



Photo : Samuel Schwyyn

Développement d'une application smartphone de positionnement différentiel GNSS en temps réel

Elisa BORLAT

Juillet 2024

Domaine Ingénierie et Architecture

Master conjoint UNIGE-HES-SO en développement territorial

Orientation Ingénierie géomatique

Directeur : Prof. Dr Sébastien GUILLAUME

Expert : Dr Daniel WILLI

Mémoire no : 1089



**UNIVERSITÉ
DE GENÈVE**

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale

Résumé

Aujourd'hui, tous les smartphones peuvent se localiser de manière absolue avec une précision de l'ordre de deux à trois mètres. Pour obtenir une meilleure précision, le positionnement différentiel peut être utilisé. L'accès aux données brutes du récepteur GNSS intégré ouvre de nouvelles perspectives pour développer des algorithmes avancés de positionnement. L'objectif de cette recherche est de réaliser une application de positionnement différentiel en temps réel.

L'application a été développée en plusieurs étapes : accès aux données brutes et calcul des pseudo-distances, accès aux éphémérides et calcul de la position des satellites, connexion et récupération des données brutes de la station de base et finalement, estimation de la position par moindres carrés avec l'estimateur de Biber. Les données ont été traitées avec le logiciel RTKLIB en mode DGPS et en mode de positionnement simple sur le code afin de les comparer avec les résultats des positions calculées avec le smartphone. Les résultats ont également été comparés à la vérité terrain.

Lors des mesures statiques réalisées, les résultats des calculs ont montré une forte dépendance au bruit de mesure élevé, attribuable à la qualité souvent médiocre des antennes embarquées dans les smartphones. La dispersion des différences de coordonnées planimétriques est de l'ordre d'un à deux mètres et de trois à quatre mètres sur les différences d'altitudes. Le biais par rapport aux coordonnées vraies est de l'ordre d'un demi-mètre pour les mesures en mode DGPS, tandis qu'il est d'un à trois mètres pour le positionnement simple.

Les résultats obtenus montrent une amélioration notable, mais la précision atteint ses limites en raison du bruit de mesure inhérent aux récepteurs GNSS intégrés aux smartphones. Des recherches futures pourraient se concentrer sur l'amélioration des antennes et des algorithmes de traitement des données pour obtenir des résultats encore plus précis.

Avant-propos

Ce travail s'inscrit dans le cadre de la formation du Master conjoint UNIGE-HES-SO en Développement Territorial (MDT), orientation Ingénierie Géomatique. Il se déroule à la fin des enseignements dispensés et permet de démontrer les qualités nécessaires à l'obtention du diplôme, à travers un travail de recherche. Il a été mené à temps partiel en parallèle d'un emploi et rédigé de septembre 2023 à juillet 2024.

Le sujet de ce travail « Développement d'une application smartphone de positionnement différentiel GNSS en temps réel » a été proposé par le Prof. Dr Sébastien GUILLAUME, professeur en géodésie, méthodes d'estimation et navigation à la Haute École d'Ingénierie et de Gestion du Canton de Vaud (HEIG-VD). Il est responsable de plusieurs modules de cours du Master et c'est à la suite à nos échanges où j'ai montré un vif intérêt pour les systèmes de navigation par satellite, qu'il m'a suggéré ce sujet. L'objectif principal avec ce travail est d'appliquer mes connaissances en positionnement par satellite en suivant le processus complet : des données brutes d'un récepteur jusqu'à l'obtention d'une position.

L'expert pour ce travail est le Dr Daniel WILLI, chef du processus *Mensuration géodésique nationale* à l'Office fédéral de topographie et auteur de la thèse *GNSS receiver synchronisation and antenna calibration*.

Ce document s'adresse principalement aux ingénieurs et spécialistes de la géomatique qui s'intéressent aux possibilités offertes par les récepteurs embarqués dans les smartphones en termes de précision. Les développeurs d'applications souhaitant intégrer une fonctionnalité de positionnement différentiel peuvent également se référer à ce travail.

La première partie de ce document présente un état de situation des technologies et possibilités actuelles dans le positionnement par satellite ainsi que les bases théoriques nécessaires à la compréhension du positionnement par satellite (chapitres 1 et 2). Le lecteur trouvera dans le chapitre 3 comment concrètement les données sont récupérées et traitées pour estimer une position et comment l'application est développée. Finalement, le chapitre 4 présente les tests qui ont été effectués et les résultats obtenus notamment en terme de précision.

Le choix de ce sujet de travail représentait un grand défi, dans la mesure où la programmation d'applications pour smartphone ne faisait pas partie des enseignements dispensés durant mes études. De plus, le langage de programmation Java, utilisé pour développer l'application, m'était inconnu au début de ce projet. Ainsi, je prie le lecteur expérimenté de bien vouloir excuser la présence de certains éléments qui pourraient lui sembler triviaux.

Le présent rapport est accompagné de l'application prototype et de l'entièreté du code source. Le code de l'application se trouve sur GitHub à l'adresse suivante : <https://github.com/elisaborlat/PrototypeDGPS>. Celle-ci ayant été développée en tant que prototype dans le cadre de ce travail et à des fins de recherche, son installation n'a pas été testée sur d'autres appareils.

Remerciements

Je tiens à exprimer mes remerciements les plus sincères et ma gratitude envers toutes les personnes qui m'ont accompagnées durant ce travail, en particulier :

- **Prof. Dr Sébastien GUILLAUME**, directeur de ce travail, pour tous ses enseignements passionnants qui m'ont donné envie de réaliser un travail en GNSS ainsi que pour son encadrement, sa disponibilité et son enthousiasme durant toute la durée de ce travail.
- **Dr Daniel WILLI** pour sa présence comme expert, sa relecture attentive du rendu intermédiaire et ses conseils avisés.
- **Prof. Yves DEILLON**, géomètre breveté fédéral, professeur à la HEIG-VD, et mon responsable direct au sein de la HEIG-VD, pour sa compréhension et sa souplesse en matière d'horaires de travail durant ce projet réalisé en parallèle de mes obligations professionnelles.
- **M. Samuel SCHWYN**, pour sa grande aide sur le terrain lors de la réalisation des tests de l'application, pour les nombreux échanges constructifs ainsi que pour les documents et scripts mis à disposition.
- **M. Kilian MOREL**, pour son soutien et la relecture du présent rapport.
- **M. Fabien DÉLÈZE**, pour son soutien durant ce travail mais aussi tout au long de notre cursus Master.

Table des matières

Introduction	1
1 État de l’art	3
2 Éléments théoriques	5
2.1 Généralités	5
2.2 Signaux GNSS	5
2.2.1 Rapport signal-bruit	7
2.3 Types de mesure	7
2.3.1 Mesure de pseudo-distance	7
2.3.2 Mesure de phase	8
2.3.3 Mesure Doppler	8
2.4 Positionnement différentiel sur le code	9
2.5 Estimation de la position	10
2.5.1 Modèle de Gauss-Markov	10
2.5.2 Positionnement différentiel sur une seule époque	11
2.5.3 Avec contraintes entre paramètres	11
2.5.4 Estimateur de Biber	12
2.5.5 Modèle stochastique	13
2.5.6 Indicateurs Dilution Of Precision (DOPs)	14
2.6 Systèmes de coordonnées	14
2.7 Systèmes temporels	16
3 Conception et développement de l’application	19
3.1 Conception de l’application	19
3.1.1 Objectifs de l’application	19
3.1.2 Fonctionnalités de l’application	19
3.1.3 Plate-forme de développement et langage de programmation	19
3.2 Développement de l’application	20
3.3 Mesures brutes du smartphone et calcul des observations	20
3.4 Messages de navigation et éphémérides	22
3.5 Données de la station de base	23
3.5.1 Méthode de transmission	23
3.5.2 Format et contenu des données de la station de base	24
3.5.3 Validation des messages RTCM	25
3.6 Traitement GNSS différentiel et asynchrone	25
3.7 Formation des doubles différences	25
3.8 Estimation de la position	26
3.9 Structure de l’application	26
3.9.1 Bases du développement d’application Android	26
3.9.2 Logique de l’application	28
3.9.3 Interface utilisateur	29
3.9.4 Fonctionnalités	31
4 Tests et évaluation de l’application	33
4.1 Validation du calcul des pseudo-distances	33
4.2 Évaluation de la précision du positionnement	33
4.2.1 Mesures statiques sur un point connu	36
4.2.2 Mesures cinématiques avec suivi de la position par tachéométrie	42

4.2.3 Résumé	46
4.3 Analyse des performances de l'application prototype	46
Conclusion	47
Bibliographie	49
Annexes	59
A Résultats des mesures cinématiques	59
A.1 Session 2	59
A.2 Session 3	62
B Caractéristiques du smartphone	65

Introduction

Contexte

Aujourd'hui, la plupart des smartphones peuvent se localiser de façon absolue, n'importe où sur Terre, grâce à la technologie GNSS, avec une précision de l'ordre de deux à trois mètres lorsque les conditions de réception sont idéales. Lorsque l'on souhaite obtenir des précisions plus élevées, il est possible d'envisager un positionnement GNSS différentiel qui combine les mesures brutes du smartphone et d'une station GNSS de référence. Actuellement, les smartphones embarquent des technologies qui donnent accès aux mesures brutes du récepteur GNSS intégré. Ainsi, le développement d'applications de positionnement GNSS différentiel peut être envisagé.

Cahier des charges

Ce travail a pour but de développer d'une application smartphone de positionnement différentiel GNSS en temps réel. Cette application devra permettre de déterminer la position du smartphone dans le cadre de référence Suisse **MN95** et **RAN95** ou **NF02**. De plus, elle devra également donner la possibilité à l'utilisateur d'accéder à des indicateurs de qualité des mesures et du positionnement. Dans un premier temps, les développements devront se concentrer sur le positionnement différentiel DGPS L1 de la constellation GPS uniquement. Par la suite, l'intégration d'une ou plusieurs constellations supplémentaires ainsi que d'autres signaux GNSS pourront être envisagés.

Les objectifs intermédiaires de ce travail sont les suivants :

1. Recherche et étude de la littérature.
2. Étude et conception de l'expérience utilisateur de la future application.
3. Développement de l'application smartphone de positionnement GNSS différentiel.
4. Mise en œuvre et tests de l'application.

