

## CHAPITRE 2

### PRÉSENTATION DU PROJET SDN « Software Defined Navigator »

#### 2.1 Historique du concept du SDR, « Software Defined Radio »

L'évolution de la radionavigation demandant beaucoup de ressources, l'optimisation des ressources, autant matérielle que financière demeure un enjeu important dans le développement des nouveaux produits et services. Le domaine de la radionavigation n'est pas le premier domaine où son évolution rapide exige une nouvelle méthodologie de travail. Le domaine des radios télécommunications a dû s'adapter à la nouvelle réalité dans les dernières décennies.

Les origines de la communication radio se situent au 19<sup>ème</sup> siècle. Tout d'abord, les équations de Maxwell qui ont permis de comprendre et d'utiliser les ondes électromagnétiques. En 1876, Alexander Graham Bell invente le téléphone, outil qui a percé près de 99 % des foyers canadiens. En 1887, Heinrich Hertz conclut ses premières expériences et découvre ce que l'on appelle maintenant les ondes radio, les ondes « hertziennes ». Enfin, Marconi, en 1896, réalise la première communication radio. Dans les années 1960 et 1970, la demande en ondes radio est tellement grande, qu'on invente de nouveau protocole de communication tel l'allocation dynamique des canaux de transmission. C'est en 1978 que Bell Telephone (US) présente le premier réseau de communication cellulaire, devenu un standard en 1982[20].

Vers la fin des années 80, le domaine des ondes radio a connu un essor comparable à celui de la radionavigation en ce moment. De cette effervescence est né un concept, celui du « Software Defined Radio » communément appelé SDR.

La définition du SDR est un système de radio communication qui utilise une partie logiciel pour la modulation et démodulation d'un signal radio [20]. Le système performe donc une majorité d'opérations sur le signal radio dans un microprocesseur, qu'il soit à l'intérieur d'un ordinateur ou sur un processeur reprogrammable. L'objectif de ce système est de produire un nouveau protocole de modulation/démodulation simplement par la compilation d'un nouveau logiciel.

Le concept idéal d'un tel système serait de raccorder un convertisseur analogique numérique à une antenne radio, tel que montré à la **Figure 8**. Les données numériques ainsi recueillies sont envoyées à un ordinateur qui, à l'aide d'un logiciel, transforme ces dernières en informations utiles à l'utilisateur. Cette partie traite de la démodulation seulement. On pourrait facilement construire une modulation radio à partir du même principe, mais en attachant un convertisseur numérique-analogique à l'antenne pour l'émission des ondes radio.

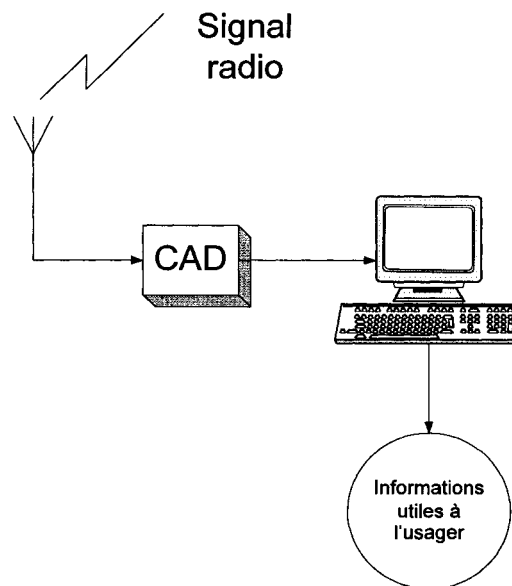


Figure 8 Concept du SDR

L'armée américaine est la première à avoir développé le concept de SDR. Ce projet s'appelait SpeakEasy[20]. Le but premier de ce projet est d'utiliser des processeurs reprogrammable afin d'émuler plus de 10 fréquences radio militaires opérant dans les bandes de fréquences entre 2 et 200 MHz. Un aspect important de ce concept est l'incorporation de nouveaux types de codages et de techniques de modulation dans le futur, afin de suivre le développement dans ces domaines.

Au début des années 90, le projet SpeakEasy entrait dans sa première phase. L'armée américaine voulait avoir une radio pouvant œuvrer dans le spectre de 2 MHz à 2 GHz afin de pouvoir communiquer avec toutes sortes de radios : les radios navales, les radios de l'aviation, les radios de l'armée de terre ainsi que les satellites. D'autres objectifs plus particuliers étaient la conception d'un autre format de signal radio en 2 semaines, et de construire une radio en connectant différents modules adaptables.

Ce projet a atteint ces objectifs, mais il a démontré quelques failles au niveau du traitement des signaux reçus, ne pouvant supporter qu'une seule communication à la fois. Le récepteur était composé d'une antenne, alimentant un amplificateur et un mélangeur. Le signal ainsi créé était amené à un convertisseur analogique numérique relié via un bus à plusieurs processeurs Texas Instrument de la série C40. Du côté de la source du signal, un convertisseur numérique analogique relié à un bus PCI était branché à une antenne radio et un amplificateur. Ce design est devenu, avec le temps, un modèle standard pour un projet de type SDR.

La deuxième phase de ce projet se voyait un peu plus audacieuse par les objectifs fixés. Le premier objectif était de construire une architecture reprogrammable plus rapide, dans une architecture ouverte, avec plusieurs canaux à protocoles différents. De plus, on constata que ce genre de système pouvait réduire les coûts et ils ont voulu faire un prototype plus petit et moins coûteux. Dans un projet basé sur un plan triennal, on obtint des résultats dans les premiers 15 mois du programme.

