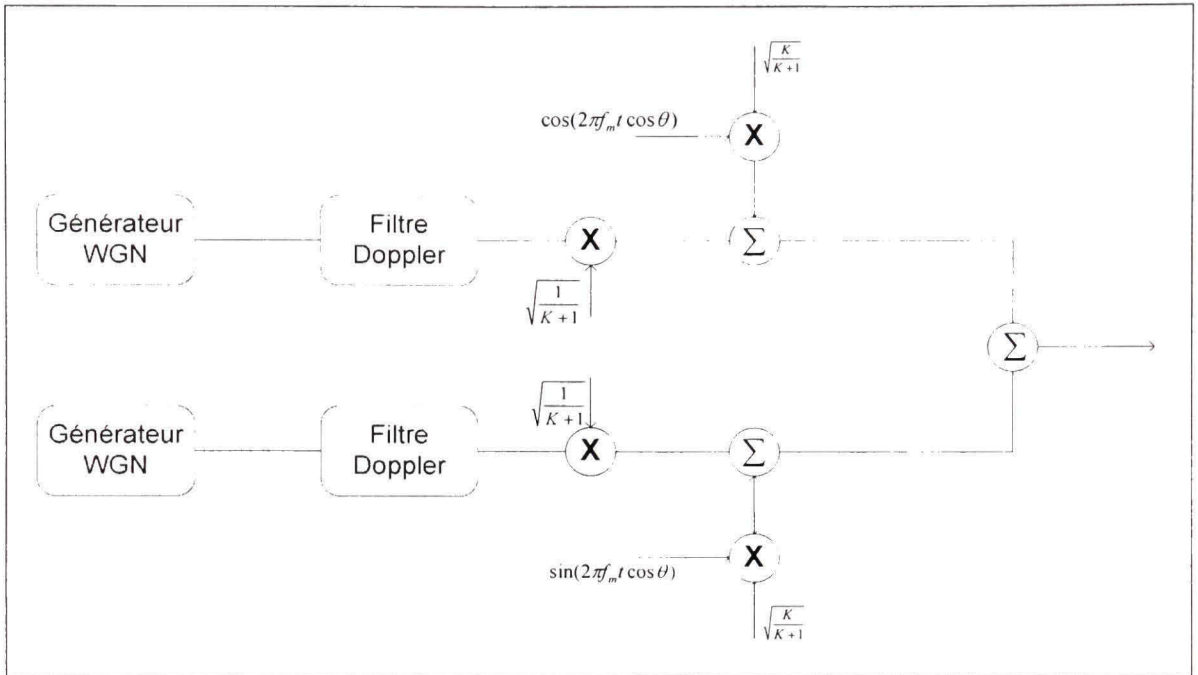


Figure 4.7 Générateur d'évanouissement Rayleigh/Rice.

Le codage de ces modules a été effectué mais nous n'avons pas été en mesure d'effectuer des tests avec ce générateur d'évanouissement sur les signaux générés par le simulateur. En effet, pour pouvoir effectuer ce genre de tests, il faut pouvoir contrôler chacune des sorties des canaux de modulation des signaux GPS, alors notre architecture IF actuelle ne le permet pas.

D'autres travaux pourront donc traiter cette partie du projet ce qui permettrait d'améliorer l'architecture générale du simulateur. Nous pouvons aussi recommander une solution plus intéressante au niveau des ressources matérielle au niveau du FPGA. En effet, nous pouvons nous restreindre à utiliser un seul générateur de bruit gaussien pour tous les canaux, comme le montre la Figure 4.8, en mettant en place un bus Série/Parallèle qui permet de garder les propriétés de la distribution gaussienne.

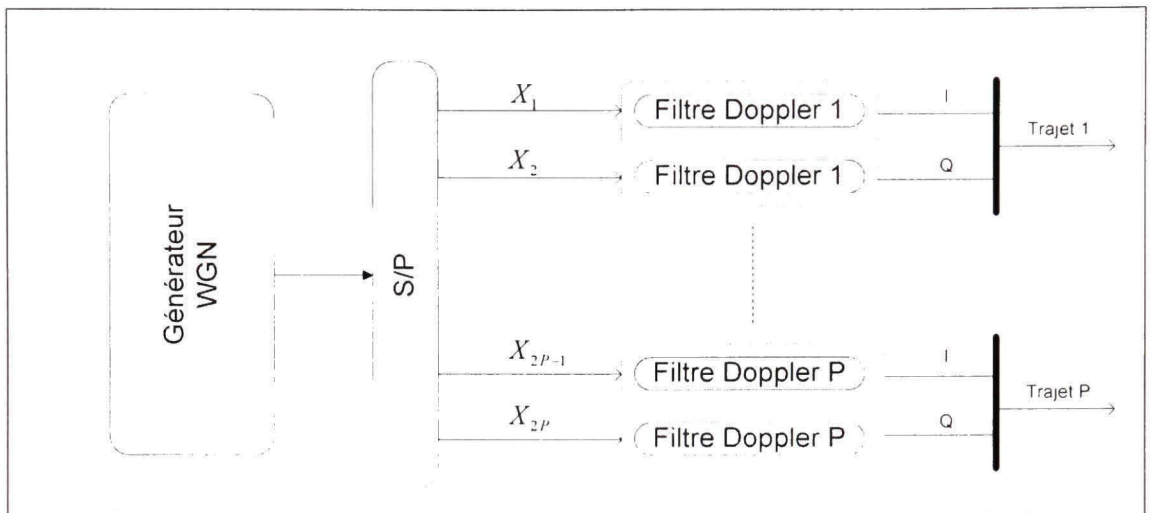


Figure 4.8 Architecture optimisée pour le générateur d'évanouissement.

4.3.3 Tests et résultats

Grâce à la nouvelle architecture logicielle du simulateur, nous avons pu effectuer des simulations incluant la modélisation des multi-trajets. Nous avons utilisé le 'Scénario_test' qui est décrit comme suit :

Constellation : *GPS (Fichier Almanach 040.alm)*

Date : *12/02/2007 à 15 :00 :30*

Position du récepteur : *Latitude N 45deg 29' 41''*

Longitude W 73deg 41' 35''

Masque d'élévation = 20 degrés.

Pour cette partie, nous avons restreint la visibilité à 5 satellites dans le but de libérer 3 canaux dans le FPGA. En effet, dans le scénario que nous avons mis en place, nous utilisons deux canaux de tests pour les multi-trajets :

- **Test numéro 1** : Satellites dupliqués : 4 et 17.
Délai = 150 mètres pour chacun des multi-trajets

Nous pouvons voir sur la Figure 4.9 l'effet des délais des multi-trajets sur la position au niveau du récepteur CMC. Nous remarquons une grande déviation de la position de plus de 10 mètres de drift après 6 minutes de simulation, alors que le drift était de l'ordre de quelques centimètres pour la même période de simulation sans multi-trajets.

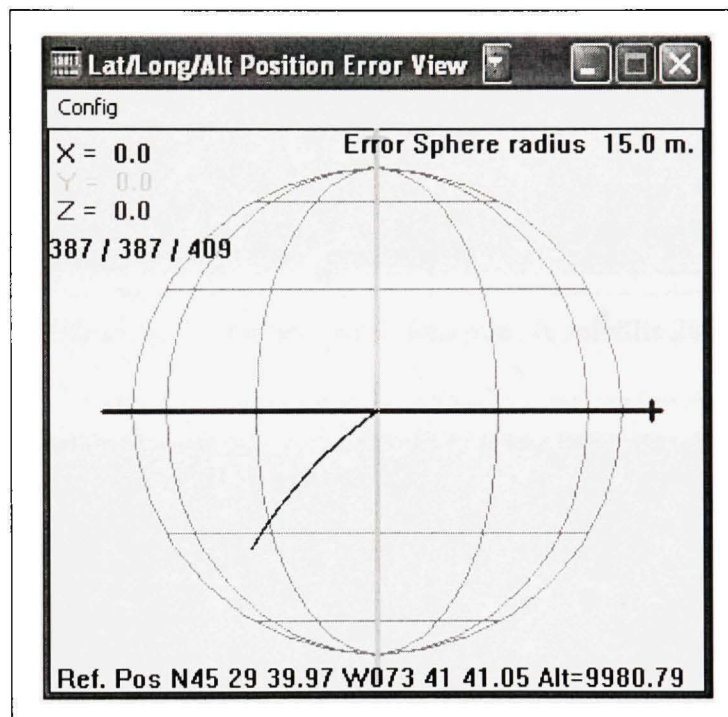


Figure 4.9 Variation de la position après 6 minutes de simulation (Délai=150m).

Les figures Figure 4.10 et Figure 4.11 montrent la différence entre le pic de corrélation au niveau du récepteur ÉTS pour un canal qui ne subit pas de multi-trajets (satellite 20) et un autre qui en subit (satellite 17). Ce dernier présente une déformation qui résulte de la sommation des deux pics de corrélations pour chacun des trajets comme expliqué précédemment dans le paragraphe 4.2.1.

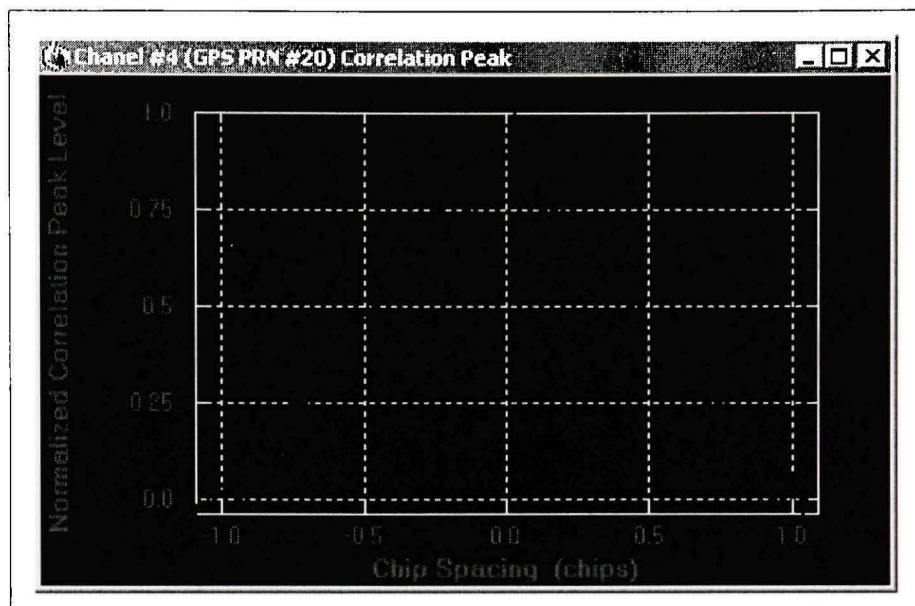


Figure 4.10 Pic de corrélation pour le satellite 20.