Chapitre 1

État de l'art

Le premier smartphone grand public équipé d'un récepteur GPS est le Nokia N95, sorti en 2007. Il a été un succès commercial et a contribué à populariser la géolocalisation sur les smartphones. Généralement, les informations accessibles aux utilisateurs se limitaient à des données élémentaires concernant les satellites, telles que le rapport signal-bruit, l'azimut et l'élévation. Il était également possible d'obtenir les positions calculées grâce aux solutions PVT (Position, Vitesse, Temps). La précision obtenue avec cette méthode était généralement de 3 à 5 mètres dans des conditions idéales, c'est-à-dire dans des zones dégagées avec une faible incidence du multi-trajet. Cependant, en présence de conditions multi-trajet défavorables, notamment dans des environnements urbains denses, la précision diminuait rapidement, atteignant souvent plus de 10 mètres. Quelques études ont cherché à améliorer la précision de la position à l'aide de capteurs externes, par exemple en envoyant des corrections à partir d'une autre antenne (Zangenehjad ; Gao, 2021). En 2014, pour la première fois, une précision centimétrique de la position a été obtenue à partir d'une antenne de smartphone (Pesyna et al., 2014). Cette étude démontre également que les observations de l'antenne du smartphone sont fortement perturbées. En effet, l'antenne est polarisée linéairement plutôt que circulairement, comme c'est le cas pour les antennes géodésiques. Cette polarisation rend les signaux captés par l'antenne du smartphone beaucoup plus sensibles aux effets de multi-trajet.

Dès 2016, l'accès aux mesures brutes GNSS est devenu possible grâce à la version 7 du système d'exploitation Android. Cette nouveauté a stimulé l'intérêt pour la géolocalisation des smartphones, entraînant une recherche accrue pour le développement de nouveaux algorithmes avec comme principal objectif d'améliorer la précision de la position (Zangenehjad ; Gao, 2021). Peu de temps après, les premiers smartphones équipés de récepteurs multi-GNSS - le Samsung S8 et le Huawei P10 -, ont été lancés sur le marché. En 2018, le Xiaomi Mi 8 est devenu le premier smartphone doté d'un récepteur double fréquence, offrant ainsi la possibilité de corriger les effets de l'ionosphère et d'améliorer la précision des positions obtenues. Ce récepteur embarqué est équipé du nouveau Broadcom BCM47755, permettant de traquer les signaux L5 pour le GPS, E5a pour Galileo, ainsi que les satellites de la constellation QZSS. Il a été suivi par une deuxième génération de récepteurs GNSS double fréquence (BCM477765) capables de suivre les constellations modernisées avec de nouveaux signaux (Zangenehjad ; Gao, 2021).

La limitation du positionnement par smartphone découle principalement de la qualité des antennes et récepteurs qui les équipent. Ces derniers sont généralement simples et économiques, ce qui entraîne des mesures brutes de qualité médiocre. Cette limitation se manifeste principalement dans le rapport signal-bruit des récepteurs de smartphones, avec un rapport C/N_0 (carrier-to-noise-power-density ratio) généralement inférieur d'environ 10 dB-Hz et plus variable par rapport à une antenne de récepteur géodésique. De plus, le bruit observé sur les mesures de code et de phase est jusqu'à 10 à 100 fois plus élevé (Zhang et al., 2018). Une corrélation significative est observée entre les résidus des mesures et le rapport C/N_0 . Ainsi, l'utilisation du rapport signal-bruit se révèle être une alternative plus pertinente pour pondérer les mesures que l'élévation des satellites, qui est généralement utilisée pour la pondération des observations avec les récepteurs géodésiques (Liu et al., 2019).

Une autre limitation au positionnement provient du mode d'économie d'énergie appelé cycle de service (*duty-cycle* en anglais) qui se traduit par des perturbations dans la cohérence des données recueillies. Par défaut, le système Android allume et éteint le récepteur GNSS afin de réduire l'utilisation de la batterie. Depuis la version 9 d'Android, il est possible de désactiver ce cycle de service. Ceci peut être fait dans les options de développement en activant l'option « forçage de toutes les mesures du GNSS ». Désactiver le cycle de service est particulièrement important pour le positionnement RTK ou PPP car ils utilisent directement les mesures de phase. En effet, le mode d'économie d'énergie des batteries engendre des ruptures de cycle et, par conséquent, des données de phase non continues, ce qui complique fortement la résolution des ambiguïtés.

Applications existantes d'enregistrement des données brutes

Avec la mise à jour de la nouvelle API *android.location* en 2016, Google lance une application appelée GNSSLogger, qui permet d'enregistrer les mesures brutes au format CSV. Dans les versions ultérieures de l'application, une nouvelle fonctionnalité est ajoutée. L'application calcule les observations de pseudo-distance, de phase et du décalage Doppler et offre la possibilité d'enregistrer ces observations directement dans un fichier au format RINEX. Aujourd'hui, de nombreuses applications offrent cette fonctionnalité d'enregistrement des données au format RINEX. Le tableau 1.1 répertorie les principales applications smartphone qui enregistrent les données brutes des satellites (Zangenehjad ; Gao, 2021).

TABLEAU 1.1 – Applications d'enregistrement de données brutes GNSS

Application	Développeur	Format de sortie
<i>GNSSLogger</i>	Google	CSV, NMEA et RINEX
<i>Geo++RINEX Logger</i>	Geo++GmbH Compagny	RINEX
<i>GalileoPVT</i>	Agence spatiale européenne (ESA)	CSV et RINEX
<i>GPSTest</i>	Université de Floride du Sud	CSV

L'enregistrement des données au format RINEX offre la possibilité de calculer la position en post-traitement à l'aide des nombreux logiciels compatibles avec ce format, tels que le logiciel open source RTKLIB. Cependant, les observations de code et de phase enregistrées par certaines de ces applications présentent des inconsistances, comme relevé dans une étude antérieure (Zangenehjad ; Jiang et al., 2023). Les résultats montrent une inconsistance entre les mesures de pseudo-distance et de phase issues des fichiers RINEX de l'application GNSSLogger et entre les mesures de phase et Doppler de l'application Geo++RINEX Logger. En revanche, les observations Doppler se sont avérées cohérentes dans les trois fichiers RINEX analysés, ne posant ainsi aucun problème de génération ni d'incohérence. Ces inconsistances dans les observations de phase et de code soulignent la nécessité de prudence dans l'utilisation des données fournies par ces applications.

Chapitre 2

Éléments théoriques

Ce chapitre présente les concepts généraux du positionnement par satellite et plus spécifiquement du positionnement différentiel. Ces concepts sont nécessaires pour comprendre comment les mesures brutes du récepteur embarqué dans le smartphone seront utilisées pour estimer la position de ce dernier. Le contenu et la nomenclature de ce chapitre reprennent les éléments des supports de cours « Systèmes de positionnement global par satellites GNSS » (2022) et « Méthodes d'estimation et réseaux géodésiques » (2022) du Prof. Dr Sébastien GUILLAUME, ainsi que du document de référence pour l'usage des mesures brutes GNSS sur les appareils Android publié par l'agence européenne GSA (2017).

2.1 Généralités

Le GNSS (*Global Navigation Satellite System*) est un ensemble de constellations de satellites qui fournissent des services de positionnement, de navigation et de synchronisation temporelle à l'échelle mondiale. Les satellites GNSS orbitent à une altitude d'environ 20 000 kilomètres au-dessus de la Terre, émettant des signaux utilisés par les récepteurs GNSS pour déterminer leur position. Le délai de transmission des signaux GNSS depuis les satellites jusqu'aux récepteurs sur Terre est d'environ 65 millisecondes.

Il y a actuellement quatre principales constellations : GPS (États-Unis), GLONASS (Russie), Galileo (Union européenne) et BeiDou (Chine). Chaque constellation fonctionne de manière coordonnée pour fournir une couverture mondiale et une disponibilité continue des signaux de positionnement et de synchronisation.

Le principe du positionnement par satellite est, à partir des signaux reçus, de mesurer la différence de temps entre la transmission et la réception du signal. Il est ensuite possible, à partir de cette différence de temps, de calculer la distance apparente entre le récepteur et un satellite appelée pseudo-distance.

$$P_A^i = (t_R^{recA} - t_T^{sat_i}) \cdot c \quad (2.1)$$

Avec :

- P_A^i Pseudo-distance entre le récepteur A et le satellite i
- t_R^{recA} Temps mesuré avec l'horloge du récepteur au moment de la réception A
- $t_T^{sat_i}$ Temps mesuré avec l'horloge du satellite au moment de la transmission du signal
- c Vitesse de la lumière dans le vide

2.2 Signaux GNSS

Chaque satellite émet des signaux qui sont des ondes électromagnétiques se propageant à la vitesse de la lumière. La fréquence de ces ondes se situe autour de 1.5 GHz, ce qui correspond à une longueur d'onde d'environ 20 cm. Un satellite émet plusieurs signaux sur des fréquences différentes. Le tableau 2.1 présente les différents signaux GPS. Aujourd'hui, les satellites GPS transmettent quatre différents signaux sur la bande L1.

Pour transmettre des informations, les ondes porteuses sont modulées. Les informations principales modulées sont le code et les données de navigation.

TABLEAU 2.1 – Signaux GPS

Bande	Signal	Fréquence [MHz]	λ [cm]	Remarque
L1	P(Y)	1575.42	19.03	Signal militaire
	C/A	1575.42	19.03	
	L1C	1575.42	19.03	Dès bloc III
	M	1575.42	19.03	Signal militaire, dès bloc IIR-M
L2	P(Y)	1227.60	24.42	Signal militaire
	L2 CM	1227.60	24.42	Dès bloc IIR-M
	L2 CL	1227.60	24.42	Dès bloc IIR-M
	M	1227.60	24.42	Signal militaire, dès bloc IIR-M
L5	I5	1176.45	25.48	Dès bloc IIF
	Q5	1176.45	25.48	Dès bloc IIF

Le code est utilisé pour déterminer le temps transmis par le satellite et d'en déduire les pseudo-distances. Pour faire cela le récepteur produit sa propre réplique du code et le synchronise avec celui transmis par le satellite. Le signal le plus couramment utilisé par les appareils grand public est le signal Coarse/Aquisition (C/A). Le code se répète périodiquement chaque milliseconde. Ainsi, lorsque le récepteur est synchronisé avec le code C/A, le temps transmis est ambigu et compris entre 0 et 1 milliseconde.

Pour résoudre cette ambiguïté, le récepteur utilise la structure du message de navigation. La figure 2.1 montre la structure du code et du message de navigation qui permet de résoudre l'ambiguïté du temps transmis par le satellite. Celle-ci est résolue lorsque le temps GPS en seconde de semaine (TOW) contenu dans chaque sous-trame est décodé.

Un message de navigation, pour la constellation GPS, est construit de 25 trames, une trame étant constituée de 5 sous-trames consécutives.

