

MA-004-175-1

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement supérieur et de la recherche scientifique

Université SAAD DAHLEB De Blida

Faculté des Sciences

Département d'Informatique



Mémoire de fin d'étude Pour l'obtention du diplôme de
MASTER En Informatique Option : Génie Logiciel

Thème

**Conception et Développement d'une
application de Positionnement indoor
pour SmartPhones sous Android**

Présenté par :

ELHADI A.Rahim

KSENTINI Dalila

Encadreur :

M^r LASLA Nouredine

Promoteur :

M^{me} ABED Hafida

Année universitaire 2012 / 2013

MA-004-175-1



Remerciements

Nos remerciements vont d'abord à notre encadreur Monsieur Noureddine LASLA pour sa disponibilité, ses précieux conseils, et ses encouragements au cours de notre travail de recherche. Ainsi qu'à notre promotrice Madame Hafida ABED pour ses conseils pertinents, sa patience et ses suggestions qui nous ont été d'une grande aide.

Nous tenons aussi à exprimer notre gratitude à l'ensemble des personnes qui ont contribué de près ou de loin à ce travail.

Dédicaces

A mes très chers parents qui ont
toujours cru en moi et n'ont jamais
cessé de me soutenir. J'espère qu'ils
trouveront dans ce travail toute ma
reconnaissance et tout mon amour.

A mes sœurs Djazia, Nabila ainsi que mon frère Adlen,
pour leur affection, compréhension et soutien.

A mes chères cousines et amis qui m'ont
épaulé tout au long de mon cursus...

Dalila

Dédicaces

A mes très chers parents qui m'ont toujours soutenu et encouragé

A mes frères Raouf, Ilyes et ma petite sœur Yousra

pour leurs compréhensions et soutiens

Au groupe Babylone Amine et Ramzy, ainsi que notre manager

Amine Aswat qui m'a beaucoup marqué cette année

A mes amis Sadek, Youcef, Redouane, et Rafik

qui m'ont épaulé tout au long de ce travail.

Rahim

Résumé

Après l'ouverture de la géo-localisation au grand public en 1993, plusieurs services sont alors apparus permettant de localiser des personnes, des objets mobiles et de gérer des flottes de véhicules. Pour ce faire, différentes techniques ont été développées dont la plus utilisée est les systèmes satellitaires tel que le GPS (Global Positionning System). D'autres techniques reposent sur les réseaux sans fil ou combine entre les deux. Ces techniques se basent sur l'installation d'une infrastructure qui peut être coûteuse de plus, parmi elles certaines deviennent non fonctionnelles dans les endroits fermés. Une autre technique appelée Navigation inertielle (INS), utilise seulement des capteurs attachés au mobile et ne nécessite aucune information externe. Cette technique n'est pas coûteuse et s'avère plus efficace pour le positionnement dans les endroits fermés.

Le but de ce projet est de développer une application de navigation mobile installée sur un Smartphone, basée sur la technique de la navigation inertielle. Pour cela, la navigation inertielle est étudiée ainsi que des séries de tests ont été effectuées afin d'examiner le niveau de précision des capteurs inertiels lors du positionnement. L'application finale est conçue à la base des résultats obtenus.

Mots-clés : INS Navigation inertielle, IMU centrale inertielle, capteurs inertiels, accéléromètre, gyroscope, boussole, navigation indoor, GPS.

Abstract

After the opening of the geolocation to the general public in 1993, several services to locate people, moving objects and manage vehicle fleets began appearing. To do this, various techniques have been developed ; the most used is the satellite systems such as GPS (Global Positioning System). Other techniques rely on wireless networks or these two combined. These techniques are based on the installation of an infrastructure that could be more expensive, some among them become non-functional in confined areas. Another technique called inertial navigation (INS), uses only sensors attached to mobile and requires no external information. This technique is not expensive and it is more effective for positioning in confined areas.

The purpose of this project is to develop a mobile navigation application installed on a Smartphone , based on the technique of inertial navigation. For this, the inertial navigation is studied as well as series of tests were conducted to examine the accuracy of inertial sensors during positioning. The final application is designed depending on the results obtained

Keywords : INS (Inertial Navigation Indoor), IMU (Inertial Measurement Unit), Inertial sensors, accelerometer, gyroscope, magnetometer, indoor navigation, GPS.

Table des matières

Résumé	v
Abstract	vi
Table des matières	vii
Table des figures	x
Liste des tableaux	xiii
Introduction générale	1
1 Etat de l'art	3
1.1 Introduction	4
1.2 Les différents systèmes de navigations :	4
1.2.1 Satellites ou GNSS : (Global Navigation Satellite System)	4
1.2.2 Les systèmes utilisant les réseaux sans fil	8
1.2.3 Les systèmes hybrides	12
1.3 Avantages et inconvénients des systèmes de positionnement	13
1.4 Le système de la navigation inertielle INS	14
1.4.1 Un référentiel inertielle	15
1.4.2 Centrale inertielle IMU (Inertial Measurement Unit)	16
1.4.3 Principe de base de la navigation inertielle	16
1.4.4 Les capteurs inertiels MEMS (Micro Electro Mechanical Systems)	17
1.4.5 Les sources d'erreurs	19
1.4.6 La fusion des capteurs	21

TABLE DES MATIÈRES

1.4.7	Les avantages et inconvénients d'INS	21
1.5	Conclusion	22
2	Tests des capteurs MEMS	23
2.1	Introduction	24
2.2	Les capteurs intégrant les Smartphones	24
2.3	Scénario de positionnement sur une dimension avec une vitesse constante	27
2.3.1	Description du scénario	27
2.3.2	Remarque et proposition	28
2.3.3	Calibrage de l'accéléromètre	31
2.3.4	Algorithme du scénario	34
2.3.5	observations des résultats du scénario	37
2.4	Scénario de positionnement sur une dimension avec une vitesse inconnue	38
2.4.1	Description du scénario	38
2.4.2	Remarque et proposition	39
2.4.3	Filtrage des données brutes	43
2.4.4	Algorithme du scénario	47
2.4.5	observations des résultats du scénario	49
2.5	Conclusion	52
3	Présentation et conception du projet	53
3.1	Introduction	54
3.2	Objectifs	54
3.3	conception du système	54
3.3.1	Les exigences de la conception	54
3.3.2	Architecture générale du système	55
3.3.3	Conception détaillée du système	57
3.4	La création de la map	66
3.5	Conclusion	68
4	Implémentation	69
4.1	Introduction	70
4.2	Systèmes d'exploitation mobiles	70
4.2.1	Ios (iPhone OS)	70
4.2.2	Windows mobile	70

TABLE DES MATIÈRES

4.2.3	Android	71
4.2.4	Symbian OS	71
4.2.5	BlackBerry OS	71
4.3	Raisons du choix d'Android	72
4.4	Présentation de la plateforme Android	73
4.4.1	L'historique d'Android	73
4.4.2	L'architecture d'Android	75
4.4.3	Le cycle de vie d'une activité	76
4.5	Outils de réalisation du projet (Environnement technique)	78
4.5.1	Environnement de développement	78
4.5.2	Eclipse	78
4.5.3	Equipement	81
4.6	Résultats de la réalisation des scénarios	83
4.7	Conclusion	86
	Conclusion et Perspectives	88
	Annexes	89
	A Les filtres	90
A.1	Filtrage numérique des données brutes	90
A.1.1	Définition et classification des filtres numériques	90
A.2	Les filtres passifs	92
A.2.1	Les filtres passe bas	92
A.2.2	Les filtres passe haut	92
A.2.3	Passe-Bande, ou sélecteur de bande ou de fréquence	92
	Bibliographie	94

Table des figures

1.1	Calcul de position d'un GPS	5
1.2	les étapes de la triangulation.	7
1.3	Signe émis par le satellite bloqué par un immeuble	8
1.4	Réseau Wi-Fi en mode infrastructure et en mode Ad-Hoc	10
1.5	Fonctionnement en quatre étapes du système de géolocalisation centré infrastructure[6].	11
1.6	Infrastructure de répéteurs	13
1.7	Un repère inertiel	15
1.8	Scéma d'un système masse ressort des 3 axes (X, Y, Z) de l'accéléromètre	18
1.9	Le gyroscope	19
2.1	Déplacement d'une personne tenant un Smartphone.	27
2.2	Etat d'un Smartphone en déplacement	28
2.3	les axes (X, Y, Z) de l'accéléromètre sur un Smartphone.	29
2.4	Graphe de l'accélération d'un mobile en état de marche et arrêt	29
2.5	Graphe de l'accélération sur l'axe Y et la somme des valeurs des axes en état de marche	30
2.6	Graphe de l'accélération sur l'axe Y et la somme des valeurs des axes en état d'arrêt	30
2.7	Graphe de la somme de la variation des accélérations du mobile sur table	32
2.8	Graphe de la somme de la variation de l'accélération du mobile sur la main	32
2.9	Graphe de la somme de la variation de l'accélération du mobile en vitesse moyenne	33

TABLE DES FIGURES

2.10	Graphe de la somme de la variation de l'accélération du mobile en grande vitesse	34
2.11	Plateforme du 2 ^{ième} scénario et l'état du déplacement du mobile	39
2.12	Graphe de l'accélération brute de l'axe Y sur la plateforme	41
2.13	Graphe de la vitesse de l'axe Y sur la plateforme	41
2.14	Graphe de la distance calculée sur la plateforme	43
2.15	Graphe du test de marche avec ALPHA 0.25	45
2.16	Graphe du test de marche avec ALPHA 0.2	46
2.17	Graphe du test de marche avec ALPHA 0.18	46
2.18	Graphe de l'accélération finale de l'axe Y avec l'utilisation des filtres sur la plateforme	49
2.19	Graphe de la vitesse finale de l'axe Y avec l'utilisation des filtres sur la plateforme	49
2.20	Graphe de la distance finale de l'axe Y avec l'utilisation des filtres sur la plateforme	50
2.21	Graphe de l'accélération de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile	51
2.22	Graphe de la vitesse de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile	51
2.23	Graphe de la distance de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile	52
3.1	Architecture générale du système	56
3.2	Le cycle de l'application du 1 ^{er} scénario	58
3.3	Diagramme du cas d'utilisation du 1 ^{er} scénario	59
3.4	Diagramme de classe du 1 ^{er} scénario	60
3.5	Diagramme d'activité du 1 ^{er} scénario	61
3.6	Cycle de l'application du 2 ^{ème} scénario	63
3.7	Le diagramme de cas d'utilisation du 2 ^{ème} scénario	64
3.8	Diagramme de classe du deuxième scénario	65
3.9	Diagramme d'activité du deuxième scénario	66
3.10	Photo réelle de la bibliothèque du C.E.R.I.S.T	67
3.11	La map de la bibliothèque du C.E.R.I.S.T	68

TABLE DES FIGURES

4.1	Parts du marché des Os mobile dans le monde en 2012 et 2013 [18] . . .	72
4.2	Evolution des versions d'Android	75
4.3	L'architecture de la plateforme Android [22]	76
4.4	Le cycle de vie d'une activité [24]	77
4.5	Le site web des outils nécessaires pour les développeurs Android [25] . .	79
4.6	Interface du SDK Manager	80
4.7	Interface d'Eclipse incluant l'ADT	81
4.8	HTC sensation Smartphone	82
4.9	La page d'accueil de l'application	83
4.10	Paramétrer vitesse et chemin vers la map	84
4.11	Galerie où sont stocké les cartes ainsi que l'interface de saisie des dimensions	85
4.12	Les détails de calcul de position du premier scénario	85
4.13	Position du mobile sur la map	86

Liste des tableaux

1.1	Avantages et inconvénients des systèmes de navigation courants [9].	14
2.1	Les capteurs pris en charge par la plateforme Android [20].	26
2.2	Résultats du premier scénario	37
2.3	Résumé des filtres passifs et filtres numériques [17]	44
3.1	Les exigences de la conception comprenant les trois sous-systèmes et leurs descriptions	55
4.1	Caractéristiques du HTC sensation [23]	82
A.1	Tableau comparatif des caractéristiques des filtres RIF et RII	92

Introduction générale

La géo-localisation ou le positionnement est le processus qui permet de positionner une personne ou un objet par rapport à un ou plusieurs points de références (de positions connues). La géo-localisation a été une préoccupation pour l'humain depuis les temps les plus primitifs, utilisant différentes techniques, commençant par les signaux de fumées passant par les compas magnétiques et arrivant de nos jours à l'utilisation des satellites.

Aujourd'hui, le GPS (Global Positioning system) est un système de localisation mondial qui a connu un grand succès dans différents domaines tels que la navigation maritime, terrestre, suivi de flotte, etc. Ce système est basé sur des récepteurs GPS qui reçoivent librement des signaux envoyés par des satellites permettant de calculer avec précision (dans les endroits ouverts) les coordonnées géographiques du récepteur. Récemment, les récepteurs GPS ont été intégrés dans la plupart des Smartphones permettant à ces derniers d'être des dispositifs de navigation facilement portables et pratiques. Cependant, le GPS est connu par sa consommation élevée d'énergie et sa faible précision dans les endroits fermés (indoor) ce qui limite son utilisation pour les dispositifs qui ont une ressource énergétique limitée, tel que les Smartphones, à l'intérieur des bâtiments. Les méthodes qui se basent sur les communications sans fil, ont été élaborées et évaluées afin de répondre aux besoins de localisation dans les endroits fermés. Ces systèmes reposent sur des réseaux sans fil installés à l'intérieur des bâtiments, et sur les propriétés liées à la propagation des signaux sans fil pour trouver la position du mobile. L'inconvénient majeur de ces systèmes est leur coût élevé ainsi que le manque de précision lié à la nature sensible et instable des communications sans fil.

La navigation inertielle est une approche totalement autonome qui se base sur une centrale d'inertie (IMU, Inertial Measurement Unit) intégrant des capteurs de vitesse et d'accélération permettant de suivre les mouvements d'un mobile. Cette approche ne nécessite aucune information externe raison pour laquelle était utilisée pour la localisa-

tion des missiles lors de la première guerre mondiale et reste toujours préférable pour la sécurité des vols et des véhicules militaires. Cette indépendance réduit considérablement la consommation d'énergie lors du processus de calcul de la position et assure la localisation dans les endroits ouverts et fermés.

L'objectif principal de ce projet est de concevoir une application de positionnement indoor, portable sur les Smartphones, et qui se base sur la navigation inertielle. La première étape de ce projet consiste à étudier la méthode de navigation inertielle ainsi que les différents capteurs qui peuvent être utilisés pour le positionnement. Dans la deuxième étape, des tests réels seront effectués afin d'examiner le niveau de précision des capteurs ainsi que leurs rôles dans les mesures de mouvement et le calcul de position. La dernière étape du projet consiste à concevoir et développer l'application de positionnement sous Android. Cette étude a été réalisée au sein du Centre de Recherche sur l'Information Technique et Scientifique (C.E.R.I.S.T).

Organisation du mémoire

Le document est organisé en quatre chapitres, le premier présente l'état de l'art. Il décrit les fondamentaux de la navigation nécessaires à la bonne compréhension du document, c.-à-d. les différents principes de la navigation par satellites, par les systèmes utilisant les réseaux sans fil, les systèmes hybrides et les systèmes de navigation inertielle. De plus, il explicite le choix sur l'étude d'une de ces solutions pour la navigation indoor. Le second chapitre comprend des tests du capteur accéléromètre sur deux scénarios de calcul de positionnement. Le troisième chapitre s'intéresse aux objectifs du projet ainsi qu'à son architecture générale et détaillée qui inclut les cycles des applications de chaque scénario et les diagrammes nécessaires pour la conception. Le dernier chapitre détaille l'implémentation de l'application en définissant les différents systèmes d'exploitation, sur quel choix s'est porté notre projet, les outils utilisés pour le réaliser, ainsi que les résultats obtenus lors de l'implémentation.

Chapitre 1

Etat de l'art

1.1 Introduction

Au fil du temps, l'homme a toujours eu besoin de savoir où il se trouve, c. à d. connaître sa position par rapport à des références connues. Aujourd'hui, avec la révolution technologique, de nombreuses techniques innovantes ont fait leurs apparitions. La navigation est la science qui regroupe l'ensemble de ces techniques et permet le calcul de la position d'un objet mobile dans l'espace. Le but de ce chapitre est de se familiariser avec les différents systèmes de navigations, en particulier le système de navigation choisi pour notre application.

1.2 Les différents systèmes de navigations :

Aujourd'hui, de nombreux systèmes de navigations et de positionnements sont disponibles et reposent sur des technologies différentes telles que les satellites, réseaux sans fil, et les systèmes hybrides qui combinent les systèmes satellites et les réseaux sans fil. Dans ce qui suit, nous présentons le principe de fonctionnement des systèmes de positionnement les plus reconnus.

1.2.1 Satellites ou GNSS : (Global Navigation Satellite System)

Les systèmes de navigation par satellite offrent un positionnement géo-spatiales avec une couverture mondiale. Dans ce qui suit, nous présentons quelques systèmes de navigation qui sont basés sur les satellites.

1.2.1.1 GPS : (Global Positioning System)

Le système de positionnement (GPS) est un système de navigation par satellite, développé par le département américain de la défense à des fins militaires. Le système a été déclaré pleinement opérationnel en 1994 . Aujourd'hui, le GPS est également utilisé à des fins civiles, comme l'arpentage, la carte de conception et bien évidemment la navigation. Le système GPS se compose de 3 segments, le segment spatial (satellites),

le segment de contrôle et le segment utilisateur (récepteur) [5]. Le secteur spatial se compose de 24 satellites en orbites, autour de 6 orbites qui ont une inclinaison de 55° par rapport à l'équateur. Les orbites sont disposés de telle manière qu'au moins 6 satellites sont toujours visibles de partout sur la surface de la terre. Les satellites GPS envoient des messages de navigations continus à un débit de 50 bits par seconde, dont l'information principale est le moment exacte d'envoi.

Le segment de contrôle est composé d'un certain nombre de stations et d'antennes, elles sont utilisées pour contrôler et surveiller la santé des satellites, et de faire les corrections nécessaires en cas de besoin, par exemple régler les horloges du satellite. Le segment utilisateur : se compose des utilisateurs militaires du service de positionnement GPS précis et les utilisateurs civils du service du positionnement standard. Le récepteur GPS est composé principalement d'une antenne, d'une horloge interne très stable, le logiciel de calcul de localisation de l'utilisateur et de la vitesse, et en général un dispositif d'affichage pour fournir les informations à l'utilisateur [5]. La Figure 1.1 montre qu'il faut au moins 4 satellites pour obtenir une position en 3 dimensions.

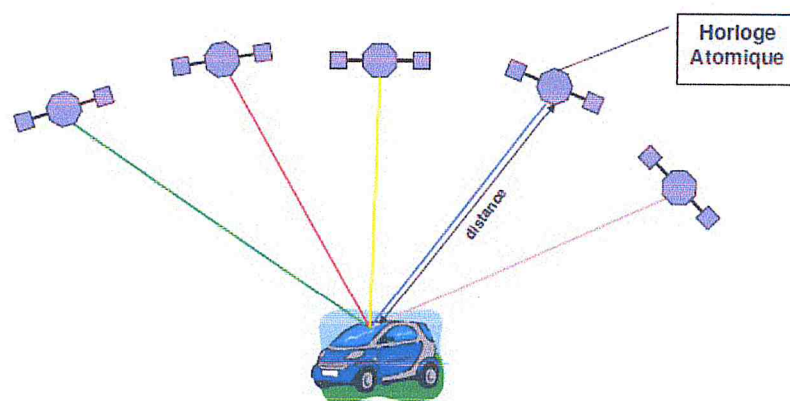


FIGURE 1.1: Calcul de position d'un GPS

1. *Principe de calcul de position :*

Les récepteurs GPS utilisent la trilatération géométrique qui combine les informations provenant de différents satellites pour prendre la position correcte. Comme mentionné ci-dessus, les messages GPS contiennent les informations sur le mo-

ment ou un message a été envoyé (informations orbitales précises), la disposition du système de l'information approximative sur les orbites et les autres satellites. Le récepteur utilise les messages contenant les éphémérides du satellite (les messages qui permettent de calculer la position du satellite) pour calculer le temps de transfert de chaque message et calcule la distance au satellite. À l'aide de la trilatération, les distances avec les satellites ainsi que l'emplacement des satellites sont utilisés pour calculer la position du récepteur [5].

L'avantage des systèmes de satellites est que les récepteurs peuvent déterminer l'attitude, la longitude et l'altitude à un haut degré de précision. Cependant, la ligne visée (ligne de mire) est nécessaire pour le bon fonctionnement de ces systèmes. Cela conduit à une incapacité d'utilisation de ces systèmes pour un environnement intérieur.

2. *La triangulation :*

La triangulation permet grâce à trois satellites fixes qui connaissent leur position exacte de déterminer la position d'un point précis sur Terre. Chaque satellite du réseau GPS possède une horloge atomique d'une grande précision [9].

Le principe de la triangulation est de mesurer la distance entre l'utilisateur et les satellites de positions connues. On définit ainsi des sphères centrées sur des satellites et dont l'intersection donne la position. Le récepteur GPS est capable d'identifier le satellite qu'il utilise à l'aide du signal pseudo aléatoire émis par chaque satellite. Il charge, à l'aide de ce signal, les informations sur l'orbite et la position du satellite [15].

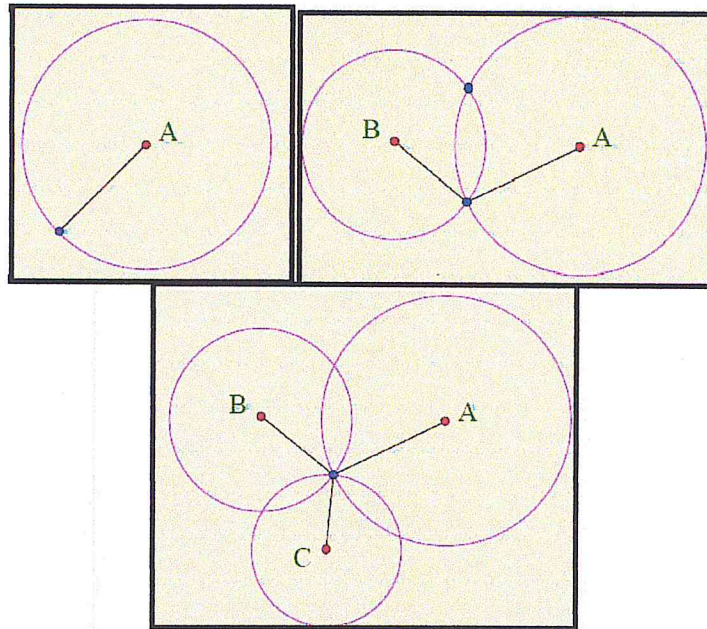


FIGURE 1.2: les étapes de la triangulation.

1.2.1.2 GLONASS (Global Navigation Satellite System)

Le GLONASS est un système de navigation Russe, développé depuis 1992, devenu opérationnel depuis 1997, et réservé à l'usage militaire. Il est composé de 24 satellites autour de 3 orbites et à l'altitude de 19130 Km. Il existe 5 stations au sol. Chaque satellite utilise sa propre fréquence ce qui augmente la résistance au brouillage. Seuls 7 satellites sont actifs actuellement [16].

1.2.1.3 Le système GALILEO / EGNOS (European Geostationary Navigation Overlay Service)

C'est un système de GNSS européen, son programme a été divisé en deux phases. La première consistait de déployer l'EGNOS (European Geostationary Navigation Overlay Service) qui devient opérationnel et son objectif était de préparer l'arrivée du GALILEO avec le développement des services compatibles. La deuxième phase était le déploiement du système GALILEO dont le lancement du premier satellite était en décembre 2005. Il a été développé pour améliorer la précision du GPS et repose sur la constellation de 30

satellites et des stations au sol, contrairement aux systèmes GPS et GLONASS c'est un système civil et non militaire [16].

Les signaux émis par les satellites ne sont pas reçus partout surtout dans les environnements fermés, comme dans les centres villes. Voir Figure 1.3.

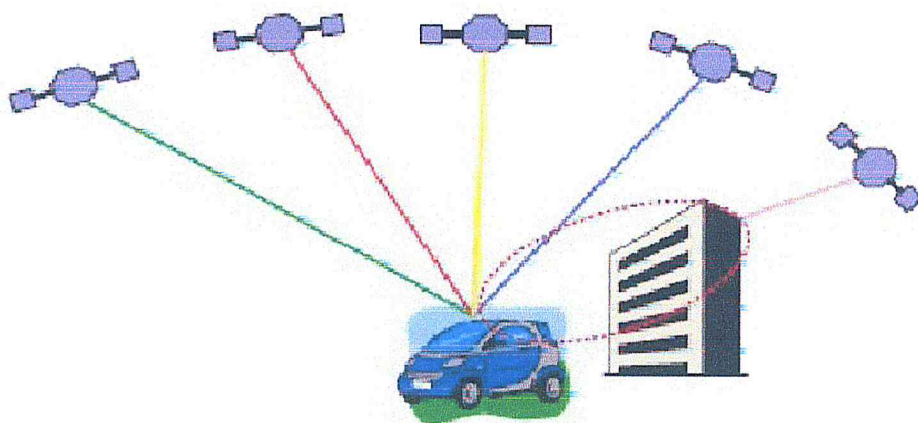


FIGURE 1.3: Signe émis par le satellite bloqué par un immeuble

1.2.2 Les systèmes utilisant les réseaux sans fil

1.2.2.1 Les systèmes basés sur les réseaux de communications cellulaires

Un réseau de communication cellulaire est un système qui permet aux téléphones mobiles de communiquer les uns avec les autres. Ce système utilise de grandes tours de téléphonie cellulaire pour une connexion sans fil des appareils mobiles. La portée des réseaux de communication cellulaire dépend de la densité des grands immeubles, des arbres et autres obstacles possibles. La portée maximale d'une tour de téléphonie cellulaire est à 35 kilomètres dans une zone rurale ouverte [3]. Cette méthode est une technique de base utilisant Cell-ID, également appelée cellule d'origine, elle fournit des services de localisation pour les utilisateurs de téléphone cellulaire [8]. Cette méthode est

basée sur la capacité d'estimation du réseau de la position d'un téléphone cellulaire, cela en identifiant la station cellulaire que le dispositif utilise à un moment précis. L'avantage de cette technique est sa distribution ubiquitaire, la mise en œuvre est facile et le fait que tous les téléphones cellulaires mobiles la supportent. La précision de cette technique est très faible en raison du fait que les tours de téléphonie cellulaire peuvent s'étendre jusqu'à 35 kilomètres ou plus [3].