L'architecture avait été séparée en différents modules :

- Contrôle du modem : module gérant la modulation et démodulation des signaux
- Processeur du signal : module servant à moduler le signal
- Processeur de cryptographie : module servant à coder le message
- Multimédia : module pour analyser la voix humaine
- Interface : module servant de GUI (*Graphical User Interface*)
- Routage : module de réseau afin de gérer les communications entre les modules
- Contrôle : module de contrôle du modèle

Ces différents modules communiquaient sur un bus PCI entre eux, d'où la nécessité d'avoir des modules de contrôle et de routage. Pour la première fois dans l'histoire du concept de SDR, on utilisait des FPGA (*Field Programmable Gate Array*) dans le modèle. À partir de ce moment, le temps de programmation du FPGA devint alors un facteur à prendre en considération.

Dès lors, on commence de plus en plus à se regrouper afin de produire un système ou une architecture commune. Cette architecture est le *Software Defined Radio* (SDR). Le principe du SDR est simple : utiliser un logiciel afin de moduler et démoduler les signaux radio avec un but ultime d'avoir un signal reçu ou transmis par le simple fonctionnement d'un logiciel. Une partie importante du traitement du signal est effectuée à l'intérieur d'un processeur (qu'il soit à l'intérieur d'un ordinateur ou monté sur une plate-forme à part), et l'analyse peut être faite autant par des instruments de mesure que par l'étude des données logicielles. La composition matérielle d'une radio « *software defined* » se définit par les outils suivants :

- Modulateur/Démodulateur superhétérodyne RF
- Convertisseur analogique numérique et convertisseur numérique analogique

Cette simplicité de montage fait en sorte que les plateformes de développement sont moins coûteuses. Et comme le traitement se fait par voie logicielle, il est plus facile de l'implémenter à travers d'autres plate-formes matérielles. Le projet GNU Radio a tiré rapidement avantage de cette fonctionnalité en distribuant à travers le monde une SDR gratuit avec tous les outils pour construire et déployer une SDR. Son aspect le plus pratique, son côté reprogrammable, permet à toute personne pouvant modifier un code informatique de créer sa propre SDR, prouvant encore une fois la flexibilité du projet.

## 2.2 Origine et approche utilisé dans le projet SDN

En 1998, le gouvernement américain a développé un plan de développement sur la radionavigation[28]. Ce rapport à été entendu et révisé par le département de la défense nationale et le département des transports. Ce rapport recommandait que le système GPS devienne le seul système de positionnement utilisé par le département des transports du gouvernement américain. Le système était perçu comme étant fiable et invulnérable, le développement d'un autre moyen de positionnement était purement inutile. Les systèmes de positionnement existant déjà, tel que LORAN (*Long Range Navigation*), ILS (*Instrument Landing System*) et NDB (*Non-Directional Beacon*) par exemple, se devaient d'être mis sur les tablettes. Les économies engendrées par cette décision devaient être importantes pour le gouvernement américain. Cependant, la vulnérabilité du système GPS a été mise en évidence et toutes les conclusions de ces rapports ont dû être révisées.

La possibilité de perdre le signal GPS implique qu'une solution alternative doit être trouvée afin de permettre à un mobile de connaître sa position en tout temps. Dans le cas d'un avion, par exemple, le positionnement est important afin d'assurer un atterrissage sécuritaire et ce, peu importe les conditions climatiques et/ou le trafic. Mais les systèmes

existants, n'ayant pas été développés pour les raisons mentionnées plus haut, ne sont pas capables de fournir l'aide requise au niveau du GPS.

Le gouvernement américain et le département de la défense se devaient donc de trouver une solution conjointe avec le département des transports afin de palier à cette mauvaise interprétation. Soit ils modernisaient le système GPS afin de le rendre moins vulnérable, soit ils investissaient dans le développement d'autres systèmes[28]. La deuxième solution n'était pas réaliste à cause de l'investissement déjà fait à travers les années sur le système GPS. La modernisation du système GPS, tel que décrite précédemment dans le texte, était donc nécessaire afin de combler ces lacunes.

Les utilisateurs civils, dont le département des transports, ont seulement accès au code C/A du signal GPS, empêchant, par le fait même, de faire une correction double fréquence du retard ionosphérique. Les signaux L1 et L2 ne fournissent pas un spectre de protection totale et, étant noyés dans le bruit, ont des puissances faibles. Ceci engendre une vulnérabilité aux niveaux des brouilleurs et aux interférences. De plus, en région urbaine, les structures de béton et d'acier provoquent des interférences et des multi-trajets. Les brouilleurs spécialisés ainsi que les multi-trajets provoquent des interférences qui empêchent la démodulation correcte du signal par le récepteur, problème qui devra être traité par notre récepteur numérique.

Deux catégories de besoins ont été considérées pour la modernisation du système GPS, et de ces catégories des pistes de solutions ont été envisagées. Pour les applications militaires, de nouveaux signaux avec séparation spectrale et une puissance augmentée pour améliorer la protection, la prévention et la préservation du système sont les besoins les plus appréciés. Dans les applications civiles, les demandes étaient beaucoup plus nombreuses :

- Augmenter la disponibilité du signal
- Augmenter la précision du signal

- Augmenter la redondance du signal
- Ajouter 2 fréquences civiles :
  - Une pour la correction ionosphérique
  - Une pour les applications « Safety-Of-Life (SOL) »

Plusieurs options ont été envisagées dont celles-ci :

- ✓ Augmentation de la puissance
- ✓ Addition d'un code militaire aux bandes de fréquences L1 et L2
- ✓ Stations au sol supplémentaires
- ✓ Augmentation du nombre de satellites

Outre l'autorisation d'un nouveau signal civil à 10.23 MHz, les signaux L1 et L2 seront disponibles aux applications civiles. Les demandes de l'aviation exigent des caractéristiques très précises et il est décidé de créer un nouveau signal pour cette application, dans la bande L5, à 1176,45 MHz. De plus, on a créé un comité gérant le système GPS, l'Interagency GPS Executive Board (IGEB), duquel les membres seraient issus des différents départements (Commerce, Intérieur, Justice, Agriculture, Etat, Transport, Défense et de la NASA) utilisant le GPS afin que les besoins de chacun soit respectés.