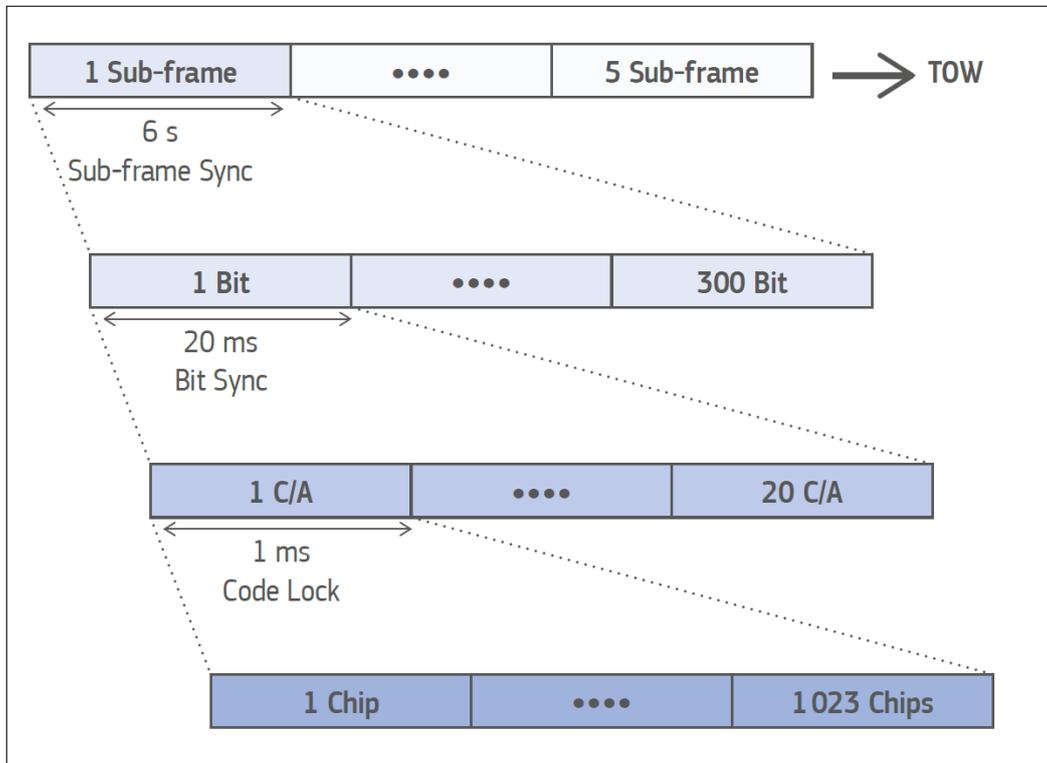


FIGURE 2.1 – Structure du code et du message de navigation du signal GPS L1 C/A (European GNSS Agency (GSA), 2017)

TABLEAU 2.2 – Statuts de synchronisation

Statuts de synchronisation	Temps
C/A code	1 ms
Bit	20 ms
Subframe	6 s
TOW	1 week

2.2.1 Rapport signal-bruit

La plupart des récepteurs estiment le rapport signal-bruit, aussi appelé signal-to-noise ratio (SNR). Cette valeur n'étant pas directement comparable entre les récepteurs, le carrier-to-noise-power density ratio noté C/N_0 est généralement utilisé. Il s'agit de la puissance de l'onde porteuse divisé par la densité du bruit dans une bande de 1 Hz. La valeur du C/N_0 vaut généralement entre 45 dB-Hz et 50 dB-Hz pour un satellite au zénith. Une grande valeur indique un signal puissant. Certaines applications utilisent parfois à tort le terme SNR à la place du carrier-to-noise ratio C/N_0 .

2.3 Types de mesure

Les trois types de mesure de base issues des signaux GNSS sont les pseudo-distances, la mesure de phase et le décalage Doppler.

2.3.1 Mesure de pseudo-distance

La pseudo-distance entre un satellite et un récepteur est déterminée à partir de la différence de temps apparente entre l'émission et la réception du signal. Chacun des satellites génère un code spécifique. Le récepteur génère une réplique de ce code basé sur sa fréquence interne et l'aligne avec le signal reçu. Avec ce décalage, il est possible de déterminer la distance apparente en multipliant la différence de temps par la vitesse de la lumière (équation 2.1).

Ces mesures sont appelées des pseudo-distances, car elles ne correspondent pas à la distance géométrique exacte qui sépare le récepteur des satellites. Elle inclut différentes erreurs à prendre en compte dont notamment les erreurs d'horloge. L'horloge interne du récepteur au moment de la réception du signal n'est pas exactement synchronisée avec les horloges des satellites. La figure 2.2 présente les différentes erreurs entre la pseudo-distance et la distance géométrique réelle qui sépare le récepteur du smartphone.

L'équation d'observation d'une pseudo-distance est la suivante :

$$P_A^i + \hat{v}_{P_A^i} = |\mathbf{x}^i - \hat{\mathbf{x}}_A| + c \cdot \hat{\delta t}_R^{recA} - c \cdot \delta t_T^{sat_i} + \delta \rho_{A,trop}^i + \delta \rho_{A,iono}^i \quad (2.2)$$

Avec :

- P_A^i Observation de pseudo-distance entre le récepteur A et le satellite i
- $\hat{v}_{P_A^i}$ Résidu sur l'observation de pseudo-distance
- \mathbf{x}^i Position du satellite i au moment de la transmission du signal
- $\hat{\mathbf{x}}_A$ Position du récepteur A au moment de la réception du signal
- $\hat{\delta t}_R^{recA}$ Erreur d'horloge du récepteur
- $\delta t_T^{sat_i}$ Erreur d'horloge du satellite
- $\delta \rho_{A,trop}^i$ Erreur due au retard troposphérique
- $\delta \rho_{A,iono}^i$ Erreur due au retard ionosphérique

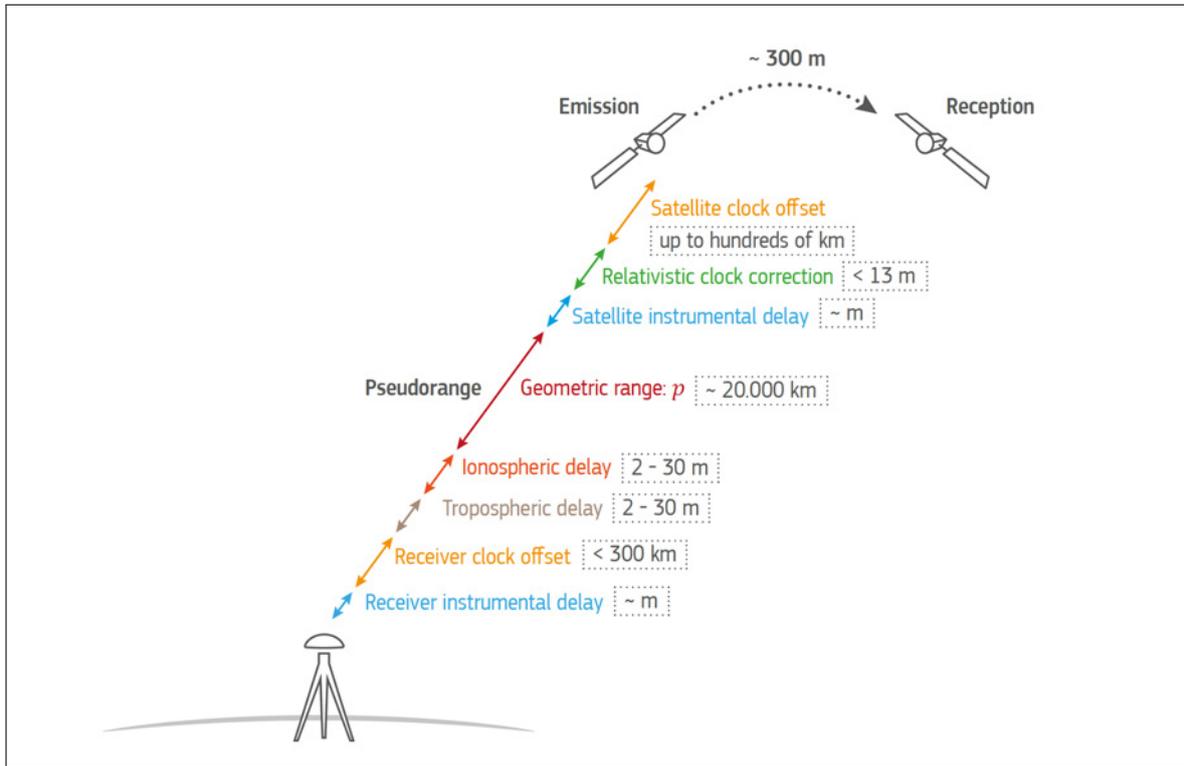


FIGURE 2.2 – Sources d’erreurs sur la mesure de pseudo-distance (European GNSS Agency (GSA), 2017)

2.3.2 Mesure de phase

Le récepteur mesure également le décalage de la phase. Cette méthode permet d’augmenter grandement la précision des mesures mais nécessite de résoudre les ambiguïtés de phase.

Ainsi, l’équation d’observation pour une mesure de phase est la suivante :

$$\lambda \cdot (\Phi_A^i + \hat{v}_{\Phi_A^i}) = |\mathbf{x}^i - \hat{\mathbf{x}}_A| - \lambda \cdot \hat{N}_A^i + c \cdot \hat{\delta}t_R^{recA} - c \cdot \delta t_T^{sat_i} + \delta \rho_{A,trop}^i + \delta \rho_{A,iono}^i \quad (2.3)$$

Avec :

- Φ_A^i Mesure de phase entre le satellite i et le récepteur A
- λ Longueur d’onde du signal observé
- $\hat{v}_{\Phi_A^i}$ Résidu sur l’observation de phase
- \hat{N}_A^i Ambiguïté de phase (ambiguïté de zéro différence)

Dans le cas du positionnement par smartphone, la résolution des ambiguïtés est rendue particulièrement compliquée par la qualité médiocre des antennes ainsi que des cycles d’économie d’énergie du système.

L’application développée dans ce travail utilise uniquement les mesures sur le code. C’est pourquoi le positionnement avec mesure de phase n’est pas développé dans le présent document.

2.3.3 Mesure Doppler

Lorsqu’une source mobile produit une onde sonore ou électromagnétique, le signal perçu par un observateur verra sa fréquence changer. Cet effet est connu sous le nom d’effet Doppler.

Dans le contexte des signaux GNSS, le décalage Doppler correspond au changement de fréquence entre le signal reçu et le signal émis par un satellite dû à la vitesse relative entre le récepteur et celui-ci.

Ainsi, l’observation Doppler exprimée en Hertz (Hz) entre un récepteur A et un satellite i est définie selon l’équation suivante :

$$D_A^i = f_{mesurées} - f^{émise} \quad (2.4)$$

Lorsque la mesure Doppler est positive, cela indique que la distance entre le récepteur et le satellite diminue (fréquence observée plus grande), tandis qu’une mesure négative indique que la distance augmente (fréquence observée plus petite).