1.2.2.2 Les systèmes basés sur le WI-FI : (Wireless Fidelity)

Le WI-FI est le nom commun de la norme IEEE 802.11, qui désigne une technologie de communication sans fil permettant l'établissement de réseaux locaux, la communication à travers un réseau sans fil est comme une communication radio bidirectionnelle :

- Un adaptateur sans fil convertit les données en signaux radios et les transmet par l'intermédiaire d'une antenne
- Un routeur sans fil reçoit le signal et le décode puis, il envoie les informations à internet en utilisant une connexion Ethernet via un câble.

Le processus fonctionne de la même manière en sens inverse, avec le routeur de réception de données à partir d'internet en les transformant en signaux radios et en les envoyant à l'adaptateur sans fil de l'appareil. Chaque routeur sans fil émet un signal qui est reçu par les dispositifs à proximité. Ces appareils ont la capacité de mesurer la force du signal. Cette force est convertie en un nombre, connu comme indicateur d'intensité du signal reçu (RSSI). Les appareils WI-FI tel que les Smartphones effectuent généralement cette conversion automatiquement, afin de fournir la puissance du signal pour les applications exécutantes sur ces derniers [10].

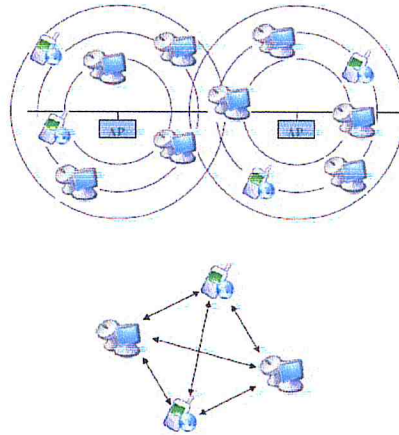


FIGURE 1.4: Réseau Wi-Fi en mode infrastructure et en mode Ad-Hoc

Le calcul de la position des mobiles est effectué par les éléments de l'infrastructure et non par le mobile lui-même (comme pour le cas du système GPS). Cette infrastructure est essentiellement composée de points d'accès (AP) écoutant le réseau afin de repérer les informations susceptibles de permettre le positionnement d'un mobile, à savoir les demandes de localisation envoyées par ce dernier. Ces demandes sont ensuite transmises à un serveur de calcul, chargé de traiter l'information de manière à déterminer la position du mobile. Les AP ne communiquant pas entre eux, ils transmettent les demandes sans se soucier d'un quelconque ordre; un serveur intermédiaire fait donc l'interface entre les AP et le serveur de calcul afin de présenter à ce dernier l'information de manière cohérente.

Sur le plan matériel, les mobiles peuvent être tout type d'appareil doté d'une interface Wi-Fi (ordinateur portable, téléphone cellulaire, PDA communicant, console de jeux portable...). Il en est de même pour les AP, à la différence que leur interface Wi-Fi doit supporter l'en-tête radiotap. Enfin, le serveur d'agrégation et le serveur de calcul peuvent être installés sur une machine plus ou moins puissante selon le nombre de mobiles à tracer. Ce système de positionnement est basé sur le principe de OWPS (Open Wireless Positioning System) qui est un système de géolocalisation en intérieur, utilisant comme support le réseau sans fil IEEE 802.11 (Wi-Fi). Il met en oeuvre plusieurs techniques de positionnement, toutes fondées sur l'analyse de la puissance des signaux Wi-Fi reçus. La Figure qui suit résume les quatre étapes de la résolution de la position d'un mobile [6].

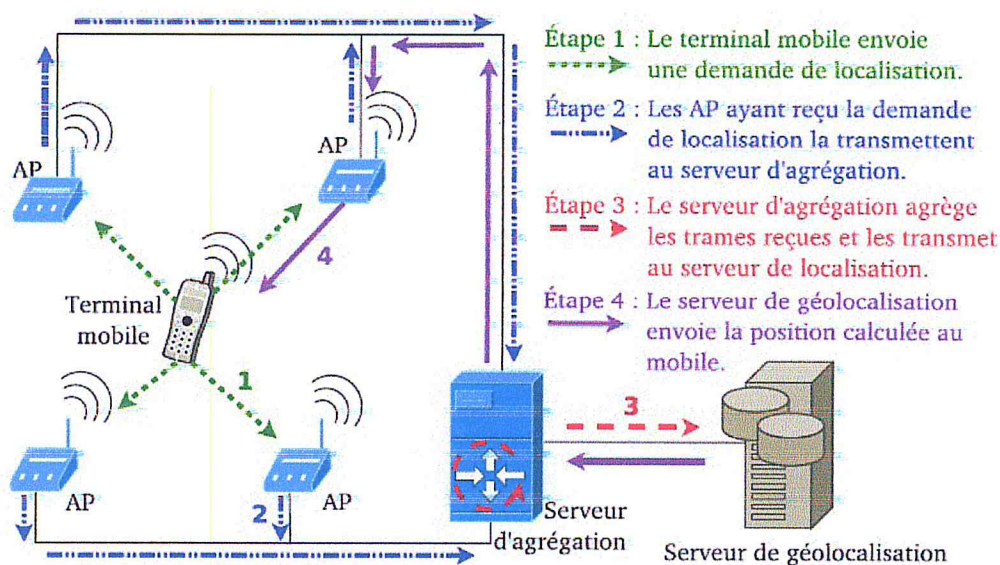


FIGURE 1.5: Fonctionnement en quatre étapes du système de géolocalisation centré infrastructure[6].

1.2.2.3 Les systèmes basés sur le Bluetooth

Le Bluetooth est une norme de technologie sans fil utilisée par deux (ou plus) de dispositifs de communication sur de courtes distances. Le grand avantage du Bluetooth est qu'il est sans fil, peu coûteux et automatique. Par rapport à la technologie infrarouge qui est un autre moyen de communication sans fil (couramment utilisé pour le système de télévision de commande à distance), le Bluetooth dispose d'un certain nombre de qualités avantageuses. Bien que les communications infrarouges sont assez fiables et ne coûtent pas très cher à construire dans un appareil, elles ont quelques inconvénients. Tout d'abord l'infrarouge est une ligne de vue technologique, par exemple, il faut pointer la télécommande vers le téléviseur pour changer les canaux. Le second inconvénient est que l'infrarouge est presque toujours une technologie de un à un. Le Bluetooth est destiné à contourner les problèmes qui viennent avec les systèmes infrarouges. Il utilise la technique Spread-Spectrum (étalement de spectre) à saut de fréquences. Ce qui permet la connexion entre plusieurs appareils simultanément, sans interférer les uns avec les autres. En outre, le Bluetooth ne nécessite pas une ligne de vue entre les appareils communicants ce qui rend la technologie utile pour contrôler plusieurs appareils dans

des pièces différentes [10]. La classe 1 Bluetooth prend en charge une vitesse de transfert maximale de 1 méga bit par seconde (Mbps). Tandis que la classe 2 Bluetooth est la norme commune sur les appareils portables qui peut gérer jusqu'à 3 Mbps, avec une distance de communication maximale de 10 mètres.

Le Bluetooth peut fournir les informations suivantes : le nom de l'appareil dans la zone, son adresse (adresse MAC), et le type de l'appareil (ordinateur, téléphone, etc). Ces informations ne permettent pas d'estimer la position exacte de l'utilisateur, mais permettent de détecter dans quelle partie l'utilisateur se trouve dans l'endroit où il est par rapport à un autre appareil.

1.2.2.4 Les systèmes basés sur l'infrarouge

L'infrarouge (IR) réseau sans fil était une technologie de pionnier dans le domaine de positionnement à l'intérieur [9]. Cependant, ce système a été confronté à plusieurs problèmes fondamentaux. Le principal défi est la portée limitée d'un réseau IR. Aussi, l'infrarouge n'a pas de méthode pour fournir des services de réseau de données. Une mise en œuvre rapide d'une technique IR est le système de badges actifs. Il s'agit d'un système de positionnement à distance dans lequel l'emplacement d'une personne est déterminé à partir du signal IR unique, émis toutes les dix secondes par un badge que la personne porte. Les signaux sont captés par des capteurs placés dans différents endroits à l'intérieur d'un bâtiment. La précision obtenue à partir de ce système est assez élevée dans les environnements intérieurs. Toutefois, le système souffre de plusieurs limitations telles que le coût d'installation du capteur en raison de la portée limitée de l'IR, le coût de l'entretien et la sensibilité du récepteur aux rayons du soleil, ce qui se produit souvent dans les chambres avec fenêtres [9].

1.2.3 Les systèmes hybrides

1.2.3.1 Utilisation des répéteurs

Son principe est de mettre en œuvre une infrastructure de répéteurs pour assurer la diffusion des signaux GPS à l'intérieur des bâtiments. Il s'agit de déployer une constellation d'émetteurs locaux. L'idée du répéteur est assez naturelle : puisque les signaux GNSS sont affaiblis par la présence de murs, de plafonds, ou tout autre obstacle, on

peut les récupérer à l'aide d'une antenne extérieure (placée par exemple sur le toit du bâtiment) et les réémettre à l'intérieur avec une amplification suffisante. Si l'on procède ainsi, le calcul donnera la position de l'antenne sur le toit. Les signaux de tous les satellites sont en effet transmis ensemble : le temps de propagation entre l'antenne d'émission du répéteur et l'antenne de réception du récepteur est le même pour tous les satellites [7]. Voir figure 1.4.

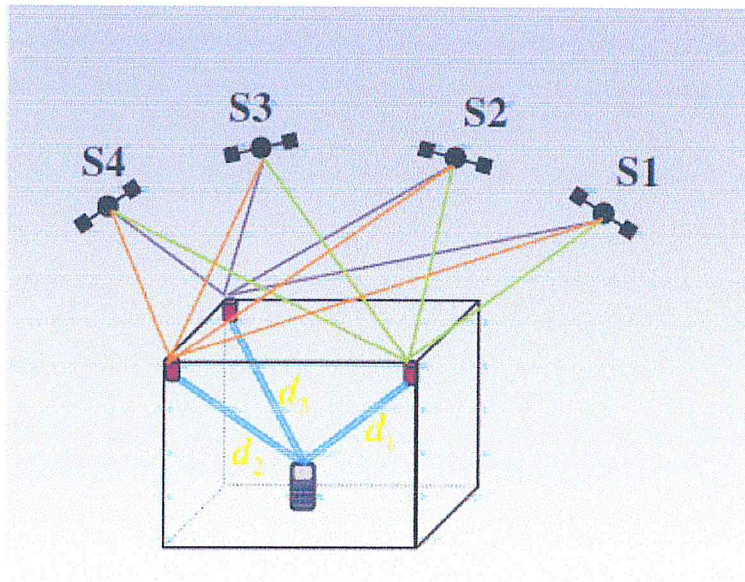


FIGURE 1.6: Infrastructure de répéteurs

1.3 Avantages et inconvénients des systèmes de positionnement

Le tableau suivant résume les avantages et les inconvénients des systèmes de navigations vu précédemment.

Système de navigation	Avantages	Inconvénients
GPS	Modéré à élevé en plein air. Haute précision. Disponibilité.	Exactitude basse jusqu'à dire manquante aux environnements à l'intérieur.
GLONASS	Augmente la résistance au brouillage.	Signaux faibles du satellite. Conçus à des fins militaires.
GALILEO	Améliore la précision du GPS. Meilleure continuité du service.	Les signaux émis par le satellite ne sont pas reçus.
Répéteurs	Homogénéité indoor et outdoor du processus de localisation.	Trop coûteux. Difficulté de déploiement.
Réseaux de communications cellulaires	Permettent d'obtenir une localisation précise en extérieur.	Précision médiocre et variable.
Wifi	Disponible dans la plupart des bâtiments. Peu coûteux.	La force du réseau peut varier en fonction de la propagation par trajets multiples.
Bluetooth	Peu coûteux.	Faible puissance.
Infrarouge	Précision assez élevée.	Coûteux.

TABLE 1.1: Avantages et inconvénients des systèmes de navigation courants [9].

Après avoir fait la comparaison des systèmes de navigation et positionnement, nous avons conclu que ces systèmes soit, ils ne sont pas fiables pour le positionnement indoor, soit ils sont trop coûteux. Pour cela, nous avons choisi d'étudier une nouvelle approche qui est la navigation inertielle.

1.4 Le système de la navigation inertielle INS

La navigation inertielle INS (Inertial Navigation System) est une technique qui utilise un ensemble de capteurs inertiels (capteurs de mouvement et capteurs de position) formant une centrale inertielle IMU (Inertial Measurement Unit). Cette dernière détermine le mouvement absolu ¹ d'un mobile. La navigation inertielle a l'avantage d'être

1. C'est un mouvement d'un corps considéré par rapport à des repères fixes

autonome c.-à-d. ne dépend d'aucun lien ni d'informations externes raison pour laquelle était utilisée pour la localisation des missiles lors de la première guerre mondiale et reste toujours préférable pour la sécurité des vols et des véhicules militaire. Cette indépendance réduit considérablement la consommation d'énergie lors du processus de calcul de la position et assure la localisation dans les endroits ouverts et fermés [5].

1.4.1 Un référentiel inertiel

Un référentiel inertiel ou galiléen est un système de coordonnées qui permet de repérer les événements et d'avoir des lois ou des équations plus simples. D'où le principe d'inertie $\sum F = 0$ (la somme des forces extérieures est nulle) sur un objet en mouvement sur une ligne droite et une vitesse constante, est vérifié [5]. Ce référentiel permet aussi à un observateur de quantifier (calculer) les positions et les déplacements.

Exemple d'un repère inertiel

Un repère inertiel $i = (X_i, Y_i, Z_i)$ est le centre de gravité de la terre. Les axes (X_i, Y_i, Z_i) (figure 1.6) ne tournent pas avec la terre et pointent vers des étoiles lointaines pour sembler fixes à un observateur terrestre. Z_i est l'axe de rotation de la terre, X_i pointe vers l'équinoxe vernal² et Y_i vient compléter le trièdre³ directe [5].

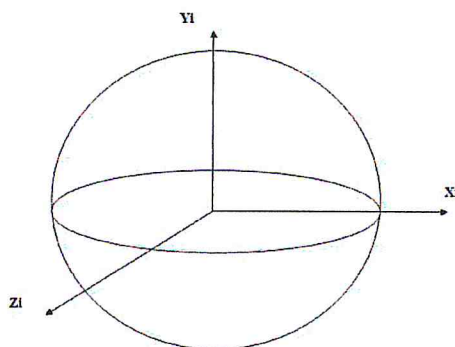


FIGURE 1.7: Un repère inertiel

2. un point d'orbite terrestre, atteint lorsque le soleil est exactement au Zenith sur l'équateur terrestre.

3. les données de trois plans deux à deux sécants selon trois droites ou axes, lesquelles sont orientées et ordonnées.

1.4.2 Centrale inertielle IMU (Inertial Measurement Unit)

C'est un appareil qui permet de nous fournir la position, la vitesse, et l'accélération d'un mobile. Il comporte six capteurs :

- Trois gyroscopes mesurant les trois composantes du vecteur de vitesse angulaire.
- Trois accéléromètres mesurant les trois composantes du vecteur de l'accélération.

Exemple. Il y a trois mouvements de translation (déplacement d'un corps d'un point à un autre) : le mouvement longitudinal (x) qui décrit un objet qui avance ou recule, le mouvement latéral (y) qui désigne un objet qui bouge de droite à gauche et enfin le mouvement vertical (z) qui est utilisé pour les corps qui montent ou descendent. Il y a ensuite trois mouvements de rotation : le roulis (R) qui désigne le mouvement de bascule de droite à gauche, le tangage (P) qui marque le mouvement d'un corps qui bascule de l'avant à l'arrière et enfin le lacet (L) qui est le pivotement de droite à gauche, comme une voiture qui prend un virage. Les accéléromètres mesurent les mouvements de translation et les gyroscopes les mouvements de rotation.

1.4.3 Principe de base de la navigation inertielle

Le calcul de position d'un mobile se fait à l'aide d'un accéléromètre qui mesure son accélération lors d'un mouvement. La vitesse d'un mobile peut être calculée en intégrant l'accélération selon la formule suivante :

$$V = \int a \cdot dt \quad (1.4.1)$$

Où V est la vitesse et a est l'accélération.

Par la suite, la position sera déterminée en intégrant la vitesse résultante selon la formule suivante :

$$P = \int V \cdot dt \quad (1.4.2)$$

Où V est la vitesse et P est la position.

1.4.4 Les capteurs inertiels MEMS (Micro Electro Mechanical Systems)

1.4.4.1 Accéléromètre

Un accéléromètre est un capteur attaché au mobile qui permet de mesurer l'accélération lors du mouvement. Le fonctionnement de l'accéléromètre est similaire au système masse-ressort (voir Figure 1.8). Les mesures capturées par l'accéléromètre représentent les forces qui sont appliquées sur ses axes (X, Y, Z) comme le montre la Figure. En règle générale, les accéléromètres mesurent l'accélération en détectant combien une masse presse sur quelque chose quand une force agit sur elle. Ces forces peuvent être statiques (force de gravité) ou dynamique (lorsque le mobile bouge ou vibre). En mesurant l'accélération dynamique, il est possible d'analyser le déplacement d'un objet [10].

Son principe est basé sur la loi fondamentale de la dynamique $\vec{F} = M.\vec{a}$, où (F : la force, M : la masse et a : l'accélération). Les accéléromètres sont distingués par deux types [12] :

1. *Les accéléromètres non asservis* : permettent de mesurer l'accélération immédiate comme les accéléromètres à détection de déplacement.
2. *Les accéléromètres à asservissement* : leur accélération est mesurée à la sortie d'une boucle à contre réaction qui comporte un correcteur améliorant la précision.

Dans le Smartphone l'accéléromètre (MEMS accéléromètre) a la forme d'une petite puce, ces accéléromètres sont de types non asservis vu leur faible coût par rapport aux accéléromètres à asservissement.

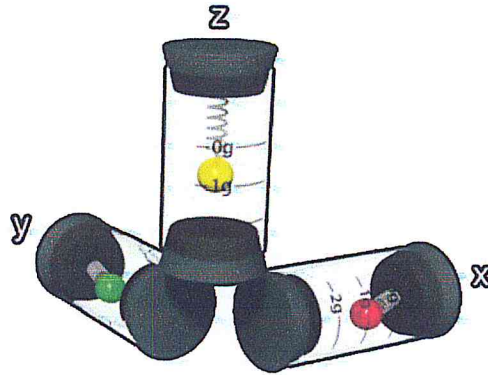


FIGURE 1.8: Scéma d'un système masse ressort des 3 axes (X, Y, Z) de l'accéléromètre

1.4.4.2 Gyroscopie

Un gyroscope permet de maintenir ou de mesurer une direction d'après le principe de conservation du moment angulaire. Un gyroscope est composé d'un rotor tournant autour d'un axe et monté sur un anneau intérieur tournant sur un axe perpendiculaire au précédent lui-même monté sur un anneau extérieur dont l'axe de rotation est perpendiculaire aux 2 précédents [11]. Voir Figure 1.8.

Le moment du gyroscope est fourni par un couple de forces : la gravité pousse vers le bas le centre de la masse tandis qu'une force égale le pousse vers le haut. Le déplacement résultant n'est pas vers le bas comme nous pourrions le supposer mais perpendiculaire à la fois à son poids et à l'axe de rotation c'est à dire dans la direction horizontale.

L'équation fondamentale qui décrit le mouvement [11] est :

$$\vec{T} = \frac{d\vec{L}}{dt} = \frac{dI\vec{\omega}}{dt} = I\vec{\alpha} \quad (1.4.3)$$

où \vec{T} est le moment du gyroscope et \vec{L} le moment cinétique, le scalaire I est le moment d'inertie, le vecteur $\vec{\omega}$ la vitesse angulaire et $\vec{\alpha}$ l'accélération angulaire. Ceci est analogue à la loi de Newton $\vec{F} = m * \vec{a}$ pour le mouvement rectiligne.

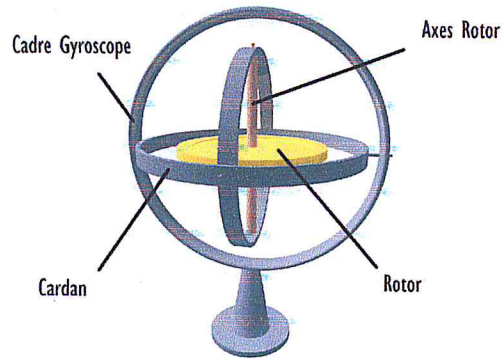


FIGURE 1.9: Le gyroscope

1.4.4.3 Boussole

Un magnétomètre est un instrument utilisé pour mesurer la résistance et / ou la direction du champ magnétique dans la région environnante de l'instrument.

Les magnétomètres peuvent être divisés en deux types de base :

- Magnétomètres scalaires : mesurent la force totale du champ magnétique auxquelles ils sont soumis.
- Magnétomètres vectoriels : ont la capacité de mesurer la composante du champ magnétique dans une direction donnée, par rapport à l'orientation spatiale de l'appareil [10].

1.4.5 Les sources d'erreurs

1.4.5.1 Accéléromètre

La source la plus importante de l'erreur d'un accéléromètre est le biais. Le biais d'un accéléromètre est le décalage de son signal de sortie à partir de la valeur réelle, en m/s^2 . Une erreur de polarisation constante ε , en cas de double intégration provoque une erreur dans la position qui est quadratique avec le temps. L'erreur accumulée est en position [2] :

$$S(t) = \varepsilon * t^2 / 2 \quad (1.4.4)$$

t est le temps de l'intégration.

Il est possible d'estimer le biais de la mesure de la moyenne à long terme de la sortie de l'accéléromètre, où il ne subit pas une accélération. Les erreurs de polarisation non corrigées sont typiquement des sources d'erreurs qui limitent les performances du dispositif [2].

1.4.5.2 Gyroscope

La polarisation d'un gyromètre est la moyenne de sortie du gyroscope quand il ne subit aucune rotation (ex : le décalage de la sortie de la valeur réelle), en °/h. Une erreur de polarisation constante ε , lorsqu'elle est intégrée, provoque une erreur angulaire qui augmente linéairement avec le temps [2] :

$$\Theta(t) = \varepsilon * t \quad (1.4.5)$$

L'erreur de polarisation constante d'un gyromètre peut être estimée. En prenant une moyenne à long terme de la sortie du gyroscope. Alors qu'il ne subit aucune rotation. Une fois que la déviation est connue, il est nécessaire de le compenser en soustrayant la déviation de la sortie [2].

Une autre erreur survenant dans les gyroscopes, qui est «l'erreur de calibrage», qui se réfère à des erreurs dans les facteurs d'échelle, les alignements, et linéarités des gyroscopes. Ces erreurs ont tendance à produire des erreurs qui sont observés pendant que le dispositif est en rotation. Ces erreurs conduisent à l'accumulation de dérive supplémentaire dans le signal intégré, dont l'amplitude est proportionnelle à la vitesse et la durée des mouvements [2].

1.4.5.3 Magnétomètre

Les deux principales sources de contamination sont les erreurs de mesure magnétique dans le capteur, et les erreurs dans la mesure de la fréquence et ferreux (contenant du fer) les matériaux dans les opérateurs et les instruments. Si le capteur est mis en rotation quand la mesure est effectuée, une erreur supplémentaire est générée[10].

1.4.6 La fusion des capteurs

La fusion des capteurs est la combinaison de mesures sensorielles provenant des données sensorielles qui dérivent de sources différentes, les informations résultantes leur utilisation est en quelque sorte mieux fusionnée qu'individuellement. Le meilleur terme dans ce cas, peut signifier plus précise, plus complète, ou plus fiable, comme la vision stéréoscopique (calcul des informations de profondeur en combinant des images bidimensionnelles à partir de deux caméras à des points de vue légèrement différents) [14].

1.4.7 Les avantages et inconvénients d'INS

les avantages d'INS

1. On obtient la position, vitesse, orientation, accélération et rotation.
2. Il a une très bonne précision sur une courte durée.
3. Le système est complètement autonome c.-à-d. Il ne dépend pas d'un dispositif extérieur ou d'une information externe.
4. Les centrales inertielles ont une taille raisonnable réduite pour envisager tout type d'application.

Les inconvénients d'INS

1. Les effets du déplacement aléatoire, bien qu'ils sont faibles dans les accéléromètres et les gyroscopes, mènent à une très forte dérive après la double intégration.
2. *Il n'a pas d'information absolue* : un INS ne peut pas s'initialiser seul c.-à-d la position et la vitesse sont toujours calculées à partir des conditions initiales qui doivent être fournies par un dispositif extérieur.
3. *Sensibilité de la gravité* : il en résulte souvent des erreurs, notamment en conséquence des erreurs d'orientations.

1.5 Conclusion

Ce chapitre a permis de connaître les différents systèmes de navigation ou positionnement outdoor et indoor, et en particulier la navigation inertielle qui utilise les différents capteurs inertiels. Dans ce qui suit, nous nous intéresserons à faire une étude plus approfondie de ces capteurs inertiels, en passant par de multiples étapes de tests afin de déterminer la méthode la plus adéquate pour le positionnement indoor.