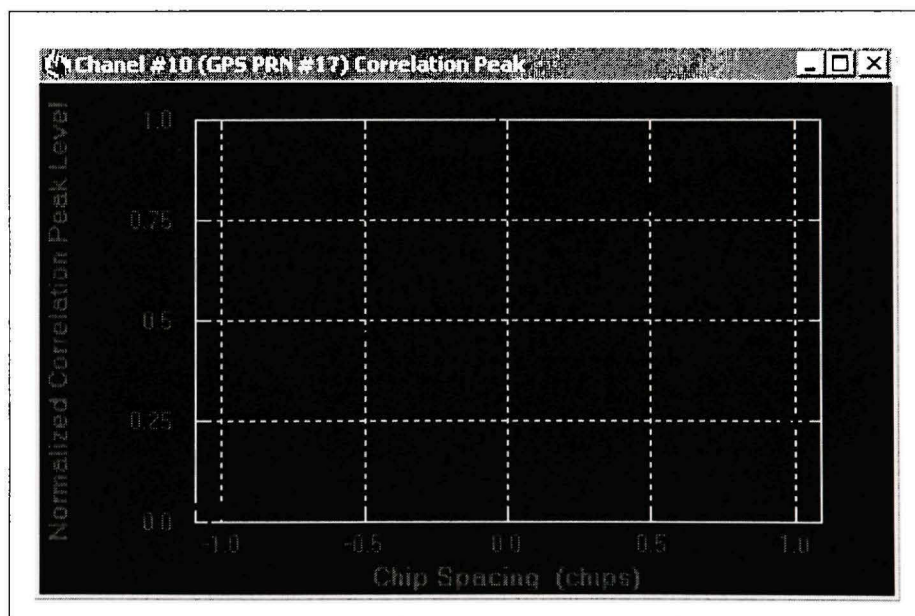


Figure 4.11 Pic de corrélation pour le satellite 17.

- **Test numéro 2 :** Satellites dupliqués 4 et 17.

Délai = 300 mètres.

Théoriquement, un délai de 300 mètres pour un signal GPS utilisant la modulation C/A, n'engendre théoriquement aucun effet sur la position du récepteur (Ilie, Landry et Constantinescu, 2005). En effet, nous pouvons voir sur la Figure 4.12 la très faible variation de la position retrouvée par le récepteur de CMC au cours du temps (6 minutes de simulation), dans un cercle de rayon 0.5 mètre. Ceci montre qu'effectivement, un délai de 300 mètres entre les multi-trajets et les signaux directs, n'a aucun effet visible sur le positionnement.

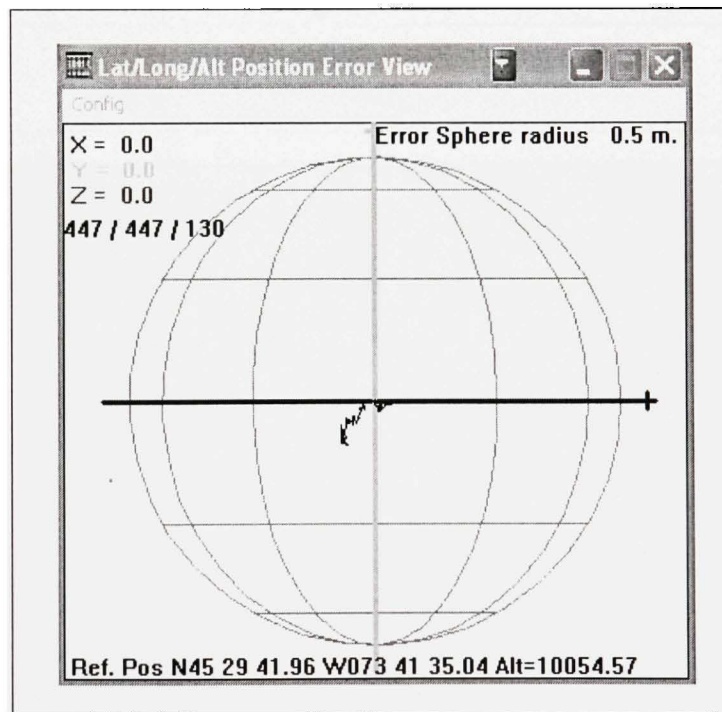


Figure 4.12 Variation de la position après 6 minutes de simulation (Délai = 300m).

Cependant, nous remarquons sur la Figure 4.13 ainsi que la Figure 4.14, la déformation des pics de corrélation pour les canaux affectés par les multi-trajets au niveau du récepteur ÉTS.

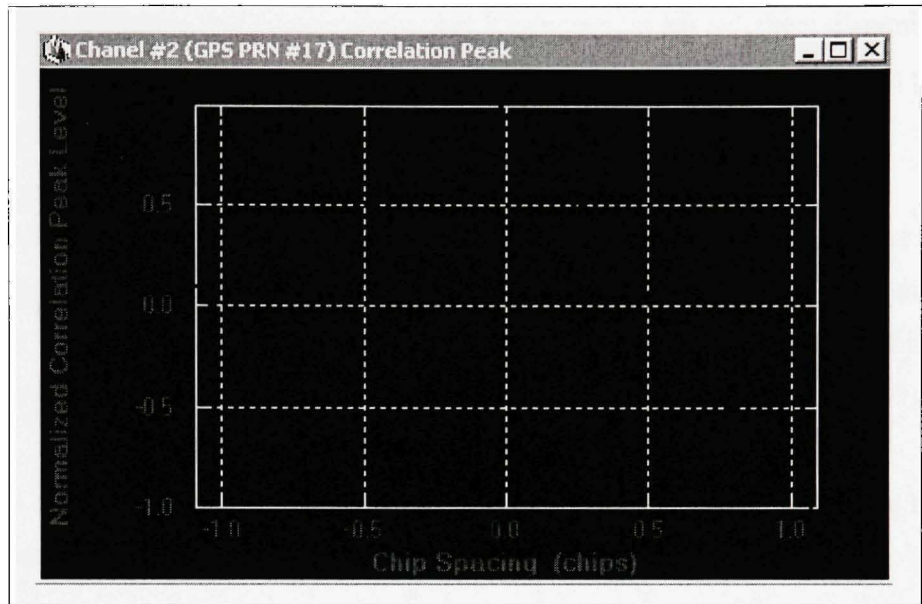


Figure 4.13 Pic de corrélation du satellite 17.

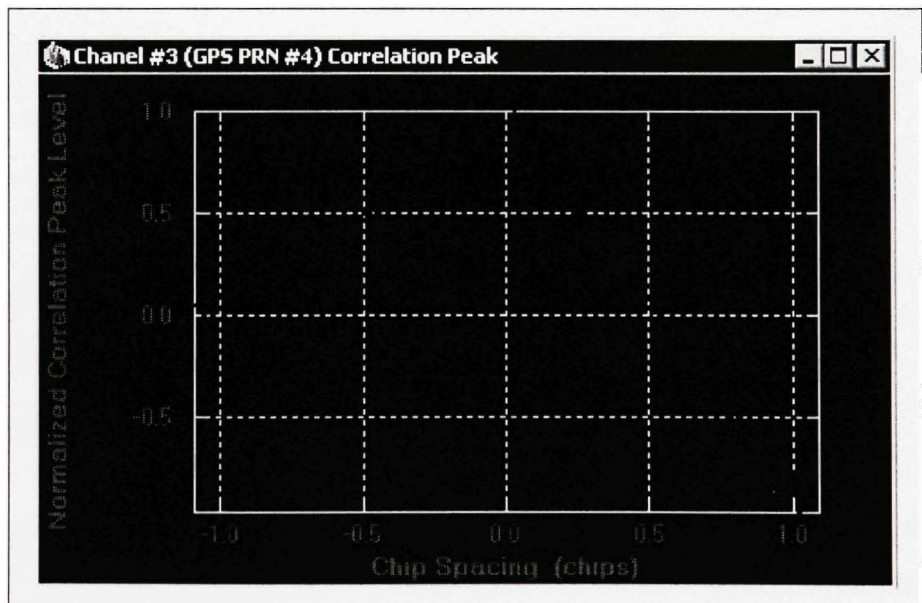


Figure 4.14 Pic de corrélation du satellite 4.

4.3.4 Modélisation optimisée des multi-trajets

L'utilisation d'un canal physique pour chaque trajet modélisé n'est sûrement pas une solution optimale, surtout quand les ressources au niveau du FPGA sont limitées. En effet, comme nous

l'avons vu dans la section précédente, seulement 8 canaux sont actuellement disponibles dans la partie IF du simulateur. Il nous faut donc trouver une autre façon de modéliser les multi-trajets sans utiliser des canaux additionnels.

La Figure 4.15 propose une architecture dans laquelle plusieurs trajets peuvent être simulés sur un seul canal physique. Ceci est possible si nous mettons en œuvre une fonction qui permet de modéliser le comportement de ces trajets pour l'appliquer sur le signal satellite initial. Le défi est d'obtenir, par la suite, les mêmes effets au niveau du récepteur (Sahmoudi, 2010).

La Figure 4.15 montre que le signal généré pour un satellite dans un contexte de multi-trajets s'exprime sous la forme :

$$r(t) = \sum_{k=1}^N \alpha_k(t) S(nT_s - \tau_k) \quad (4.8)$$

où :

- $r(t)$: Le signal résultant pour un satellite donné.
- N : Le nombre de trajets pour le satellite simulé.
- $\alpha_k(t)$: L'atténuation pour chacun des trajets.
- τ_k : Le retard pour chacun des trajets.
- S : Le signal initial généré par le simulateur pour le satellite simulé.
- T_s : La période d'échantillonnage.

L'idée est donc de remplacer cet ensemble de signaux qui représentent chacun un trajet, par un seul signal qui, d'un point de vue mathématique, représente l'ensemble des trajets et qui s'exprime tel que :

$$r(t) = A_m \times S(t - \tau_m) \quad (4.9)$$

où :

- A_m est une fonction qui permet de représenter les atténuations pour tous les trajets.
- τ_m est une fonction qui permet de représenter les retards pour tous les trajets.

Ces fonctions ne seront pas des représentations exactes, mais plutôt des approximations qui permettront d'obtenir des résultats similaires en remplaçant plusieurs canaux par un seul canal. Il faut aussi noter que cette méthode ne pourra pas être efficace pour un grand nombre de trajets. Il faut donc effectuer une étude théorique approfondie qui permet de définir les conditions dans lesquelles cette approximation est valable.

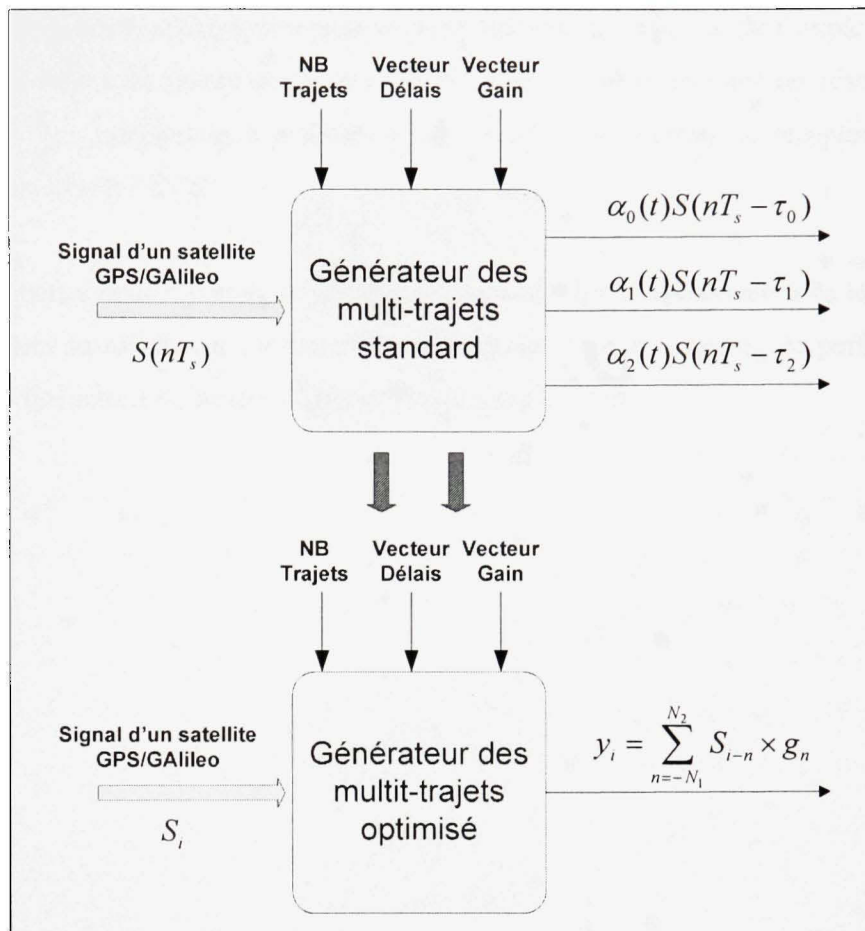


Figure 4.15 Optimisation de l'utilisation des canaux pour les multi-trajets.