Le programme GPS III offrira un service encore plus complet. Son objectif est de satisfaire les besoins autant civils que militaires. Une architecture nouvelle et une meilleure valeur d'acquisition entreront en jeu et répondront aux besoins civils pour le positionnement spatial, la navigation et le système de référence temporel jusqu'en 2030. L'armée de l'air américaine désire que GPS III offre un potentiel de meilleure robustesse aux brouilleurs en ajoutant 2 bandes pour le code M sur les canaux L1 et L2.

Tous ces changements impliquent une évolution technologique relativement importante, autant au niveau matériel qu'au niveau logiciel. En fait, au niveau matériel, la demande est sans cesse grandissante. Pensons seulement aux largeurs de fréquence sur lesquelles on travaille. À des fréquences dépassant le gigahertz, on utilise des bandes de fréquences d'une largeur de 10 MHz.

En comparant l'évolution de la radionavigation et de la radio communication, on se rend compte de plusieurs similitudes :

- ✓ Demande grandissante de plusieurs domaines
- ✓ Évolution rapide de la technologie
- ✓ Incorporation du système dans plusieurs technologies existantes
- ✓ Demande de nouveaux services
- ✓ Demande d'augmentation de la qualité et de la robustesse du signal
- ✓ Création d'un organisme mondial pour la gestion des technologies

À l'intérieur du message transmis par les satellites, il y aura des différences au niveau du codage et de la distribution des données. Par exemple, dans le cas du GPS, il y a le code C/A et le code M tandis qu'avec Galileo, le code BOC (« Binary Offset Carrier ») sera privilégié. Donc, comme on le verra plus tard à la section 3.4, dans l'architecture des nouveaux récepteurs, chaque composante sera dédiée à une tâche en particulier puisque les signaux sont si spécifiques.

Pour l'intégration d'un nouveau service sur une technologie existante, deux options s'offrent aux chercheurs et/ou programmeurs, tel que montré à la **Figure 9** :

1. Acheter tout le matériel requis afin de développer un prototype,
2. Trouver une plate-forme reprogrammable, alliant logiciel et matériel pour continuer le développement.



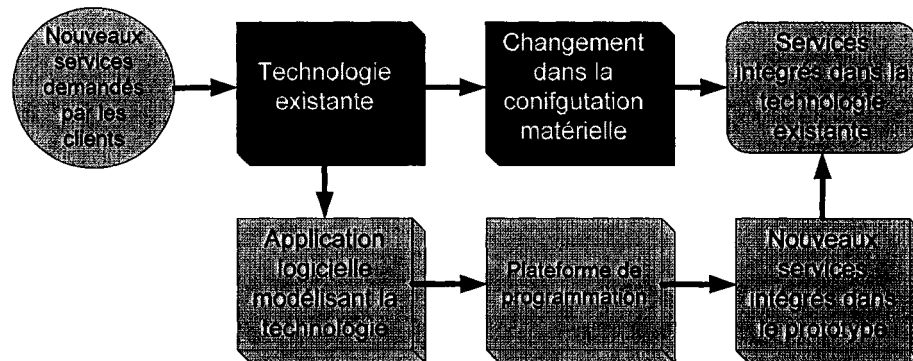


Figure 9 Schéma bloc de l'intégration d'un nouveau service

La première solution engendre des coûts assez élevés. Comme il y a peu de redondances dans la partie démodulation, le récepteur devra être composé de plusieurs parties indépendantes. De plus, les systèmes étant en évolution constante, il est difficile de prédire exactement le fonctionnement des systèmes de positionnement. Par exemple, pour le signal de Galileo, la Communauté Européenne hésite encore entre quelle modulation BOC (Binary Offset Carrier) sera utilisée pour moduler le signal. Cependant, en mai 2006, une décision a été prise sur le type de modulation que supportera Galileo. Mais comme nos travaux étaient déjà effectués, nous ne tiendrons pas compte de cette décision dans ce travail. Il n'est pas impossible aussi qu'il y ait des modifications au niveau des services offerts. Chaque changement ou modifications apportées entraîne un changement dans le modèle et, par conséquent, dans les composantes requises pour la conception du prototype. Avec la valeur des pièces actuellement sur le marché et l'assemblage, le coût de cette option est assez élevé et peut satisfaire certaines compagnies mais comporte plusieurs points à éviter, tel la limite de développement et la difficulté d'adaptation.

La deuxième option, favorisée par les compagnies de développement en radiocommunication, constitue l'approche « software defined ». La combinaison

matérielle et logicielle engendre certains problèmes mais ceux-ci sont contournables par l'élaboration de protocoles de communication. Déjà, plusieurs fabricants de matériel investissent dans le développement d'outils de communication entre leurs composantes et un ordinateur. L'utilisation d'une plateforme de programmation unique permet aux développeurs de tester plusieurs aspects sans avoir à apprendre de nouveau les différents protocoles de communication. La plateforme doit cependant répondre à des caractéristiques précises et doit satisfaire aux normes les plus exigeantes de la technologie afin d'assurer un développement adéquat.

Dans le cadre de ce mémoire, l'approche « software defined » comporte plusieurs avantages. Au niveau des coûts, nous devons travailler avec un budget limité dans un cadre de laboratoire de recherche universitaire. L'achat d'une seule plateforme reprogrammable nous permet de réduire les coûts de recherche et développement. De plus, elle permet un prototypage rapide de notre modèle de communication satellite. Dans la prochaine section, nous vous présenterons l'environnement de travail et la plateforme de développement.