Chapitre 2

Tests des capteurs MEMS

2.1 Introduction

Afin de mieux comprendre le fonctionnement des capteurs inertiels ainsi que leurs utilisations dans le processus de positionnement, un ensemble de tests sur deux scénarios a été effectués. Le premier scénario avait pour but la détection du mouvement d'une personne mobile qui porte un Smartphone dans ses mains, en supposant que la position de son départ et la vitesse de son déplacement sont connus au préalable. Dans ce scénario, l'accéléromètre est le capteur utilisé pour détecter les moments d'arrêts et de marches de la personne mobile afin de la positionner et suivre son mouvement. Dans le deuxième scénario, la vitesse est supposée inconnue et l'accéléromètre est donc utilisé pour calculer la vitesse de déplacement et la distance parcourue par le mobile. Dans la suite de ce chapitre, nous présentons un résumé sur l'ensemble des capteurs de navigation qui sont intégrés dans les Smartphones ainsi que les détails des tests effectués sur les deux scénarios.

2.2 Les capteurs intégrant les Smartphones

Aujourd'hui, presque tous les appareils fonctionnant sous le système d'exploitation Android sont équipés de capteurs capables de mesurer le mouvement, l'orientation ainsi que les conditions environnementales [20]. Ces capteurs sont capables de se procurer des données précises et exactes, ainsi ils permettent aux programmeurs de calculer les mouvements de l'appareil en trois dimensions, et son positionnement. On peut également inspecter son milieu ambiant.

Il existe trois types de capteurs qui sont pris en charge par la plateforme Android :

1. *Capteurs de mouvements* : Ils permettent de mesurer les différents types de forces comme l'accélération, la gravité et la rotation.
2. *Capteurs environnementaux* : Ils nous fournissent des informations sur les conditions environnementales comme la température de l'air ambiant et la pression, l'éclairage et l'humidité, ces valeurs sont récupérées à l'aide du baromètre, photomètre, et les thermomètres.

3. *Capteurs de positions* : Ils sont en mesure de calculer la position physique de l'appareil à l'aide de capteurs d'orientation et de magnétomètre.

Tous ces types de capteurs sont accessibles à travers l'Android Sensor Framework [4]. Certains d'entre eux sont basés sur le matériel (Hardware), tandis que d'autres sont des éléments sont intégrés dans le Smartphone (Software). Les capteurs basés sur le Hardware sont l'accéléromètre et le magnétomètre, quant aux capteurs basés sur le Software, appelés aussi capteurs virtuels, fondent leurs données sur un ou plusieurs capteurs Hardware, comme l'accélération linéaire ainsi que la gravité [20]. Tous les capteurs pris en charge par la plateforme Android sont résumés dans le tableau ci-dessous.

Capteur	Type	Description	Utilisation
Accéléromètre	Hardware	Mesure l'accélération des forces appliquées sur les 3 axes (X, Y, Z) de l'appareil, en incluant la force de gravité, elle est mesurée en m/s^2 .	Détection de mouvements (secousses, inclinaison, etc.).
Gyroscope	Hardware	Mesure la rotation en terme de vitesse (vitesse angulaire) autour de chaque axe (X, Y, Z), mesurée en radian/seconde.	Détection de rotation (spin, tourner, etc.).
Accélération linéaire	Software ou Hardware	Mesure l'accélération des forces appliquées sur les 3 axes (X, Y, Z) de l'appareil, en excluant la force de gravité, elle est mesurée en m/s^2 .	Calcul de la vitesse du mobile.
Orientation	Software	Mesure le degré de rotation d'un mobile autour de 3 axes (X, Y, Z).	Déterminer la position du mobile.
Gravité	Software ou Hardware	Mesure la force de gravité appliquée sur le mobile des 3 axes (X, Y, Z), mesurée en m/s^2 .	Détection de mouvement.
Magnetic_field (magnétomètre)	Hardware	Mesure le champ magnétique des 3 axes (X, Y, Z) en μ Tesla.	Pour la création d'une boussole.
Light (lumière)	Hardware	Mesure le niveau de luminosité en Lux.	Contrôle de luminosité de l'écran.
Température	Hardware	Mesure la température ambiante d'une pièce en degrés Celsius.	Contrôle de température de l'air.
Proximité	Hardware	Mesure la proximité d'un objet par rapport à l'écran d'affichage d'un mobile en Cm.	Déterminer si le téléphone est détenu jusqu'à l'oreille.
Vecteur de rotation	Software ou Hardware	Mesure l'orientation de l'appareil en fournissant les éléments du vecteur de rotation de l'appareil.	La détection de mouvement et de rotation.
Humidité relative	Hardware	Mesure l'humidité relative ambiante en pourcent (%).	La surveillance de point rosé absolu, et l'humidité relative.
Pressure (pression)	Hardware	Mesure la pression de l'air ambiant en hPa ou mbar.	Le suivi de changement de pression d'air.

TABLE 2.1: Les capteurs pris en charge par la plateforme Android [20].

2.3 Scénario de positionnement sur une dimension avec une vitesse constante

Le but de cette section est l'utilisation du capteur accéléromètre pour la détection de mouvement d'une personne tenant dans les mains un Smartphone, et ainsi tracer sa trajectoire.

2.3.1 Description du scénario

Ce scénario est une simplification du processus de suivi du mouvement, la personne mobile dans ce cas marche avec une vitesse et un point de départ connus au début du mouvement et sa trajectoire est tracée sur une seule dimension, donc l'accéléromètre est utilisé seulement pour détecter les états de marches et d'arrêts de la personne mobile. Le point de départ peut être obtenu avec un réseau Wifi, GPS s'il est disponible ou bien manuellement, dans le cas d'une marche normale la vitesse est généralement constante

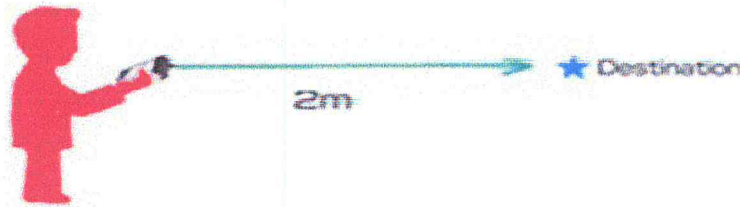


FIGURE 2.1: Déplacement d'une personne tenant un Smartphone.

Afin de pouvoir effectuer l'expérience citée précédemment, l'accéléromètre nous permet de distinguer les moments de marches et d'arrêts. Nous sommes amenés alors à utiliser un chronomètre pour la mesure des durées de ces moments et calculer par la suite la distance parcourue par le smart phone.

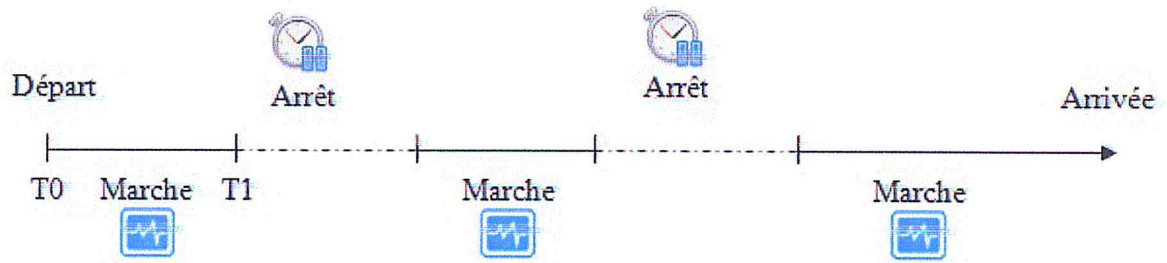


FIGURE 2.2: Etat d'un Smartphone en déplacement

2.3.2 Remarque et proposition

L'accéléromètre comporte 3 axes (X, Y, Z) voir Figure 1.3, et mesure l'accélération des forces appliquées sur chaque axe. Qui dépend de la position du Smartphone l'accélération de l'un ou de plusieurs axes est influencée par la force de gravité, par exemple lorsque le Smartphone est tenu horizontalement sur la main, l'axe Z de l'accéléromètre est affecté par la force de gravité et indique une accélération différente de nul même au repos.

Dans ce cas, l'utilisation des valeurs de l'accélérations des axe (X,Y,Z) ou leurs somme, nous mène à des résultats non fiables pour la détection des mouvements et des arrêts.

Pour pallier à ce problème, nous avons considéré qu'il est nécessaire d'utiliser la variation de l'accélération entre deux valeurs continues, en d'autre terme, nous avons remarqué que lorsque la personne mobile est en état de marche, il y'a des secousses ou vibrations, les variations de l'accélération de ces vibrations vont détecter les états de marches ou d'arrêts, c'est-à-dire la somme des variations des 3 axes (X, Y, Z). La variation sera calculée avec la formule suivante :

$$Variation = |AccélérationPrécédente - AccélérationActuelle| \quad (2.3.1)$$

Où l'AccélérationPrécédente est l'ancienne accélération et l'AccélérationActuelle est l'accélération qui change de valeur à chaque seconde.

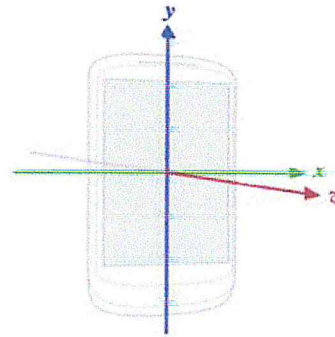


FIGURE 2.3: les axes (X, Y, Z) de l'accéléromètre sur un Smartphone.

L'axe X est horizontal et pointe vers la droite, l'axe Y est vertical et pointe vers le haut, et l'axe Z pointe vers l'extérieur de la face avant de l'écran. Le graphe suivant illustre les valeurs de l'accélérations en marche et arrêt de chaque axe ainsi que leur somme.

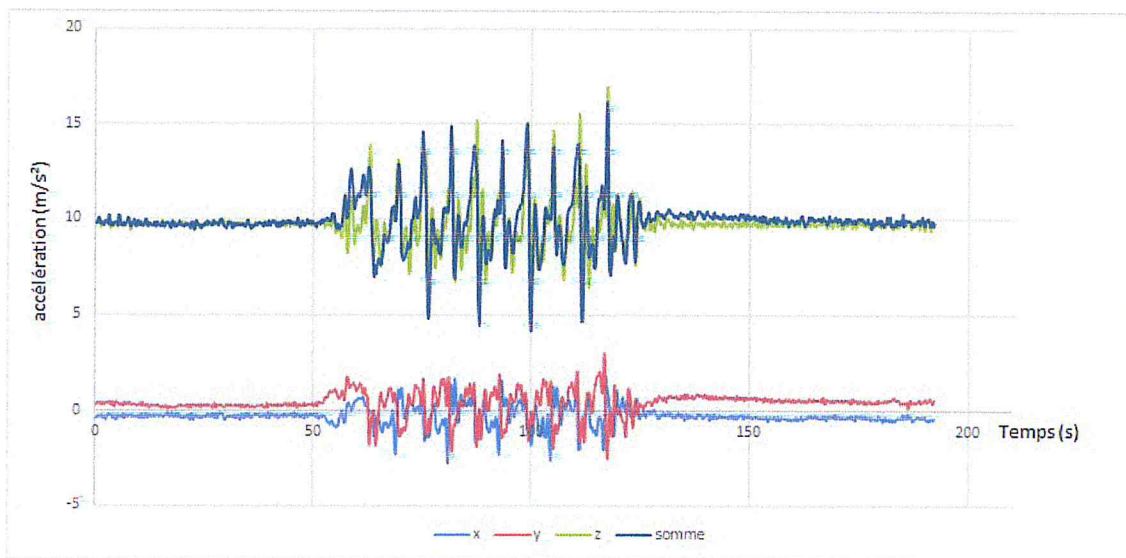


FIGURE 2.4: Graphe de l'accélération d'un mobile en état de marche et arrêt .

Les figures qui suivent sont un exemple qui montre la difficulté de distinguer les moments de marches et d'arrêts à partir des valeurs de l'accélération de l'axe Y ainsi que la somme des accélérations des 3 axes (X, Y, Z). La première indique l'accélération de l'axe Y en état de marche, et la deuxième indique l'accélération de l'axe Y en état

d'arrêt en inclinant le Smartphone de 40°, afin de montrer l'effet ou l'influence de la gravité sur l'axe de Y qui fausse les résultats de la détection du mouvement .

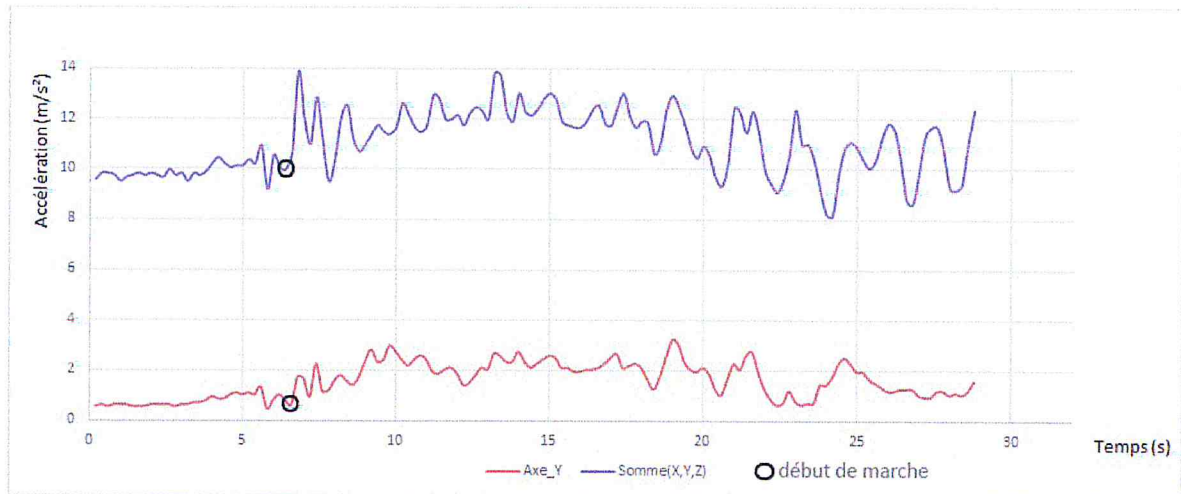


FIGURE 2.5: Graphe de l'accélération sur l'axe Y et la somme des valeurs des axes en état de marche

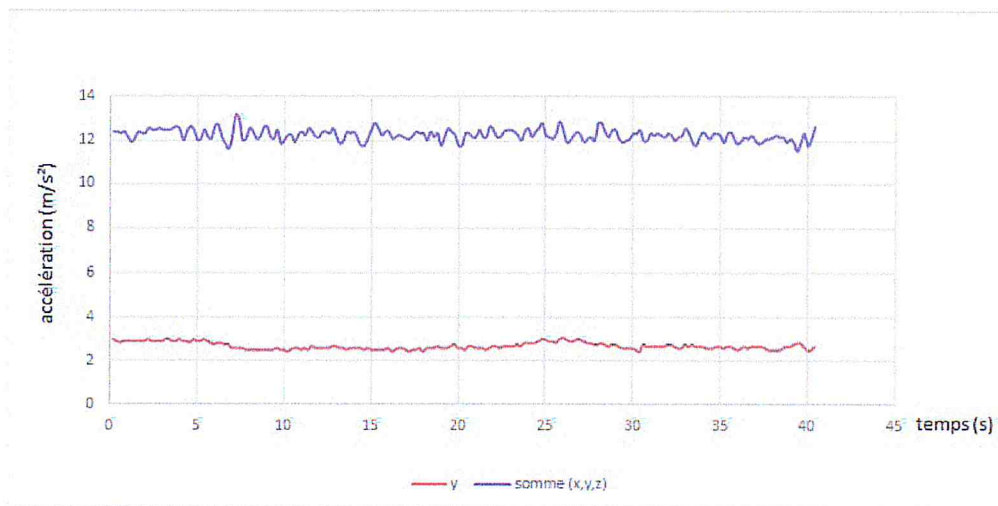


FIGURE 2.6: Graphe de l'accélération sur l'axe Y et la somme des valeurs des axes en état d'arrêt

On remarque sur le premier graphe que la valeur de l'accélération en état de marche augmente jusqu'à « $3m/s^2$ » sur l'axe Y et « $13,95 m/s^2$ » sur la somme des valeurs des axes. Et dans le deuxième graphe, les valeurs de l'accélération sur l'axe Y prennent presque

les mêmes valeurs soit dans un état d'arrêt ou état de marche, ainsi que la somme des valeurs des axes. Ce qui ne nous empêche de distinguer les vrais états de marche et d'arrêt.

2.3.3 Calibrage de l'accéléromètre

Au repos la valeur de l'accélération devrait être nulle, cette valeur n'est pas tout le temps enregistrée par le capteur accéléromètre cela est dû à sa sensibilité. Par exemple si le capteur se trouve dans la main d'une personne en état d'arrêt, le capteur enregistre quand même une accélération résultante du moindre tremblement de la main. Une phase de calibrage est alors envisagée pour pouvoir distinguer les états de marches et d'arrêts.

Pour réaliser le calibrage, nous avons réalisé plusieurs études pour avoir plus de précision :

1. Smartphone en état stationnaire sur table.
2. Smartphone en état stationnaire sur la main.
3. Smartphone en état de marche avec une vitesse moyenne.
4. Smartphone en état de marche avec une vitesse rapide.

2.3.3.1 Mobile en état stationnaire sur table

Le graphe suivant nous fourni la somme des variations des axes (X, Y, Z) par rapport au temps, ces valeurs ont été calculés à partir de la formule vue précédemment, qui est : $Variation = |AccélérationPrécédente - AccélérationActuelle|$. Nous pouvons dire que la variation de l'accélération est entre 0 m/s^2 et 0.70 m/s^2

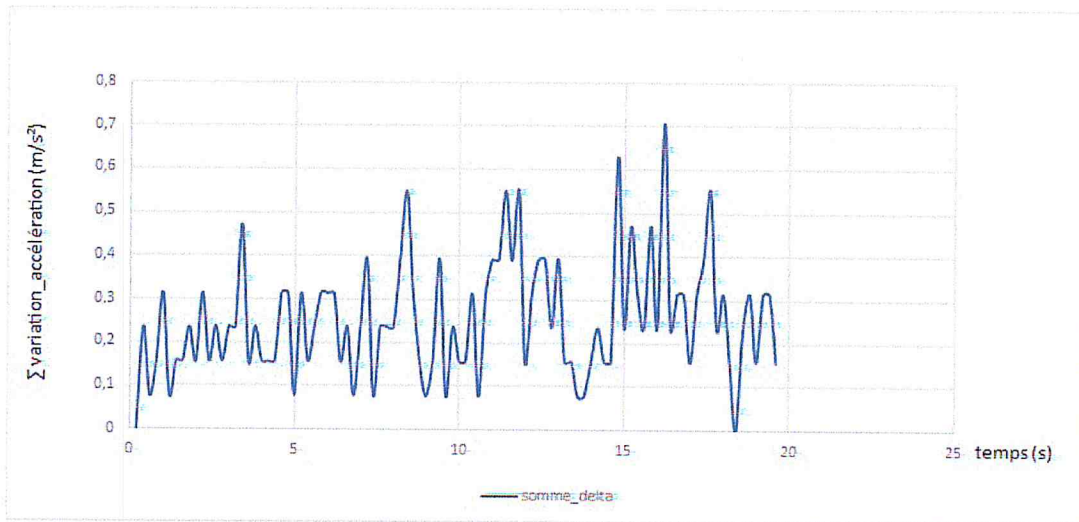


FIGURE 2.7: Graphe de la somme de la variation des accélérations du mobile sur table

2.3.3.2 Mobile en état stationnaire sur la main

La Figure suivante représente les valeurs des sommes des variations de l'accélération des axes (X, Y, Z) par rapport au temps. Ces valeurs varient entre 0.01m/s² et 0.382 m/s², dans ce cas là, nous pouvons dire que l'accéléromètre commence à détecter des mouvements à partir de 0.382m/s² et ce sur la main.

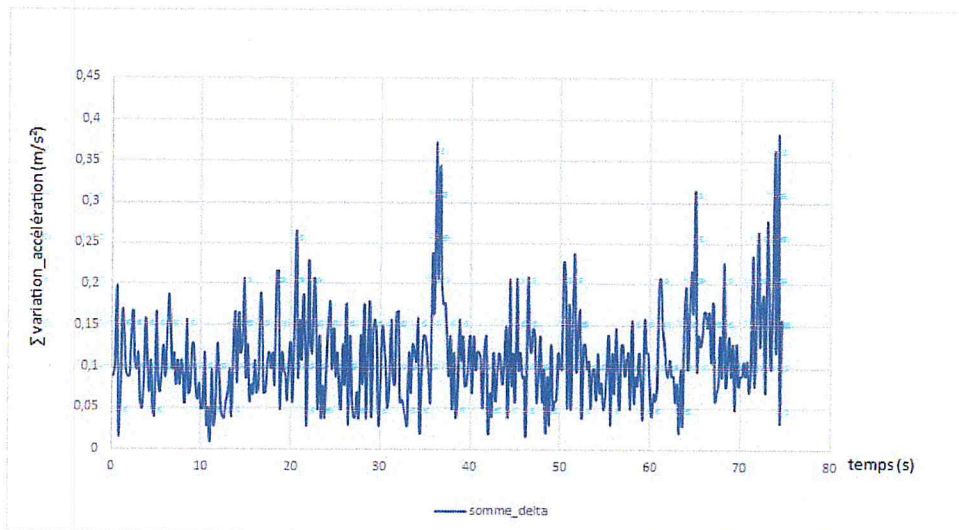


FIGURE 2.8: Graphe de la somme de la variation de l'accélération du mobile sur la main

2.3.3.3 Mobile en vitesse moyenne

Le schéma qui suit est aussi une illustration des valeurs de la somme des variations de l'accélération des trois axes (X, Y, Z), mais cette fois le mobile est en marche avec une vitesse moyenne. La somme des variations est entre 0 m/s² et 1.099 m/s². Nous remarquons qu'à chaque augmentation de la vitesse ou chaque mouvement du smartphone la variation change de valeur, ce qui indique le début d'un mouvement ou marche.

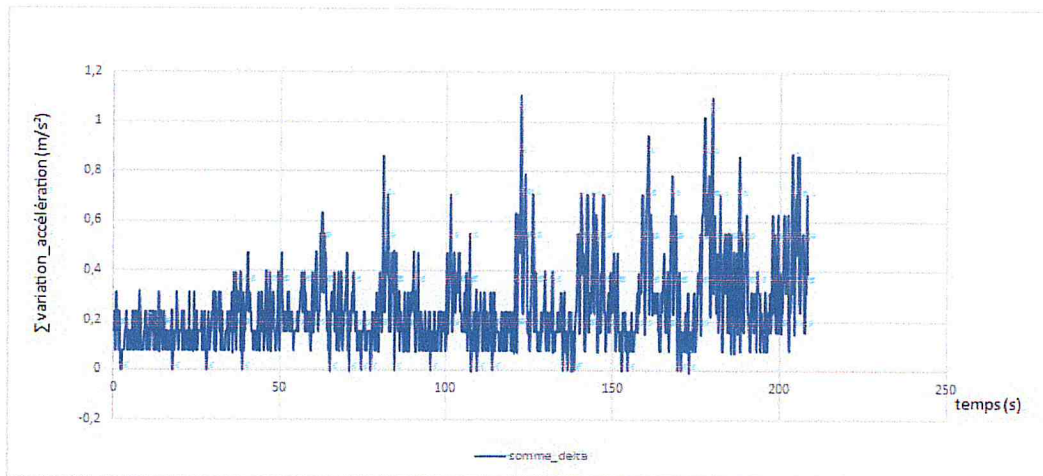


FIGURE 2.9: Graphe de la somme de la variation de l'accélération du mobile en vitesse moyenne

2.3.3.4 Mobile en grande vitesse

Quand le mobile est en grande vitesse, la variation de l'accélération des 3 axes varie entre 0 m/s² et 4.003 m/s², la figure suivante représente le graphe de cet état.

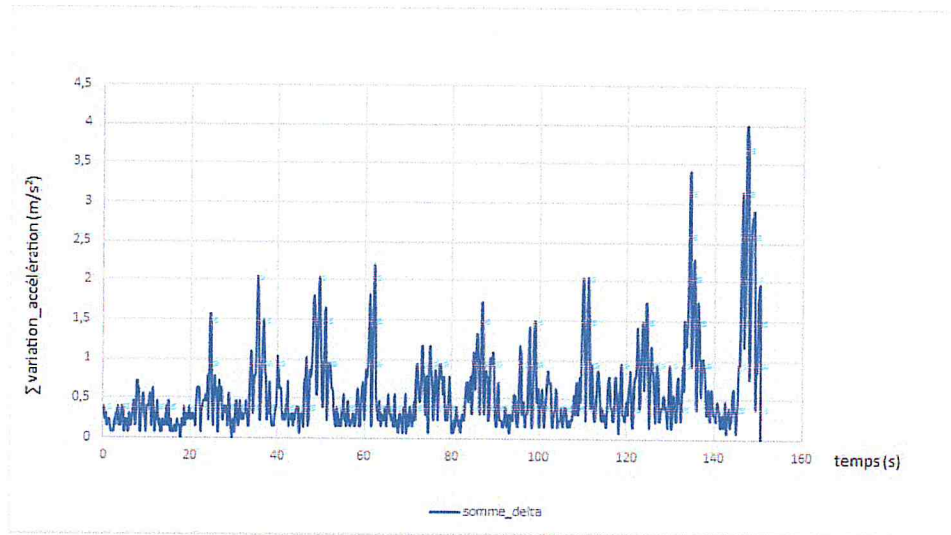


FIGURE 2.10: Graphe de la somme de la variation de l'accélération du mobile en grande vitesse

2.3.3.5 Résultats

Après avoir étudié les quatre états du mobile : mobile sur table et main (état arrêté), mobile en marche vitesse moyenne et grande, nous pouvons conclure que si la somme des variations des 3 axes (X, Y, Z) est supérieure à 0.7 m/s^2 , le Smartphone est en état de marche et si elle est inférieure, le Smartphone est en état d'arrêt. Pour mettre en pratique ce qui a été fait précédemment, nous avons écrit un algorithme pour la détection des états de marches et d'arrêts du mobile, en passant par le calibrage.