2.4 Positionnement différentiel sur le code

Le positionnement différentiel repose sur l'utilisation des mesures de deux récepteurs pour former des combinaisons entre les pseudo-distances, appelées double différence. Cette approche offre l'avantage notable d'éliminer les erreurs d'horloge des satellites et des récepteurs lorsque la ligne de base entre ces deux récepteurs est courte (moins de 50 km). Les erreurs liées à l'ionosphère et à la troposphère sont également considérées comme identiques si la ligne de base est courte et s'éliminent.

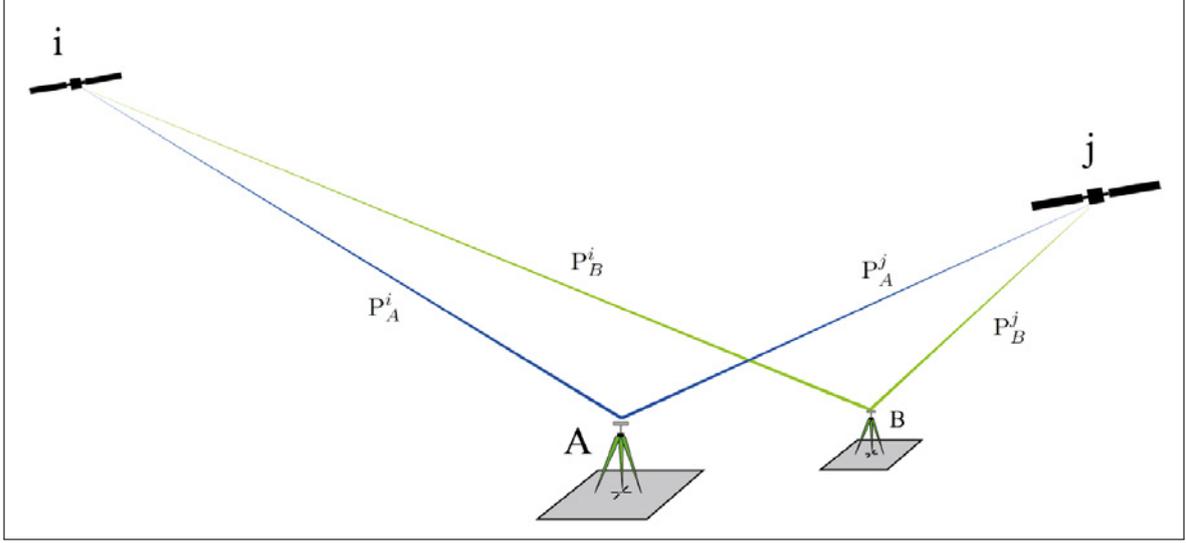


FIGURE 2.3 – Principe du positionnement différentiel (Guillaume, 2022a)

Zéro différence

Pour former les doubles différences, il faut partir des observations de pseudo-distance, appelées également zéro différence dans le contexte du positionnement différentiel.

Selon l'équation 2.2, voici les observations de pseudo-distance de deux récepteurs A et B avec le satellite i et j . Les résidus ne sont pas explicitement notés dans les équations, par mesure de simplicité.

$$P_A^i = |\mathbf{x}^i - \mathbf{x}_A| + c \cdot \delta t_R^{recA} - c \cdot \delta t_T^{sat_i} + \delta \rho_{A,trop}^i + \delta \rho_{A,iono}^i \quad (2.5)$$

$$P_A^j = |\mathbf{x}^j - \mathbf{x}_A| + c \cdot \delta t_R^{recA} - c \cdot \delta t_T^{sat_j} + \delta \rho_{A,trop}^j + \delta \rho_{A,iono}^j \quad (2.6)$$

$$P_B^i = |\mathbf{x}^i - \mathbf{x}_B| + c \cdot \delta t_R^{recB} - c \cdot \delta t_T^{sat_i} + \delta \rho_{B,trop}^i + \delta \rho_{B,iono}^i \quad (2.7)$$

$$P_B^j = |\mathbf{x}^j - \mathbf{x}_B| + c \cdot \delta t_R^{recB} - c \cdot \delta t_T^{sat_j} + \delta \rho_{B,trop}^j + \delta \rho_{B,iono}^j \quad (2.8)$$

Simple différence

La simple différence est formée de la différence entre les mesures de deux récepteurs vers le même satellite. Cette étape permet de s'affranchir de l'erreur d'horloge du satellite.

$$P_{AB}^i = P_B^i - P_A^i = |\mathbf{x}^i - \mathbf{x}_B| - |\mathbf{x}^i - \mathbf{x}_A| + c \cdot \delta t_R^{recB} - c \cdot \delta t_R^{recA} \quad (2.9)$$

$$P_{AB}^j = P_B^j - P_A^j = |\mathbf{x}^j - \mathbf{x}_B| - |\mathbf{x}^j - \mathbf{x}_A| + c \cdot \delta t_R^{recB} - c \cdot \delta t_R^{recA} \quad (2.10)$$

De plus, lorsque la ligne de base est courte, les signaux traversent la même atmosphère et il est admis que les délais troposphériques et ionosphériques sont équivalents. Ainsi la formation de simples différences élimine également ces délais.

$$\delta \rho_{A,trop}^i \cong \delta \rho_{B,trop}^i$$

$$\delta \rho_{A,iono}^i \cong \delta \rho_{B,iono}^i$$

Double différence

La double différence est constituée de la différence entre deux simples différences, soit entre les récepteurs A et B avec un premier satellite i et entre les récepteur A et B avec un second satellite j .

$$P_{AB}^{ij} + \hat{v}_{P_{AB}^{ij}} = |\mathbf{x}^j - \mathbf{x}_B| - |\mathbf{x}^j - \mathbf{x}_A| - |\mathbf{x}^i - \mathbf{x}_B| + |\mathbf{x}^i - \mathbf{x}_A| \quad (2.11)$$

Cette deuxième différence permet d'éliminer l'erreur d'horloge des récepteurs.

Pour former une observation de double différence, il est nécessaire d'avoir deux satellites. Ainsi, avec n satellites, il est possible de former $n-1$ doubles différences.

2.5 Estimation de la position

La position du récepteur est estimée par la méthode des moindres carrés. Cette méthode permet d'estimer les paramètres inconnus de façon optimale à partir des observations.

La solution de l'estimation par les moindres carrés est celle qui minimise la fonction suivante :

$$\min \hat{\mathbf{v}}^T \cdot \mathbf{P} \cdot \hat{\mathbf{v}} \quad (2.12)$$

2.5.1 Modèle de Gauss-Markov

Pour effectuer une estimation avec le modèle de Gauss-Markov, chacune des observations doit pouvoir s'écrire en fonction des paramètres inconnus.

$$\mathbf{l} + \hat{\mathbf{v}} = \mathbf{f}(\hat{\mathbf{x}}) \quad (2.13)$$

Avec :

- \mathbf{l} Vecteur des observations
- $\hat{\mathbf{v}}$ Vecteur des résidus estimés
- $\hat{\mathbf{x}}$ Vecteur des inconnues à estimer
- $\mathbf{f}(\hat{\mathbf{x}})$ Fonction des inconnues (équations d'observations)

Une fois l'ensemble des équations d'observation posé, celles-ci sont linéarisées autour des paramètres approchés.

$$\mathbf{l} + \hat{\mathbf{v}} = \mathbf{f}(\overset{\circ}{\mathbf{x}}) + \left. \frac{\partial \mathbf{f}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \right|_{\overset{\circ}{\mathbf{x}}} \cdot \delta \hat{\mathbf{x}} \quad (2.14)$$

La matrice jacobienne, formée par les dérivées partielles des fonctions par rapport aux inconnues, est définie de la façon suivante :

$$\mathbf{A} = \left(\begin{array}{cccc} \frac{\partial f_1}{\partial \hat{x}_1} & \frac{\partial f_1}{\partial \hat{x}_2} & \cdots & \frac{\partial f_1}{\partial \hat{x}_u} \\ \frac{\partial f_2}{\partial \hat{x}_1} & \frac{\partial f_2}{\partial \hat{x}_2} & \cdots & \frac{\partial f_2}{\partial \hat{x}_u} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \hat{x}_1} & \frac{\partial f_n}{\partial \hat{x}_2} & \cdots & \frac{\partial f_n}{\partial \hat{x}_u} \end{array} \right) \bigg|_{\overset{\circ}{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_3}} \quad (2.15)$$

Avec le vecteur des observations réduites :

$$\delta \mathbf{l} = \mathbf{l} - \mathbf{f}(\overset{\circ}{\mathbf{x}}) \quad (2.16)$$

Le système d'équations peut s'écrire sous la forme matricielle :

$$\delta \mathbf{l} + \hat{\mathbf{v}} = \mathbf{A} \cdot \delta \hat{\mathbf{x}} \quad (2.17)$$

Le vecteur des paramètres inconnus $\hat{\mathbf{x}}$ est relié au vecteur d'accroissement des paramètres inconnus $\delta \hat{\mathbf{x}}$ par :

$$\hat{\mathbf{x}} = \overset{\circ}{\mathbf{x}} + \delta \hat{\mathbf{x}} \quad (2.18)$$

Le vecteur d'accroissement est calculé de la manière suivante :

$$\delta \hat{\mathbf{x}} = (\mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{P} \cdot \delta \mathbf{l} \quad (2.19)$$

L'ajustement par la méthode des moindres carrés consiste à estimer les paramètres par itérations successives jusqu'à ce que le vecteur des accroissements ne varie presque plus. Il est également nécessaire de définir un nombre maximum d'itérations afin d'interrompre l'ajustement lorsque celui-ci ne converge pas.

La matrice des cofacteurs des paramètres inconnus $\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ est calculée selon l'équation suivante :

$$\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = (\mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \quad (2.20)$$

Calcul des résidus et de l'unité de poids empirique

Une fois que le calcul a convergé, il est possible de calculer les résidus et les résidus normés :

$$\hat{\mathbf{v}} = \mathbf{A} \cdot \delta \hat{\mathbf{x}} - \delta \mathbf{l} \quad (2.21)$$

L'écart-type de l'unité de poids empirique est calculé à partir des résidus :

$$s_0 = \sqrt{\frac{\hat{\mathbf{v}}^T \cdot \mathbf{P} \cdot \hat{\mathbf{v}}}{n - u}} \quad (2.22)$$

2.5.2 Positionnement différentiel sur une seule époque

Lors d'un calcul différentiel, les coordonnées du point de base sont supposées connues et fixes. Ainsi, le vecteur des paramètres inconnus n'est composé que des coordonnées du récepteur A . Les erreurs d'horloge des récepteurs s'annulent d'ailleurs dans un calcul de doubles différences.