### **2.3 Environnement de travail et outils utilisés dans le projet SDN**

L'environnement de travail est la clé du succès dans ce type de projet. La première étape est l'analyse du système. Cette étape permettra de déterminer les outils qui seront les plus adéquats et les plus performants pour la réalisation de notre prototype. Cette étape sera abordée plus en détail dans le CHAPITRE 4 mais, pour cette section, nous en garderons les grandes lignes. Tel que présenté dans le CHAPITRE 1, le signal GPS est en constante évolution mais les bases restent inchangées. Premièrement, la fréquence de la porteuse se situe à exactement à 1575.42 MHz. Une condition imposée par cette caractéristique est la fréquence d'opération (ou fréquence d'échantillonnage) des composantes électroniques contenues sur la plate-forme. Comme ce signal est porteur du code et des données, on doit, à son tour, l'échantillonner adéquatement afin de respecter

le théorème de Nyquist et recouvrir les données. Cette facette engendre une fréquence minimum au niveau des opérations.

Deuxièmement, dans notre modèle de récepteur GPS, nous utilisons plusieurs calculs arithmétiques, dont certains sont plutôt complexes. Ce facteur deviendra limitatif lorsque nous travaillerons en temps réel et devra être tenu en compte pour les différents délais que nous obtiendrons dans nos résultats.

Troisièmement, la rapidité des communications entre les différentes cartes devient importante. Puisque nous devons échantillonner à très haute fréquence et utiliser des algorithmes complexes, les protocoles de communication se doivent d'être rapides, efficaces et simples.

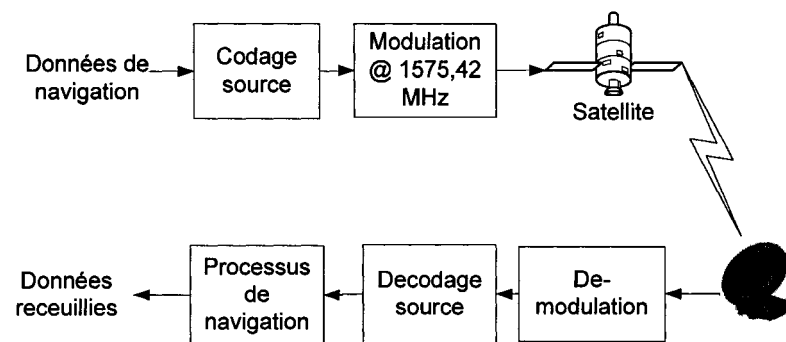


Figure 10 Chaîne de communication GPS

À la **Figure 10**, nous pouvons voir le système complet que nous voulons modéliser, des données de navigation envoyées jusqu'au décodage de ces données. Les buts poursuivis par notre projet sont les suivants :

1. Simuler et traiter en cosimulation (tel que montré à la **Figure 14** plus loin dans cette section) et en temps réel la source

## 2. Simuler et traiter en cosimulation et en temps réel le récepteur

On remarque à la **Figure 10** que nous avons de l'interaction entre plusieurs signaux (par exemple, codage, modulation, décodage,...). Ces processus devront être tenus en compte lors de l'analyse de notre modèle. Plusieurs processus ne requièrent pas les mêmes caractéristiques, certains étant plus complexe que d'autres. Par exemple, le taux d'échantillonnage devra être élevé dans certain cas (la démodulation), alors que la complexité des calculs sera la limitation d'un autre processus (le processus de navigation).

À travers la multitude d'offre de processeurs qui se trouve sur le marché, le choix d'un processeur demande une analyse plus poussée. Cette analyse, et le choix, seront basés sur quatre attributs de notre produit final : complexité, vitesse, prise de décision et algorithmes mathématiques. La complexité de notre système se mesure par deux caractéristiques : la diversité et le nombre de tâches que l'on doit exécuter. Pour un récepteur GPS, non seulement il y a beaucoup de tâches à exécuter mais elles vont de la modulation à l'algorithme complexe des discriminateurs. La vitesse aura aussi un rôle important à jouer, puisque nous allons traiter des signaux à fréquence élevée. Un processeur rapide est essentiel comme cible à notre méthodologie. De plus, les algorithmes mathématiques complexes contenus à l'intérieur de notre modèle seront nombreux, le processeur doit être capable de supporter des calculs mathématiques. De la démodulation d'un signal, jusqu'au calcul de l'erreur de phase par la méthode de la puissance avance moins retard normalisée, la liste des algorithmes mathématiques utilisés est longue. Ces algorithmes seront présentés au CHAPITRE 3.

Les valeurs recherchées dans les processeurs sont donc les suivants en ordre d'importance :

1. Vitesse
2. Complexité

3. Algorithmes mathématiques
4. Prise de décision

La partie prise de décision est une option secondaire sur laquelle on ne s'attardera pas. Les valeurs des résultats des calculs mathématiques effectués à l'intérieur du modèle serviront à guider les boucles au niveau des décisions à prendre. Cependant, un autre critère peut être ajouté à ce modèle, soit la mémoire disponible.

Basé sur ces critères de sélections, 2 choix possibles s'imposent : un FPGA (pour la vitesse et les algorithmes mathématiques) et un DSP (pour la complexité). Les avantages de chacun de ces processeurs pour notre modèle sont présentés brièvement dans les sections 4.4.1 et 4.4.2.