2.3.4 Algorithme du scénario

Pour la réalisation du 1er scénario, nous avons élaboré un algorithme, qui nous permet de détecter le mouvement et l'arrêt. Cet algorithme va contenir les variables suivantes :

- *Marche* : le compteur du temps de marche.
- *Arret* : le compteur du temps d'arrêt.
- *Accélération_x*, *Accélération_y*, *Accélération_z* : les accélérations des trois axes (X, Y, Z) calculées à partir de l'accéléromètre.

- *Accélération_nouv_x*, *Accélération_nouv_y*, *Accélération_nouv_z* : les nouvelles accélérations des trois axes (X, Y, Z) qui changent à chaque seconde, calculées aussi à partir de l'accéléromètre.
- *Delta_x*, *Delta_y*, *Delta_z* : les variations des accélérations des trois axes (X, Y, Z).
- *Temps_marche* : le temps de marche du chemin parcouru.
- *Temps_arrêt* : le temps d'arrêt du chemin parcouru.
- *Etat* : un booléen qui indique l'état du mobile (marche/arrêt).

Algorithme 2.1 Algorithme du premier scenario

```
Début
Acceleration_x , Acceleration_y , Acceleration_z ;
Acceleration_nouv_x , Acceleration_nouv_y , Acceleration_nouv_z ;
Delta_x , Delta_y , Delta_z ;
Marche , ArretTemps_marche=0 ;
Temps_arrêt=0 ;
Etat=0 ; // 0 pour arrêt et 1 pour marche
  Si          Etat= 0;
    Alors
      Acceleration_x = Acceleration_nouv_x ;
      Acceleration_y = Acceleration_nouv_y ;
      Acceleration_z = Acceleration_nouv_z ;
      Etat = 1 ;

  Sinon // Etat en marche
    Alors
      Delta_x = | Acceleration_x - Acceleration_nouv_x | ;
      Delta_y = | Acceleration_y - Acceleration_nouv_y | ;
      Delta_z = | Acceleration_z - Acceleration_nouv_z | ;

      Acceleration_x = Acceleration_nouv_x ;
      Acceleration_y = Acceleration_nouv_y ;
      Acceleration_z = Acceleration_nouv_z ;

  Fsi

  Si      Delta_x + Delta_y + Delta_z > 0.7
// condition début de marche
    Alors
      Marche ;

  Sinon
    Alors
      Arret ;

  Fsi
Temps_marche = Temps_marche + Marche ;
Temps_arret = Temps_arret + Arret ;
Fin
```

2.3.5 observations des résultats du scénario

Après avoir implémenté l'algorithme vu précédemment, nous avons testé ce premier scénario, le tableau suivant résume les résultats obtenus :

Distance réelle (Mètre)	11	22	33
Distance obtenue (Mètre)	10.92	21.87	31.72
Erreur obtenue ε (Mètre)	0.08	0.13	1.28

TABLE 2.2: Résultats du premier scénario

Les distances réelles ont été données avant de commencer le test. Afin de pouvoir mettre en place le scénario, nous nous sommes déplacés sur une trajectoire linéaire, sans secousse ni vibration, et nous nous sommes arrêtés deux fois durant le test, pour distinguer les moments de marches et d'arrêts. Le résultat obtenu des distances n'est pas exacte et cela est dû aux erreurs de précision du capteur l'accéléromètre. Nous pouvons remarquer que lorsque la distance augmente l'erreur ε augmente avec. Dans la prochaine section, nous réaliserons le 2^{eme} scénario.

2.4 Scénario de positionnement sur une dimension avec une vitesse inconnue

Cette section présente un nouveau scénario, dans lequel nous supposons que la direction et le point de départ sont connus. Quant à la vitesse, elle est calculée au fur et à mesure avec les accélérations enregistrées par l'accéléromètre, afin de tracer une trajectoire linéaire.

2.4.1 Description du scénario

Ce scénario réalise une expérience plus avancée que l'ancienne. Cette dernière a pour objectif de calculer la vitesse lors du mouvement du mobile et cela fait la différence avec le 1^{er} scénario où la vitesse est paramétrée au lancement du test, alors que ce scénario doit être en mesure de calculer la vitesse en utilisant les valeurs d'accélérations fournies par l'accéléromètre.

Pour la réalisation de ce scénario, nous l'avons divisé en deux expériences ; la première expérience nécessitait la fabrication d'une plateforme qui comporte :

- Une barre de deux mètres.
- Un Smartphone fixé sur un objet pouvant se déplacer sur la barre.

Cette plateforme simplifie la réalisation de l'expérience par l'élimination des secousses dues aux mouvements. Pour effectuer ce test, on glissera l'objet qui porte le Smartphone pendant un instant t jusqu'à un mètre, après un arrêt de 5 secondes on repousse l'objet jusqu'à atteindre la fin de la barre, (voir figure 2.11).

La deuxième expérience consiste à tenir le Smartphone dans la main de la personne mobile et maintenir une direction linéaire par rapport à l'axe Y. Le processus sera comme suit : une marche de quatre mètres ensuite un arrêt, puis une marche de quatre mètres, (voir figure 2.1).

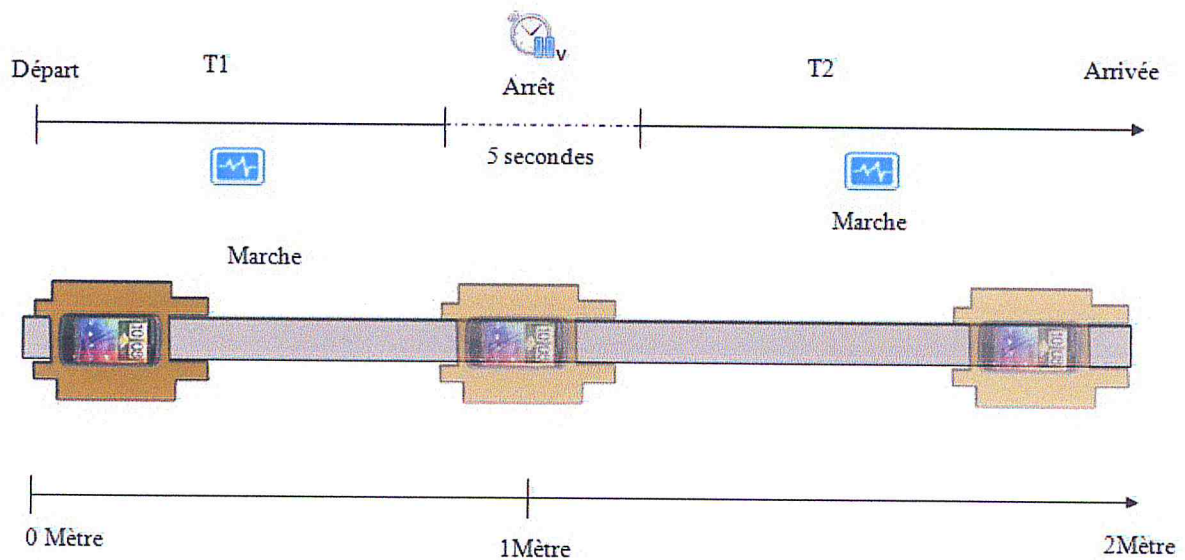


FIGURE 2.11: Plateforme du 2^{ième} scénario et l'état du déplacement du mobile

2.4.2 Remarque et proposition

L'accélération, la vitesse et la distance (position) sont trois termes étroitement liés, pour étudier la connexion entre ces trois concepts, nous allons introduire quelques formules à l'aide des notations suivantes :

$p(t)$ = Position d'un objet à l'instant t

$v(t)$ = Vitesse d'un objet à l'instant t

$a(t)$ = Accélération d'un objet à l'instant t

La première étape consiste à établir une relation entre la vitesse et l'accélération. Mathématiquement, l'accélération est définie par la dérivée du vecteur de vitesse en fonction du temps [13].

$$a(t) = \frac{dv}{dt} \quad (2.4.1)$$

Entre autre, nous pouvons appliquer le théorème fondamental du calcul qui permet d'écrire la relation entre la vitesse et l'accélération sous la forme suivante :

$$\int_{t1}^{t2} a(t)dt = v(t2) - v(t1) \quad (2.4.2)$$

Enfin, si l'on pose le temps initial $t1=0$ et le temps final $t2=T$, l'intégrale aurait la formule ci-dessous :

$$\int_0^T a(t)dt = v(T) - v(0) \quad (2.4.3)$$

Par la suite, en intégrant l'accélération nous allons trouver la vitesse à l'instant T

$$\begin{aligned} at|_0^T &= v(T) - v(0) \\ aT &= v(T) - v(0) \\ v(T) &= v(0) + aT \end{aligned} \quad (2.4.4)$$

D'après ce qu'on a vu, nous pouvons dire que l'accélération résultante de l'accéléromètre lors du mouvement sera utilisée avec la formule déduite des formules précédente

$$V_n = \sum_{i=0}^n (a_i * dt) \quad (2.4.5)$$

Les graphes ci-dessous montre l'accélération à l'état brute de l'axe Y et la vitesse calculée à partir de l'accélération sur la plateforme vu précédemment.

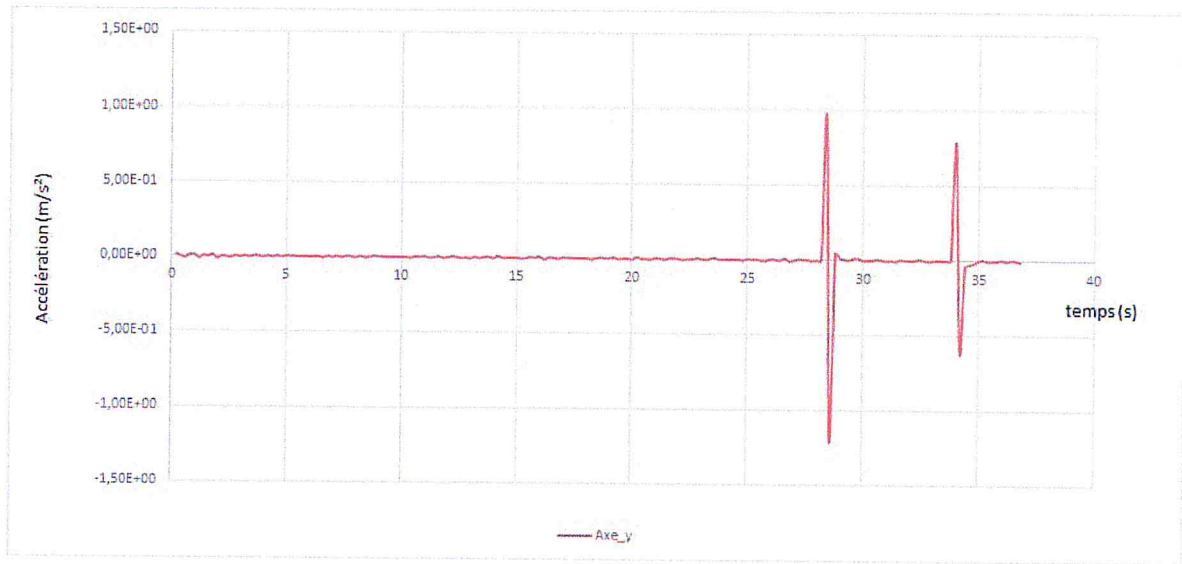


FIGURE 2.12: Graphe de l'accélération brute de l'axe Y sur la plateforme

Dans le graphe ci-dessus, nous remarquons que l'accélération varie entre -1.5m/s^2 et 1m/s^2 dans le premier mouvement et entre -0.6m/s^2 et 0.7m/s^2 dans le deuxième mouvement.

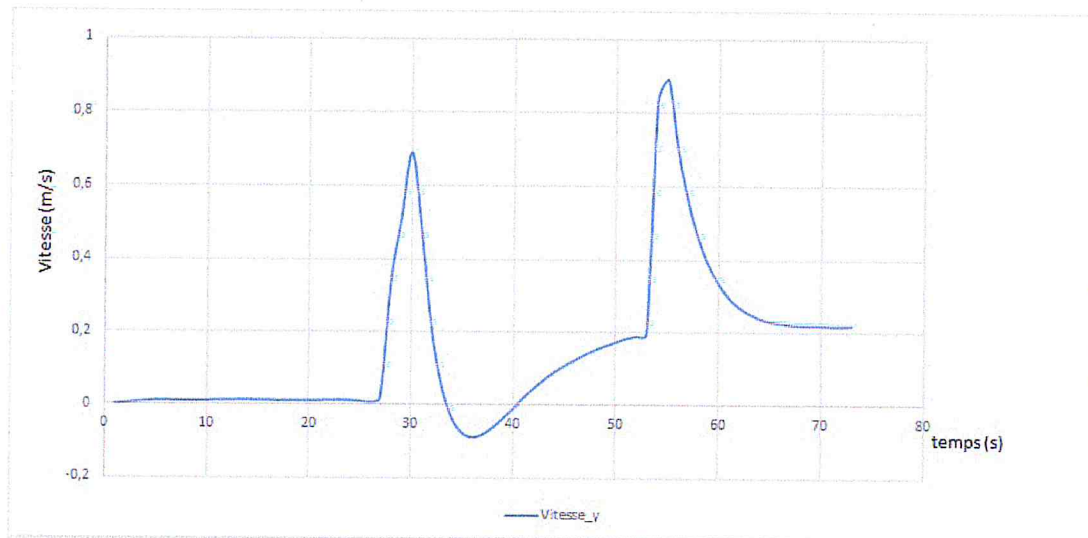


FIGURE 2.13: Graphe de la vitesse de l'axe Y sur la plateforme

La vitesse vue sur le graphe précédent est calculée par l'intégration de l'accélération à l'état brute, nous remarquons que la courbe semble correcte, ainsi que nous pouvons

dire que deux mouvements exercés sur la barre ont été détectés.

Après le calcul de la vitesse, nous allons passer à la seconde étape qui consiste à définir la relation entre la position et la vitesse, comme se fut le cas pour l'accélération et la vitesse. Nous pouvons aussi dériver le vecteur de position en fonction du temps pour obtenir la vitesse au temps t [13].

$$v(t) = \frac{p(t)}{dt} \quad (2.4.6)$$

Ainsi, d'après le théorème fondamental du calcul, on obtient la formule suivante, le résultat peut être considéré comme le déplacement d'un objet entre les temps t_1 et t_2 .

$$\int_{t_1}^{t_2} v(t)dt = p(t_2) - p(t_1) \quad (2.4.7)$$

Comme vu précédemment, nous allons planifier le processus entre le temps 0 et T , ainsi l'intégral devient comme suit :

$$\int_0^T v(t)dt = p(T) - p(0) \quad (2.4.8)$$

Ensuite nous allons intégrer la vitesse pour obtenir la position ci-dessous où le résultat final $p(t)$ représente le déplacement d'un objet entre le temps 0 et T .

$$\begin{aligned} \int_0^T (v(0) + aT)dt &= p(T) - p(0) \\ (v(0)T + \frac{1}{2}aT^2)|_0^T &= p(T) - p(0) \\ v(0)T + \frac{1}{2}aT^2 &= p(T) - p(0) \\ p(T) &= p(0) + v(0)T + \frac{1}{2}aT^2 \end{aligned} \quad (2.4.9)$$

Pour faire plus simple, nous allons calculer la distance ou bien la position p avec l'intégration de la vitesse calculée précédemment sur le temps de déplacement t avec la formule déduite suivante :

$$p_n = \sum_{i=0}^n (v_i dt + \frac{1}{2}a_i dt) \quad (2.4.10)$$

Le graphe suivant présente la distance calculée à partir de l'intégration de la vitesse :

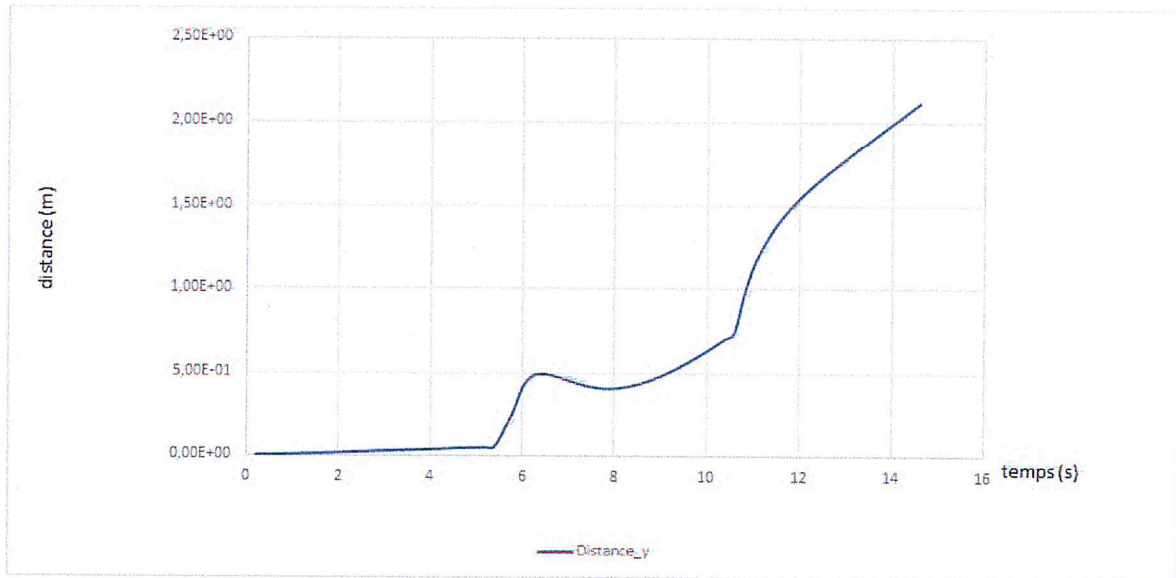


FIGURE 2.14: Graphe de la distance calculée sur la plateforme

Après l'observation des résultats du graphe de la distance sur la plateforme, nous remarquons que la distance augmente avec le temps, mais les valeurs ne sont pas identiques à la réalité et cela est dû aux bruits. Pour cela, nous avons déduit que le cas où le Smartphone sera tenu dans la main aura des erreurs encore plus grandes, vu que là où on a éliminé le plus de secousses sur la plateforme les résultats étaient toujours incorrects. Pour pallier ce problème nous avons fait des recherches qui nous ont mené au filtrage, ce que nous allons voir en détail par la suite du chapitre.

2.4.3 Filtrage des données brutes

Un filtre est un circuit électronique linéaire dont le comportement dépend de la fréquence. Autrement dit il atténue certaines composantes d'un signal et en laisse passer d'autres [17], (Voir Annexe A).

2.4.3.1 Types de filtre

On distingue plusieurs types de filtres parmi eux les filtres passifs et numériques. Ces types sont résumés dans le tableau ci-dessous.

Type de filtre		Description
Filtres passifs	Passe haut	Le filtre passe haut ne laisse passer que les hautes fréquences du signal d'entrée et filtre les basses fréquences.
	Passe bas	Le filtre passe bas ne laisse passer que les basses fréquences du signal et filtre les hautes fréquences.
Filtres numériques	Filtre à réponse impulsionnelle finie (RIF)	Appelé filtre non récursif car sa sortie dépend de l'entrée du signal. Sa réponse impulsionnelle se stabilisera ultimement à zéro.
	Filtre à réponse impulsionnelle infinie (RII)	Appelé filtre récursif car sa sortie dépend à la fois du signal d'entrée et du signal de sortie. Sa réponse impulsionnelle est une valeur variée qui prend beaucoup de temps à se stabiliser jusqu'à dire ne se stabilisera jamais et même à l'infini.

TABLE 2.3: Résumé des filtres passifs et filtres numériques [17]

Dans notre cas, le filtrage consistera à éliminer les bruits dont les valeurs de l'accélération ne sont pas fiables. Ces bruits sont causés par des secousses lors du déplacement du mobile, le filtrage se fera en deux phases :

- D'après le tableau 2.3, nous avons choisis d'utiliser deux filtres, le premier est le filtre passe haut qui va nous aider à parvenir au calcul de l'accélération qui est linéaire en éliminant l'effet de la gravité de l'accélération. Le mobile ne subira aucune force de gravité.

Le code ci-dessous sera utilisé [21]


```
private float [] highPass(float x, float y, float z)
{
float [] filteredValues = new float [3];
    gravity [0] = ALPHA * gravity [0] + (1 - ALPHA) * x;
    gravity [1] = ALPHA * gravity [1] + (1 - ALPHA) * y;
    gravity [2] = ALPHA * gravity [2] + (1 - ALPHA) * z;
    filteredValues [0] = x - gravity [0];
    filteredValues [1] = y - gravity [1];
    filteredValues [2] = z - gravity [2];
return filteredValues;
}
```

Où ALPHA est une valeur de pondération du filtre qui est entre 0 et 1. Cette variable associe un poids à l'ancienne valeur filtrée, dans notre cas nous l'avons fixé à 0.25 sur la plateforme, pour la deuxième expérience là où le Smartphone est tenu sur la main nous avons remarqué que la valeur de ALPHA n'indique pas des valeurs correctes après le filtrage pour cela nous sommes passés par plusieurs tests pour pouvoir faire le choix de ALPHA les valeurs essayées sont respectivement 0.25, 0.2, 0.18. les graphes suivant sont l'illustration des test faits pour le choix de ALPHA.

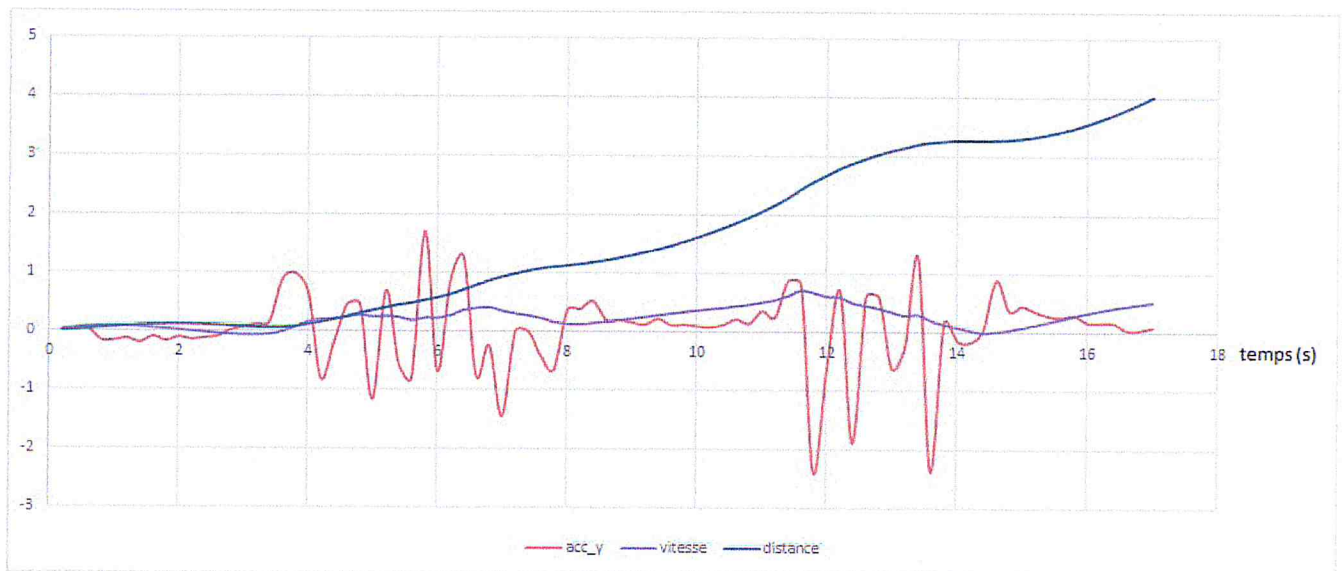


FIGURE 2.15: Graphe du test de marche avec ALPHA 0.25

Le processus était de marcher en tenant le Smartphone jusqu'à 4 mètres ensuite un

arrêt, puis remarcher 4 mètres encore, là nous remarquons que la distance est la moitié de ce qu'on a parcourue. On peut dire que pour l'instant 0.25 ne convient pas.

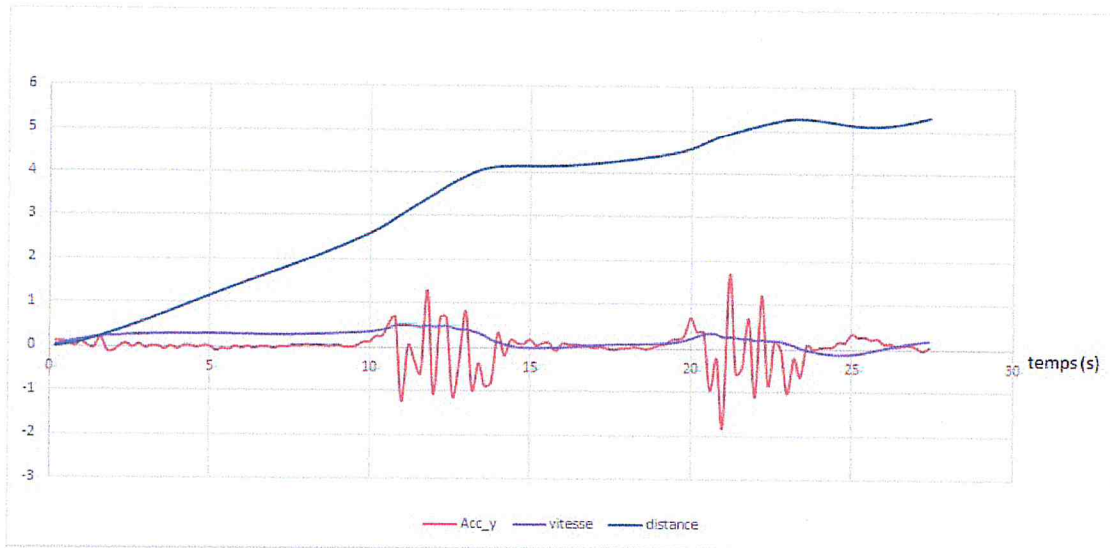


FIGURE 2.16: Graphe du test de marche avec ALPHA 0.2

Le deuxième graphe avec ALPHA=0.2 nous montre qu'on a parcourue jusqu'à 4 mètres ensuite un arrêt ensuite marche jusqu'à presque 6 mètres mais n'arrive pas jusqu'à 8 mètres. Pour cela nous allons refaire le test avec ALPHA= 0.18 voir figure 2.17.