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x}_A \\ \hat{y}_A \\ \hat{z}_A \end{pmatrix} \quad (2.23)$$

Le vecteur des observations est formé avec les observations de double différence entre deux satellites et deux récepteurs.

$$\mathbf{l} = \begin{pmatrix} P_{AB}^{ij} \\ P_{AB}^{ik} \\ P_{AB}^{il} \\ \vdots \\ P_{AB}^{iz} \end{pmatrix} \quad (2.24)$$

La matrice \mathbf{A} est la suivante :

$$\mathbf{A} = \begin{matrix} & \hat{x}_A & \hat{y}_A & \hat{z}_A \\ \begin{matrix} P_{AB}^{ij} \\ P_{AB}^{ik} \\ P_{AB}^{il} \\ \vdots \\ P_{AB}^{iz} \end{matrix} & \begin{pmatrix} x & x & x \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{pmatrix} \end{matrix} \quad (2.25)$$

2.5.3 Avec contraintes entre paramètres

Il est possible d'ajouter au modèle fonctionnel de Gauss-Markov des contraintes sur les paramètres estimés.

En fixant comme contraintes les coordonnées connues de la position du smartphone, il est possible d'évaluer directement les résidus sur les observations de double différence. L'avantage est de pouvoir observer directement le comportement des doubles différences calculées et de vérifier qu'il n'y ait pas d'erreur dans le calcul de ces observations.

En ajoutant les équations de contraintes $\mathbf{c}(\hat{\mathbf{x}})$, le modèle devient :

$$\begin{cases} \mathbf{l} + \hat{\mathbf{v}} &= \mathbf{f}(\hat{\mathbf{x}}) \\ 0 &= \mathbf{c}(\hat{\mathbf{x}}) \end{cases} \quad (2.26)$$

Les équations de contraintes sont linéarisées autour des paramètres approchés de la même manière que les équations d'observations :

$$\begin{cases} \mathbf{l} + \hat{\mathbf{v}} &= \mathbf{f}(\hat{\mathbf{x}}) + \left. \frac{\partial \mathbf{f}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \right|_{\hat{\mathbf{x}}} \cdot \delta \hat{\mathbf{x}} \\ 0 &= \mathbf{c}(\hat{\mathbf{x}}) + \left. \frac{\partial \mathbf{c}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \right|_{\hat{\mathbf{x}}} \cdot \delta \hat{\mathbf{x}} \end{cases} \quad (2.27)$$

Elles peuvent être réécrites sous forme matricielle :

$$\begin{cases} \delta \mathbf{l} + \hat{\mathbf{v}} &= \mathbf{A} \cdot \delta \hat{\mathbf{x}} \\ \mathbf{t} &= \mathbf{C} \cdot \delta \hat{\mathbf{x}} \end{cases} \quad (2.28)$$

avec les vecteurs suivants :

$$\delta \mathbf{l} = \mathbf{l} - \mathbf{f}(\overset{\circ}{\mathbf{x}}) \qquad \mathbf{t} = -\mathbf{c}(\overset{\circ}{\mathbf{x}}) \quad (2.29)$$

Avec les coordonnées connues du récepteur A fixées comme contraintes, les trois équations de contraintes écrites explicitement sont les suivantes :

$$0 = \hat{x}_A - \check{x}_A \quad (2.30)$$

$$0 = \hat{y}_A - \check{y}_A \quad (2.31)$$

$$0 = \hat{z}_A - \check{z}_A \quad (2.32)$$

Le vecteur \mathbf{t} est alors le suivant :

$$\mathbf{t} = \begin{pmatrix} \check{x}_A - \overset{\circ}{x}_A \\ \check{y}_A - \overset{\circ}{y}_A \\ \check{z}_A - \overset{\circ}{z}_A \end{pmatrix} \quad (2.33)$$

La matrice jacobienne formée par les dérivées partielles des équations de contraintes par rapport aux inconnues se calcule de la façon suivante, avec c le nombre d'équations de contraintes :

$$\mathbf{C} = \begin{pmatrix} \frac{\partial c_1}{\partial \hat{x}_1} & \frac{\partial c_1}{\partial \hat{x}_2} & \cdots & \frac{\partial c_1}{\partial \hat{x}_u} \\ \frac{\partial c_2}{\partial \hat{x}_1} & \frac{\partial c_2}{\partial \hat{x}_2} & \cdots & \frac{\partial c_2}{\partial \hat{x}_u} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_c}{\partial \hat{x}_1} & \frac{\partial c_c}{\partial \hat{x}_2} & \cdots & \frac{\partial c_c}{\partial \hat{x}_u} \end{pmatrix} \bigg|_{\overset{\circ}{\hat{x}}_1, \overset{\circ}{\hat{x}}_2, \dots, \overset{\circ}{\hat{x}}_3} \quad (2.34)$$

Ce qui donne la matrice \mathbf{C} suivante :

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.35)$$

L'estimation du vecteur des paramètres inconnus se fait par itérations successives comme pour le modèle de Gauss-Markov.

$$\overset{\circ}{\mathbf{x}}_{k+1} = \overset{\circ}{\mathbf{x}}_k + \delta \hat{\mathbf{x}}_k \quad (2.36)$$

Avec le vecteur des accroissements $\delta \hat{\mathbf{x}}$ qui se calcule de la façon suivante :

$$\begin{pmatrix} \delta \hat{\mathbf{x}} \\ \mathbf{k} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{A}^T \cdot \mathbf{P} \cdot \delta \mathbf{l} \\ \mathbf{t} \end{pmatrix} \quad (2.37)$$

Dès que l'itération a convergé, les résidus peuvent être calculés de façon identique à la méthode Gauss-Markov (équation 2.21). L'écart-type empirique de l'unité de poids s_0 et la matrice des cofacteurs des inconnues $\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ peuvent être calculés avec les formules suivantes :

$$s_0 = \sqrt{\frac{\hat{\mathbf{v}}^T \cdot \mathbf{P} \cdot \hat{\mathbf{v}}}{n - u + c}} \quad (2.38)$$

$$\begin{pmatrix} \mathbf{N} & \mathbf{C}^T \\ \mathbf{C} & 0 \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{pmatrix} \quad (2.39)$$

avec :

$$\mathbf{N} = \mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A} \quad (2.40)$$

2.5.4 Estimateur de Biber

L'estimateur par moindres carrés classique n'est pas robuste. Ainsi, une mauvaise observation influencera directement les résultats. C'est pour cette raison qu'il a été choisi d'implémenter l'estimateur de Biber dans l'application pour le calcul de la position. Cet estimateur permet de réduire le poids des observations considérées comme mauvaises quand leur résidu normé dépasse un certain intervalle $[-c, +c]$.

$$\rho(\hat{w}) = \begin{cases} \frac{1}{2} \hat{w}^2 & \text{si } \|\hat{w}\| < c \\ c \cdot \|\hat{w}\| - \frac{1}{2} c^2 & \text{si } \|\hat{w}\| \geq c \end{cases} \qquad \psi(\hat{w}) = \begin{cases} \hat{w} & \text{si } \|\hat{w}\| < c \\ c & \text{si } \|\hat{w}\| \geq c \end{cases} \quad (2.41)$$

Marche-à-suivre pour la M-estimation avec l'estimateur de Biber

1. Calculer une première solution par la méthode des moindres carrés traditionnelle :

$$\delta\hat{\mathbf{x}}_{(0)} = (\mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{P} \cdot \delta\mathbf{l} \quad (2.42)$$

2. A partir de cette étape, nous considérons nous trouver à l'itération (k), soit à la première convergence de la solution par les moindres carrés.

3. Calculer les vecteurs des résidus :

$$\hat{\mathbf{v}}^{(k)} = \mathbf{A} \cdot \delta\hat{\mathbf{x}}^{(k)} - \delta\mathbf{l} \quad (2.43)$$

4. Calculer les vecteurs des résidus normés :

$$\hat{\mathbf{w}} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} \frac{v_1}{\sigma_0 \cdot \sqrt{q_{\hat{v}\hat{v}}}_{11}}} \\ \frac{v_2}{\sigma_0 \cdot \sqrt{q_{\hat{v}\hat{v}}}_{22}}} \\ \vdots \\ \frac{v_n}{\sigma_0 \cdot \sqrt{q_{\hat{v}\hat{v}}}_{nn}}} \end{pmatrix} \quad (2.44)$$

avec la matrice des cofacteurs des résidus qui se calcule par :

$$\mathbf{Q}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} = \mathbf{Q}_{\mathbf{ll}} - \mathbf{A} \cdot \mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \cdot \mathbf{A}^T \quad (2.45)$$

5. Construire la matrice de poids \mathbf{W} :

$$\mathbf{W}^{(k)} = \begin{pmatrix} \mathbf{W}_1 & 0 & \dots & 0 \\ 0 & \mathbf{W}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{W}_n \end{pmatrix} \quad (2.46)$$

6. Calculer la M-estimation en intégrant la matrice des poids :

$$\delta\hat{\mathbf{x}}_{(k)} = (\mathbf{A}^T \cdot \mathbf{W}^{(k)} \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{W}^{(k)} \cdot \mathbf{P} \cdot \delta\mathbf{l} \quad (2.47)$$

7. Itérer à partir du point trois jusqu'à ce que le vecteur des paramètres à estimer ne varier plus.

2.5.5 Modèle stochastique

La matrice de variance-covariance des observations est la suivante :

$$\mathbf{K}_{\mathbf{ll}} = \begin{pmatrix} \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \dots & \sigma_{l_1 l_n} \\ \sigma_{l_2 l_1} & \sigma_{l_2}^2 & \dots & \sigma_{l_2 l_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{l_n l_1} & \sigma_{l_n l_2} & \dots & \sigma_{l_n}^2 \end{pmatrix} \quad (2.48)$$

avec :

σ_{l_i} écart-type de l_i

$\sigma_{l_i l_j}$ covariance entre l_i et l_j

La matrice des cofacteurs des observations $\mathbf{Q}_{\mathbf{ll}}$ est calculée à partir de la matrice de variance-covariance $\mathbf{K}_{\mathbf{ll}}$:

$$\mathbf{Q}_{\mathbf{ll}} = \frac{1}{\sigma_0^2} \mathbf{K}_{\mathbf{ll}} \quad (2.49)$$

$$\mathbf{P} = \mathbf{Q}_{\mathbf{ll}}^{-1} \quad (2.50)$$

2.5.6 Indicateurs Dilution Of Precision (DOPs)

1. GDOP : indicateur global sur la géométrie

$$\text{GDOP} = \sqrt{(q_{\hat{x}\hat{x}} + q_{\hat{y}\hat{y}} + q_{\hat{z}\hat{z}} + q_{\hat{t}\hat{t}})} \quad (2.51)$$

2. PDOP : indicateur sur le positionnement 3D

$$\text{PDOP} = \sqrt{(q_{\hat{x}\hat{x}} + q_{\hat{y}\hat{y}} + q_{\hat{z}\hat{z}})} \quad (2.52)$$

3. HDOP : indicateur sur le positionnement planimétrique (composante horizontale)

$$\text{HDOP} = \sqrt{(q_{\hat{x}\hat{x}} + q_{\hat{y}\hat{y}})} \quad (2.53)$$

4. VDOP : indicateur sur le positionnement altimétrique (composante verticale)

$$\text{VDOP} = \sqrt{(q_{\hat{z}\hat{z}})} \quad (2.54)$$

5. TDOP : indicateur sur l'erreur d'horloge du récepteur

$$\text{TDOP} = \sqrt{(q_{\hat{t}\hat{t}})} \quad (2.55)$$

Les indicateurs DOP sont calculés à partir des paramètres estimés et donnés dans le système de coordonnées de notre estimation soit dans un système géocentrique.