Au niveau de la vitesse, le problème peut se scinder en deux : la vitesse de l'horloge et les performances des fonctions. Le FPGA, sur ce point, l'emporte sur le DSP. Les fonctions peuvent être optimisées rapidement avec un minimum de coups d'horloge. Les DSP peuvent combler ce problème avec des bus haute vitesse mais ils n'atteignent pas la vitesse du FPGA. Les algorithmes mathématiques peuvent être optimisés sur les FPGA pour obtenir une rapidité des plus efficace. Le problème sur ce point réside dans la tête du programmeur puisque l'optimisation doit être faite par programmation, augmentant ainsi le temps de développement du processus. Sur l'autre point, les performances des fonctions, le DSP est mieux équipé que le FPGA car ses performances dépendent moins de la programmation mais plutôt des attributs matériels du processeurs. Au niveau de la complexité, le DSP possède des *pipelines* aidant la distribution des données, ce qui permet de pouvoir utiliser les bonnes ressources au bon moment. Et le DSP est rempli d'applications embarquées qui permettent l'introduction d'algorithmes complexes à l'intérieur des calculs.

Puisque les avantages sont intéressants dans le cas des deux différents processeurs, la combinaison des deux devient une avenue à envisager. Pour notre récepteur GPS, la vitesse de fonctionnement et la complexité sont des critères équivalents. Avec des protocoles de communications efficaces et une gestion optimale du transport des données optimale, la combinaison des deux processeurs répondrait exactement aux besoins identifiés. L'échange de données entre les processeurs pourrait avoir un impact minimal sur les performances du système et permettrait à notre plate-forme de tirer profit des caractéristiques des processeurs en implémentant des algorithmes complexes et rapides. La décision prise dans le cadre du projet est donc d'allier DSP et FPGA pour maximiser et optimiser les performances du récepteur GPS. De plus, en pensant à l'avenir du projet, la conciliation des deux processeurs permet d'envisager le développement de nouveaux systèmes combinés, tel que Galileo ou Glonass, sur cette même plate-forme, optimisant l'investissement monétaire effectué sur cette carte.

### **2.3.1 Matériel utilisée dans le projet**

D'après les caractéristiques et objectifs énumérés plus haut, une recherche de plateforme de développement au niveau matériel fut entreprise. Il existe, sur le marché des cartes de développement, plusieurs possibilités qui peuvent répondre à nos besoins compte tenu de notre budget. Voici un résumé de nos besoins :

- Fréquence d'échantillonnage élevée
- Convertisseur analogique-numérique à large bande
- Processeur à algorithme complexe
- Entrée analogique
- Grande capacité d'espace mémoire
- Processeur à haute vitesse
- Port série (pour communiquer avec l'ordinateur)

Le choix d'une plateforme est crucial dans la démarche du projet et plusieurs options et compagnies ont été étudiées. La plateforme retenue est produite par Sundance Inc[14]. La plateforme se compose de quatre (4) parties :

- 1- Carte d'acquisition de données
- 2- Carte modulaire avec FPGA
- 3- Carte modulaire avec un DSP
- 4- Carte-mère

Le fait que la plateforme se compose de quatre (4) cartes nous donne une option de plus que prévue. En effet, puisque nous avons trois (3) modules distincts, une flexibilité au niveau de l'adaptation de la plateforme de développement offre la possibilité de mettre à jour le système, tel qu'observé par Avrunin et ass [6]. En effet, avec l'évolution grandissante des signaux et les changements continuels, certaines cartes choisies pourraient s'avérer rapidement désuètes. Cependant, des cartes modulaires mieux adaptées s'interchangeraient facilement avec la carte actuelle. Le fait de changer seulement une carte modulaire et non la plateforme entière est un atout majeur pour l'évolution du projet car il diminue grandement les coûts de développement.

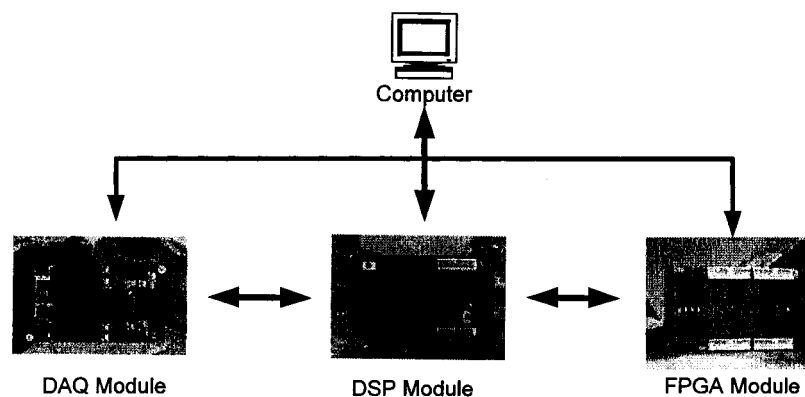


Figure 11 Interaction entre les cartes modulaires

La Figure 11 nous montre l'interaction entre les différentes cartes modulaires et l'ordinateur. Ces communications se font par l'intermédiaire d'une carte-mère sur laquelle sont déposées et branchées les cartes modulaires. De plus, une fois les modules fixés sur la carte mère, cette dernière est fixée sur le bus PCI de l'ordinateur afin de communiquer avec lui. Cette carte a une horloge interne de 33 MHz sur son bus PCI, une possibilité d'y installer quatre cartes modulaires et de la mémoire à partager entre ces différents modules pour la configuration.

### 2.3.2 Logiciels requis

Suite à la sélection du support matériel requis, les logiciels de programmation, de génération de code et de communication deviennent donc indispensables. Ces différents logiciels sont utilisés afin de convertir notre modèle en langage machine compréhensible pour les processeurs dans lesquels les programmes iront. Avant de déterminer les différents logiciels que nous utiliserons, nous devons entrevoir les différentes étapes par lesquelles notre modèle passera afin de terminer en langage machine.