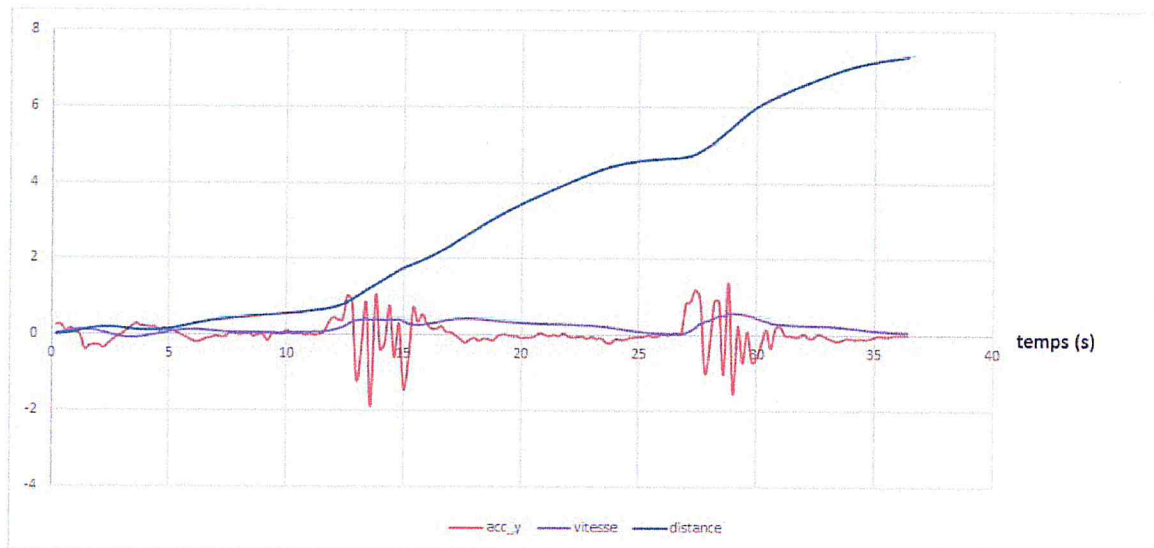


FIGURE 2.17: Graphe du test de marche avec ALPHA 0.18

Le dernier graphe nous a montré qu'on a marché jusqu'à 4 mètres ensuite un arrêt ensuite une autre marche jusqu'à presque 8 mètres. Pour conclure le choix de ALPHA s'est porté sur ces trois tests et la valeur la plus correcte pour notre test est 0.18.

- Le deuxième filtre choisi est le filtre à *réponse impulsionnelle infinie (RII)* qui est donné par la formule suivante [17] :

$$F(k) = \alpha \left(\frac{1}{N+1} \sum_{n=0}^N E(k-n) \right) + \beta \left(\frac{1}{M} \sum_{m=0}^M F(k-m) \right) \quad (2.4.11)$$

avec $0 < \alpha \leq 1$ et $\beta = 1 - \alpha$

2.4.4 Algorithme du scénario

Pour la réalisation du deuxième scénario, nous avons mis en place un algorithme qui permet de calculer la vitesse et la distance en utilisant les filtres pour le calcul de l'accélération. Cet algorithme utilisera les variables suivantes :

- *Accel_x; Accel_y; Accel_z* : Les accélérations brutes des trois axes (X, Y, Z) calculées à partir de l'accéléromètre.
- *Lin_x; Lin_y; Lin_z* : Les accélérations linéaire des axes .
- *Gravité[]* : La gravité de chaque axe (X, Y, Z).
- *ALPHA* : Une valeur de pondération utilisée pour le filtre passe haut.
- *Coef* : Un coefficient utilisé pour le filtre RII (Réponse Impulsionnelle Infinie).
- *Accel_filtre* : L'accélération finale de l'axe Y calculée.
- *Etat* : Un booléen qui indique si on est en arrêt ou en marche.
- *t* : Le temps du processus pour le calcul de la vitesse et la distance.
- *Vitesse* : La vitesse calculée à partir de l'accélération de l'axe Y.
- *Distance* : La distance parcourue.

Ainsi que les méthodes suivantes :

- *AccelerationLinéaire()* : cette méthode prend en compte le calcul de l'accélération de chaque axe en excluant la gravité .
- *AccelerationFiltre()* : cette méthode calcule l'accélération finale de l'axe de Y en utilisant le filtre passe haut.
- *CalculVitesse()* : La méthode qui calcule la vitesse.
- *CalculDistance()* : La méthode qui calcule la distance.

Algorithme 2.2 Algorithme du deuxieme scenario

```
Début
Accel_x , Accel_y , Accel_z ;
Gravité [] ;
Vitesse = 0 ; distance = 0 ; Alpha , Coef;
etat= 0 ; // arret
Si ( etat=1 ) // marche
{
    accel_lin_y = AccélérationLinéaire ( accel_x , accel_y , accel_z ) ;
    accel_filtre = AccélérationFiltre ( accel_filtre , alpha , accel_lin_y ) ;
    vitesse = CalculeVitesse ( accel_Filtre , dt ) ;
    distance = CalculeDistance( vitesse , accel_filtre , dt ) ;
}
AccélérationLinéaire ( float Lin_x , float Lin_y , float Lin_z )
{
    gravité [0] = ALPHA * gravité [0] + (1 - ALPHA) * Accel_x;
    gravité [1] = ALPHA * gravité [1] + (1 - ALPHA) * Accel_y;
    gravité [2] = ALPHA * gravité [2] + (1 - ALPHA) * Accel_z;
    Lin_x = Accel_x - gravité [0];
    Lin_y = Accel_y - gravité [1];
    Lin_z = Accel_z - gravité [2];
Return Lin_y ;
}
AccélérationFiltre ( float OldAccel , float Coef , float Val )
{
Return ( OldAccel * Coef ) + (( 1- Coef) * val) ;
}
CalculeVitesse( float acc , temps time)
{
return Vitesse + (acc * time) ;
}
CalculeDistance( float accel , float vitesse , temps t)
{
return Distance + ((Vitesse * t ) + ((accel *( t*t ) /2 ))) ;
}
Fin
```

2.4.5 observations des résultats du scénario

Après avoir effectué les étapes citées précédemment nous avons obtenus les résultats suivants :



FIGURE 2.18: Graphe de l'accélération finale de l'axe Y avec l'utilisation des filtres sur la plateforme

Comme on a cité précédemment, on fera les mêmes étapes pour le calcul de distance en passant par la vitesse

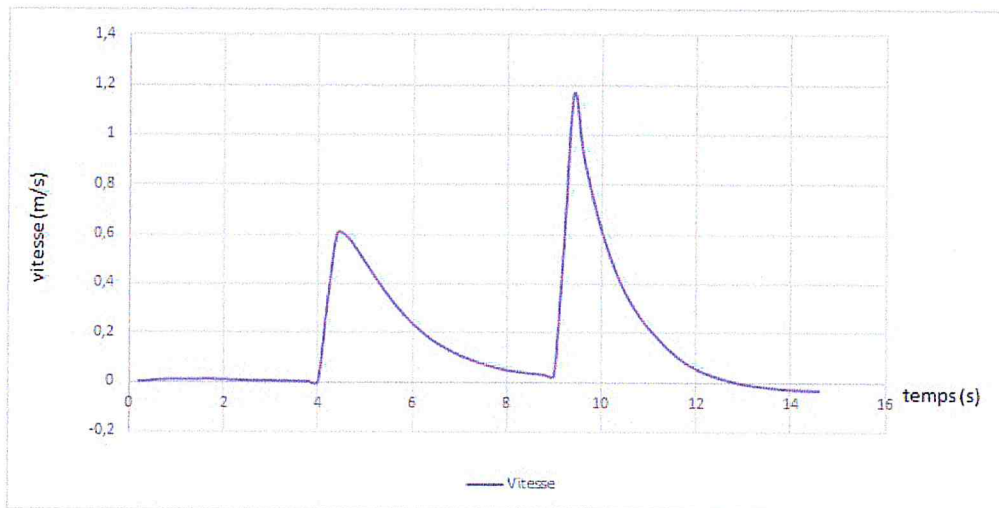


FIGURE 2.19: Graphe de la vitesse finale de l'axe Y avec l'utilisation des filtres sur la plateforme



Chapitre 2 Tests des capteurs MEMS

D'après le graphe présenté ci-dessus, nous pouvons dire qu'il y'a eu deux mouvements et un arrêt entre ces deux mouvements, pour l'instant tout semble correcte. Par la suite on calcule la distance, par l'intégration de la vitesse sur le temps t , selon l'équation précédente, voir la Figure 2.17.

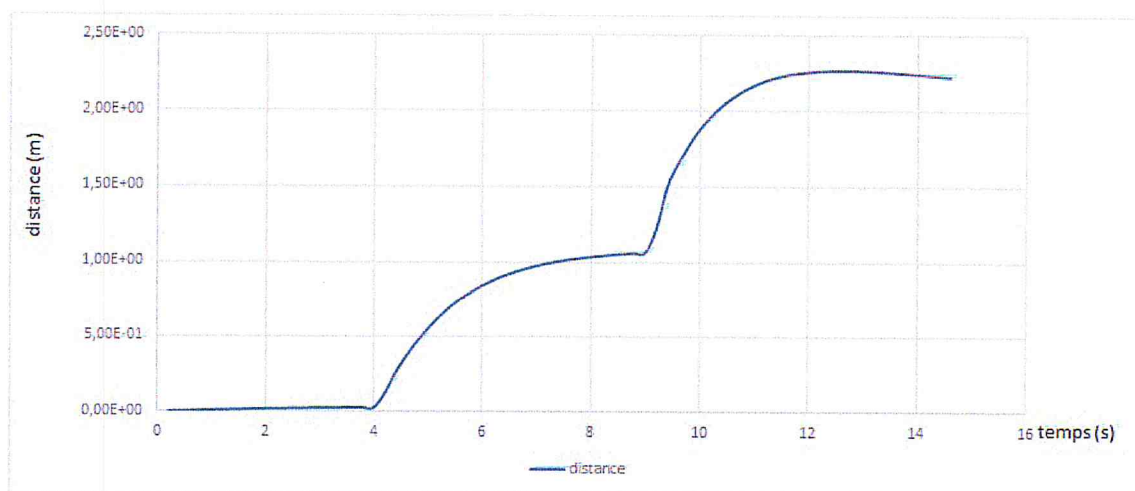


FIGURE 2.20: Graphe de la distance finale de l'axe Y avec l'utilisation des filtres sur la plateforme

Le graphe nous indique que la distance calculée sur la plateforme en utilisant les filtres approche de la réalité, nous remarquons le moment d'arrêt au début, ensuite la marche d'un mètre, puis arrêt, et enfin une autre marche d'un mètre, et cela est très proche de la réalité.

Par la suite nous verrons les résultats de la deuxième expérience où l'on tient le Smartphone dans la main. Le graphe suivant représente l'accélération de l'axe Y après l'utilisation des filtres.

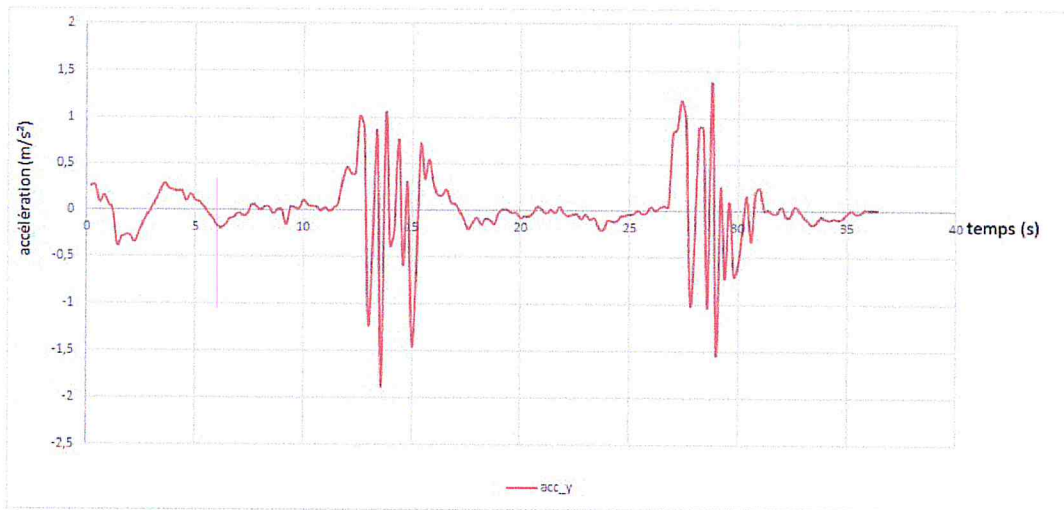


FIGURE 2.21: Graphe de l'accélération de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile

Sur le graphe nous remarquons une variété de valeurs de l'accélération et cela est dû aux secousses, mais les valeurs semblent correctes. Ensuite nous verrons le graphe de la vitesse calculée à partir de l'intégration de l'accélération résultante du graphe précédent.

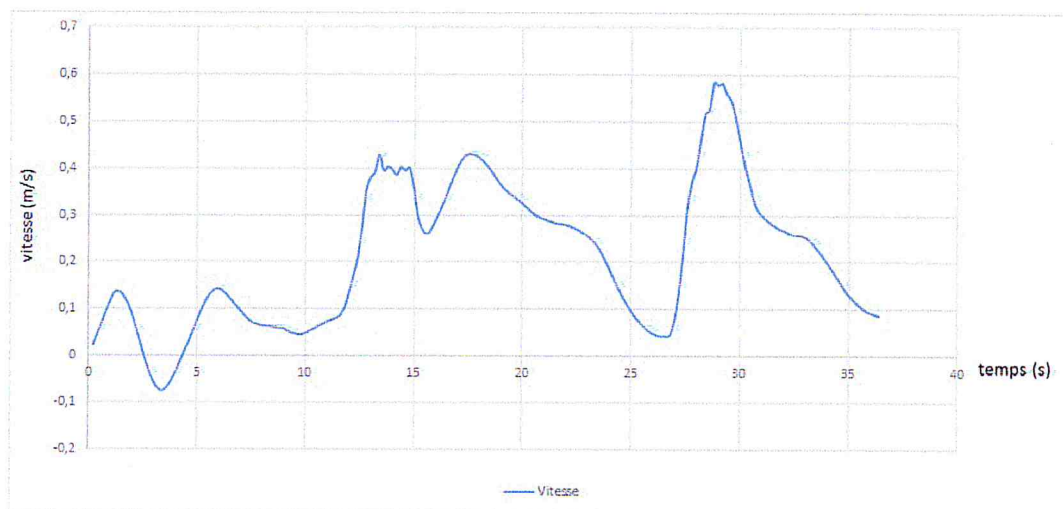


FIGURE 2.22: Graphe de la vitesse de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile

Ensuite nous calculerons la distance à partir de l'intégration de la vitesse résultante du graphe précédent.

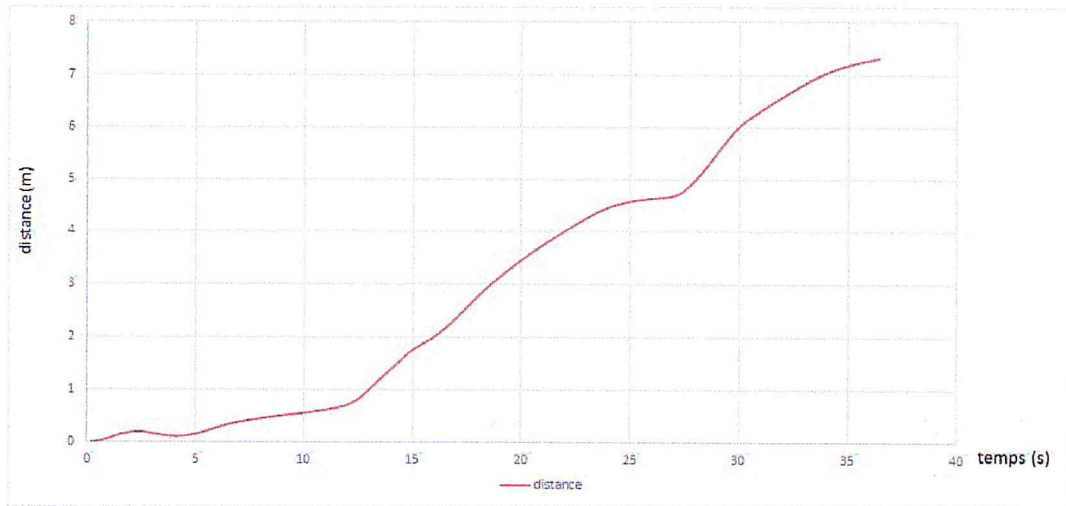


FIGURE 2.23: Graphique de la distance de l'axe Y en utilisant les filtres avec Smartphone tenu sur la main d'une personne mobile

Le graphique de la distance en utilisant les filtres, et en tenant le Smartphone dans la main, nous indique la distance parcourue par la personne mobile, qui rapproche à huit mètres et cela est très proche de la réalité, nous remarquons qu'il y'a des erreurs mais ces erreurs ne sont pas aussi grandes.

2.5 Conclusion

Ce chapitre nous a permis de procéder à deux tests essentiels du projet, le premier consiste à faire l'étude du capteur accéléromètre pour le but de tracer une trajectoire d'une personne mobile tenant un Smartphone dans la main et en marchant vers une direction linéaire tout en paramétrant sa vitesse qui doit être constante. Le 2^{ème} test consiste à reprendre le même principe du premier test en rajoutant un processus de calcul de la vitesse et en passant par un filtrage de données pour obtenir des résultats plus adéquats. Dans le prochain chapitre nous allons nous intéresser à la présentation et la conception du projet.

Chapitre 3

Présentation et conception du projet

3.1 Introduction

Ce chapitre est consacré à la présentation d'un aperçu général du projet, ainsi que la définition de ses objectifs, et ses besoins. La partie conception comportera une conception générale du système et une conception détaillée de chaque scénario.

3.2 Objectifs

L'objectif principal de ce projet est d'utiliser et de maîtriser la navigation inertielle à travers la réalisation d'une application de positionnement indoor. Cette maîtrise permet par la suite de répondre aux besoins des différentes applications de positionnement. Dans ce mémoire, nous avons conçu une application qui met en pratique les deux scénarios décrits dans le chapitre précédent.

Pour le bon fonctionnement du projet, le système doit répondre à ces deux conditions :

1. L'appareil doit être en mesure de guider l'utilisateur le long de son chemin.
2. La disposition d'une interface utilisateur intuitive.

L'objectif principal sera atteint à travers la conception et l'intégration d'un certain nombre de sous-systèmes. La première condition nécessite l'utilisation des algorithmes définis dans le chapitre précédent. Pour être en mesure de conduire l'utilisateur à sa destination finale.

La 2^{ème} condition est la création des interfaces ou bien des maps qui correspondent aux endroits voulus où l'utilisateur pourra rechercher des destinations souhaitées. Une galerie contiendra une liste de cartes ou maps souhaitées.

3.3 conception du système

3.3.1 Les exigences de la conception

La conception comprend trois sous-systèmes. Ces systèmes et leurs fonctionnalités sont résumées dans le tableau 3.1.

Positionnement	La localisation de l'utilisateur par rapport à son point de départ.
Mapping	Faire le mapping entre la position calculée et son endroit réel dans la map (carte).
Interface Utilisateur	Permettre l'accès aux utilisateurs à toutes les fonctionnalités disponibles.

TABLE 3.1: Les exigences de la conception comprenant les trois sous-systèmes et leurs descriptions

3.3.2 Architecture générale du système

L'architecture générale du système est résumée dans le schéma qui suit

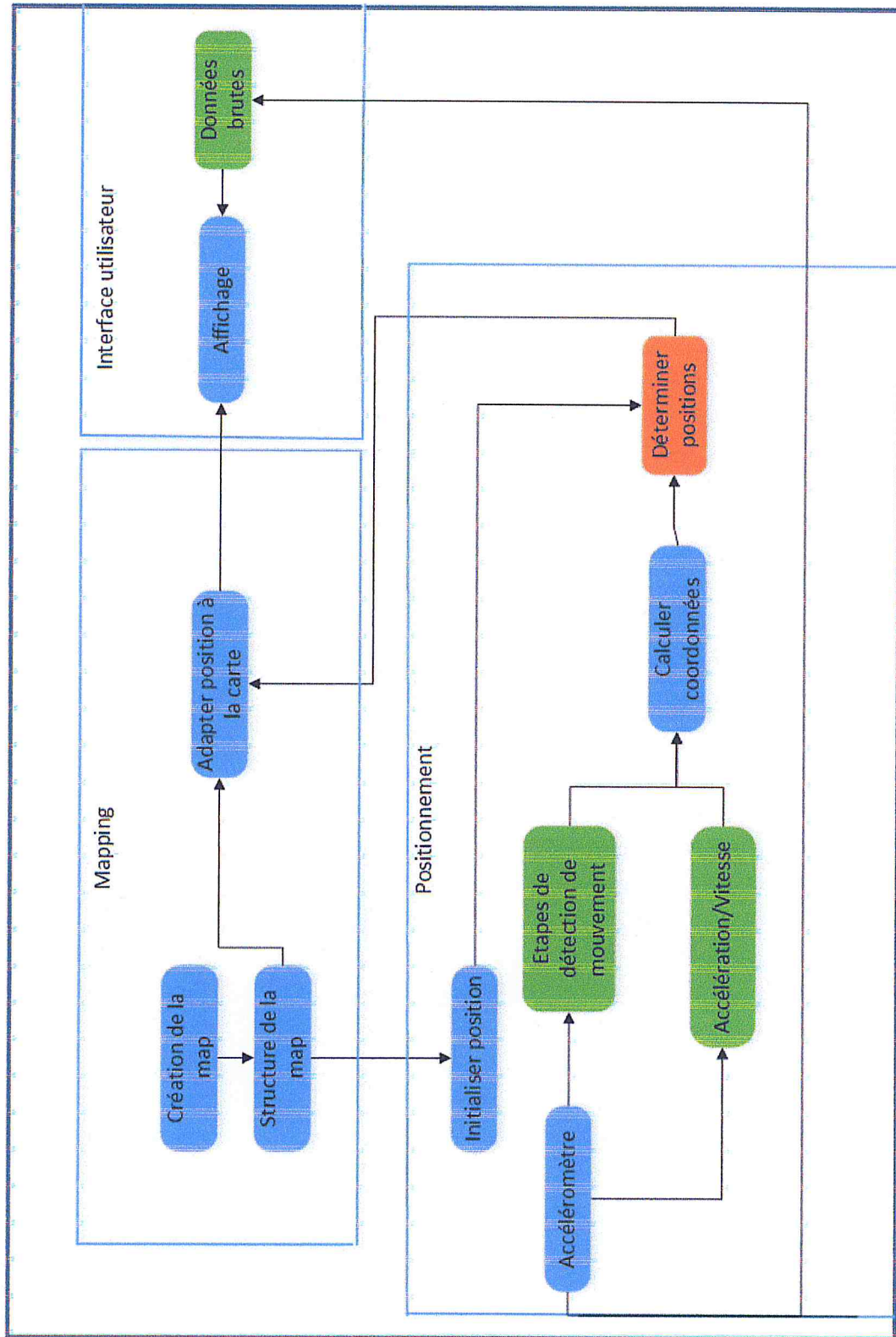


FIGURE 3.1: Architecture générale du système

La Figure ci-dessus montre la structure de l'application à réaliser. Elle est structurée en trois sous-systèmes. Le sous-système de traitement principale appelé positionnement, s'occupe de la lecture des valeurs données par les capteurs (accéléromètres, gyroscope, et magnétomètre). Dans le cadre de notre projet, nous avons utilisé seulement l'accéléromètre pour la réalisation des deux scénarios vu précédemment. Le gyroscope et le magnétomètre n'ont pas été nécessaires dans leur réalisation.

Le sous-système mapping traite les cartes, leurs structures, et adapte la position estimée sur la carte. L'interface utilisateur affiche les informations traitées par le sous-système positionnement, et ainsi affiche la position de l'utilisateur.

3.3.3 Conception détaillée du système

3.3.3.1 Fonctionnalité de l'application du premier scénario

Le premier scénario utilise les fonctionnalités suivantes :

Affichage des accélérations brutes

Il consiste à afficher les valeurs de l'accélération pour analyser le comportement de l'accéléromètre.

Paramètres

Ils consistent à introduire la vitesse, le point de départ et la direction comme paramètres avant de commencer la navigation.

Map

L'utilisateur peut choisir la carte ou la map à utiliser pour la charger dans l'application. Pour ce premier scénario nous avons créé une carte de la bibliothèque du centre de recherche C.E.R.I.S.T.

Calibrage

Avant l'utilisation des données de l'accélération, cette dernière doit être calibrée (voir définition calibrage chapitre 2).

Detection de mouvement

C'est l'étape où nous devons distinguer l'état de marche et l'état d'arrêt à l'aide des valeurs de l'accélération calibrée.

Calcul de position

C'est une fonction où on calcule la position en intégrant la vitesse paramétrée au début.

Affichage de la position sur la map

Une fois que l'utilisateur démarre la marche en direction linéaire, la trajectoire commence à se dessiner jusqu'à l'arrêt de l'application.