Pour connaître la précision locale, il est nécessaire de calculer les indicateurs de précision dans un système de coordonnées topocentriques.

S'il est possible d'exprimer la position topocentrique en fonction du vecteur des paramètres estimés $\mathbf{Q}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$, alors il est possible de faire une propagation analytique de la matrice de variance-covariance :

$$\mathbf{K}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} = \mathbf{F} \cdot \mathbf{K}_{\hat{\mathbf{q}}\hat{\mathbf{q}}} \cdot \mathbf{F}^T \quad (2.56)$$

avec :

$\mathbf{K}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}$ matrice de variance-covariance de la position dans un système topocentrique

\mathbf{F} matrice jacobienne pour passer du système géocentrique à topocentrique

$\mathbf{K}_{\hat{\mathbf{q}}\hat{\mathbf{q}}}$ matrice de variance-covariance des paramètres estimés de la position

La transformation d'un système géocentrique à topocentrique s'effectue par une rotation entre les deux systèmes de coordonnées. Ainsi, on obtient la matrice \mathbf{F} suivante où λ et ϕ sont les coordonnées de la position estimée calculée en coordonnées ellipsoïdales.

$$\mathbf{F} = \begin{pmatrix} -\sin(\phi) \cdot \cos(\lambda) & -\sin(\lambda) \cdot \sin(\phi) & \cos(\phi) \\ \sin(\lambda) & \cos(\lambda) & 0 \\ \cos(\phi) \cdot \cos(\lambda) & \cos(\phi) \cdot \sin(\lambda) & \sin(\phi) \end{pmatrix} \quad (2.57)$$

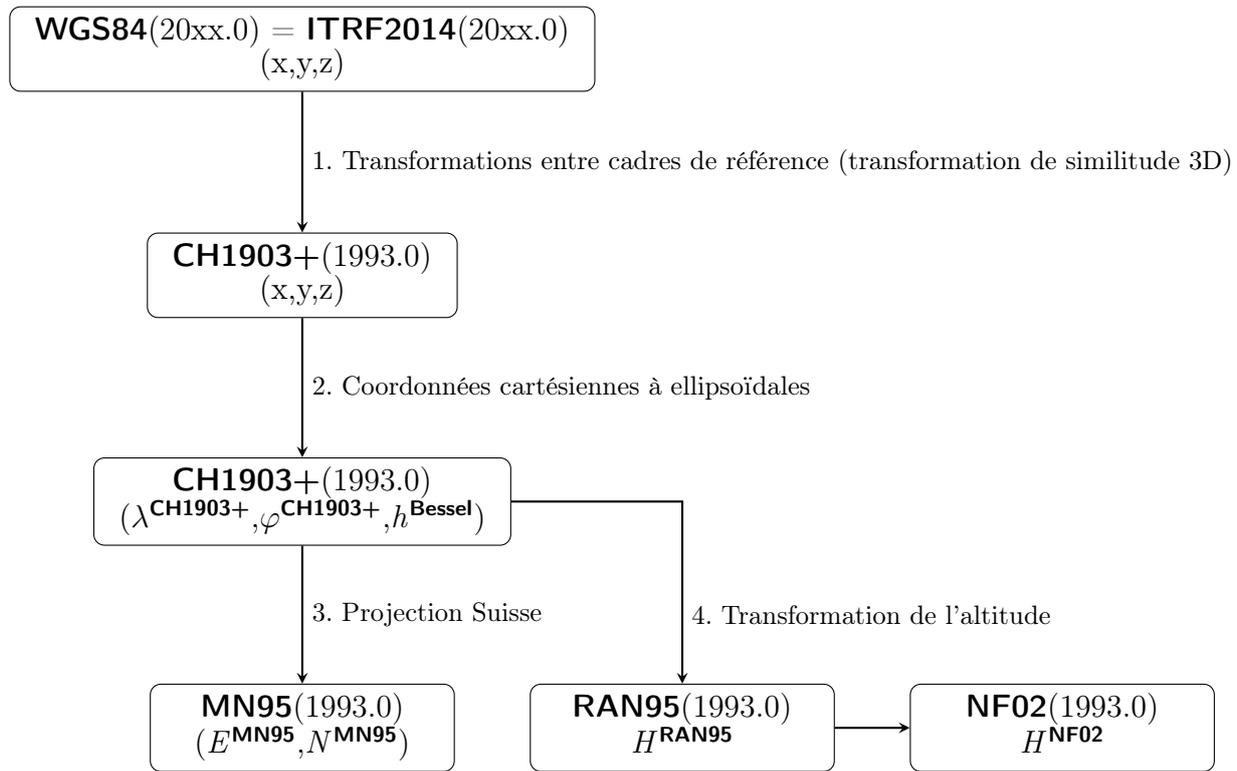
2.6 Systèmes de coordonnées

Le système de référence fondamental utilisé en géodésie et pour le positionnement par GNSS est le système de référence terrestre **ITRS** (*International Terrestrial Reference System*). C'est un système géocentrique dont les axes sont fixes par rapport à la surface de la Terre (Guillaume, 2021), appelé aussi **ECEF** de l'anglais *earth-centered, earth-fixed*.

Les coordonnées des satellites sont données dans le système **ITRS**, excepté pour les orbites GPS calculées avec les éphémérides qui sont données dans le système **WGS84**. Actuellement, l'écart entre les systèmes **ITRS** et **WGS84** est d'environ un centimètre. Il est donc admis, pour ce travail, que les coordonnées **ITRS** sont équivalentes aux coordonnées **WGS84**.

Dans la description des fichiers RINEX, les coordonnées de la position de l'antenne sont des coordonnées géocentriques et le système de référence recommandé est **ITRS**.

Le schéma 2.4 décrit le processus de conversion des coordonnées **WGS84** ellipsoïdales en coordonnées projetées **MN95** :

FIGURE 2.4 – Étapes pour la transformation des coordonnées **WGS84** à **MN95**

Les différentes étapes du processus sont décrites ci-dessous :

1. Transformation des coordonnées cartésiennes **WGS84** en **CH1903+**. Cette transformation regroupe plusieurs transformations de similitudes 3D entre cadres de référence. Les paramètres de la transformation de similitude sont publiés par les organisations nationales et internationales de géodésie lorsqu'un nouveau cadre de référence est publié. Voici les différentes étapes intermédiaires :

$$\begin{aligned}
 \mathbf{WGS84}(20xx.0) &\approx \mathbf{ITRF2014}(20xx.0) \\
 &\Rightarrow \mathbf{ITRF1993}(20xx.0) \\
 &\Rightarrow \mathbf{ETRF1993}(20xx.0) \\
 &\Rightarrow \mathbf{ETRF1993}(1993.0) \\
 &\Rightarrow \mathbf{CHTRF95}(1993.0) \\
 &\Rightarrow \mathbf{CH1903+}(1993.0)
 \end{aligned}$$

2. Calcul des coordonnées ellipsoïdales **CH1903+/Bessel** :

$$\lambda = \text{atan2}(y, x) \quad (2.58)$$

$$\varphi = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}} \cdot \frac{1}{1 - \frac{R_N}{R_N + h} \cdot e^2}\right) \quad (2.59)$$

Où :

$$h = \frac{\sqrt{x^2 + y^2}}{\cos \varphi} - R_N \quad (2.60)$$

$$R_N = \frac{a}{\sqrt{1 - e^2 \cdot \sin^2 \varphi}} \quad (2.61)$$

Avec :

- R_N Rayon de courbure de la section normale à l'ellipsoïde [m]
- a Demi-grand axe de l'ellipsoïde de Bessel 6378137.000 [m]
- e Excentricité de l'ellipsoïde de Bessel : 0.08169686312225269 [-]

3. Calcul des coordonnées projetées suisses dans le cadre de référence **MN95** :

- (a) Tout d'abord, la première projection conforme de l'ellipsoïde sur la sphère est effectuée en déplaçant le méridien d'origine λ_0 à l'ancien observatoire de Berne :

$$\phi_{\text{sph}} = 2 \cdot \arctan \left[k \cdot \tan^\alpha \left(\frac{\pi}{2} + \frac{\phi}{2} \right) \cdot \frac{1 - e \cdot \sin \phi}{1 + e \cdot \sin \phi} \right]^{\frac{\alpha \cdot e}{2}} - \frac{\pi}{2} \quad (2.62)$$

avec :

$$\begin{aligned} \lambda_{\text{sph}}, \phi_{\text{sph}} &: \text{Coordonnées sphériques normales} \\ \lambda_0 &= 7^\circ 26' 22.50'' = 7.4395833333^\circ \\ \alpha &= 1.0007291384304 \\ k &= 1.0030714396280 \end{aligned}$$

- (b) Ensuite, une rotation est appliquée selon le méridien d'origine pour déplacer le centre de projection sur le parallèle ϕ_0 de l'ancien observatoire de Berne :

$$\begin{aligned} \lambda_{\text{sph}}^t &= \arctan \left(\frac{\sin \lambda_{\text{sph}}}{\sin \phi_0 \cdot \tan \phi_{\text{sph}} + \cos \phi_0 \cdot \cos \lambda_{\text{sph}}} \right) \\ \phi_{\text{sph}}^t &= \arcsin (\cos \phi_0 \cdot \sin \phi_{\text{sph}} - \sin \phi_0 \cdot \cos \phi_{\text{sph}} \cdot \cos \lambda_{\text{sph}}) \end{aligned} \quad (2.63)$$

- (c) Enfin, la deuxième projection conforme de Mercator sphérique normale est utilisée pour obtenir les coordonnées projetées MN95 :

$$\begin{pmatrix} E(\lambda, \varphi) \\ N(\lambda, \varphi) \end{pmatrix} = \begin{pmatrix} E_0 + R_{\text{sph}} \cdot \lambda_{\text{sph}}^t \\ N_0 + R_{\text{sph}} \cdot \ln \left[\tan \left(\frac{\pi}{4} + \frac{\phi_{\text{sph}}^t}{2} \right) \right] \end{pmatrix} \quad (2.64)$$

Avec :

$$\begin{aligned} \lambda_{\text{sph}}^t, \phi_{\text{sph}}^t &: \text{Coordonnées sphériques obliques} \\ \phi_0 &= 46^\circ 54' 27.83324846'' = 46.9077314579^\circ \\ R_{\text{sph}} &= 6,378,815.90365\text{m} \end{aligned}$$

et :

$$\begin{pmatrix} E_0 \\ N_0 \end{pmatrix} = \begin{cases} +2'600'000 \text{ [m]} \\ +1'200'000 \text{ [m]} \end{cases}$$

4. Transformation de l'altitude ellipsoïdale dans les cadres de référence altimétriques suisses **RAN95** et **NF02** :

L'altitude ellipsoïdale est liée à l'altitude orthométrique **RAN95** par la relation suivante, avec N la cote du géoïde :

$$H^{\text{RAN95}} = h^{\text{Bessel}} - N \quad (2.65)$$

Le modèle de géoïde utilisé est le CHGeo2004 (Marti, 1997). Il est disponible ¹ sur le site de l'Office fédéral de topographie.