1. Définition de notre modèle de base.
2. Changement du modèle pour le passage en temps réel : au niveau du processeur digital et au niveau du FPGA.
3. Génération des codes adaptés au DSP et au FPGA.
4. Encapsulation des codes à l'intérieur des protocoles de communication pour les processeurs.
5. Activation des codes et retour d'information.

Il n'existe pas, pour notre projet, un logiciel ayant la capacité de faire toutes ces étapes. Il nous faudra analyser notre modèle et les caractéristiques des logiciels afin de faire correspondre les deux. Ainsi, nous devons diviser notre travail en différents processus et sous processus. Cependant, le fait de faire intervenir plusieurs logiciels peut entraîner



des problèmes, au niveau de la compatibilité entre eux. Il n'est pas acquis que les fichiers résultant d'un logiciel de niveau hiérarchique supérieur soient compatibles avec celui des niveaux inférieurs. L'interaction entre 2 logiciels entraîne systématiquement des erreurs de communications ou d'échange de données. La réduction au minimum du nombre d'interaction favorisera un développement plus rapide et une diminution du nombre d'erreurs dans l'échange de données.

Pour la simulation de notre récepteur numérique GPS, la suite Simulink de Matlab, créée par MathWorks, possède des bibliothèques détaillées qui nous permettent de diviser les processus de notre modèle en de simples opérations. Cette division des processus en tâches simples facilitera notre tâche au niveau de l'implémentation puisque les algorithmes utilisés seront simples à leur tour. De plus, l'environnement Matlab/Simulink offre rapidité, précision et flexibilité. Rapidité puisqu'il permet de simuler rapidement notre modèle, facilitant la validation de celui-ci. De plus, nous pouvons utiliser des fréquences d'échantillonnage élevées ce qui augmentera la précision de notre modèle. Comme plusieurs compagnies travaillent en collaboration avec MathWorks, plusieurs options s'offrent afin de travailler en temps réel. Que ce soit avec Real-Time Workshop, ou d'autres bibliothèques, une panoplie d'options est incluse dans la suite de Matlab, offrant ainsi une flexibilité dans notre choix de matériel.

Maintenant, en analysant notre modèle, nous en arrivons à la conclusion que nous allons devoir séparer notre modèle en deux. Les diverses raisons seront présentées dans la section 4.3. Pour des raisons de complexité d'algorithme et de fréquence d'horloge, nous aurons deux cibles telles que nous avons vu plus haut, un FPGA et un DSP. La **Figure 12** montre l'interaction des logiciels utilisés à travers les étapes d'implémentations.

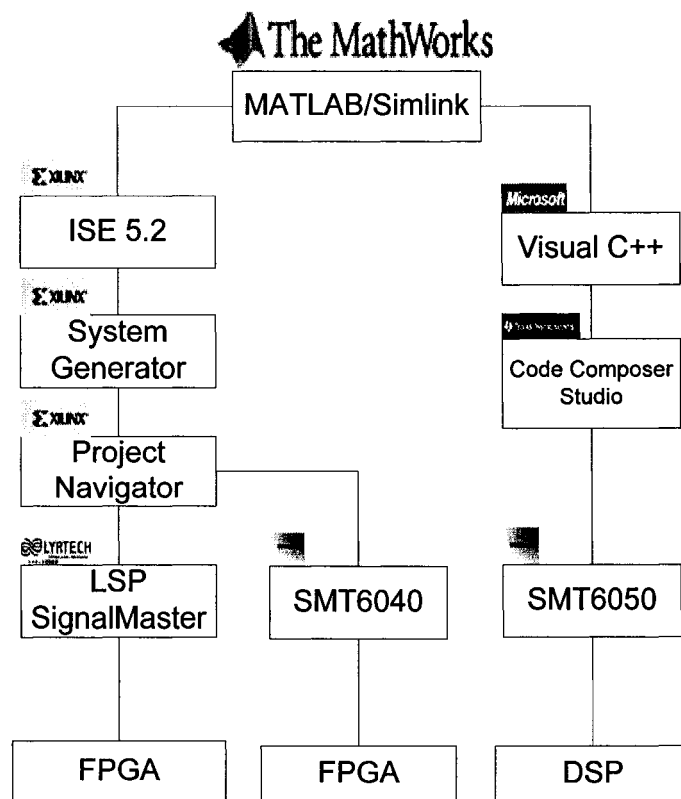


Figure 12 Interaction entre les différents logiciels

De toutes les compagnies qui font la production des FPGA, Xilinx Inc. a développé une série d'outils logiciels faciles à utiliser et adaptés au FPGA qu'ils produisent. La suite de Xilinx, appelé ISE, comporte plusieurs éléments aidant à la génération, à la validation et à la compréhension des codes binaires qui seront générés pour l'implémentation du FPGA.

Xilinx incorpore à l'intérieur des bibliothèques de Matlab/Simulink, une bibliothèque de blocs adaptés et préparés à être transformés en code binaire. Cette bibliothèque nous servira de base afin de préparer et de modifier le modèle Simulink de notre chaîne de

communication GPS. Les blocs inclus offrent les éléments de conceptions nécessaires afin d'implanter notre modèle.

La partie System Generator de la suite Xilinx génère les codes VHDL. À partir de ces codes, le logiciel Project Navigator, de la suite Xilinx ISE, permet de transformer les codes VHDL en code binaire. Avant de créer le code binaire, le logiciel nous permet aussi de visualiser les emplacements et les différents blocs utilisés directement sur une carte, représentant le FPGA. Ainsi, le concepteur peut facilement voir, non seulement l'espace requis par son système, mais aussi les différents branchements à travers le processeur et le programmeur peut ainsi optimiser, manuellement, l'emplacement du code à l'intérieur du matériel.