La figure qui suit est un résumé des états de l'application du premier scénario

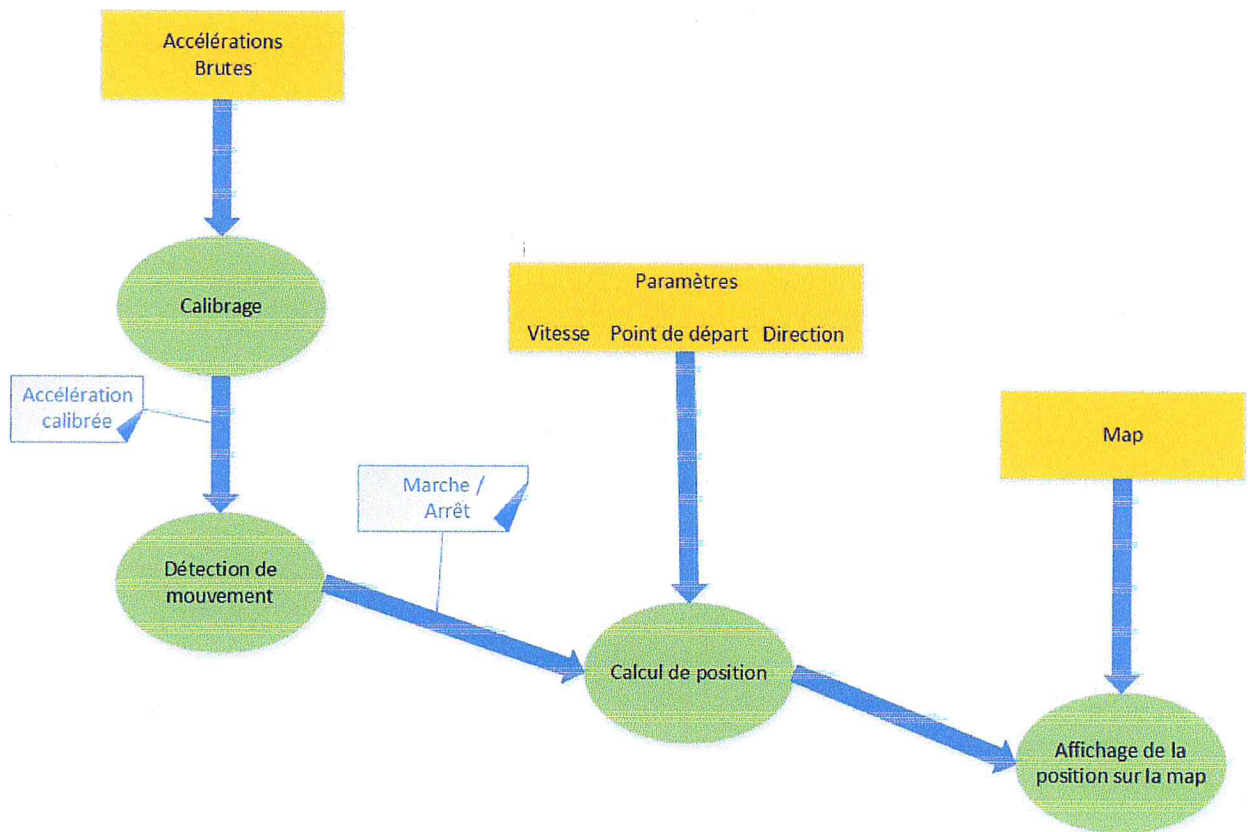


FIGURE 3.2: Le cycle de l'application du 1^{er} scénario

3.3.3.2 Diagramme du cas d'utilisation du premier scénario

La Figure 3.3 présente le diagramme du cas d'utilisation du premier scénario. Nous avons un seul acteur qui est l'utilisateur (celui qui utilise le Smartphone). Il commence par paramétrer la vitesse, ensuite le pas sera calculé automatiquement. Par la

suite la distance sera calculée. Après on sélectionne une map ou carte pour faire enfin mapper la position de l'utilisateur à la carte.

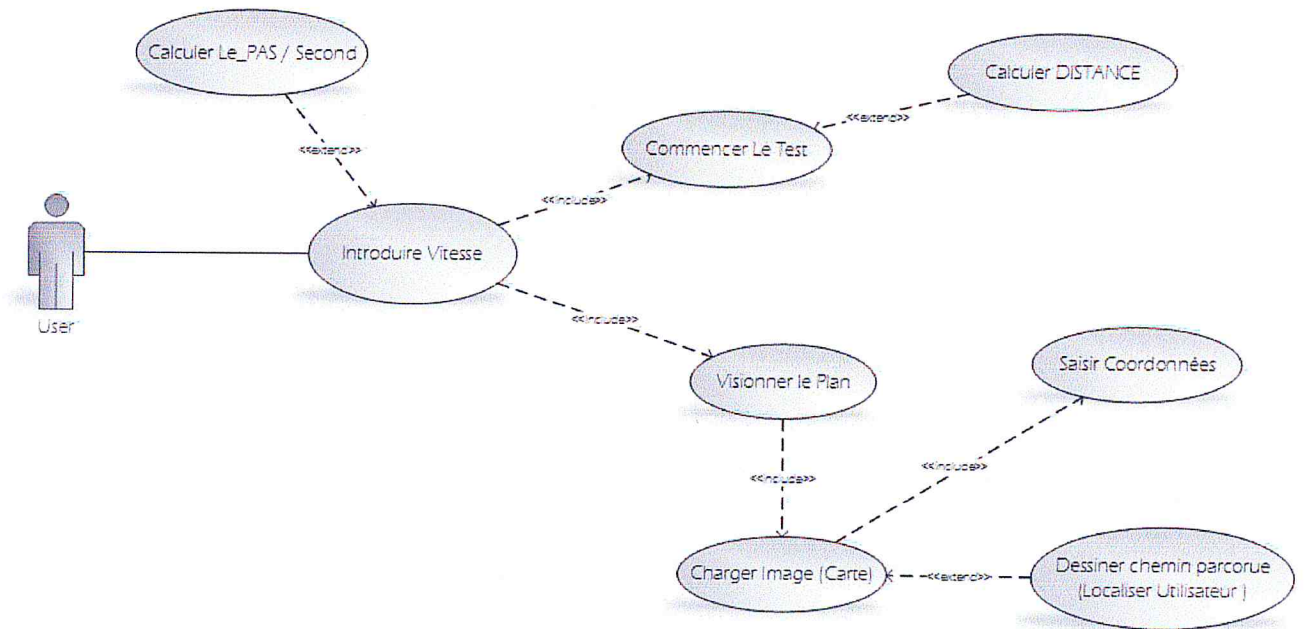


FIGURE 3.3: Diagramme du cas d'utilisation du 1^{er} scénario

3.3.3.3 Diagramme de classe du premier scénario

Le diagramme de classe du premier scenario contiendra trois classes :

- **Capteurs** : qui inclut tous les types de capteurs dont on a besoin. Elle aura `id_Capteur` comme identifiant, et aura pour méthode `onSensorChanged ()` qui retourne les valeurs de l'accélération, et la méthode `Calibrer ()` pour le calibrage du capteur.
- **Positionnement** : C'est la classe qui s'occupe du calcul de la position. Elle a comme attributs, Etat, Temps, Accélération, Vitesse et Distance. L'attribut Etat représente l'état de la personne mobile (Arrêt/ marche) et permet d'estimer le temps de marche et le temps d'arrêt par le biais de la méthode `CalculTemps()`. La valeur de l'attribut état est décidée selon l'accélération mesurée par le biais de la méthode `CalculPosition()`.
- **Vue** : contient les layouts nécessaires pour le scénario et les cartes pour l'affichage de la position de l'utilisateur. Elle possède deux méthodes ; `onDraw ()` et `ArrièrePlan`

().

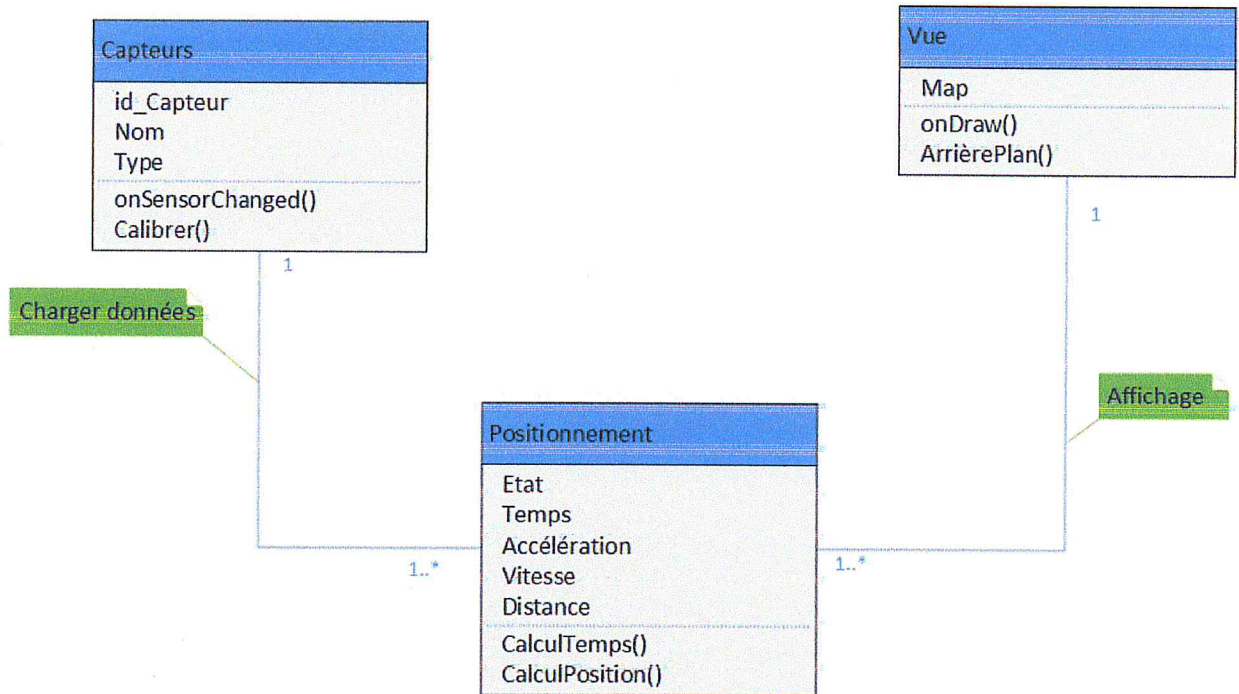


FIGURE 3.4: Diagramme de classe du 1^{er} scénario

3.3.3.4 Diagramme d'activité du premier scénario

La Figure 3.5 représente le diagramme d'activité du premier scénario.

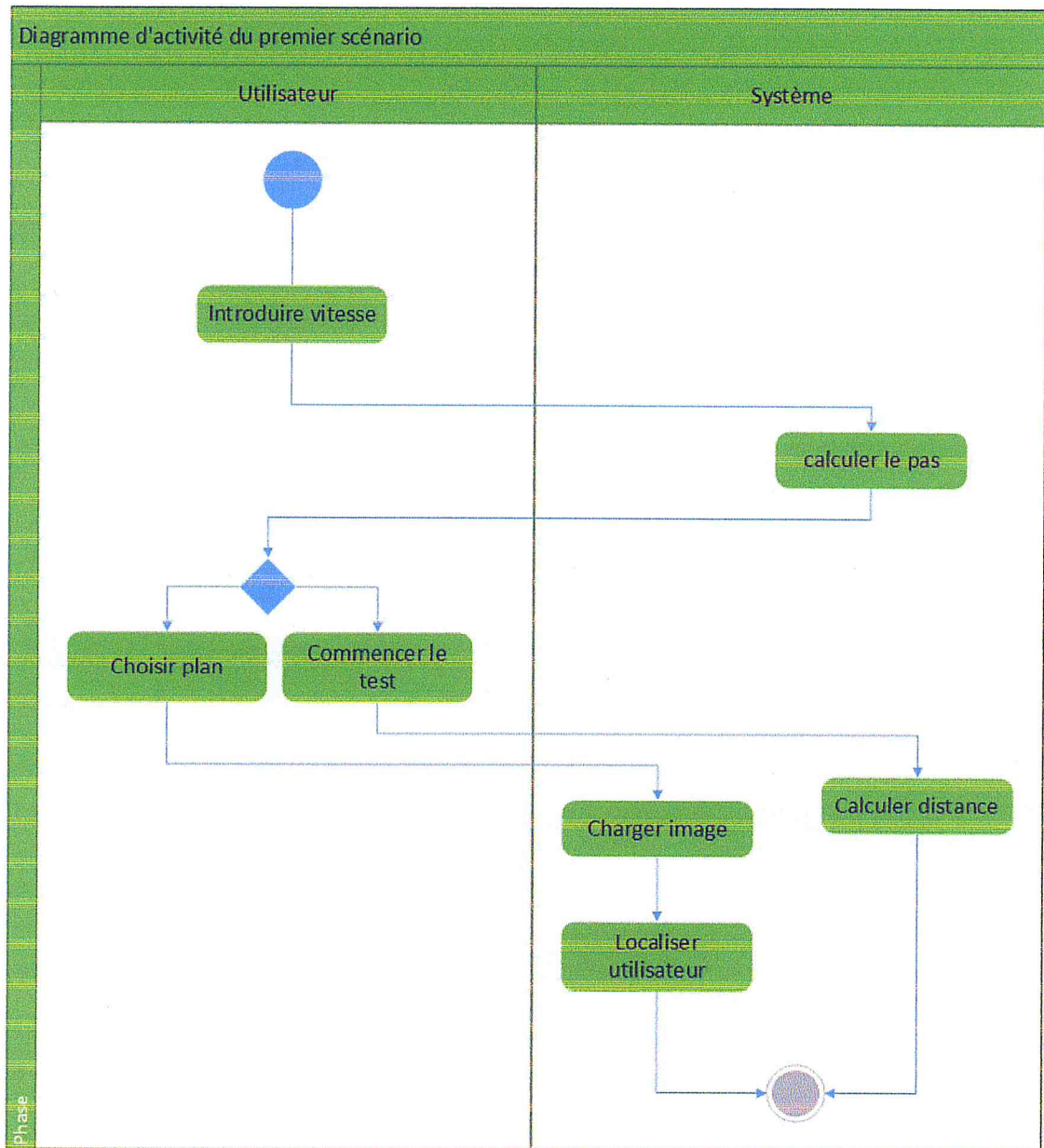


FIGURE 3.5: Diagramme d'activité du 1^{er} scénario

3.3.3.5 Fonctionnalité de l'application du deuxième scénario

Les fonctionnalités nécessaires pour mettre en place le deuxième scénario sont résumées dans les points suivants :

Affichage des accélérations

Cette fonctionnalité permet de récupérer et afficher les valeurs retournées par l'accéléromètre afin de les analyser.

Paramètres

Dans ce scénario seules le point de départ et la direction seront pris comme paramètres avant de commencer le scénario.

Map

La même map sera utilisée que celle qui a été utilisée dans le premier scénario, c.-à-d. le plan de la bibliothèque du centre de recherche C.E.R.I.S.T. Notons que différentes map pourront être utilisées dans notre application.

Filtrage avec le passe haut

Cette étape consistera à utiliser le filtre passe haut qui élimine l'effet de gravité et donc calculera l'accélération linéaire.

Filtrage avec le RII (Réponse Impulsionnelle Infinie)

C'est la seconde étape du filtrage, on utilisera le filtre à réponse impulsionnelle infinie RII qui filtrera les bruits dûs aux mouvements, et ainsi donner les valeurs exactes de l'accélération.

Calcul de vitesse

Cette fonction consiste à calculer la vitesse en intégrant l'accélération filtrée.

Calcul de position

Elle consiste à calculer la distance parcourue en intégrant la vitesse résultante du calcul précédent.

Affichage de la position sur la map

Une fois que l'utilisateur commence à marcher en direction linéaire, sa trajectoire sera tracée par l'application.

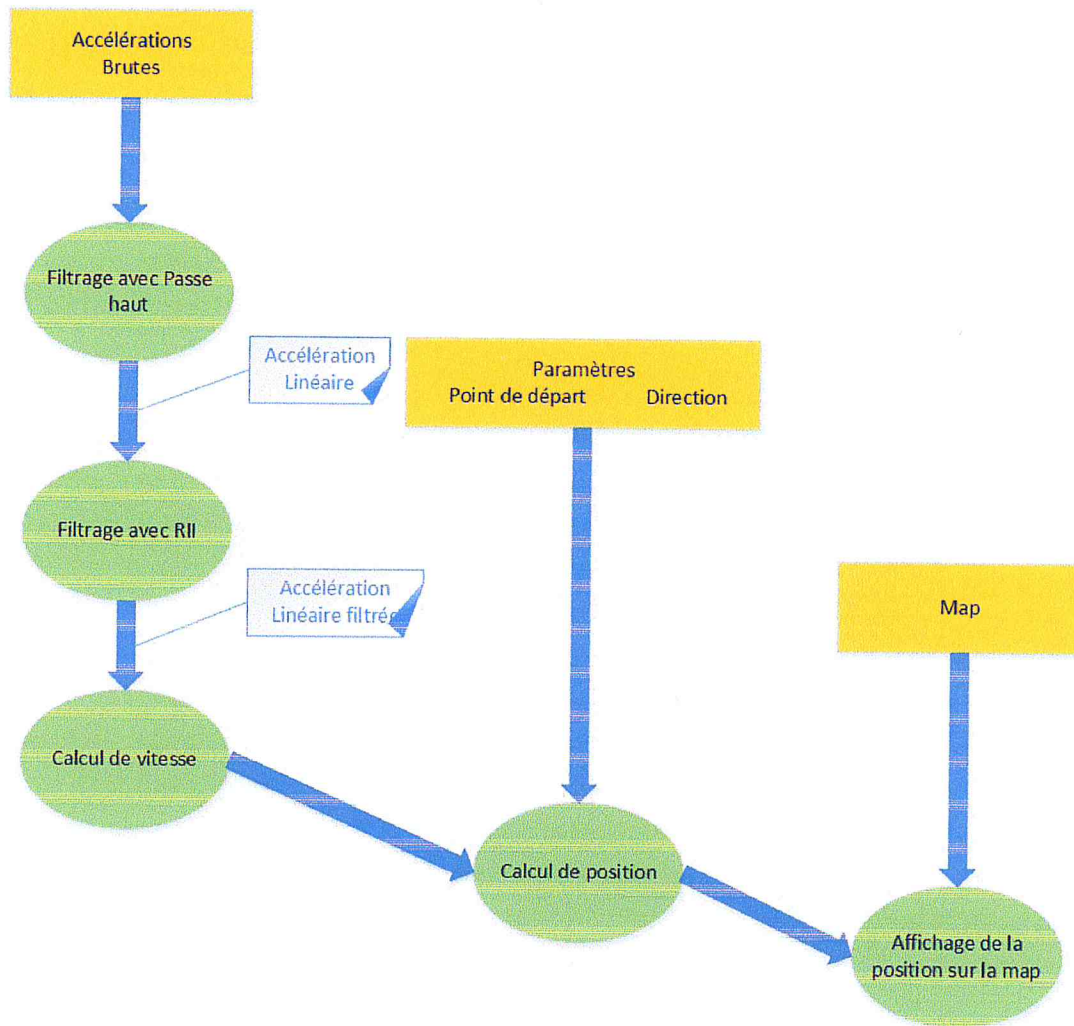


FIGURE 3.6: Cycle de l'application du 2^{ème} scénario

3.3.3.6 Diagramme du cas d'utilisation du deuxième scénario

Le diagramme de cas d'utilisation inclut un seul acteur qui est l'utilisateur, quand il lance l'application, il choisit la map qui sera chargée. Une fois le teste commence, la vitesse sera calculée, ensuite la distance, et enfin mapper la position sur la carte. La Figure qui suit représente le diagramme du cas d'utilisation du deuxième scénario.

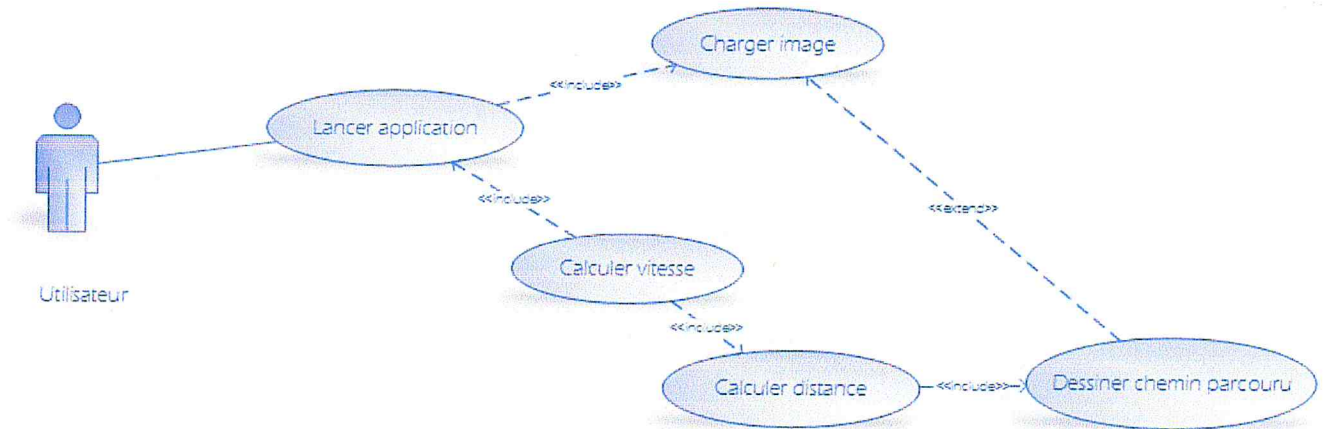


FIGURE 3.7: Le diagramme de cas d'utilisation du 2^{ème} scénario

3.3.3.7 Diagramme de classe du deuxième scénario

Le diagramme de classe du deuxième scénario contiendra trois classes :

- **Capteurs** : qui inclus tous les types de capteurs dont on a besoin. Elle aura `id_Capteur` comme identifiant, et aura pour méthode `onSensorChanged()` qui retourne les valeurs de l'accélération, et la méthode `AccélérationLinéaire` pour filtrer l'accélération brute avec le filtre passe haut.
- **Positionnement** : C'est la classe qui s'occupe du calcul de la position. Elle a comme attributs, `Etat`, `Temps`, `Accél_lin`, `Vitesse` et `Distance`. L'attribut `Etat` représente l'état de la personne mobile (Arrêt/ marche). Elle aura pour méthode `AccélérationFiltre()` qui filtre l'accélération linéaire. Pour le calcul de la vitesse, `CalculVitesse()` et `CalculDistance()`.
- **Vue** : contient les layouts nécessaires pour le scénario et les cartes pour l'affichage de la position de l'utilisateur. Elle possède deux méthodes ; `onDraw()` et `ArrièrePlan()`.

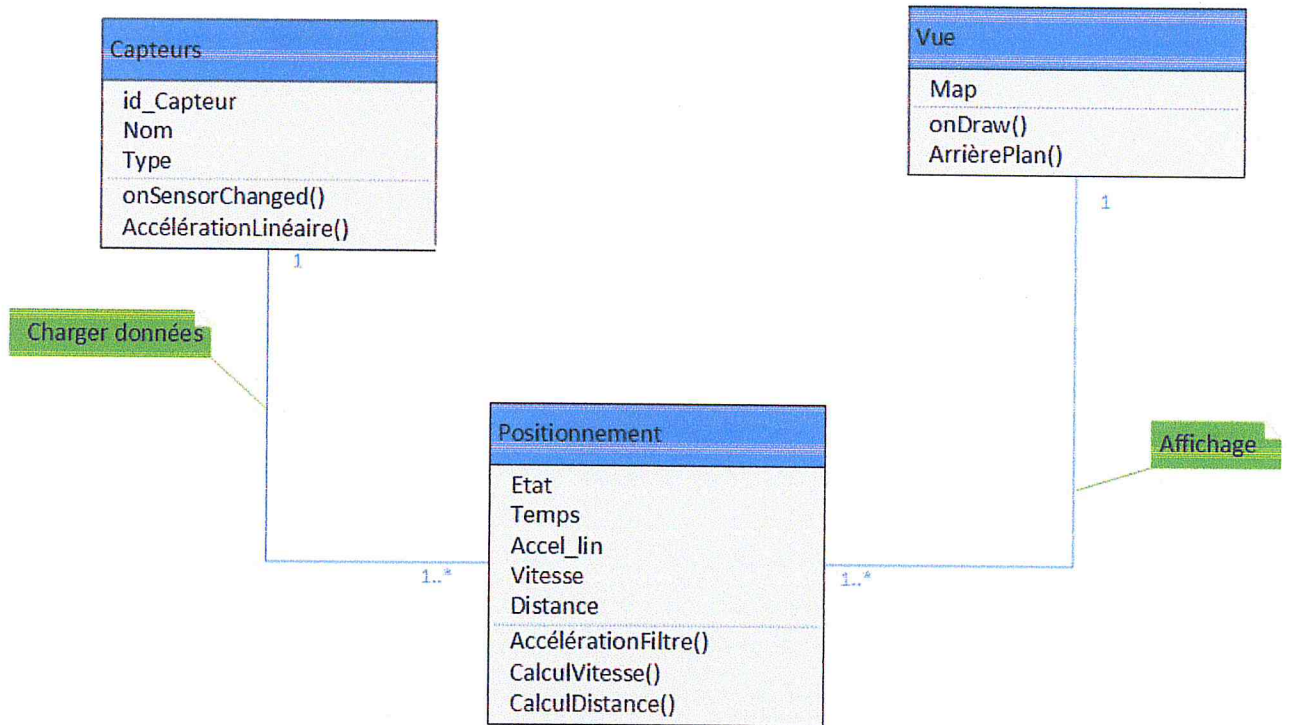


FIGURE 3.8: Diagramme de classe du deuxième scénario

3.3.3.8 Diagramme d'activité du deuxième scénario

La Figure 3.9 représente le diagramme d'activité du deuxième scénario.