2.7 Systèmes temporels

La compréhension des différents systèmes de référence du temps est essentielle dans le concept de positionnement GNSS puisque la position est calculée à partir de la différence de temps entre la transmission du signal par le satellite et sa réception.

De plus, chaque constellation utilise son propre système de temps. Seule le temps GPS et sa relation avec les autres systèmes de temps sera décrit dans cette section puisque l'application utilise uniquement les satellites GPS.

1. <https://www.swisstopo.admin.ch/fr/geoid-fr>, consulté le 20 juillet 2024

Temps TAI

Le Temps Atomique International (TAI) est une échelle de temps très précise, continue et uniforme. Il est calculé à partir de l'ensemble des horloges atomiques de haute précision réparties dans le monde entier.

Temps UTC

Le temps universel coordonné UTC est le système de temps mondial. Il est aligné à l'échelle du temps TAI, mais ajusté selon les variations de la rotation terrestre. Cela se réalise concrètement par l'ajout de secondes intercalaires (en anglais : *Leap second*) lorsque le temps UTC s'éloigne trop du calendrier solaire.

Au moment de la rédaction de ce document, le nombre de secondes intercalaires est de 37 secondes.

$$Leapsecond_{UTC-TAI} = 37s$$

Temps UNIX

Le temps UNIX est défini comme le nombre de secondes écoulées depuis le 1^{er} janvier 1970 à 00:00:00 UTC, sans tenir compte des secondes intercalaires. Il est couramment utilisé pour représenter les dates et heures en informatique et en programmation.

Par exemple, le 1^{er} janvier 2024 à minuit UTC, le temps UNIX est de 1704067200 secondes. Ce nombre représente le nombre total de secondes écoulées depuis le 1^{er} janvier 1970.

Temps GPS

Le temps GPS ou GPST est une échelle de temps continue basée sur des horloges atomiques sur Terre dans les stations de contrôle ainsi que dans les satellites. Il a commencé à minuit entre le 5 et le 6 janvier 1980 UTC. A cette date, le nombre de secondes intercalaires entre le TAI et UTC était de 19 secondes.

Dans la communauté GNSS, le terme de seconde intercalaire est parfois utilisé pour définir les secondes d'écart entre le temps GPS et le temps UTC.

$$Leapsecond = Leapsecond_{UTC-TAI} - 19s$$

Au moment de la rédaction de ce document, la différence entre le temps UTC et GPS est donc de 18 secondes.

La différence entre le temps UNIX et le temps GPS est constante puisque les deux ont la même échelle de temps continue. Elle vaut 315964800 secondes ce qui correspond au nombre de secondes entre le début du temps UNIX et le début du temps GPS.

Le temps GPS est souvent exprimé en numéro de semaine de l'anglais *Week Number* (WN) et seconde de semaine GPS, de l'anglais *Time Of Week* (TOW). Le TOW est le nombre de seconde de la semaine GPS en cours donc à partir de 00:00 le dimanche. Sa valeur est donc comprise entre 0 et 604800 secondes, ce qui correspond au nombre de secondes écoulées durant une semaine complète.

Le tableau 2.3 regroupe les différentes relations entre les systèmes de temps utilisés dans ce travail :

TABLEAU 2.3 – Relations entre les systèmes de temps

Systèmes	Relation
GPST-UTC	GPST = UTC + 18s
GPST-UNIX	GPST = UNIX + 315964800s + 18s
GPST-TAI	GPST = TAI - 19s

Dates juliennes modifiées

De façon générale et indépendamment du système temporel utilisé, les dates du calendrier grégorien ne sont pas pratiques à utiliser pour effectuer des calculs. C'est pourquoi il est de coutume, lorsqu'il faut manipuler des temps, de transformer les dates calendaires en dates juliennes modifiées.

Chapitre 3

Conception et développement de l'application

3.1 Conception de l'application

3.1.1 Objectifs de l'application

Le but premier de l'application est de déterminer la position du smartphone en temps réel par positionnement différentiel sur le code. La spécificité de cette application est de pouvoir obtenir des coordonnées dans le cadre de référence suisse **MN95** et **RAN95** ou **NF02**.

3.1.2 Fonctionnalités de l'application

L'application prévoit les fonctionnalités suivantes :

- **Établissement de la connexion avec une station de base**
L'application permet de se connecter à n'importe quelle station de base qui diffuse ses données via NTRIP. Pour cela, l'utilisateur peut entrer l'adresse IP, le numéro du port, le point de montage et les informations d'authentification (nom d'utilisateur et mot de passe) sur l'application.
- **Visualisation des statuts des satellites GPS**
L'application permet d'afficher les satellites visibles et leur statut en temps réel. Les informations sont représentées sous forme d'un tableau avec l'identifiant, la constellation, les signaux traqués, le rapport signal-bruit, l'azimut et l'élévation des satellites. Le nombre de satellites total utilisé pour le calcul de la position par rapport au nombre de satellites visibles est indiqué.
- **Calcul de la position**
La position calculée du smartphone est mise à jour en temps réel.
- **Indicateurs de précision**
La qualité des signaux et la qualité de la position (erreur moyenne) sont fournis à l'utilisateur. Les indicateurs PDOP, HDOP et VDOP sont indiqués.
- **Enregistrement des données**
L'application permet d'enregistrer dans un fichier les données brutes du smartphone aux formats CSV et RINEX. Elle peut également enregistrer un fichier de résultats avec la position du smartphone calculée par DGPS.

L'application ne prévoit pas de fonctionnalité permettant à l'utilisateur de modifier les paramètres de calcul de la position, tels que l'angle de masque.

3.1.3 Plate-forme de développement et langage de programmation

Le langage de programmation utilisé pour le développement de l'application est Java.

L'application est développée sur Android Studio qui est l'environnement de développement intégré (IDE) officiel conçu et distribué par Google. La version du logiciel est la dernière en date du présent document, à savoir Android Studio Giraffe.

3.2 Développement de l'application

Le développement de l'application est décomposé en différents blocs correspondant aux éléments nécessaires minimums pour déterminer la position du smartphone par positionnement différentiel, soit les observations de double différence, la position des satellites et de la station de base (figure 3.1). Pour calculer les observations de double différence, il est nécessaire d'obtenir les pseudo-distances du smartphone ainsi que de la station de base. Pour calculer la position des satellites, il faut avoir accès aux données d'éphémérides.

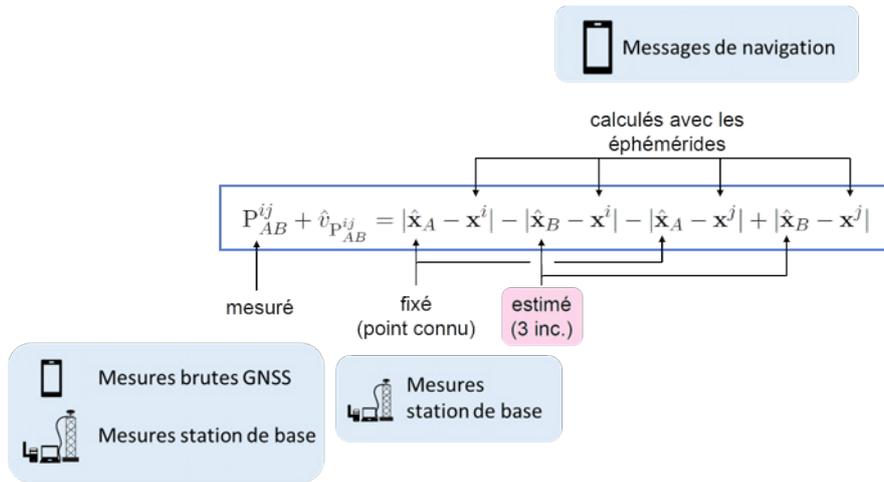


FIGURE 3.1 – Éléments nécessaires minimums pour déterminer la position du smartphone

Le développement de l'application est donc divisé selon les blocs suivants :

- Récupération des données brutes du smartphone à partir de l'API *android.location* et calcul des observations brutes de pseudo-distance, de mesure de phase et du décalage Doppler.
- Récupération des messages de navigation à partir de l'API *android.location* et calcul de la position des satellites.
- Connexion avec la station de base et récupération des mesures brutes de pseudo-distance et de sa position.
- Traitement GNSS différentiel et asynchrone.
- Formation des doubles différences.
- Estimation de la position du smartphone.

3.3 Mesures brutes du smartphone et calcul des observations

Cette section est basée sur le document de référence pour l'usage des mesures brutes GNSS sur les appareils Android publié par l'agence européenne GSA (2017).

Le système Android offre diverses fonctionnalités permettant aux utilisateurs d'accéder aux mesures brutes du récepteur GNSS via une API. Les API varient d'une version d'Android à l'autre. À partir d'Android 7, l'accès aux données brutes se fait via *android.location*. Cette dernière ne fournit pas directement les observations de pseudo-distance, de mesure de phase et de Doppler. Pour calculer ces observations, il faut récupérer plusieurs variables provenant des classes *GnssClock* et *GnssMeasurement* dont une partie est décrite dans les tableaux 3.1 et 3.2.