4.3.5 Conclusion

Ce chapitre a été l'occasion de parler de la modélisation des multi-trajets dans les signaux générés par le simulateur GPS/Galileo.

Une première partie nous a permis de parler des modèles théoriques mentionnés dans la littérature, ainsi que de l'effet des multi-trajets au niveau des récepteurs GPS. Nous avons aussi fait le point sur la modélisation des multi-trajets dans les simulateurs commerciaux.

Dans la deuxième partie, nous avons parlé de la modélisation, ainsi que de l'implémentation des modèles multi-trajets au niveau du simulateur. Nous avons aussi présenté les résultats des tests effectués avec ces modèles et la validation des résultats en utilisant le récepteur CMC et le récepteur développé à l'ETS.

Une dernière partie nous a permis d'introduire une possibilité d'optimisation de la modélisation des multi-trajets au niveau du simulateur, ce qui laisse place à beaucoup de perspectives pour les personnes qui seront impliquées dans ce projet dans le futur.

CONCLUSION

La mise en œuvre d'un simulateur de signaux GNSS a été beaucoup plus complexe que prévu. Dans le chapitre 1, nous avons décrit les systèmes GPS et Galileo, ce qui nous a permis de comprendre les fondements théoriques du positionnement par satellite. Nous avons aussi parcouru, lors d'un premier aperçu, les différentes composantes logicielles et matérielles de notre simulateur de constellation.

Dans le chapitre 2, nous avons fait une description détaillée de la partie logicielle du simulateur. Cette partie représentait pour nous la partie la plus complexe à valider. D'abord, il fallait faire une description des étapes de configuration d'un scénario et des paramètres utilisés lors de la simulation. Ensuite, nous avons parcouru les différents algorithmes de calculs en temps réel utilisés dans la partie logicielle du simulateur que nous avons pu valider en utilisant des données préenregistrées par un récepteur GPS. La convergence vers une position stable, au niveau du récepteur, en utilisant notre simulateur a été une étape à la fois difficile et cruciale, car la génération de signaux satellites nécessite des connaissances relatant à différents domaines tel que les télécommunications, la géomatique et la programmation logicielle.

La seconde étape du projet consistait à mettre en place une nouvelle architecture logicielle novatrice qui permet d'atteindre un niveau de performances hautement supérieur à la première version. En effet, nous avons mis en œuvre un logiciel ainsi qu'une structure qui répond aux normes de développement de produits commerciaux compétitifs tels que l'extensibilité, la réutilisabilité et la simplicité de tests. Il est cependant à noter que seule la constellation GPS L1 a été validée dans le cadre de ce projet, mais le travail effectué va permettre d'inclure d'autres constellations sans aucune difficulté.

La dernière partie de ce projet consiste à la modélisation des multi-trajets dans les signaux générés par le simulateur. Une étude théorique montre d'abord l'importance de cette modélisation, ainsi que l'extrême sensibilité des récepteurs GPS actuels à ce phénomène. Ensuite, nous avons présenté une première méthode par laquelle nous avons pu modéliser des

multi-trajets au sein du simulateur et effectuer ainsi quelques tests. Cette modélisation initiale qui n'est cependant pas efficace en termes de ressources matérielles utilisées, nous a poussés à proposer une nouvelle méthode dont l'implémentation n'a pas été effectuée, mais dont les résultats vont probablement être très intéressants au niveau des performances.

Pour conclure, rappelons que ce projet fait partie d'une panoplie de projets de communications sans fil au sein du LACIME, et qui nous a permis d'explorer de nouvelles perspectives pour le futur, tel que l'intégration de nouvelles constellations, des systèmes d'augmentations tels que SBAS ou encore l'utilisation des systèmes d'information géographique pour prévoir la nature des canaux, et effectuer des modélisations intelligentes en temps réel.

ANNEXE I

Calcul des positions des satellites

Voici les étapes qui permettent de calculer les positions des satellites à un instant donné t selon (ARINC, 2001):

1) Calcul du mouvement moyen corrigé : $n = \sqrt{\frac{\mu}{A^3}} + \Delta n$

2) Calcul de l'anomalie moyenne : $M_k = M_0 + n \times t_k$

3) Calcul de l'anomalie excentrique : $M_k = E_k - e \times \sin(E_k)$

On résout cette équation par la méthode de Newton (Sauriol, 2007)

4) Calcul de l'anomalie vraie : $V_k = a \tan 2(\sqrt{1-e^2} \times \sin(E_k), \cos(E_k) - e)$

$a \tan 2$ représente la fonction d'arc-tangente à quatre quadrants.

5) L'argument de latitude : $\phi_k = V_k + \omega$

6) Calcul des termes de correction d'orbite :

$$\delta u_k = C_{us} \sin(2\phi_k) + C_{uc} \cos(2\phi_k)$$

$$\delta r_k = C_{rs} \sin(2\phi_k) + C_{rc} \cos(2\phi_k)$$

$$\delta i_k = C_{is} \sin(2\phi_k) + C_{ic} \cos(2\phi_k)$$

7) Calcul de l'argument de latitude corrigé : $u_k = \phi_k + \delta u_k$

8) Calcul du rayon corrigé : $r_k = A(1 - e \times \cos E_k) + \delta r_k$

9) Calcul de l'inclinaison corrigée : $i_k = i_0 + \delta i_k$

10) Calcul de la longitude du nœud ascendant corrigé : $\omega_k = \omega_0 + (\omega_{dot} - \omega_{terre}) \times t_k - \omega_{terre} \times t_0 a$

où ω_{terre} est la vitesse angulaire de rotation de la terre (rad/s)

Nous pouvons à partir des paramètres ($\omega_k, r_k, \varphi_k, i_k$) définir la position d'un satellite sur son orbite. Nous pouvons aussi décrire cette position dans un référentiel cartésien ECEF en utilisant les équations suivantes :

$$1) x_p = r_k \times \cos u_k$$

$$2) y_p = r_k \sin u_k$$

$$3) X_{sat} = x_p \cos \omega_k - y_p \cos i_k \sin \omega_k$$

$$4) Y_{sat} = x_p \sin \omega_k - y_p \cos i_k \cos \omega_k$$

$$5) Z_{sat} = y_p \sin i_k$$

ANNEXE II

Paramètres générés par la DLL MATLAB

- *ConstParamNavMessage* : contient les variables dont les valeurs ne changent pas au cours de la simulation.

ConstParam_NavMess_GPS_L1	
• <u>SVHealth SVs Dec GPS L1</u>	• <u>C rs Dec GPS L1</u>
• <u>e Dec GPS L1</u>	• <u>C rc Dec GPS L1</u>
• <u>t oa Dec GPS L1</u>	• <u>C us Dec GPS L1</u>
• <u>i 0 Dec GPS L1</u>	• <u>C uc Dec GPS L1</u>
• <u>OmegaDot Dec GPS L1</u>	• <u>C is Dec GPS L1</u>
• <u>sqrt A Dec GPS L1</u>	• <u>C ic Dec GPS L1</u>
• <u>Omega 0 Dec GPS L1</u>	• AODO_Dec_GPS_L1
• <u>omega Dec GPS L1</u>	• DataID_Dec_GPS_L1
• <u>M 0 Dec GPS L1</u>	• alpha_0_Dec_GPS_L1
• <u>a f0 Dec GPS L1</u>	• alpha_1_Dec_GPS_L1
• <u>a f1 Dec GPS L1</u>	• alpha_2_Dec_GPS_L1
• <u>Delta n Dec GPS L1</u>	• alpha_3_Dec_GPS_L1
• <u>a f2 Dec GPS L1</u>	• beta_0_Dec_GPS_L1
• <u>IDot Dec GPS L1</u>	• beta_1_Dec_GPS_L1
• <u>delta_i_Dec_GPS_L1</u>	• beta_2_Dec_GPS_L1
• <u>URAIindex_Dec_GPS_L1</u>	• beta_3_Dec_GPS_L1
• <u>IODC Dec GPS L1</u>	• AntiSpoof_SVs_Dec_G PS_L1
• <u>IODE Dec GPS L1</u>	• AI_SVs_Dec_GPS_L1
• <u>T_GD_Dec_GPS_L1</u>	• ERD_SVs_Dec_GPS_L1
	• A_1_Dec_GPS_L1
	• A_0_Dec_GPS_L1

Figure A2.1 Le tableau des paramètres constants.

- *VarParamNavMessage* : contient les variables dont les valeurs changent au cours de la simulation.

● TOW_Dec_GPS_L1	● WN_t_Dec_GPS_L1
● SubFrameID_dec_GPS_L1	● Delta_t_LS_Dec_GPS_L1
● WeekNumber_Dec_GPS_L1	● WN_LSF_Dec_GPS_L1
● <u>t_oc Dec GPS L1</u>	● DN_Dec_GPS_L1
● <u>t_oe Dec GPS L1</u>	● Delta_t_LSF_Dec_GPS_L1
● WN_a_Dec_GPS_L1	● NavMess_Actual{1}.TOW_Actual
● t_ot_Dec_GPS_L1	

Figure A2.2 Tableau des paramètres variables.

- *RangeParam* : contient les variables calculées relatives aux distances entre le satellite et le récepteur.

● <u>PseudorangesErr GPS L1</u>
● <u>StdPseudorange GPS L1</u>
● <u>DeltaPseudoranges GPS L1</u>
● <u>StdDeltaPseudorange GPS L1</u>
● <u>Doppler GPS L1</u>
● <u>CNO GPS L1 Step</u>

Figure A2.3 Tableau des paramètres de distances.

- *VisibilityParam* : un tableau de valeur booléennes qui indique pour chaque satellite s'il est visible ou pas.
- *TimeParam* : une structure de données qui contient les paramètres relatifs au temps de la simulation ainsi qu'à sa durée.

ANNEXE III

Message de navigation Galileo

Le message de navigation Galileo est généré selon (ESA, 2006) la structure I/NAV qui est utilisée pour les canaux E1 et E5b. Nous avons aussi entamé le calcul des champs CRC, et l'encodage FEC du message de navigation, mais qui n'ont pas encore été intégrés dans le projet. Deux types de pages sont définis :

- Une page nominale qui a une durée de 2 secondes
- Une page d'alerte qui a une durée de 1 seconde

La structure de la page nominale se fait selon le tableau suivant :

E5b-I						E1B									
Even/odd=0	Type	Data i (1/2)				Tail	Total	Even/odd=1	Type	Data j (2/2)	Reserved	CRC _j	Reserved	Tail	Total
1	1	112				6	120	1	1	16	64	24	8	6	120
Even/odd=1	Type	Data i (2/2)	Reserved	CRC _i	Reserved	Tail	Total	Even/odd=0	Type	Data j+1 (1/2)				Tail	Total
1	1	16	64	24	8	6	120	1	1	112				6	120

Figure A3.1 Allocation des bits selon la structure nominale I/NAV.