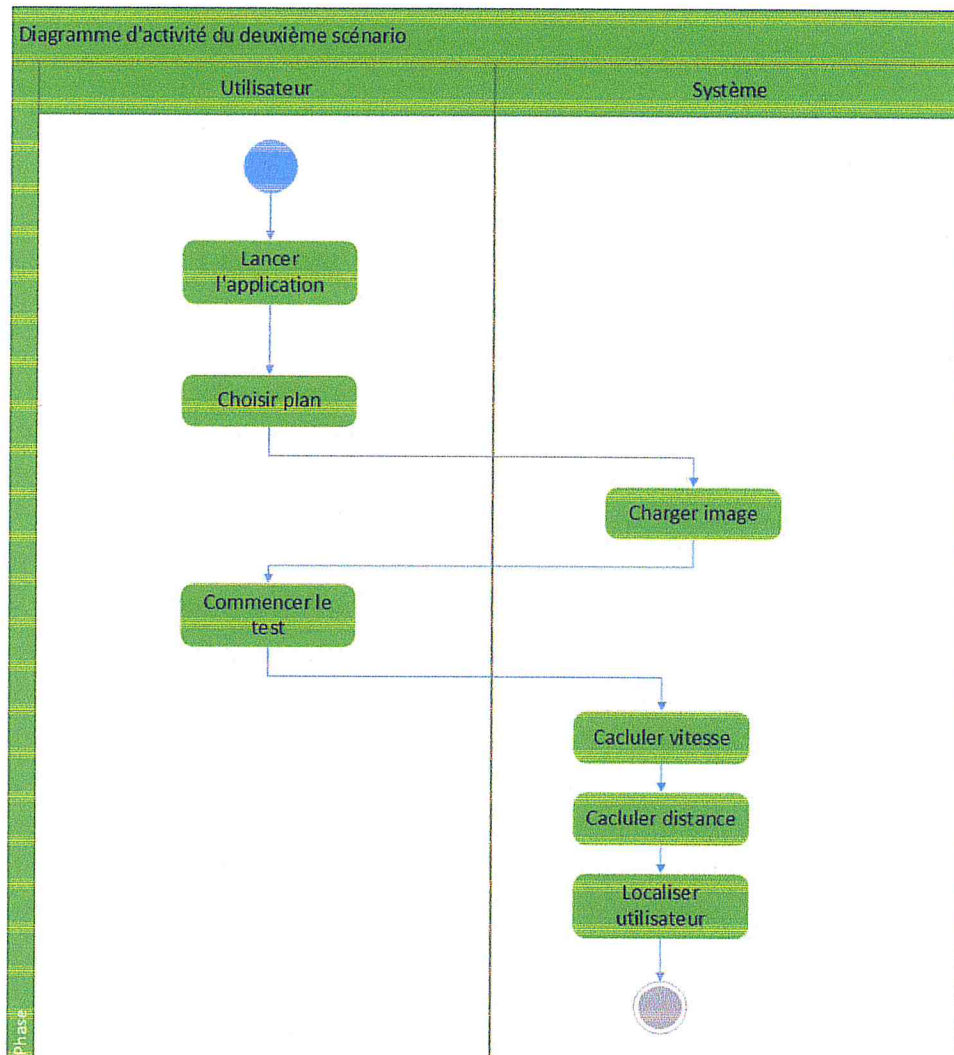


FIGURE 3.9: Diagramme d'activité du deuxième scénario

3.4 La création de la map

La création des maps ou cartes consiste à recueillir des informations qui décrivent la structure de l'endroit voulu pour l'estimation de la position de l'utilisateur. Ces informations seront converties sous une forme utilisable par d'autres procédés. Pour notre cas nous avons créé une map de la bibliothèque du centre de recherche C.E.R.I.S.T vu que nos recherches et essais se sont faites dans ce centre. Cette bibliothèque de compose de tables alignées et de six piliers. Ces derniers ont été mis sur la carte en précisant

leurs dimensions exactes en réel. La map est sous le format JPEG, l'application propose alors de faire la conversion de dimension de la bibliothèque du mètre au pixel. La Figure suivante représente une photo réelle de la bibliothèque du C.E.R.I.S.T.



FIGURE 3.10: Photo réelle de la bibliothèque du C.E.R.I.S.T

L'image qui correspond à la photo réelle de la bibliothèque du C.E.R.I.S.T, et qui sera chargée dans le téléphone, est présentée dans la Figure suivante :

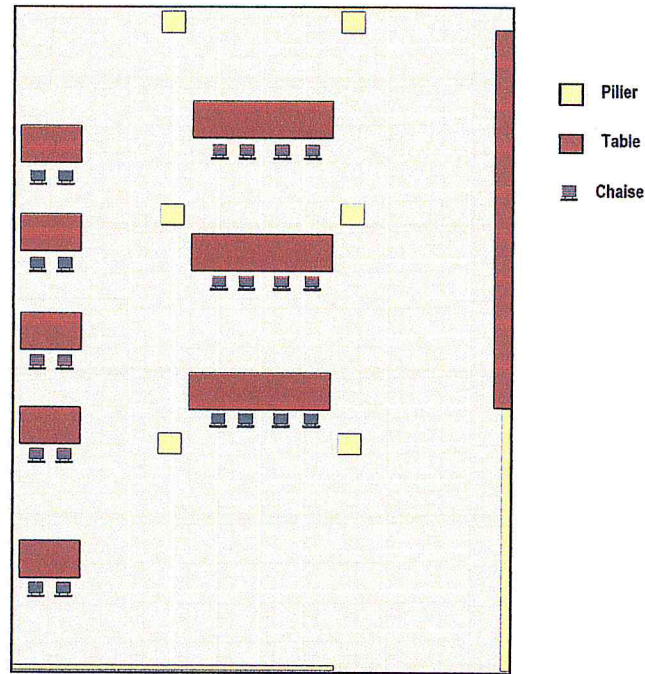


FIGURE 3.11: La map de la bibliothèque du C.E.R.I.S.T

3.5 Conclusion

Dans ce chapitre nous avons présenté la conception de notre application. Cette conception a été faite à l'aide d'une présentation détaillée du diagramme du cas d'utilisation, de classes et d'activités. Par la suite nous montrerons l'implémentation détaillée de notre application qui était faite à la base de cette conception, ainsi que les outils nécessaires pour sa réalisation.

Chapitre 4

Implémentation

4.1 Introduction

Dans ce chapitre, nous présentons les différents systèmes d'exploitations existants dans le marché des téléphones mobiles intelligents Smartphones ; les raisons du choix du système pour le projet ; suivi d'une étude approfondie de ce dernier. De plus, une présentation des différents outils utilisés pour le développement de ce projet, ainsi que les résultats obtenus.

4.2 Systèmes d'exploitation mobiles

Un système d'exploitation mobile est une plateforme logicielle sur laquelle les autres programmes appelés «programmes d'applications» peuvent s'exécuter sur des appareils mobiles tels que les PDA (personnal digital assistant) [1], les téléphones cellulaires, Smartphones, . . . etc. Parmi les systèmes d'exploitation nous avons :

4.2.1 ios (iPhone OS)

Anciennement appelé «iPhone OS», est un système d'exploitation conçu par Apple, qui est dérivé de Mac OS X. Il fonctionne sur iPhone, iPod Touch ou iPad et a pris une part significative du marché [1].

4.2.2 Windows mobile

développé par Microsoft pour les Smartphones et Pocket PC. Il propose des applications basiques comme email, Internet, chat et multimédia, etc., les applications tierces développées avec .NET Compact Framework sont également supportées [1].

Le 15 février 2010 Microsoft a lancé un nouveau système d'exploitation pour mobile, Windows Phone 7. Il intègre des fonctionnalités média sociaux tel Facebook et Twitter. Et en 2012 Microsoft a lancé le Windows Phone 8.

4.2.3 Android

Android est développé par l'Open Handset Alliance, il a été annoncé en 2007. C'est par ailleurs un système « ouvert », contrairement à ses deux concurrents principaux Microsoft et Apple. C'est un point extrêmement important car cela signifie que la communauté de développeurs qui contribue à son développement est naturellement plus disposée à investir sur un système ouvert que sur un système rigide et soumis à des cahiers des charges trop contraignants [1].

4.2.4 Symbian OS

Est un système d'exploitation plus utilisé pour des Smartphones et PDA, qui a été conçu par Symbian Ltd. Symbian OS fournit les fonctionnalités essentielles du système d'exploitation, notamment le cœur du système, ainsi que les API communes et une interface utilisateur de référence. Il a été adopté par différents fabricants de téléphones portables de 2G et 3G (Nokia, Sony Ericsson, Motorola, Samsung, etc.). Les principales interfaces utilisateur sont S60 série (pour clavier numérique), UIQ (pour écran tactile), S80 série (pour clavier alphanumérique), etc. [1].

4.2.5 BlackBerry OS

Est un système d'exploitation qui fonctionne sur le Smartphone BlackBerry. Il permet aux développeurs de mettre en place des applications en utilisant les APIs BlackBerry, mais toute application doit être signée numériquement par le compte RIM du développeur [1].

La Figure ci-dessous représente les parts du marché des OS mobile dans le monde en 2012 et 2013

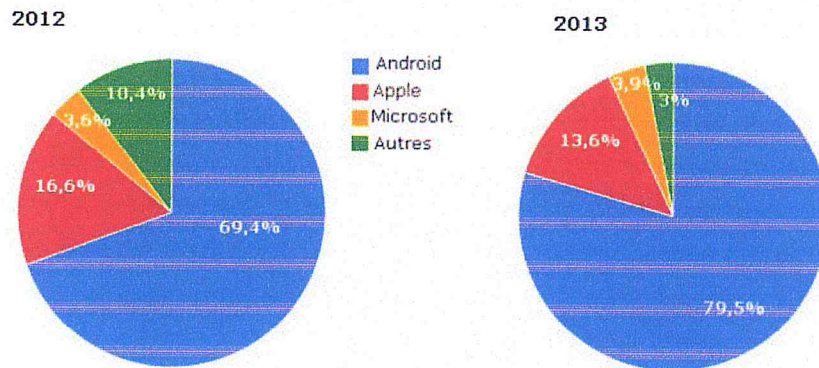


FIGURE 4.1: Parts du marché des Os mobile dans le monde en 2012 et 2013 [18]

D'après cette Figure, nous remarquons qu'Android prend la plus grande place du marché, même s'il est majoritaire, il doit en effet faire face à la concurrence d'Apple, son principal adversaire, avec son ios pour les iPhones. De plus, Google doit aussi compter sur la concurrence de RIM (Blackberry), qui reste très implantée dans les entreprises, et Microsoft avec Windows Mobile, qui espère avec Windows Mobile 7, gagner quelques parts de marché. Quant à Nokia et son système Symbian, il est en nette perte de vitesse car il n'a pas beaucoup évolué ces dernières années .

Notre choix du système d'exploitation mobile pour le développement de notre application a été basé sur l'étude qui faite précédemment de ces plusieurs systèmes d'exploitation, et la plateforme Android a été privilégiée pour les raisons que nous allons voir dans la prochaine partie.

4.3 Raisons du choix d'Android

Précédemment, nous avons cité que la plateforme Android est notre environnement de développement pour notre application, et pour cela nous allons expliquer les raisons de ce choix :

- Android est une nouvelle plateforme en code source ouverte.
- Android est une plateforme puissante, moderne, sûre et ouverte.

- Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Les utilisateurs peuvent adapter les applications à leur besoin.

4.4 Présentation de la plateforme Android

Android est un OS pour téléphone mobile et tablette tactile, promu par Google et l'Open Handset Alliance qui comprend plus de 35 constructeurs, fournisseurs de logiciel, et Opérateurs. Il concurrence des plateformes telles que l'iOS d'Apple, Windows Mobile de Microsoft, RIM OS intégré dans les BlackBerry de Research In Motion, ou encore Symbian et MeeGo de Nokia. La plateforme Android est un OS basé sur GNU/Linux entièrement gratuit, sous licence open source Apache 2. Le kit de développement (SDK) et le code source d'Android sont disponibles depuis novembre 2007 en version 1.0. La version actuelle des sources est la Jelly bean 4.2.2 qui est sortie en 11 février 2013. Android a une très grande communauté de développeurs qui produisent des applications diverses et variées pour étendre les fonctionnalités du système d'exploitation [1]. Il y a actuellement plus de 100 000 applications sur l'Android Market, ce qui en fait le deuxième environnement de développement le plus populaire, derrière iOS. Les applications sont écrites pour la plupart en Java, et peuvent utiliser le hardware et les fonctionnalités du système via des bibliothèques Java développées par Google. Le fait que le code soit complètement ouvert, permet à de nombreux développeurs de le modifier pour y ajouter des fonctionnalités ou corriger des bugs, et ainsi de proposer à la communauté des builds personnalisés, souvent plus avancés que les versions officielles proposées par les constructeurs.

4.4.1 L'historique d'Android

En juillet 2005, Google a acquis Android, Inc., une petite startup qui développait des applications pour téléphones mobiles. C'est à ce moment là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'avait des données sûres à propos des marchés dans lesquels ils allaient se positionner. Après ce rachat fait par Google, une équipe dirigée par Andy Rubin, un ancien d'Android Inc, a

commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA soit créé officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que Google développait des applications mobiles de son moteur de recherche, qu'elle développait un nouveau téléphone mobile, etc [1].

En 2007, le 5 novembre, l'OHA a été officiellement annoncée, ainsi que son but : développer des standards open sources pour appareil mobile. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6.

En octobre 2008, apparaît la première version d'Android qui n'avait pas reçu de nom. Cette version s'est avérée être la β du système. La version 1.5 Cupcake corrigea le manque d'API et rendit le système plus utilisable. Depuis, *Android 1.6*, *2.0* et *2.1* ont apporté d'importantes améliorations respectivement sur les fonctionnalités et sur l'interface graphique du système. *Android 2.2 Froyo* a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. Aussi, Google annonce que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8.

Android 3.0 Honeycomb est spécialement étudié pour les tablettes tactiles. Les premiers modèles devraient être annoncés au CES 2011. On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont sûrement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.

Par la suite, le 19 octobre 2011 *Android 4.0 Ice Cream Sandwich* a été lancé qui crée le premier terminal embarquant le 4.0, ainsi que la réunification de l'interface pour les Smartphones et les tablettes. Il ajoute aussi les supports d'écrans 720p (HD), ainsi que l'amélioration de la reconnaissance vocale.

Vient après en juin 2012 *Android 4.1 jelly Bean* qui rapporte de nouvelles fonctionnalités aux notifications, améliore aussi la fluidité des animations dans l'interface et ajoute aussi de nouvelles possibilités à Android Beam (NFC).

Android 4.2 devrait arriver très vite pour rajouter encore plus de fonctionnalités aux terminaux. Pour le développement, ces nouvelles versions d'Android devraient proposer de nouveaux composants permettant de réaliser des applications avec une ergonomie plus adaptée aux tablettes tactiles. La Figure qui suit une représentation des versions

d'Android à travers le temps [19].

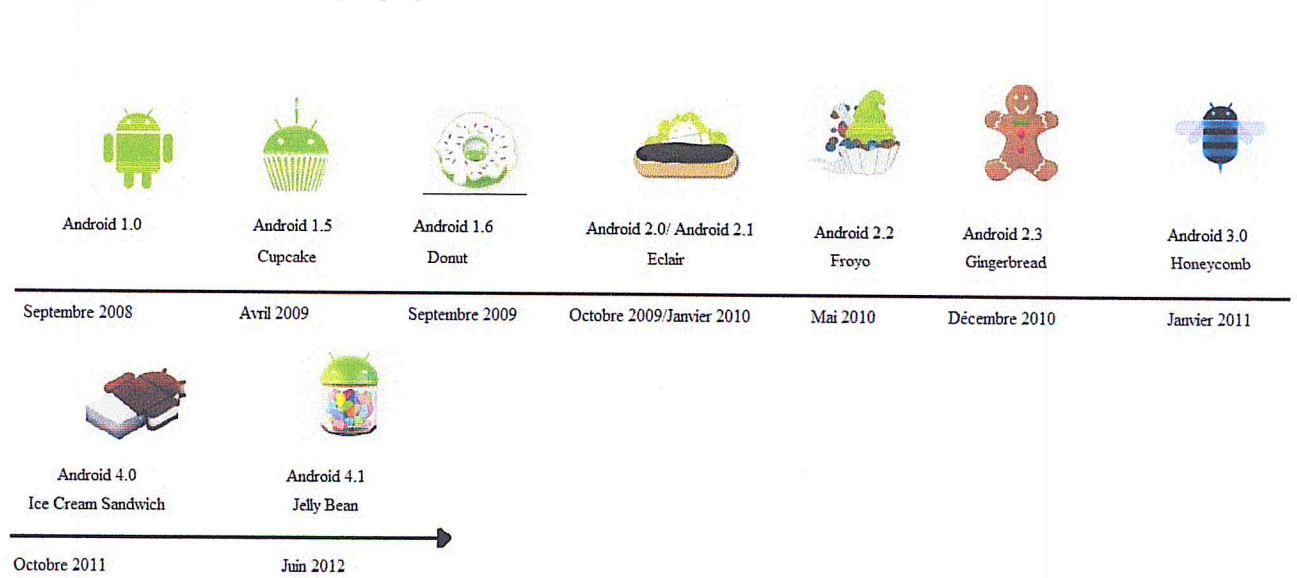


FIGURE 4.2: Evolution des versions d'Android

4.4.2 L'architecture d'Android

Le diagramme suivant illustre les composants principaux du *système d'exploitation Android*

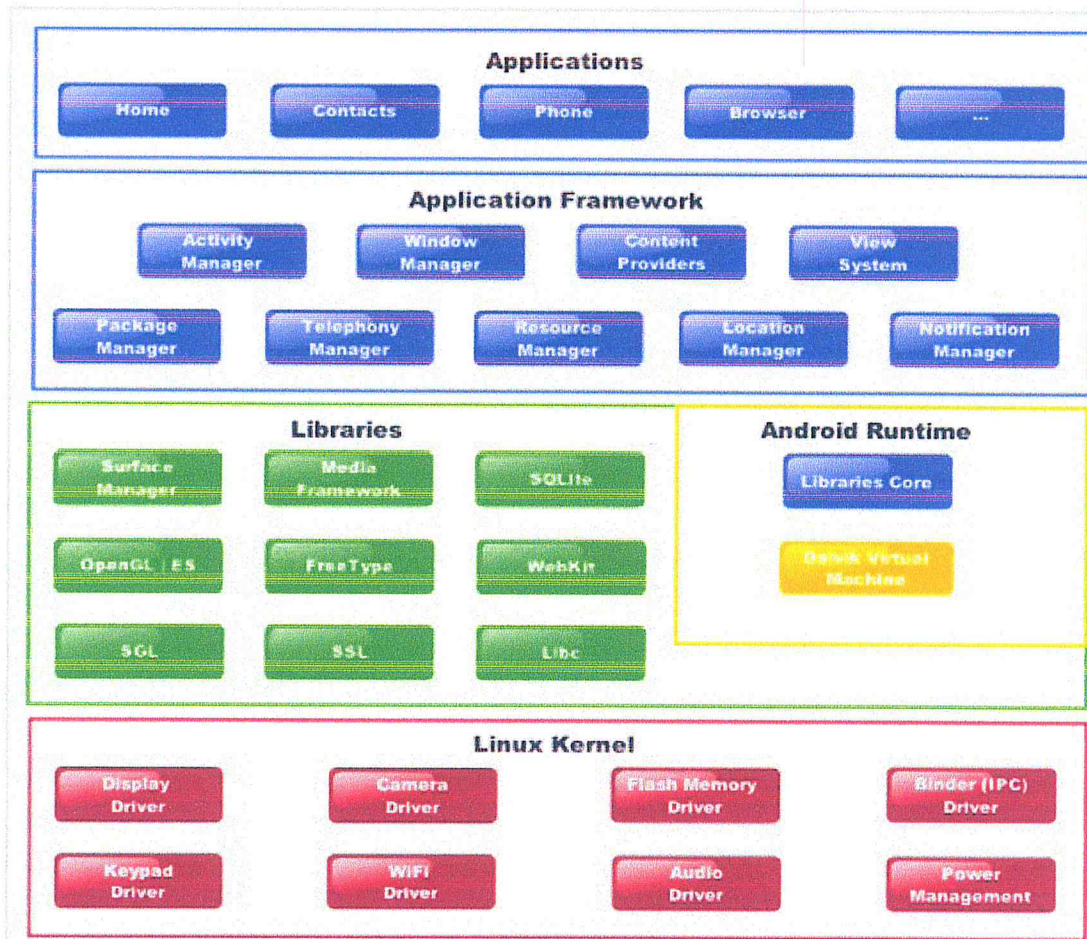


FIGURE 4.3: L'architecture de la plateforme Android [22]

Android se base sur un noyau Linux 2.6. Le SDK Android possède une bibliothèque de librairie de plusieurs classes java de base pour plusieurs types d'application (exemple : OpenGL|ES pour la 3D, SSL pour les protocoles de sécurité, etc.). Une application Android se repose sur un Framework qui facilite l'utilisation des classes de base et sert d'interface entre les "Librairies" et les applications.

4.4.3 Le cycle de vie d'une activité

Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faci-

litér la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité.

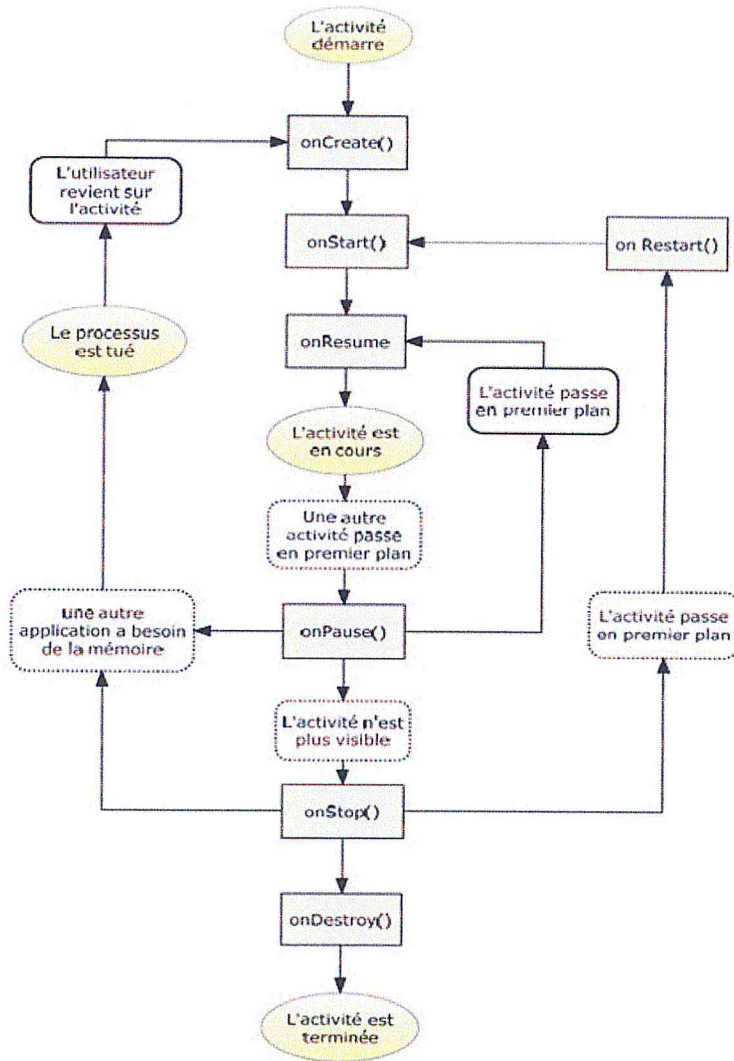


FIGURE 4.4: Le cycle de vie d'une activité [24]

Quand l'activité démarre : la méthode *onCreate* est appelée. Pendant l'utilisation d'une activité, l'utilisateur presse la touche Accueil, ou bien l'application téléphone, qualifiée comme prioritaire et qui interrompt son fonctionnement par un appel téléphonique entrant. L'activité est arrêtée (appel de *onStop*), le développeur détermine l'impact sur l'interface utilisateur, par exemple la mise en pause d'une animation puisque

l'activité n'est plus visible. Une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause (appel de *onRestart*, *onStart*). L'activité reste trop longtemps en pause, le système a besoin de mémoire, il détruit l'activité (appel de *onDestroy*). *onPause* et *onResume* rajoutent un état à l'activité, puisqu'ils interviennent dans le cas d'activités partiellement visibles, mais qui n'ont pas le focus. La méthode *onPause* implique également que la vie de cette application n'est plus une priorité pour le système. Donc si celui-ci a besoin de mémoire, l'Activité peut être fermée. Ainsi, il est préférable, lorsque l'on utilise cette méthode, de sauvegarder l'état de l'activité dans le cas où l'utilisateur souhaiterait y revenir avec la touche Accueil [24].

4.5 Outils de réalisation du projet (Environnement technique)

4.5.1 Environnement de développement

4.5.2 Eclipse

Notre projet est implémenté en Java, l'environnement le plus adéquat à notre application est l'Eclipse qui est l'Environnement de Développement Intégré ou IDE le plus performant, il est Open source et gratuit. L'Eclipse est libre, extensible, universel et polyvalent. Il permet de créer des projets mettant en œuvre n'importe quel langage de programmation, Il est écrit en Java à l'aide de la bibliothèque graphique SWT d'IBM. Et ce langage grâce à des bibliothèques spécifiques est utilisé pour écrire des extensions. L'architecture de L'IDE Eclipse est totalement développée autour de la notion des plugins (Un plugin est un module qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités). Toutes les fonctionnalités d'Eclipse sont développées en tant que plugins.

Le langage Java d'Eclipse est privilégié pour le développement des applications Androids. Et pour la réalisation de ce fait, Google a conçu un plugin pour Eclipse. L'Eclipse est téléchargeable à partir du site <http://eclipse.org>.

4.5.2.1 Prise en main de l'environnement Android

Pour commencer le travail de développement nous devons appréhender le plugin SDK Android. Comme a été dit précédemment, Google a mis en place un plugin pour Android et un grand nombre d'outils pour encourager le développement d'Android. Pour obtenir le SDK le site <http://Developer.Android.com> est recommandé pour les développeurs Android. Ce site présente toutes les instructions d'installation et d'utilisations des différents outils comme le SDK, NDK etc... , il propose également des tutoriels et articles concernant le développement des applications Android, ainsi que les références des API. La Figure qui suit est le portail des développeurs Android le site cité précédemment.