### **2.3.3 Concepts fondamentaux et raffinements**

#### **2.3.3.1 Démarche d'implémentation pas-à-pas**

Tel que vu précédemment, par la chaîne d'interaction logicielle, nous avons certaines étapes à suivre avant de réussir à implémenter le système dans un processeur. Tout d'abord, nous devons bien paramétrer notre modèle. Certains de ces paramètres auront des répercussions importantes, non seulement dans la performance du système au niveau de la simulation, mais aussi au niveau des performances du système en temps réel puisque ces derniers interviendront dans la génération des codes.

La **Figure 13** nous montre l'interaction, autant logicielle que matérielle, de l'implémentation de notre système.

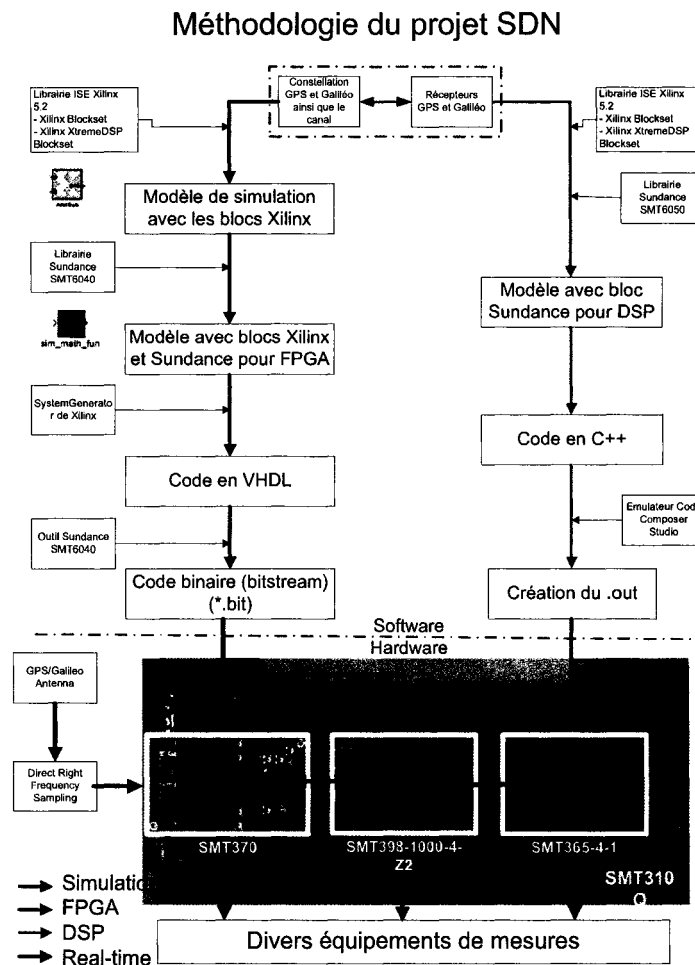


Figure 13 Méthodologie de travail pour l'implémentation d'un récepteur temps réel

La démarche exposée dans la **Figure 13** peut non seulement s'appliquer à notre projet, mais aussi à tout autre projet de réalisation de modèle en temps réel. Il est à noter que cette démarche est utilisée dans des entreprises oeuvrant dans le domaine. Ceci est une marque que notre méthodologie a fait ses preuves et qu'elle est adaptée à nos besoins.

La première partie, en vert sur la figure, représente la partie modélisation du système avec Matlab/Simulink. Elle est la base du développement, ce qui fait qu'on la retrouve la première sur le graphique. Une erreur à ce niveau, tel qu'on le verra plus tard, se

répercute dans tout le reste du processus. Cependant les simulations et l'analyse des performances du modèle en simulation, les erreurs peuvent être diagnostiquées à ce niveau et être corrigées avant l'implémentation. Ainsi, nous nous attarderons plus longuement sur les explications et les paramètres concernant cette partie. Le chapitre 3 explorera et expliquera le système du projet, c'est-à-dire la chaîne de communication GPS considérée. Pour cette section, seule la démarche sera expliquée. Les chapitres suivants montreront les différentes étapes une par une utilisées dans ce projet afin de compléter l'implémentation.

Une fois le système modélisé avec Matlab/Simulink et fonctionnant de manière réaliste, il est important de bien connaître les différentes bibliothèques de Xilinx et de Sundance. On peut remarquer que la **Figure 12** et la **Figure 13** se doivent d'être superposées puisqu'elles sont complémentaires, la démarche utilisant les logiciels afin de réaliser les différentes étapes. Afin de compléter le passage en temps réel du système, certains blocs sont particuliers et importants. Plusieurs paramètres doivent être respectés sur chacun de ces blocs, que ce soit le nombre de bits après la virgule, le nombre de bits total ou la fréquence d'échantillonnage. Évidemment, le début de cette étape se situe au niveau de l'étude théorique du système à implémenter. On en extrait les grandes caractéristiques et les limitations, ce qui aidera à modéliser le système. Dans le chapitre 3 du présent ouvrage, l'étude théorique d'une communication GPS est présentée, et jumelée avec les caractéristiques énumérées précédemment complète l'étude théorique.