Chaque sous-trame est générée selon une certaine séquence définie dans la figure suivante :

T, (GST, sync.) [s]	E5b sub frame ID	E5b page	E5b content		E1B content		E1B page	E1B sub frame ID
0	N	Even	Word 1 (1 2)		Spare Word (2 2)	Reserved	Odd	N-1
1	N	Odd	Word 1 (2 2)	Reserved	Word 2 (1 2)		Even	N
2	N	Even	Word 3 (1 2)		Word 2 (2 2)	Reserved	Odd	N
3	N	Odd	Word 3 (2 2)	Reserved	Word 4 (1 2)		Even	N
4	N	Even	Word 5 (1 2)		Word 4 (2 2)	Reserved	Odd	N
5	N	Odd	Word 5 (2 2)	Reserved	Word 6 (1 2)		Even	N
6	N	Even	Word 7 (1 2)		Word 6 (2 2)	Reserved	Odd	N
7	N	Odd	Word 7 (2 2)	Reserved	Word 8 (1 2)		Even	N
8	N	Even	Word 9 (1 2)		Word 8 (2 2)	Reserved	Odd	N
9	N	Odd	Word 9 (2 2)	Reserved	Word 10 (1 2)		Even	N
10	N	Even	Reserved		Word 10 (2 2)	Reserved	Odd	N
11	N	Odd	Reserved		Reserved		Even	N
12	N	Even	Reserved		Reserved		Odd	N
13	N	Odd	Reserved		Reserved		Even	N
14	N	Even	Reserved		Reserved		Odd	N
15	N	Odd	Reserved		Reserved		Even	N
16	N	Even	Reserved		Reserved		Odd	N
17	N	Odd	Reserved		Reserved		Even	N
18	N	Even	Reserved		Reserved		Odd	N
19	N	Odd	Reserved		Reserved		Even	N
20	N	Even	Word 2 (1 2)		Reserved		Odd	N
21	N	Odd	Word 2 (2 2)	Reserved	Word 1 (1 2)		Even	N
22	N	Even	Word 4 (1 2)		Word 1 (2 2)	Reserved	Odd	N
23	N	Odd	Word 4 (2 2)	Reserved	Word 3 (1 2)		Even	N
24	N	Even	Word 6 (1 2)		Word 3 (2 2)	Reserved	Odd	N
25	N	Odd	Word 5 (2 2)	Reserved	Word 5 (1 2)		Even	N
26	N	Even	Spare Word (1 2)		Word 5 (2 2)	Reserved	Odd	N
27	N	Odd	Word (2 2)	Reserved	Spare Word (1 2)		Even	N
28	N	Even	Spare Word (1 2)		Spare Word (2 2)	Reserved	Odd	N
29	N	Odd	Spare Word (2 2)	Reserved	Spare Word (1 2)		Even	N
30	N-1	Even	Word 1 (1 2)		Spare Word (2 2)	Reserved	Odd	N

Figure A3.2 Structure de la sous-trame nominale I/NAV.

ANNEXE IV

Schéma détaillé de la partie IF

La Figure A6.1 montre le schéma de modulation des signaux utilisé dans la partie IF du simulateur. Il s'agit d'un modulateur universel utilisé pour les signaux GPS L1, GPS L5, Galileo E1, Galileo E5a, Galileo E5b.

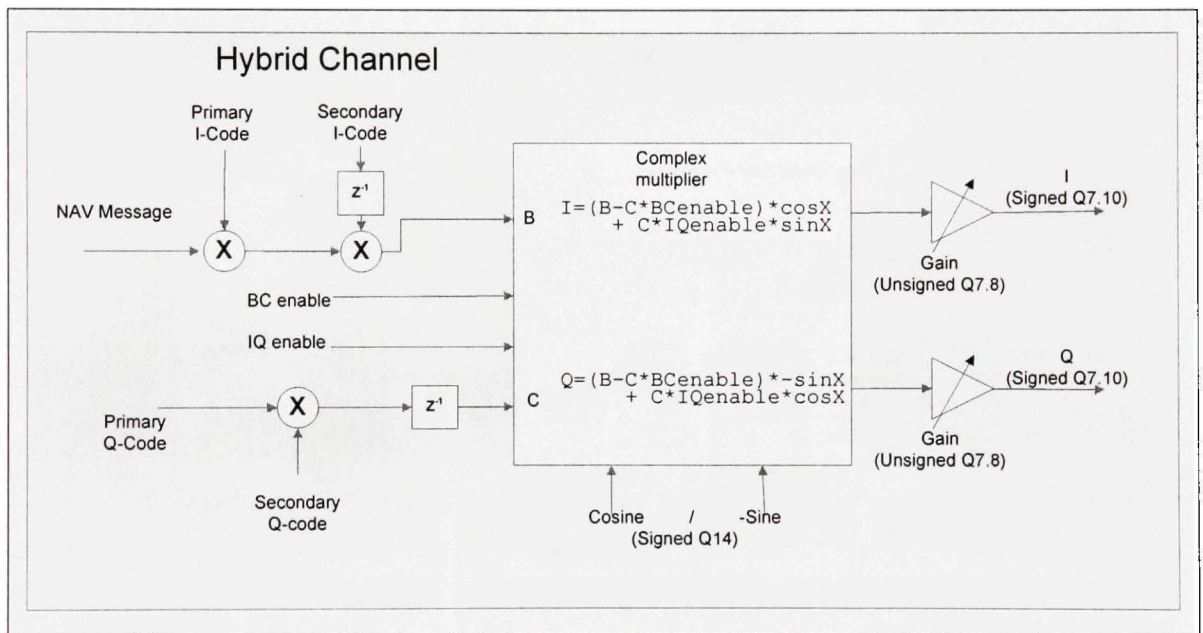


Figure A4.1 Schéma de modulation des signaux GNSS.

La figure A6.2 montre le fonctionnement plus détaillé du modulateur qui récupère les données provenant de la partie logicielle du simulateur en utilisant les bus XMEM et FSB :

Le bus XMEM permet de mettre à jour les valeurs des fréquences de la porteuse et du code d'étalement, ainsi que les gains de chaque canal.

Le bus FSB permet de mettre à jour les messages de navigation, ainsi que les phases de la porteuse et les registres de contrôle des canaux.

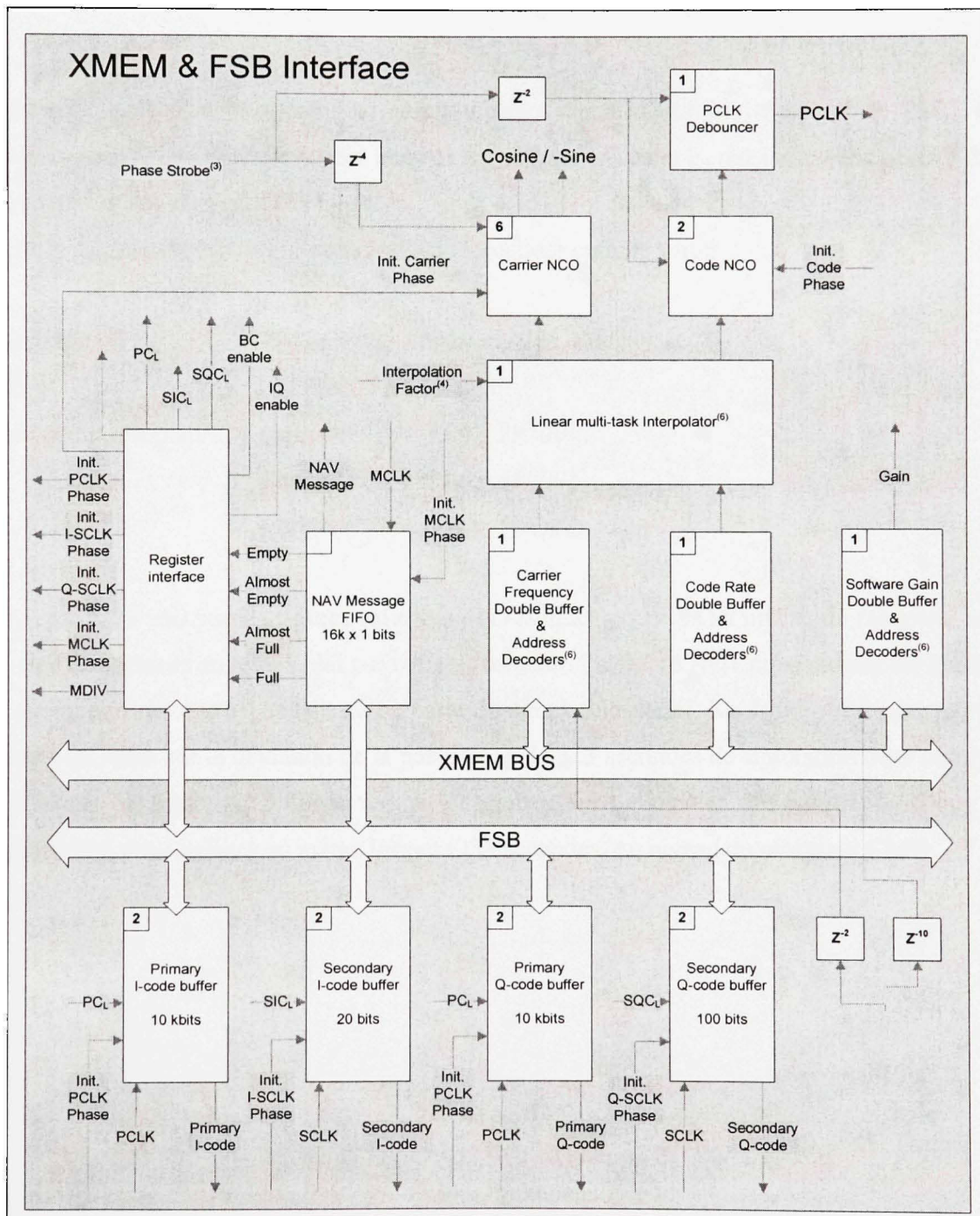


Figure A4.2 Les bus XMEM et FSB.

ANNEXE V

Résultats de la DLL MATLAB

Dans cette annexe, nous verrons les résultats retrouvés au niveau du récepteur de CMC. Ces résultats nous permettent de valider tous les algorithmes de calculs, ainsi que ceux utilisés dans la génération des signaux GPS L1.

Le premier scénario est défini au niveau du simulateur comme suit :

Constellation : *GPS (Fichier Almanach 040.alm)*

Date : *12/02/2007 à 15 :00 :30*

Position du récepteur : *Latitude N 45deg 29' 00''*

Longitude W 73deg 41' 00''

Masque d'élévation = 5 degrés.

Nous pouvons voir sur les figures suivantes les résultats retrouvés au niveau du récepteur avec les signaux générés en temps réel par notre simulateur. La figure A7.1 nous montre les données de navigation qui correspondent au scénario de simulation défini. La figure A7.2 nous permet d'avoir une idée sur la déviation de la position après 323 secondes de simulation dans un rayon de 1 mètre. La figure A7.3 donne une vision globale sur l'allocation des canaux, ainsi que sur les positions des satellites au même instant (323 secondes) au niveau du récepteur CMC.

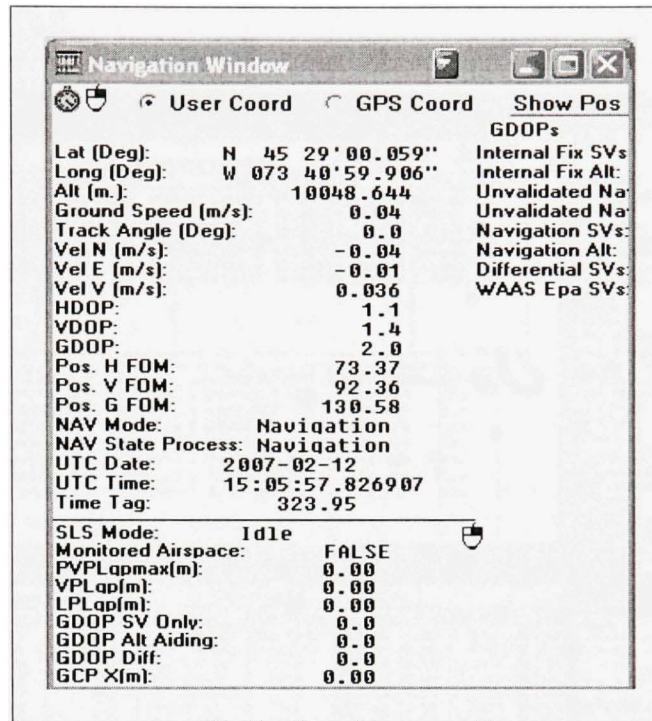


Figure A5.1 Les données de navigation (Scénario 1).