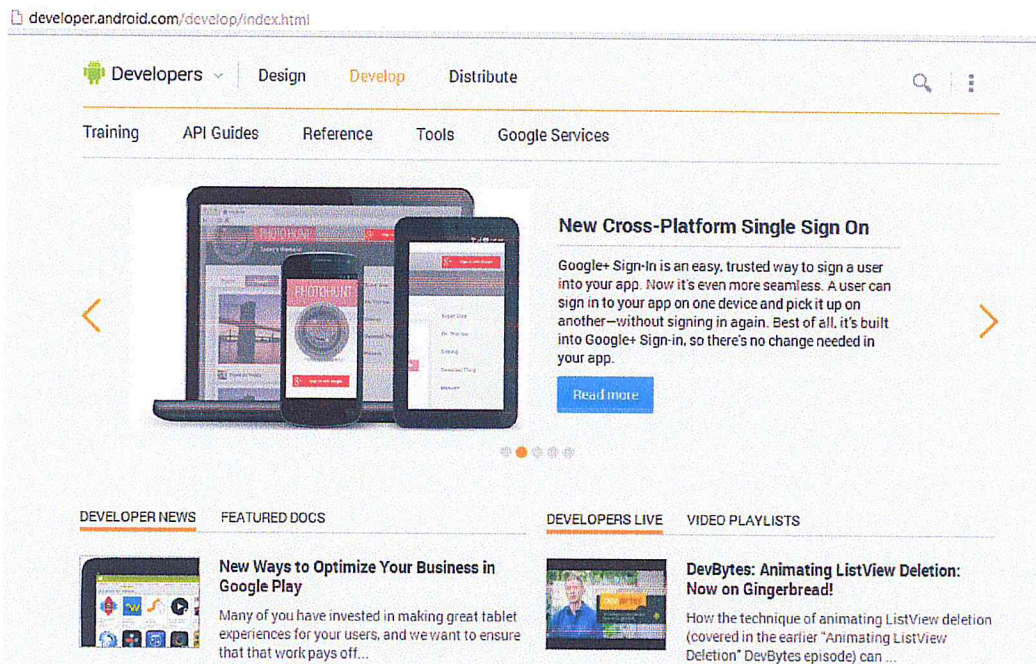


FIGURE 4.5: Le site web des outils nécessaires pour les développeurs Android [25]

4.5.2.2 SDK Android

Le SDK est l'outil le plus important, il permet de télécharger tous les outils indispensables pour le développement des applications, il permet aussi de télécharger toutes les versions d'Android, ainsi que les versions de Google APIs pour intégrer les fonctionnalités liées aux services Google tels que MAP etc. On peut aussi télécharger

de la documentation Java Doc. Son fonctionnement est similaire aux gestionnaires de paquets de linux [26]. La Figure suivante est l'interface du SDK Manager qui nous permet d'installer les différentes versions d'Android ainsi que d'autres outils.

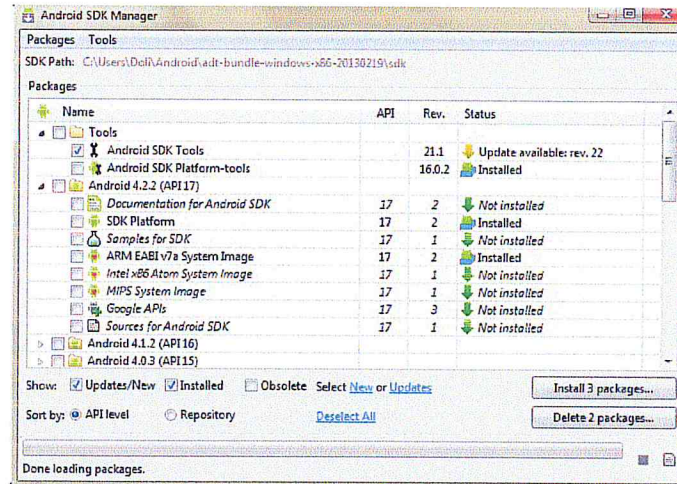


FIGURE 4.6: Interface du SDK Manager

4.5.2.3 ADT pour Eclipse

L'ADT (Android Development tools) est très complets et facile à l'utilisation et à la manipulation de la conception graphique d'interface utilisateur, debug distant sur un téléphone, la gestion de l'architecture de fichier d'une application etc. Dans le cadre de notre projet, nous avons utilisé la dernière version d'Eclipse, Juno et le plugin ADT de Google.

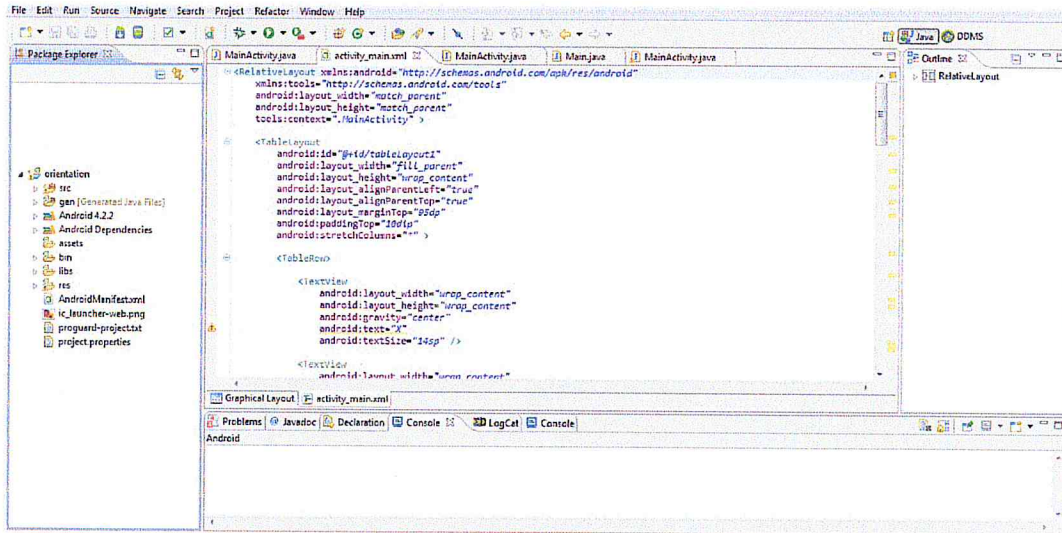


FIGURE 4.7: Interface d'Eclipse incluant l'ADT

4.5.3 Equipement

Pour la réalisation de notre projet, il était nécessaire d'utiliser un Smartphone, ce Smartphone a été le HTC sensation, qui est conçu par HTC corporation, ce téléphone est conçu pour supporter des applications et services basés sur la localisation. Il est muni du système d'exploitation Android 4.0 « Ice Cream Sandwich » OS [23]. Son utilisation va être tout le long du projet pour sa disponibilité de la totalité des capteurs dont nous avons besoin.

Le tableau suivant va présenter quelques caractéristiques du Smartphone HTC sensation

HTC Sensation	
Dimensions	Hauteur : 12.61cm Largeur : 6.54cm Epaisseur : 1.13cm Poids : 148g DAS : 0.358w/kg
Processeur	Qualcomm MSM8260 Snapdragon
Système d'exploitation	Android 4.0.4 Ice Cream Sandwich
Mémoire	Memoire : 1Gb Rom - 768Mb Ram Mémoire extensible
Ecran	4.3" TFT_LCD
Connectivité	Jack 3.5 m USB 2.0 Sortie tv via micrp USB (MHL)
Capteurs	Accéléromètre , Gyroscope, Digital compass, Capteur de proximité, Capteur de lumière.

TABLE 4.1: Caractéristiques du HTC sensation [23]



FIGURE 4.8: HTC sensation Smartphone

4.6 Résultats de la réalisation des scénarios

Ce projet a pour but de faciliter la navigation intérieure (indoor), pour pouvoir réaliser ce projet nous sommes passés par deux étapes chacune étudiait une expérience et chaque expérience dépendait de la précédente. Le projet a regroupé deux scénarios, le premier consiste à calculer la position d'une personne mobile en utilisant un accéléromètre et en paramétrant la vitesse, le point de départ et la direction qui est linéaire au début du scénario. Le deuxième aura le même principe que le premier sauf que seules le point de départ et la direction seront paramétrés et la vitesse sera calculée au long du processus. La fusion de ces deux scénarios nous a donné les résultats suivants :



FIGURE 4.9: La page d'accueil de l'application

La Figure 4.9 représente l'accueil de l'application, il propose une liste de choix des scénarios, c.-à-d. choisir soit le premier scénario de positionnement « détecter l'état du mobile » ou bien le deuxième scénario positionnement avec « Vitesse et position ».

Si l'on choisit le premier, nous aurons les deux interfaces suivantes : la première nous indique d'introduire la vitesse à utiliser tout au long du processus elle est entre 0 et 1

m/s, la deuxième propose un chemin pour atteindre la galerie où l'utilisateur choisit une carte qui correspond à l'endroit voulu.

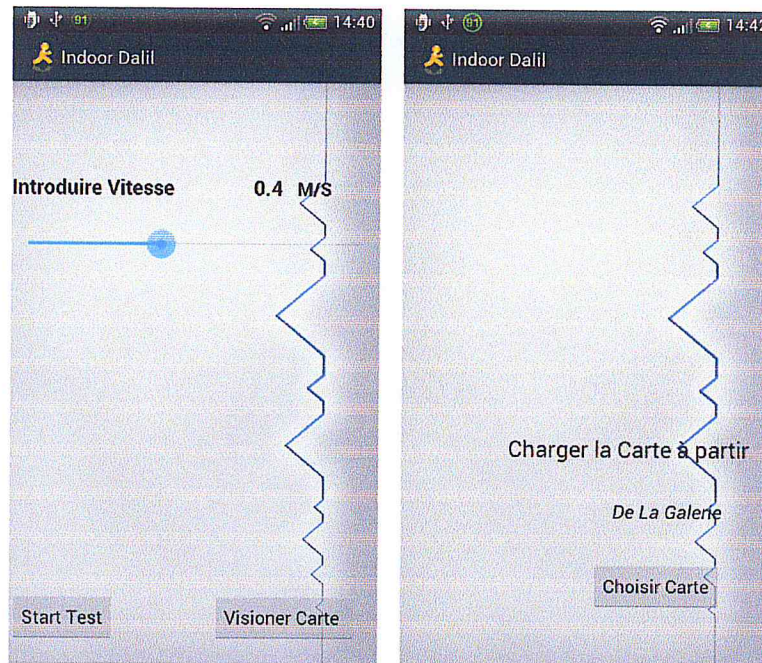


FIGURE 4.10: Paramétrer vitesse et chemin vers la map

Dans ce qui suit on présente la galerie du Smartphone où le dossier des maps a été stockée. Une fois la map chargée nous devons paramétrer la carte en faisant la conversion des distances réelles de l'endroit en pixel et cela en sélectionnant la hauteur de l'endroit voir Figure 4.11.

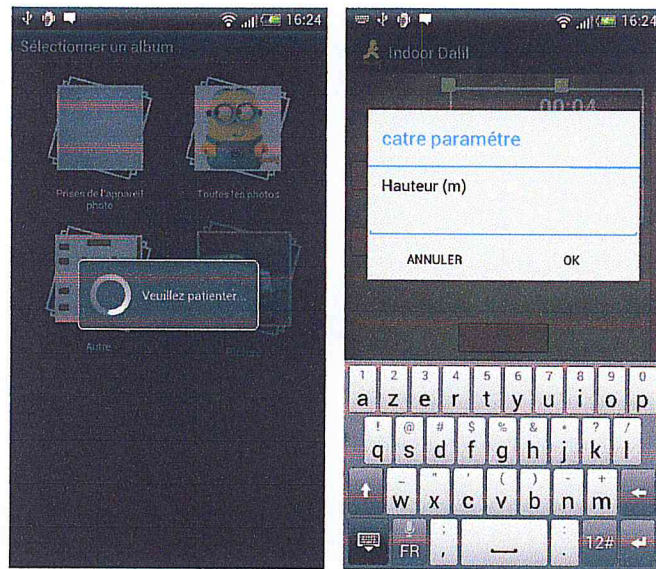


FIGURE 4.11: Galerie où sont stocké les cartes ainsi que l'interface de saisie des dimensions

Ensuite, l'utilisateur aura le choix entre voir les détails du processus qui expriment les temps où l'utilisateur était en marche ou en arrêt ainsi que la distance parcourue voir Figure 4.12.

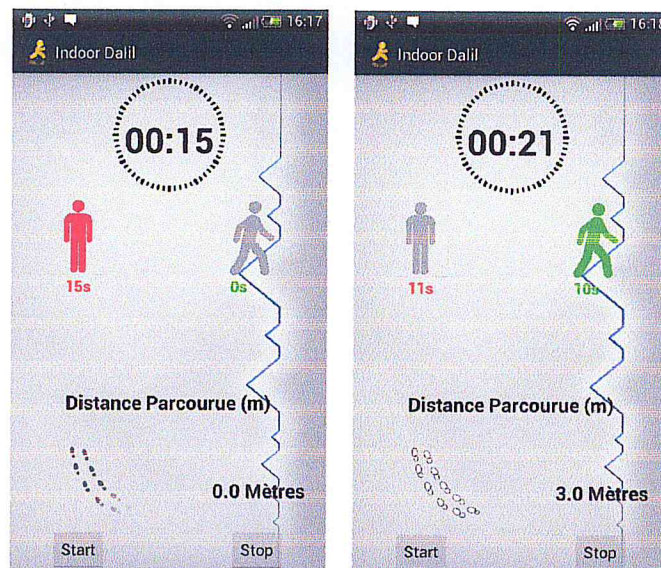


FIGURE 4.12: Les détails de calcul de position du premier scénario

Dans les interfaces précédentes, le bonhomme en vert indique que l'utilisateur est en état de marche et le bonhomme en rouge indique que l'utilisateur est en état d'arrêt. Les pas représentent la distance parcourue. Si l'utilisateur choisit le deuxième scénario c-à-d. vitesse et calcul de position il aura les mêmes interfaces détaillées que celui du premier scénario à l'exception du calcul de la vitesse qui n'est pas introduite au début du processus, voir Figure 4.12.

Enfin l'utilisateur aura la map affichée avec plus de détails, en indiquant sa position sur la carte, ainsi que le temps, la distance, et la vitesse seront affichés en haut à droite. le cercle blanc représente la position initiale de l'utilisateur voir Figure 4.13.

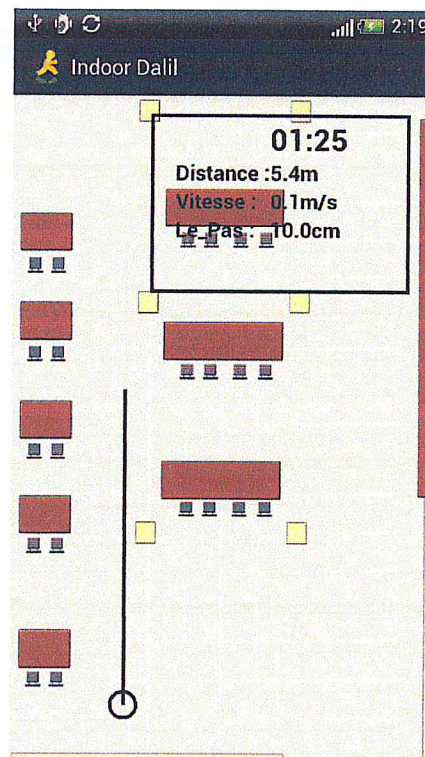


FIGURE 4.13: Position du mobile sur la map

4.7 Conclusion

Ce dernier chapitre a présenté les différents systèmes d'exploitations mobiles connus, une présentation du système Android, ainsi que les outils de réalisation du

projet (Eclipse, ADT...). Lors de la réalisation de l'application, nous avons rencontré quelques difficultés parmi elles, la compréhension du comportement de l'accéléromètre, et la correction des valeurs de l'accélération mesurées par ce dernier, en passant par le calibrage et le filtrage. Et à la fin de ce chapitre, une présentation des détails de l'application développée et une illustration de son utilisation avec quelque interfaces.

Conclusion et Perspectives

Dans ce mémoire, nous avons mis en avant les différents systèmes de navigation ainsi que leurs avantages et inconvénients. Nous avons étudié aussi en détails la navigation inertielle et nous avons vue qu'elle est la plus adéquate pour la navigation dans les endroits fermés. La navigation inertielle se base sur l'utilisation d'un ensemble de capteurs attachés au mobile. Ces capteurs mesurent certaines grandeurs physiques liées aux mouvements du mobile et qui permettent le calcul de sa position.

Nous avons effectué des tests sur ces capteurs principalement l'accéléromètre afin d'analyser les mesures retournées par ce dernier, et les utiliser dans le processus du calcul de position. Les tests ont été effectués sur deux scénarios différents, dans le premier nous avons supposé que la vitesse et la direction du mobile sont connues au préalable, par contre dans le deuxième scénario la vitesse est inconnue. L'obtention de la position du mobile n'était possible qu'après l'analyse, le calibrage, et le filtrage des mesures de l'accélération capté par l'accéléromètre. Et les résultats obtenus s'avèrent logiques et très proches des valeurs réelles des distances et de la vitesse.

Pour la mise en place de ces résultats, nous avons défini et conçu une application mobile pour le positionnement indoor. Cette application fut développée sous le système d'exploitation pour Smartphone Android, elle permet à l'utilisateur de visualiser le tracé de sa trajectoire dans un endroit spécifique (Map), différentes Map peuvent être chargées par l'utilisateur selon son besoin.

Comme perspective à ce travail considéré comme brique de base, nous étudions l'utilisation du capteur gyroscope et magnétomètre pour l'estimation de la position du mobile même dans le cas de mouvements dans différentes directions

Annexes

Annexe A

Les filtres

Un filtre est un circuit électronique linéaire dont le comportement dépend de la fréquence[17]. Autrement dit il atténue certaines composantes d'un signal et en laisse passer d'autres, dans notre cas l'utilisation des filtre étaient en raison d'éliminer les bruits et secousses dus aux mouvements du mobile.

A.1 Filtrage numérique des données brutes

A.1.1 Définition et classification des filtres numériques

Un filtre numérique est défini mathématiquement comme un opérateur ou une transformation qui modifie une séquence d'entrée $E(k)$ en une séquence de sortie $F(k)$. On peut représenter cette transformation par un opérateur T tel que $F(k) = TE(k)$ et en donner l'équation mathématique ou son schéma fonctionnel [17]. Il existe deux types de filtres numériques qui sont :

1. *Filtres à Réponse Impulsionnelle Finie (RIF)* : ce type de filtre est dit fini car sa réponse impulsionnelle se stabilisera ultimement à zéro. Il est aussi appelé filtre non récursif car sa sortie dépend uniquement de l'entrée du signal et il n'y a pas de contre-réaction. Un exemple simple de sommation d'une série de N données puis le calcul de leur moyenne est équivalent à appliquer un filtre RIF à coefficient constant $1/N$ à ces données. Ce type de filtre correspond à un échantillon de sortie $F(k)$ qui dépend uniquement des échantillons d'entrées $E(k), E(k-1), E(k-2), \dots E(k-N)$,

il est par construction toujours stable et sa forme générale d'ordre N est décrite par l'équation mathématique suivante :

2. *Filtres à Réponse Impulsionnelle Infinie (RII)* : ce type de filtre est dit infini car sa réponse impulsionnelle est une valeur oscillatoire qui prend beaucoup de temps à se stabiliser et défois même elle ne se stabilisera jamais, et ce, même à l'infini. Il est aussi appelé filtre récursif car sa sortie dépend à la fois du signal d'entrée et du signal de sortie, il possède ainsi une boucle de contre-réaction. Il correspond par cela à une structure bouclée où l'échantillon de sortie $F(k)$ dépend non seulement des échantillons d'entrées $E(k), E(k-1), E(k-2), \dots, E(k-N)$ mais aussi des sorties précédentes $F(k-1), F(k-2), \dots, F(k-M)$, il y a donc une réinjection de la sortie vers l'entrée comme le montre sa forme générale d'ordre (N, M) décrite par l'équation mathématique suivante :

$$F(k) = \alpha \left(\frac{1}{N+1} \sum_{n=0}^N E(k-n) \right) + \beta \left(\frac{1}{M} \sum_{m=0}^M F(k-m) \right)$$

$$\text{avec } 0 < \alpha \leq 1 \text{ et } \beta = 1 - \alpha$$

La valeur des coefficients α et β fixera la nature du filtre RII, *passé-bas* qui filtre les composantes hautes fréquences et récupère les composantes basses fréquences ou *passé-haut* qui filtre les composantes basses fréquences et récupère les composantes hautes fréquences. Le tableau suivant présente un résumé des caractéristiques des filtres précédents RIF et RII :

<i>Caractéristiques</i>	<i>Filtre RIF</i>	<i>Filtre RII</i>
Récurtivité	Non récurtif	Récurtif
Sélectivité	Faible	Élevée
Ordre	Élevé	Faible
Nombre d'opérations	Élevé	Faible
Mémoire nécessaire	Élevée	Faible
Stabilité	Absolue	Limitée
Précision des coefficients	Raisnable	Élevée
Filtres adaptatifs	Possible	Difficile

TABLE A.1: Tableau comparatif des caractéristiques des filtres RIF et RII

A.2 Les filtres passifs

A.2.1 Les filtres passe bas

Le filtre passe bas est un intégrateur à haute fréquence qui laisse passer les basses fréquences.

A.2.2 Les filtres passe haut

Le filtre passe haut est un dérivateur à basse fréquence qui laisse passer les hautes fréquences.

A.2.3 Passe-Bande, ou sélecteur de bande ou de fréquence

Le filtre passe bande atténue les hautes et basses fréquences et ne laisse passer qu'une bande de fréquence.

Coupe-Bande, ou réjecteur de bande ou de fréquence

Le filtre coupe bande atténue une bande limitée de fréquences.

Bibliographie

- [1] GARIN, Florent, « Android Developer des applications mobiles pour les googles phones », Dunod, 213 pages, octobre 2009.
- [2] WOODMAN, O. J. « An introduction to inertial navigation ». Technical Report, University of Cambridge, Laboratoire d'informatique 2007.
- [3] ZEIMPEKIS Vasileios, M. GIAGLIS George, and LEK George , « A taxonomy of indoor and outdoor positioning techniques for mobile location services », SIGecom exchange, 2003.
- [4] AEBI, Frédérique : « Autonomous Indoor Navigation, Implementation of an Autonomous Navigation System on Android », thèse de master, University of Fribourg(Switzerland) Décembre 2012.
- [5] De Boer, Jean-Remi : « Capteurs MEMS, Optimisation des méthodes de traitement capteurs, de navigation et d'hybridation », thèse de doctorat, Université de Toulouse France, 12 janvier 2012.
- [6] CYPRIANI Matteo, HENRIET Adrien, CANALDA Philippe, SPIES François, « Etude de l'interception et du positionnement de trafic Wi-Fi dans un environnement hétérogène », Rapport de recherche, Laboratoire d'informatique de l'université FRANCHE-COMPTE, Janvier 2011.
- [7] VERVISCH.P Alexandre, « Etude de systemes de positionnement en interieur utilisant des mesures de phase du code ou de phase de la porteuse de signaux de navigation par satellite », thèse de doctorat, Université de Pierre et Marie Curie, 2 juillet 2010.
- [8] BILL R, CAP C, KOFAHL M, and MUNDT T, « Indoor and Outdoor Positioning in Mobile Environments », Geographical Information Sciences, pp. 91-98, 2004.

BIBLIOGRAPHIE

- [9] MANH HUNG V. Le, SARAGAS Dimitris, WEBB Nathan : « Indoor Navigation System for Handheld Devices », thèse de master, Worcester Polytechnic institute, Worcester Massachusetts, USA, 22 octobre 2009.
- [10] SHALA Ubjed , RODRIGUEZ Angel, « Indoor Positioning using Sensor-fusion in Android devices », thèse de master, University of Kristianstad Sweden, 2011.
- [11] GUERIN, J, JALLIFFIER, V, « Le gyroscope », cours Application Mathématique, Ecole Polytechnique MONTREAL, juin 2009.
- [12] STAUFFER, Jean Mi , « Perspectives et application des technologie MEMS, Exemple concret Les accéléromètres », 23 mars 2012.
- [13] SEGUIN Marc, DESCHENEAU Julie, TARDIF Benjamin, « Physique XXI », Edition du Renouveau Pédagogique, 2010.
- [14] ELMENREICH, W. « Sensor Fusion in Time-Triggered Systems », thèse de doctorat de PhD. Vienna University of Technology.2002.
- [15] JONQUIERIE, J.L : « Système de navigation par satellite », Ecole Nationale de l'Aviation Civile , 1994.
- [16] DELOT, Thierry : « Techniques de positionnement en environnement mobile », Laboratoire LAMIH-UMR CNTS 8530, Université de valenciennes,2006.
- [17] MUDRY, Freddy : « Traitement des signaux », Haute Ecole d'Ingenierie et de Gestion du Cauton de Vaud, Novembre 2011.
- [18] PERELSTEIN, Laszlo, « Android équipe quatre téléphones sur cinq vendus dans le monde », La Tribune, « En ligne », <http://www.latribune.fr/technos-medias/electronique/20130809trib000779898/android-equipe-quatre-telephones-sur-cinq-vendus-dans-le-monde.html> , 9 août 2013.(Consulté le 30 août 2013).
- [19] Android- Help « En ligne ». <http://www.android-help.fr/article/les-versions-Android> (Consulté le 20 juin 2013).
- [20] Android - Sensors Overview « En ligne ». http://developer.android.com/guide/topics/sensors/sensors_overview.html(Consulté le 28 mai 2013).

BIBLIOGRAPHIE



- [21] Android- Sensors Motion « En ligne » http://developer.android.com/guide/topics/sensors/sensors_motion.html (Consulté le 20 août 2013).
- [22] Architecture Android « En ligne » <http://javamind-fr.blogspot.com/2012/10/architecture-android-le-noyau-linux.html> (Consulté le 20 juin 2013).
- [23] HTC références « En ligne » <http://www.htc.com/fr/> (Consulté le 28 juillet 2013).
- [24] Introduction-Programmation-Android « En ligne » <http://www.tutos-android.com/introduction-programmation-android> (Consulté le 12 mai 2013).
- [25] Portail developpeur Android « En ligne » <http://developer.android.com/index.html> (Consulté le 20 juin 2013).
- [26] Références SDK developpeur « En ligne » <http://developer.android.com/sdk/index.html> (Consulté le 20 juin 2013).