Après la simulation validée avec des résultats réels et probants, pour suivre la méthodologie, la séparation de notre modèle est exigée afin de déterminer la marche à suivre. Évidemment, tel qu'expliqué plutôt, les processeurs ciblés étant différents, ils impliquent une marche à suivre non seulement qui diffère mais qui fait intervenir différents logiciels. Les caractéristiques de notre modèle nous imposent souvent le processeur ciblé et réduit les choix du programmeur. Tel qu'il sera expliqué plus loin, certains processus requièrent une attention particulière lors de cette étape. Ensuite, selon

la cible choisie, divers logiciels ou partie de logiciels interviennent. Ainsi, pour la génération d'un code binaire, nous devons faire appel à différentes bibliothèques de Simulink que Xilinx a développé. Ces bibliothèques possèdent des outils adaptés afin de faciliter le passage en temps réel de notre modèle. À l'instar de Xilinx, Texas Instruments, un des leaders mondiaux dans le domaine des processeurs numériques, utilise des logiciels spécifiques pour convertir notre modèle en un code C++. Cependant, du côté de Texas Instruments, aucun changement n'est requis afin de passer le modèle en C++. Par contre, Sundance a développé des outils pour faciliter la communication une fois que le modèle est en assembleur. Ces blocs doivent donc apparaître au niveau du modèle, ce qui nous force à modifier notre modèle exactement comme dans le cas où le processeur ciblé est un FPGA.

Une fois les changements faits, il nous est donc possible de déterminer sur quelle branche de la marche à suivre nous nous trouvons. Du côté DSP, par la génération automatique du code C++ relié à notre modèle, le logiciel Code Composer Studio prend ce code et le traduit en langage machine, soit l'assembleur, pour que le DSP puisse l'exécuter. De plus, ce logiciel prend vraiment le contrôle des communications avec le DSP car c'est à partir de son écran de contrôle que nous pouvons commander l'exécution de notre code. Ceci est une différence majeure entre le DSP et le FPGA, puisque pour ce dernier, une fois le code implémenté dans le processeur, il s'exécute automatiquement. Cette particularité fera de notre DSP le système de démarrage de notre modèle en temps réel.

Une des difficultés de cette démarche reste la communication entre la partie logicielle et la partie matérielle. Il faut s'assurer que les protocoles de communication soit parfaits puisqu'une erreur dans l'implémentation peut rendre notre modèle tout à fait inopérant Il faut donc respecter la composition des processeurs afin d'établir un protocole qui nous permettra de les configurer. Puisque nous avons deux différents processeurs (DSP et FPGA), les protocoles diffèrent pour chacun d'entre eux. Tel que mentionné plus haut,

Texas Instruments a inclus ce protocole avec le logiciel Code Composer Studio. Par contre, pour le FPGA, ce n'est pas le fabricant qui a conçu le système de communication mais bien Sundance, le développeur de la plate-forme. Ceci implique une interaction de plus entre notre modèle et le processeur ciblé, augmentant ainsi la complexité de la tâche. Il est important de noter que plus nous avons d'étapes, plus il est probable que nous introduisons des erreurs.

Une fois notre modèle dans les processeurs, nous pouvons donc tester le système en co-simulation. Cette étape nous permet de valider l'implémentation à l'intérieur des processeurs. La **Figure 14** illustre le principe de co-simulation et les relations qu'il y a entre le matériel intervenant.

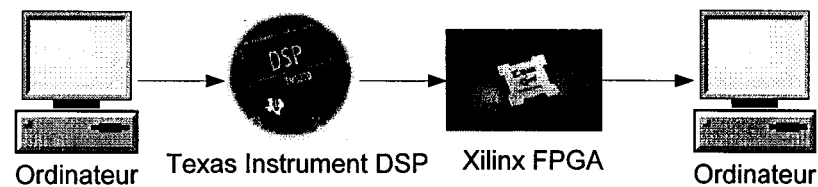


Figure 14 Principe de co-simulation

Par la co-simulation, les données sont générées par l'ordinateur, via un logiciel tel Matlab/Simulink, et le traitement est fait à l'intérieur des processeurs. Les données traitées sont ensuite redirigées vers le logiciel qui nous permet d'analyser les performances du système. Ainsi, l'analyse ainsi faite sert à dépister plus rapidement les erreurs pouvant avoir lieu, dans un environnement rapide et avec un minimum d'interaction. Il est plus facile de cette façon d'adapter le système et le modèle aux caractéristiques des processeurs.

Le système implémenté dans les deux processeurs est alors prêt à être testé en temps réel. Mais pour ce faire, il faut être capable de récupérer les signaux générés par les processeurs. Il nous faut donc des convertisseurs analogique-numérique rapides. La

troisième carte de notre plate-forme, nommée SMT370 sur la **Figure 13**, sert à acquérir ces signaux afin de pouvoir les analyser en temps réel. L'analyse du fonctionnement en temps réel, toujours à partir de signaux générés à partir du logiciel Matlab, permet d'établir les limites du système. Ainsi, lors de la prochaine étape, il sera plus facile de déterminer l'environnement de notre système.

La dernière étape de notre méthodologie permute les signaux générés avec les signaux réels de la constellation satellitaire GPS. L'antenne GPS sera branchée sur une tête RF, qui servira à moduler le signal dans la bande passante active des convertisseurs analogique-numériques. Le signal IF sera donc traité par nos processeurs et les données comparées aux éphémérides disponibles sur le web.

## 2.4 Conclusion

Le principe de radionavigation étant assez récent dans le monde moderne, son développement est encore à l'état embryonnaire. À l'image du développement radio (qui engendra l'éclosion des cellulaires), le principe de « Software Defined Navigator » est essentiel au développement des récepteurs de positionnement. Mais pour y arriver, une méthodologie doit être développée pour identifier les éléments nécessaires à la conception d'un récepteur de position purement logiciel. L'avenir de cette technologie est important si l'on se réfère à ce qui s'est produit au niveau des ondes radio. Au cours de ce mémoire, nous avons développé et testé cette méthodologie et ce principe. Les chapitres suivants font des conclusions et des projets futurs de ce projet du « Software Defined Navigator ».