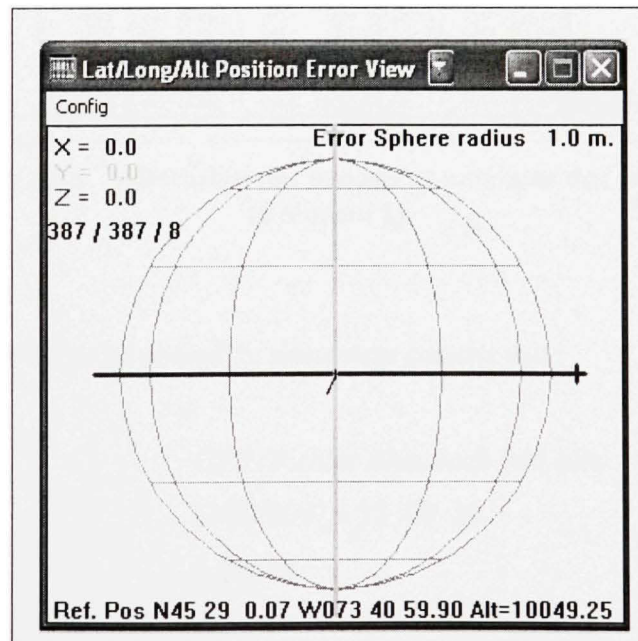


Figure A5.2 Niveau d'erreur de la position (Scénario 1).

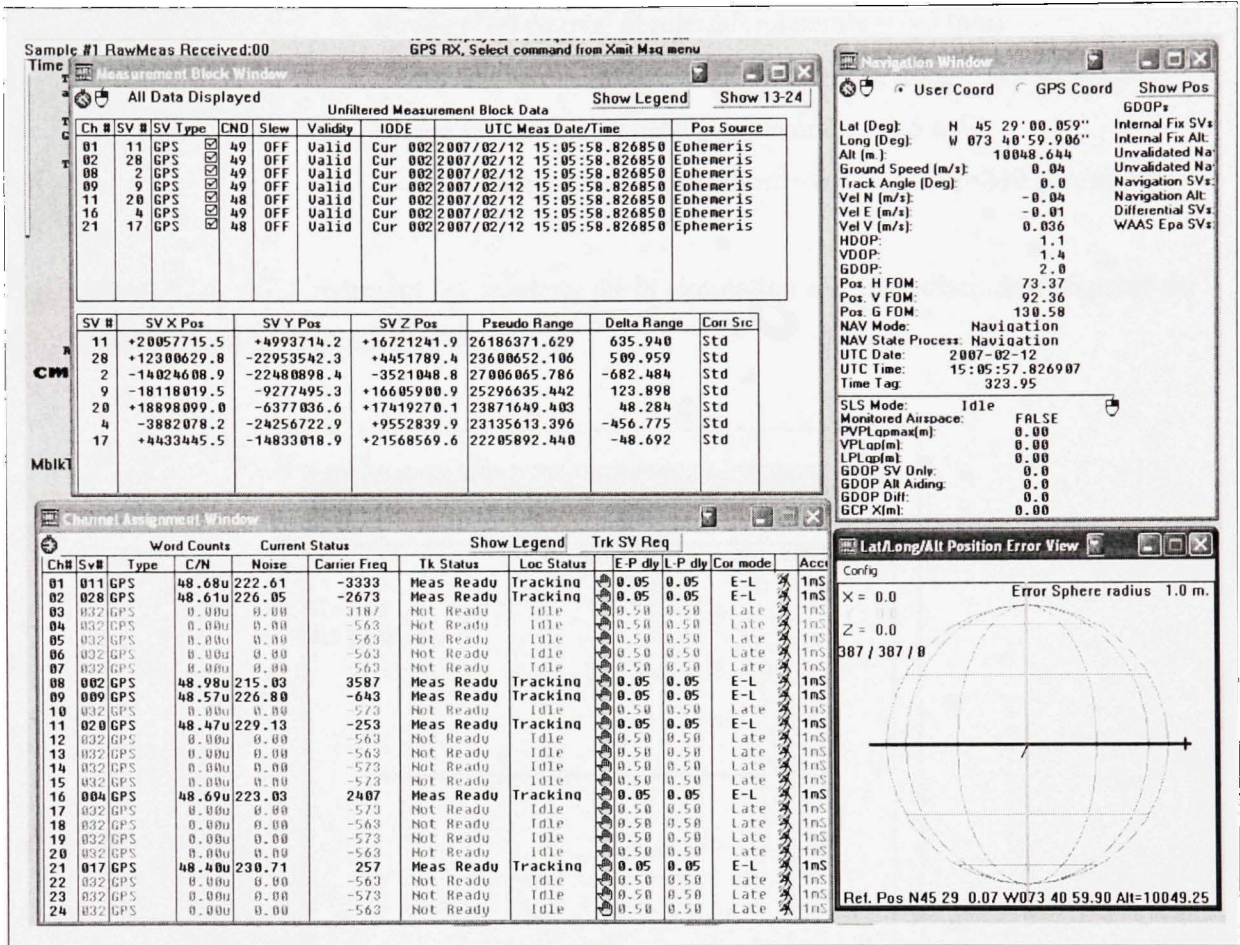


Figure A5.3 Allocation des canaux et positions des satellites (Scénario 1).

Le second scénario est défini au niveau du simulateur comme suit :

Constellation : GPS (Fichier Almanach 040.alm)

Date : 12/02/2007 à 15 :00 :30

Masque d'élévation : 5 degrés

Position initiale du récepteur : Latitude N 45deg 29' 00''

Longitude W 73deg 41' 00''

Trajectoire du récepteur :
 Ligne droite (accélération = 5 m/s, durée = 80 secondes)
 Virage (180 degrés, accélération latérale = 0.5 m/s)
 Ligne droite (accélération = 0 m/s, durée=240 secondes)
 Virage (90 degrés, accélération latérale = 0.5 m/s)
 Ligne droite (accélération = 0 m/s, durée=240 secondes)

Les figures A7.4, A7.5 montrent les résultats de la simulation sur l'interface du récepteur de CMC comme expliqué précédemment.

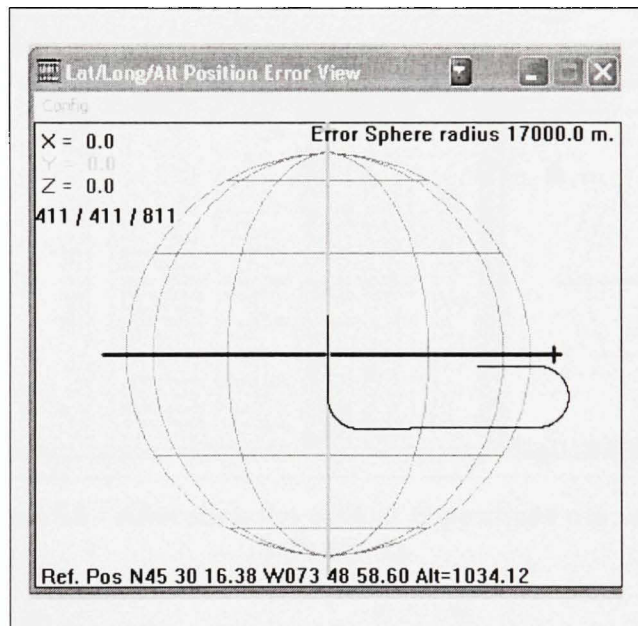


Figure A5.4 Niveau d'erreur de la position (Scénario 2).

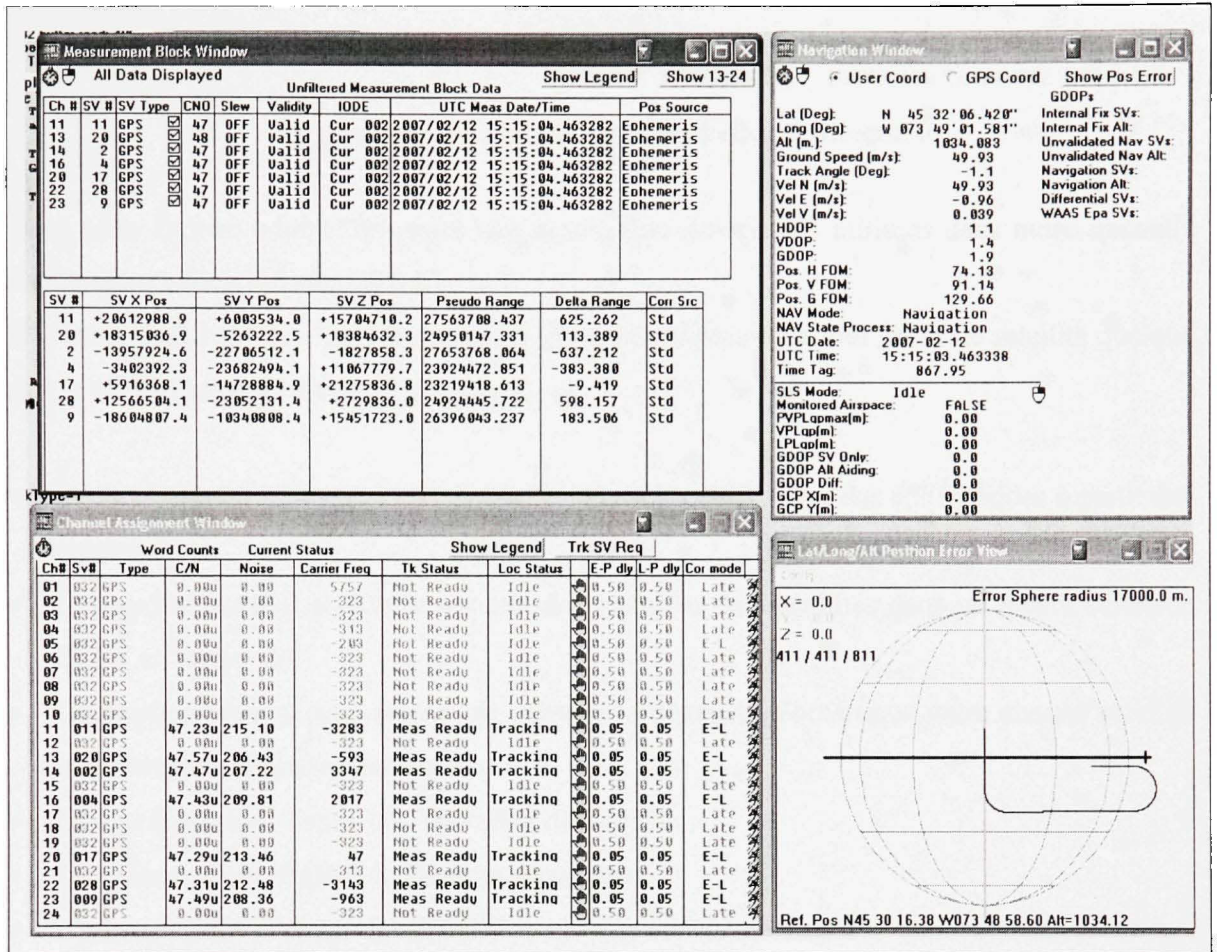


Figure A5.5 Allocation des canaux et positions des satellites (Scénario 2).

ANNEXE VI

Diagramme des classes de la nouvelle architecture

Dans cette annexe nous allons faire une description des classes utilisées dans notre nouvelle architecture logicielle pour GPS L1.

La classe 'Satellite' contient les méthodes qui sont utilisées pour tout genre de satellite quelque soit la constellation à laquelle il appartient :

- *UpdateEphemerisData*: permet de mettre à jour les paramètres des éphémérides à partir des almanachs.
- *UpdateSatPosition*: permet de calculer les positions des satellites dans un repère cartésien ECEF en temps réel.
- *UpdateGeometricRange*: permet de calculer la distance géométrique entre chaque satellite et le récepteur à chaque instant.
- *UpdateElevation*: Calcul de l'élévation du satellite.
- *UpdateAzimuth*: Calcul de l'azimut du satellite.
- *UpdateVisibility*: Permet de définir si le satellite est visible ou pas.

Ces méthodes sont expliquées en détails dans la section 2.2.1 du mémoire.

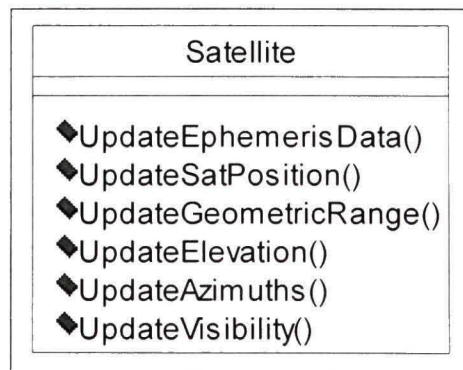


Figure A6.1 Classe Satellite.

La classe 'GPS' est une classe dérivée ou 'fille' de la classe 'Satellite'. Elle hérite donc toutes ses méthodes. D'autre part, 'GPS' implémente la méthode 'Generate_Nav_Message' qui est spécifique aux satellites appartenant à la constellation GPS, et qui permet de générer les sous-frames et les trames du message de navigation selon (ARINC, 2001).

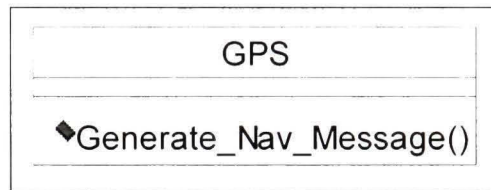


Figure A6.2 Classe GPS.

La classe 'GPS_L1' est une classe dérivée de la classe 'Frequency'. La classe 'Frequency' est une classe abstraite qui n'implémente aucune méthode. Toutes ses méthodes sont virtuelles et peuvent être implémentées de différentes manières dans les sous-classes comme 'GPS_L1', 'GPS_L5' ou 'Galileo_E1'...

Ces méthodes sont :

- 'UpdateTroposphericDelay' : Permet de calculer le délai troposphérique selon le modèle d'EGNOS.
- 'UpdateIonosphericDelay' : Permet de calculer le délai ionosphérique selon le modèle de Klobuchar.
- 'UpdateSVClockBias' : Permet de calculer le biais d'horloge du satellite.
- 'UpdatePseudorange' : Effectue le calcul de la pseudo-distance entre le satellite et le récepteur.
- 'UpdateDoppler' : calculer la valeur de Doppler à chaque instant pour chaque satellite.
- 'IFComputation' : Effectuer les calculs des paramètres permettant la génération des signaux GPS L1.

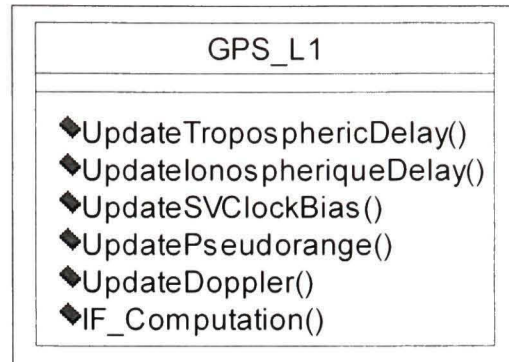


Figure A6.3 Classe GPS L1.

La classe '*ConstResultData*' contient les résultats, calculés en temps réel, des paramètres en rapport avec les positions des satellites et leurs mouvements.

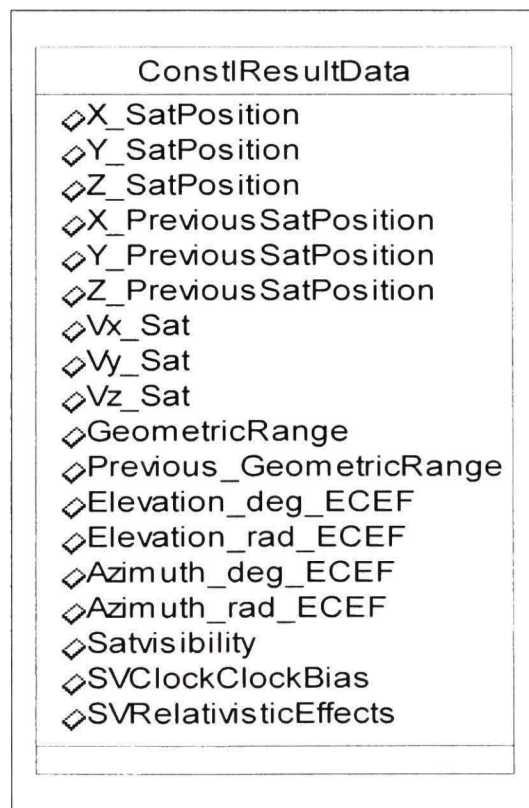


Figure A6.4 Classe résultats au niveau des satellite.

La classe 'RangeParam' contient les données calculées en temps réel, en rapport avec la pseudo-distance entre chaque satellite et le récepteur.

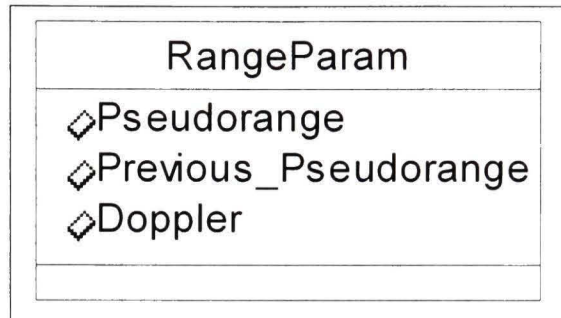


Figure A6.5 Classe des résultats au niveau des distances.

ANNEXE VII

Interface graphique de la nouvelle architecture logicielle

La nouvelle interface graphique donne la possibilité à l'utilisateur de définir un ou plusieurs scénarios à la fois. Ceci peut s'avérer très utile si nous voulons effectuer la comparaison des résultats retrouvés avec différentes valeurs de paramètres qu'on modifie d'un scénario à un autre (Figure A8.1):

- Paramètres relatifs au temps et durée de la simulation.
- Paramètres de la constellation (orbitaux, perturbations...).
- Paramètres du véhicule et sa trajectoire.
- Paramètres relatifs à l'atmosphère.
- Modélisation des multi-trajets.
- Contrôle de la puissance des signaux RF et du bruit.

Lors du lancement de notre application logicielle, nous avons mis en place un 'Wizard' qui permet à l'utilisateur de configurer d'une manière très rapide et très simple, les paramètres principaux d'un scénario :

- Nom du scénario et espace de travail (Figure A8.3).
- Choix de la constellation et chargement des fichiers Almanachs (Figure A8.4).
- Choix du véhicule (Figure A8.5).
- Choix du temps et de la durée de la simulation (Figure A8.6).

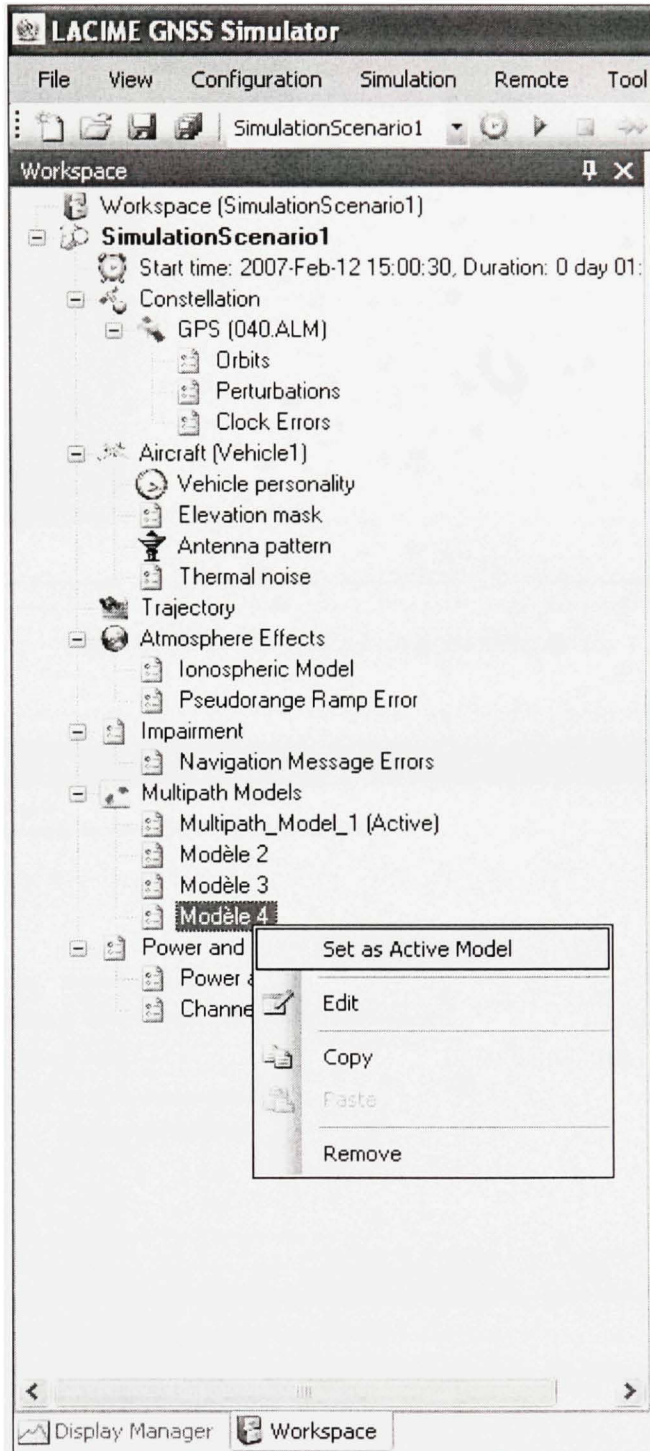


Figure A7.1 Espace de travail et configuration d'un scénario.



Figure A7.2 Page d'accueil du 'Wizard'.

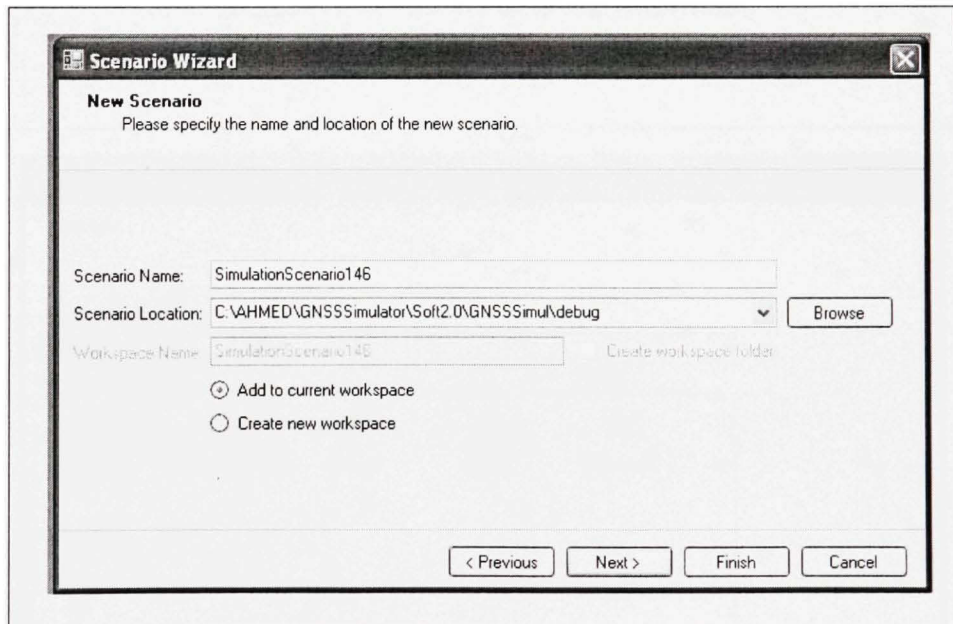


Figure A7.3 Nom du scénario et espace de travail.

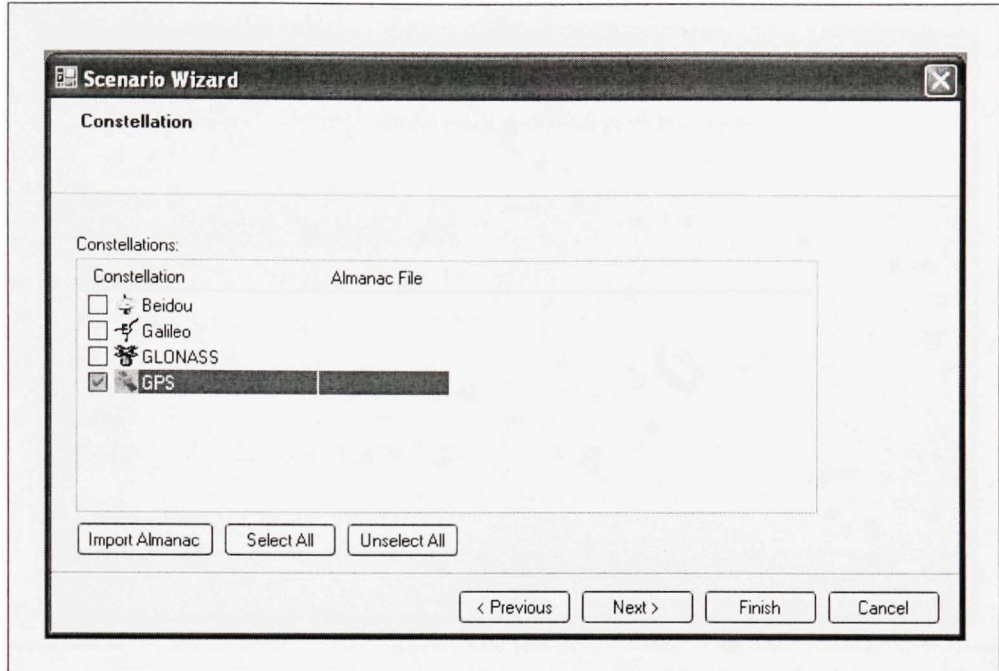


Figure A7.4 Choix de la constellation.

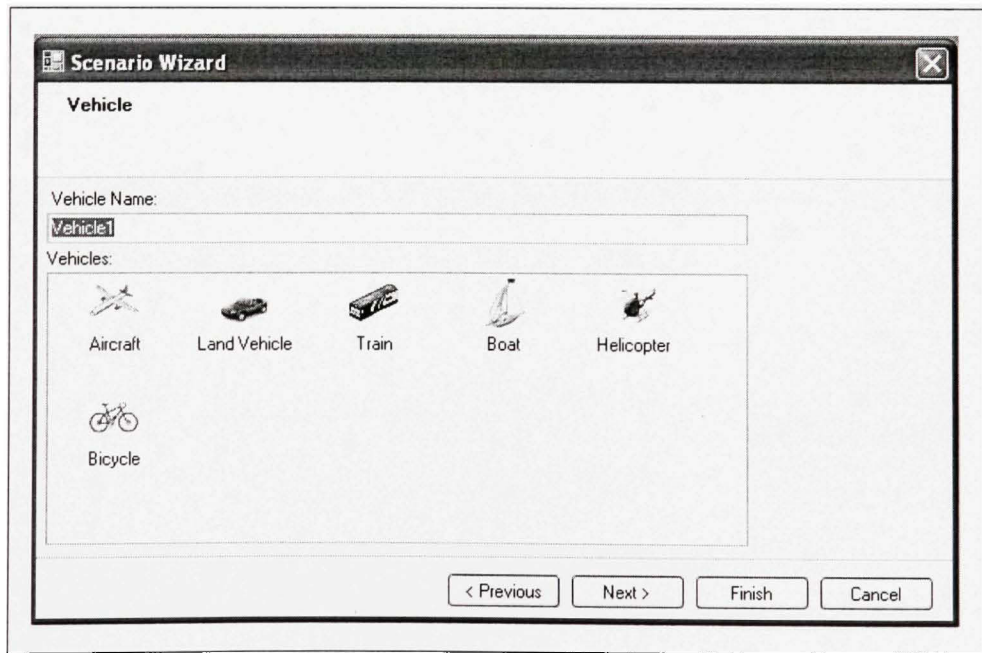


Figure A7.5 Choix du véhicule.

The image shows a software dialog box titled "Scenario Wizard" with a close button in the top right corner. The main heading is "Start Time and Duration" followed by the instruction "Please specify the time to start the simulation and the duration that the simulation is going to run." The dialog contains several input fields: "Start time" with a date dropdown set to "2009-10-26" and a time spinner set to "15:46:20"; "WN rollover" with a spinner set to "1"; "Week number" with a spinner set to "531"; "TOW" with a spinner set to "143180"; and "Duration" with a spinner set to "0" followed by the text "day(s)" and another spinner set to "00:01:00". At the bottom, there are four buttons: "< Previous", "Next >", "Finish", and "Cancel".

Scenario Wizard

Start Time and Duration
Please specify the time to start the simulation and the duration that the simulation is going to run.

Start time
Start time: 2009-10-26 15:46:20

WN rollover: 1

Week number: 531

TOW: 143180

Duration
Duration: 0 day(s) 00:01:00

< Previous Next > Finish Cancel

Figure A7.6 Choix du temps et de la durée de la simulation.

Constellation Configuration (SimulationScenario144)

Motion Orbit | Perturbations | Clock Errors

Constellation: GPS
 Satellite: 1
 Copy Paste

Almanac file 040.ALM information
 Number of satellites 32
 Reference time N/A

Present in almanac/ephemeris file

Reference time (WN and TOW [s]) 390 61440 Apply to Constellation

Root semi major axis ($A^{1/2}$) 5153.685059 meters^{1/2}

Eccentricity (e) 0.0066676140

Argument of perigee (W) -1.7576745850 rad

Mean anomaly (M0) -2.5607105310 rad

Inclination (I0) 0.9896297290 rad

Longitude of ascending node (W0) -2.1899077520 rad

Mean motion difference (Delta N) 0 rad/s

Inclination rate (I $\dot{0}$) 0.0000000000 rad/s

Right ascension rate (W $\dot{0}$) 0 rad/s

Mean motion difference rate (Delta N $\dot{0}$) 0 rad²

Load Ephemeris file Save ephemeris file Load Almanac file Save almanac file

Load Configuration Save Configuration OK Cancel

Figure A7.7 Paramètres orbitaux.

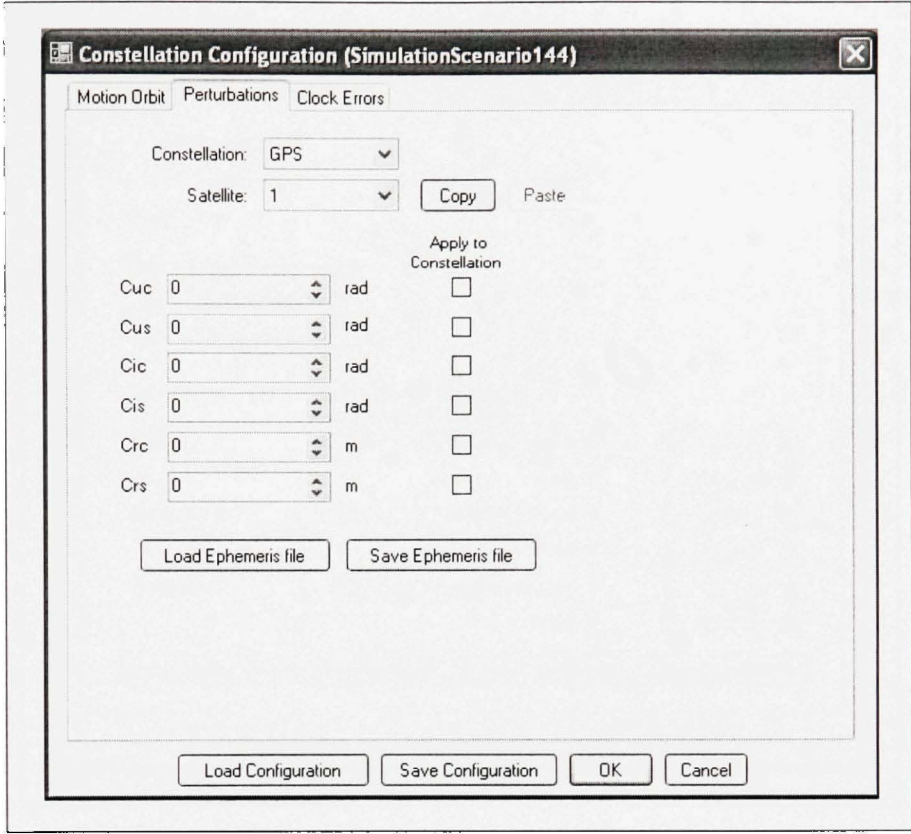


Figure A7.8 Paramètres de perturbations.

Constellation Configuration (SimulationScenario144)

Motion Orbit | Perturbations | **Clock Errors**

Constellation: GPS
Satellite: 1 [Copy] [Paste]

Clock Errors

Af0: 0,000106811523 s Apply to Constellation
Af1: 0,000000000003 s/s
Af2: 0,000000000000 s/s²

Clock divergence terms (effective times wit start of run)

Delta Af0: 0 s effective from day 0 to day 0
Delta Af1: 0 s/s effective from day 0 to day 0
Delta Af2: 0 s/s² effective from day 0 to day 0

[Load Almanac file] [Save Almanac file] [Load Ephemeris file] [Create Ephemeris file]

[Load Configuration] [Save Configuration] [OK] [Cancel]

Figure A7.9 Paramètres de correction d'horloge.

Vehicle personality

Linear dynamic limits

Max speed: 600,00 m/s
Max acceleration: 50,00 m/s²
Max jerk: 500,00 m/s³

Lateral dynamic limits

Max acceleration: 69,30 m/s²
Max jerk: 500,00 m/s³

Yaw limits

Max rate: 7,00 deg/s
Max acceleration: 70,00 deg/s²
Max jerk: 700,00 deg/s³

Pitch limits

Max angle: 1,5700 deg
Max rate: 7,00 deg/s
Max acceleration: 70,00 deg/s²
Max jerk: 700,00 deg/s³

Roll limits

Max angle: 3,1415 deg
Max rate: 7,00 deg/s
Max acceleration: 70,00 deg/s²
Max jerk: 700,00 deg/s³

[Load] [Save] [OK] [Cancel]

Figure A7.10 Personnalité du véhicule.

Ionospheric model

Frequency:

Satellite:

Applied to RF signal

Alpha 0	<input type="text" value="0,000000005"/> s	<input type="checkbox"/>	Apply to all satellites	Beta 0	<input type="text" value="79872"/> s	<input type="checkbox"/>	Apply to all satellites
Alpha 1	<input type="text" value="0,000000015"/> s/semicircle	<input type="checkbox"/>		Beta 1	<input type="text" value="65536"/> s/semicircle	<input type="checkbox"/>	
Alpha 2	<input type="text" value="-0,000000060"/> s/semicircle ²	<input type="checkbox"/>		Beta 2	<input type="text" value="-65536"/> s/semicircle ²	<input type="checkbox"/>	
Alpha 3	<input type="text" value="-0,000000060"/> s/semicircle ³	<input type="checkbox"/>		Beta 3	<input type="text" value="-393216"/> s/semicircle ³	<input type="checkbox"/>	

Broadcast into navigation data

Alpha 0	<input type="text" value="0,000000005"/> s	<input type="checkbox"/>	Apply to all satellites	Beta 0	<input type="text" value="79872"/> s	<input type="checkbox"/>	Apply to all satellites
Alpha 1	<input type="text" value="0,000000015"/> s/semicircle	<input type="checkbox"/>		Beta 1	<input type="text" value="65536"/> s/semicircle	<input type="checkbox"/>	
Alpha 2	<input type="text" value="-0,000000060"/> s/semicircle ²	<input type="checkbox"/>		Beta 2	<input type="text" value="-65536"/> s/semicircle ²	<input type="checkbox"/>	
Alpha 3	<input type="text" value="-0,000000060"/> s/semicircle ³	<input type="checkbox"/>		Beta 3	<input type="text" value="-393216"/> s/semicircle ³	<input type="checkbox"/>	

Figure A7.11 Modélisation ionosphérique.

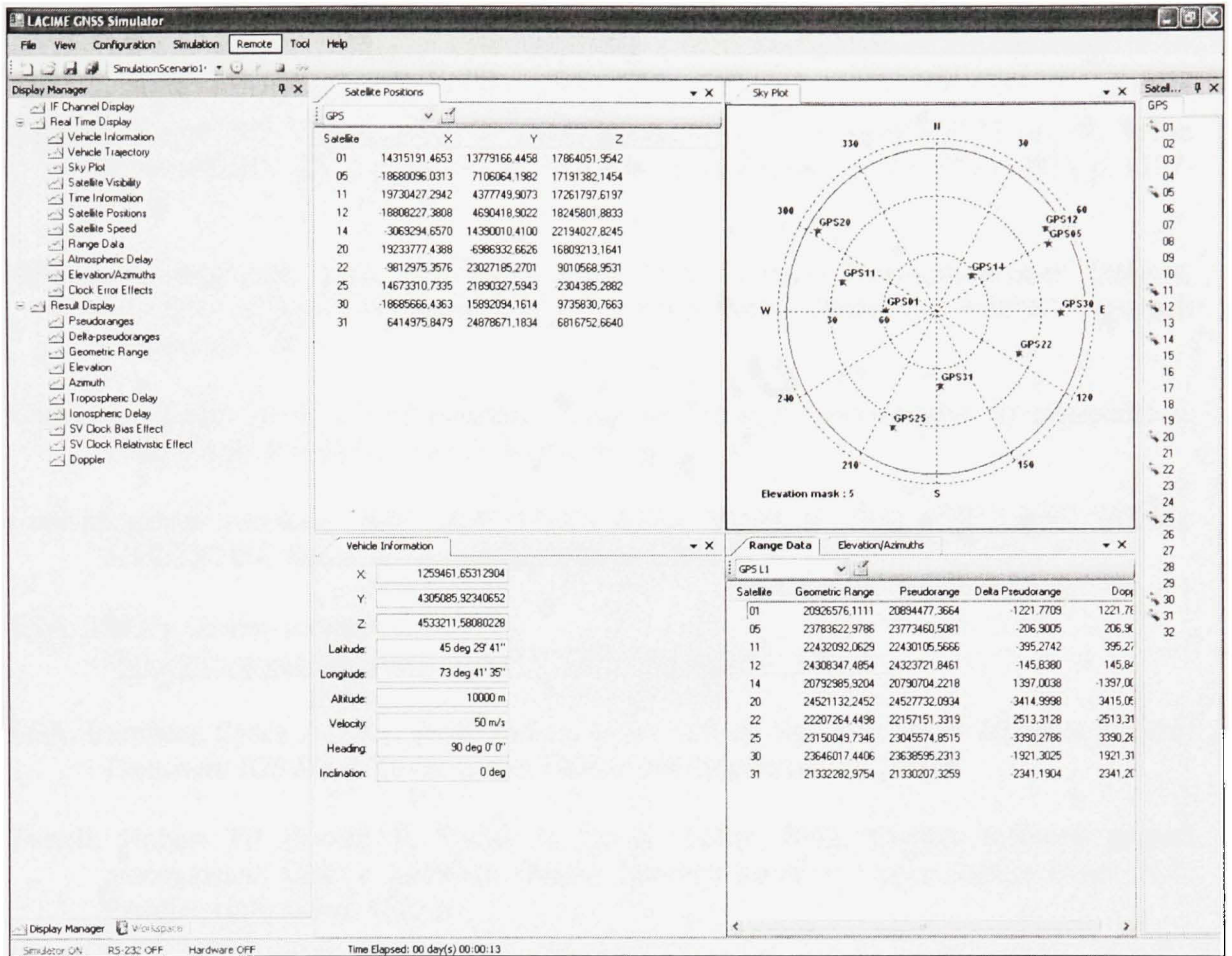


Figure A7.12 Résultats de simulation en temps réel.

BIBLIOGRAPHIE

- Akturan, R, et W.J Vogel. 1997. « Path diversity for LEO satellite-PCS in the urban environment ». *IEEE transactions Antennas Propagation*, vol. 45, n° 7 (1997), p. 1107-1116.
- ARINC, Incorporated. 2001. *NAVSTAR GPS Space Segment/Navigation User Inteface, Interface Control Document (ICD)*. El Segundo, California: ARINC Research Corporation, 70 p.
- Chaabane, Walid. 2010. « Implémentation d'une bibliothèque de simulation de trajectoire ». Tunis, École Polytechnique de Tunis, 100 p.
- Constantinescu, Aurelien. 2006. *SOFTWARE USER MANUAL FOR THE GNSS SIGNAL SIMULATOR*. École de Technoloogie Supérieure, 87 p.
- ESA. 2009. « Galileo services ». http://www.esa.int/esaNA/SEMTHVXEM4E_galileo_0.html.
- ESA, European Space Agency. 2006. *Galileo Open Service Signal In Space Interface Control Document (OS SIS ICD)*. Bruxelles: Galileo Joint Undertaking, 192 p.
- Futrell, Robert T., Donald F. Shafer et Linda Shafer. 2002. *Quality software project management*. Coll. « Software Quality Institute series ». Upper Saddle River, N.J.: Prentice Hall, xxxvi, 1639 p.
- Ilie, Iurie. 2003. « Récepteur Hybride GPS/GALILÉO : Simulation et analyse de robustesse avec le filtre FADP ». Montréal, École de Technologie Supérieure, 198 p.
- Ilie, Iurie, René Jr. Landry et Aurelien Constantinescu. 2005. « Simulation of GPS and Galileo Architectures for Anti-jamming and Multipath Analysis ». *Canadian Aeronautics and Space Journal*, vol. 51, n° 1 (March 2005), p. 13-22.
- Jeng-Kuang, Hwang, Li Jeng-Da, Chung Rih-Lung et Chen Chen-Yu. 2006. « Efficient Structure for FPGA Implementation of a Configurable Multipath Fading Channel Emulator ». In *Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on*. p. 481-484.
- Jeng-Kuang, Hwang, Lin Kuei-Horng, Li Jeng-Da et Deng Juinn-Horng. 2007. « Fast FPGA prototyping of a multipath fading channel emulator via high-level design ». In *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*. p. 168-171.

- Karasawa, Y, k Minamisono et T Matsudo. 1995. « A propagation channel model for personal mobile-satellite services ». In *Progress of Electromagnetic Research Symposium of the European Space Agency (ESA)* (July 1995). p. 11-15. Noordwijk, The Netherlands.
- Kelly, Joseph M., Michael S. Braasch et Michael F. DiBenedetto. 2003. « Characterization of the effects of high multipath phase rates in GPS ». *GPS Solutions*, vol. 7, n° 1, p. 5-15.
- Klobuchar, John A., et Joseph M. Kunches. 2001. « Eye on the Ionosphere ». *GPS Solutions*, vol. 5, n° 1, p. 68-69.
- Klukas, R., G. Lachapelle, C. Ma et G. I. Jee. 2003. « GPS signal fading model for urban centres ». *Microwaves, Antennas and Propagation, IEE Proceedings -*, vol. 150, n° 4, p. 245-252.
- Loo, Chun. 1985. « A statistical model for a land mobile satellite link ». *IEEE transactions on vehicular technology*, vol. 34, n° 3, p. 122-127.
- Loukil, Rami. 2010. « Conception et mise en œuvre de module de contrôle de puissance de génération de bruit pour conditionner les signaux RF d'un simulateur de constellation GPS et Galileo ». Montréal, École de Technologie Supérieure, 151 p.
- Monnerat, Michel. 2007. *Procédés d'acquisition accélérée des signaux satellites*. Lucent, Alcatel (inv.). En ligne 1 804 070 A1. Base de données sur les brevets. <<http://www.freepatentsonline.com/EP1804070.pdf>>. Consulté le 27 Octobre 2009.
- Porter, Michael E. 1986. *L'avantage concurrentiel : comment devancer ses concurrents et maintenir son avance*. Paris: InterÉditions, 647 p.
- Rappaport, Theodore S. 2002. *Wireless communications : principles and practice*, 2nd. Coll. « Prentice Hall communications engineering and emerging technologies series ». Upper Saddle River, N.J.: Prentice Hall PTR, xxiii, 707 p.
- Sahmoudi, M., Kouki, A., Landry R. 2010. « A new approach for mitigating Carrier Phase Multipath Errors in Multi-GNSS Real-Time Kinematic Receivers ». *International Conference on Acoustics, Speech and signal processing 2010*, paper ID 4276.
- Sauriol, Bruno. 2007. « GNSS Simulator Technical Note ». Montréal, École de Technologie Supérieure, 5 p.
- Sauriol, Bruno. 2007. « Mise en oeuvre en temps réel d'un récepteur hybride GPS-GALILEO ». Montréal, École de Technologie Supérieure, 308 p.
- SPIRENT, Communications. 2006. « SimGEN Software User Manual ». <<http://www.spirent.com/>>.

Tsui, James Bao-yen. 2005. *Fundamentals of global positioning system receivers : a software approach*, 2nd. Coll. « Wiley series in microwave and optical engineering ». New Jersey: John Wiley & Sons Inc., xvi, 352 p.

US Department of Homeland Security, Navigation Center. 2009. « Definition Of a YUMA Almanac ». <<http://www.navcen.uscg.gov/GPS/almanacs.htm>>.

Weiss, Jan Peter. 2007. « Modeling and characterization of multipath in global navigation satellite system ranging signals ». Colorado, University of Colorado, 203 p.

WORLD, GPS. 2009. « L5 signals ». <<http://www.gpsworld.com/defense/news/l5-gps-signal-now-being-transmitted-irm-20-satellite-7079>>.