TABEAU 3.1 – Liste des variables de la classe *GnssClock*

Champ	Description
<i>TimeNanos</i>	Valeur de l'horloge interne du récepteur en nanoseconde
<i>BiasNanos</i>	Biais de l'horloge en sub-nanoseconde
<i>FullBiasNanos</i>	Différence entre l'horloge interne du récepteur et le vrai temps GPS depuis le 6 janvier 1980

TABLEAU 3.2 – Liste des variables de la classe `GnssMeasurement`

Champ	Description
<code>TimeOffsetNanos</code>	Décalage temporel auquel la mesure a été prise
<code>State</code>	État de synchronisation actuel du signal
<code>ReceivedSvTimeNanos</code>	Temps reçu du satellite GNSS, au moment de la mesure
<code>PseudorangeRateMetersperSecond</code>	Vitesse de variation de la mesure de code en m/s
<code>AccumulatedDeltaRangeMeters</code>	Distance cumulée depuis la dernière réinitialisation du système
<code>CarrierFrequencyHz</code>	Fréquence du signal en Hz

Temps GPS

Le temps GPS n'est pas directement fourni par Android, mais il est possible de le calculer à partir de l'horloge interne du récepteur GNSS, ainsi que de la différence de temps totale par rapport au temps GPS depuis le 6 janvier 1980. Cette dernière est estimée par le récepteur à partir des signaux reçus. Le temps GPS pour chaque époque de mesure se calcule donc de la façon suivante :

$$GPSTime = TimeNanos - (FullBiasNanos + BiasNanos) \quad (3.1)$$

L'horloge interne du récepteur, récupérée via `TimeNanos`, bat à une fréquence d'environ 1 Hz dès que les mesures GNSS sont activées sur le smartphone. Elle se réinitialise lorsque l'écran se verrouille ou lorsque l'application est complètement fermée. En comparaison avec les récepteurs géodésiques classiques, le temps GPS pour chaque époque de mesure ne bat pas parfaitement la seconde et il est possible d'avoir des différences de plusieurs millisecondes.

Le récepteur estime la différence totale entre son horloge interne et le temps GPS récupéré via `FullBiasNanos`. Ce biais total est donc négatif et vaut environ -1390000000 secondes en janvier 2024. Pour vérifier que le temps GPS peut être calculé, il est nécessaire de s'assurer que ce biais total a bien été estimé par le récepteur via la valeur de `hasFullBiasNano` qui doit être vérifiée.

Mesure de pseudo-distance

Comme décrit dans la section 2.3.1, la pseudo-distance peut être calculée à partir de la différence de temps entre le temps transmis et le temps reçu du signal, multiplié par la vitesse de la lumière. Le nom des variables est modifié dans ce chapitre de manière à suivre le document de référence de l'agence européenne GSA (2017), mais aussi car les temps mesurés par le récepteur sont rattachés à des systèmes de temps différents selon la constellation.

Ainsi l'observation de pseudo-distance est notée :

$$\rho = (t_{Rx} - t_{Tx}) \cdot 10^{-9} \cdot c \quad (3.2)$$

Avec :

- ρ Observation de pseudo-distance [m]
- t_{Rx} Temps au moment de la réception du signal [ns]
- t_{Tx} Temps au moment de la transmission du signal [ns]
- c Vitesse de la lumière dans le vide [m/s]

Pour calculer les pseudo-distances, les temps au moment de la réception et de la transmission doivent être exprimés dans le même système de temps. Généralement, c'est le temps GPS qui est utilisé comme système de temps de référence.

Le temps transmis par le satellite t_{Tx} s'obtient directement via la classe `GnssMeasurement` :

$$t_{Tx} = ReceivedSvTimeNanos \text{ [ns]} \quad (3.3)$$

Ce temps dépend du type de satellite. Par exemple, pour les satellites appartenant à la constellation GPS, le temps transmis est relatif au début de la semaine GPS en cours (TOW). Il est donc compris entre 0 et 604 800 secondes. Il est systématiquement nécessaire de vérifier que le récepteur a pu résoudre l'ambiguïté de temps (voir section 2.2) et cela peut être fait en vérifiant que le TOW a été entièrement décodé dans le statut de suivi du satellite via `State`.

Le temps au moment de la réception du signal est calculé selon l'équation suivante :

$$t_{Rx_{GNSS}} = TimeNanos + TimeOffsetNanos - (FullBiasNanos(1) + BiasNanos(1)) \text{ [ns]} \quad (3.4)$$

Seules les premières valeurs de *FullBiasNanos* et *BiasNanos* doivent être utilisées pour calculer tous les temps de réception jusqu'à ce qu'il y ait une discontinuité dans l'horloge interne du récepteur.

Finalement, pour calculer la pseudo-distance, le temps de réception est converti dans le même système de temps que le temps transmis, ici en TOW, en prenant le module de temps de réception par rapport au nombre de nanosecondes contenues dans une semaine.

$$t_{Rx} = \text{mod}(t_{Rx_{GNSS}}, NumberNanoSecondsWeek) \text{ [ns]} \quad (3.5)$$

Mesure de phase

La mesure de phase est déterminée de la manière suivante :

$$\varphi = AccumulatedDeltaRangeMeters/\lambda \quad (3.6)$$

Avec λ la longueur d'onde du signal déterminée avec la fréquence du signal obtenue via *CarrierFrequencyHz* et la vitesse de la lumière : $\lambda = c/f$.

La mesure de phase doit être utilisée uniquement si le statut obtenu via *AccumulatedDeltaRangeState* est valide.

Mesure Doppler

Le décalage Doppler se calcule à partir de la vitesse de variation de la pseudo-distance à l'époque de mesure obtenue en mètre par seconde :

$$dopplershift = -PseudorangeRateMetersperSecond/\lambda \quad (3.7)$$

3.4 Messages de navigation et éphémérides

Chaque satellite reçoit depuis les antennes au sol ses données de navigation qu'il peut ensuite renvoyer à l'utilisateur sous la forme d'un message de navigation (voir section 2.2). Les messages de navigation contiennent toutes les informations nécessaires au calcul de la position des satellites. Cela inclut notamment les paramètres des éphémérides, les corrections des horloges, les paramètres du modèle ionosphérique et les almanachs.

L'API *android.location* permet d'accéder aux messages de navigation via la classe *GnssNavigationMessage* décrite dans le tableau 3.3.

Voici un exemple de message de navigation de l'on obtient avec la classe *GnssNavigationMessage*.

```

1  GnssNavigationMessage:
2  Type           = GPS L1 C/A
3  Svid           = 32
4  Status         = ParityPassed
5  MessageId      = 11
6  SubmessageId   = 5
7  Data           = {34, -64, -3, 41, 38, 44, -115, 32, 18, -62, -116, 72, 3, -61, -33, -15,
                    63, 79, -64, 33, 40, 67, 60, 40, 38, -12, 35, -33, 38, -25, 1, -93, 61, 41, -36, -118, 45,
                    63, -24, 64 }
```

TABLEAU 3.3 – Liste des variables de la classe *GnssNavigationMessage*

Champ	Description
<i>Type</i>	Type du message de navigation (TYPE_GPS_L1CA : 257)
<i>Svid</i>	Identifiant du satellite
<i>Status</i>	Statut du message de navigation (contrôle de parité)
<i>MessageId</i>	Identifiant du message
<i>SubmessageId</i>	Identifiant du sous-message correspondant au type
<i>Data</i>	Données du message de navigation

Chaque message de navigation correspond à un satellite et un type de signal spécifique.

Décodage des messages de navigation

Les données des messages de navigation sont obtenues en octets selon le format big endian ou MSB de l'anglais *Most Significant Byte*. Le format des données de navigation diffère en fonction de la constellation et du type de signal. La documentation d'Android fournit la description du format pour GPS L1 C/A suivante :

Pour GPS L1 C/A, chaque sous-message contient 10 mots de 30 bits. Chaque mot (30 bits) doit être inséré dans les 30 derniers bits d'un mot de 4 octets (sauter B31 et B32), avec le MSB en premier, pour un total de 40 octets, couvrant une période de temps de 6, 6 et 0.6 secondes, respectivement.

Le décodage des messages de navigation est effectué à partir du module pseudorange de l'application GNSS-Logger qui propose une implémentation pour le décodage des messages de navigation des satellites GPS. Les éphémérides sont directement récupérées à partir des messages de navigation brutes. Les messages sont ensuite stockés dans la classe intitulée `GpsNavigationMessageStore` qui permet d'extraire les informations contenues dans les messages de navigation GPS à partir des octets bruts. Cette classe fournit une méthode pour passer les octets bruts et une autre pour construire et remplir une nouvelle classe intitulée `GpsNavMessageProto` avec les différents éphémérides pour chaque satellites.

De façon plus détaillée, l'application GNSSLogger réalise les étapes suivantes :

- Récupération des messages de navigation brutes de la classe `GnssNavigationMessage`.
- Traitement d'un nouveau message avec vérification que le message est de type GPS et qu'il contient des données brutes (longueur de 40 octets).
- Décodage des octets selon l'identifiant du sous-message.
 - Sous-trame n° 1 : contient les paramètres de correction sur les horloges des satellites, le numéro de la semaine GPS.
 - Sous-frames n° 2 et 3 : données des éphémérides nécessaires pour déterminer la position précise des satellites.
 - Sous-frames n° 4 et 5 : almanach et données sur l'état de santé des satellites GPS, données de correction ionosphérique et différence de temps entre l'UTC et l'heure GPS.

Récupération des éphémérides à partir d'une source externe

Il est également possible d'obtenir les données des éphémérides à partir d'une source externe. De façon générale, dès lors que des informations proviennent d'une source autre que les signaux des satellites GNSS, le terme GNSS assisté ou A-GNSS est utilisé. Il est généralement plus rapide pour un appareil de récupérer la totalité des données de navigation depuis un serveur que de décoder les messages de navigation directement à partir des signaux des satellites.

Google a développé dans ce but un serveur selon le protocole SUPL (*Secure User Plane Location*) à partir duquel il est possible d'obtenir les données des éphémérides. Celui-ci est utilisé dans l'application GNSSLogger.

L'application prototype est développée en utilisant uniquement les données des signaux des satellites.

3.5 Données de la station de base

3.5.1 Méthode de transmission

Pour calculer sa position, le smartphone a besoin de connaître les coordonnées précises de la station de base ainsi que les données brutes mesurées par celle-ci. Il faut donc que la station de base transmette en temps réel ses mesures brutes et que la station itinérante les reçoive. La transmission des données est effectuée par le réseau internet avec le protocole de communication NTRIP.

Protocole NTRIP

Le protocole de communication utilisé pour la transmission des données est le NTRIP, en anglais *Networked Transport of RTCM via Internet Protocol*. Ce protocole est open-source et largement utilisé pour la diffusion de données GNSS.

Le protocole de transmission NTRIP fonctionne sur la base des standards HTTP avec d'un côté le serveur qui envoie les données et de l'autre le client qui les reçoit. Une authentification entre le serveur et le client est nécessaire.

Une fois l'authentification vérifiée, le serveur transmet la chaîne de caractères « ICY 200 OK ».

Pour lire le flux d'octets, la classe `InputStream` est utilisée. Elle permet de lire le flux octet par octet. La valeur de l'octet est un entier compris entre 0 et 255. Si aucun octet n'est disponible, car c'est la fin du flux de données, alors la fonction renvoie -1.

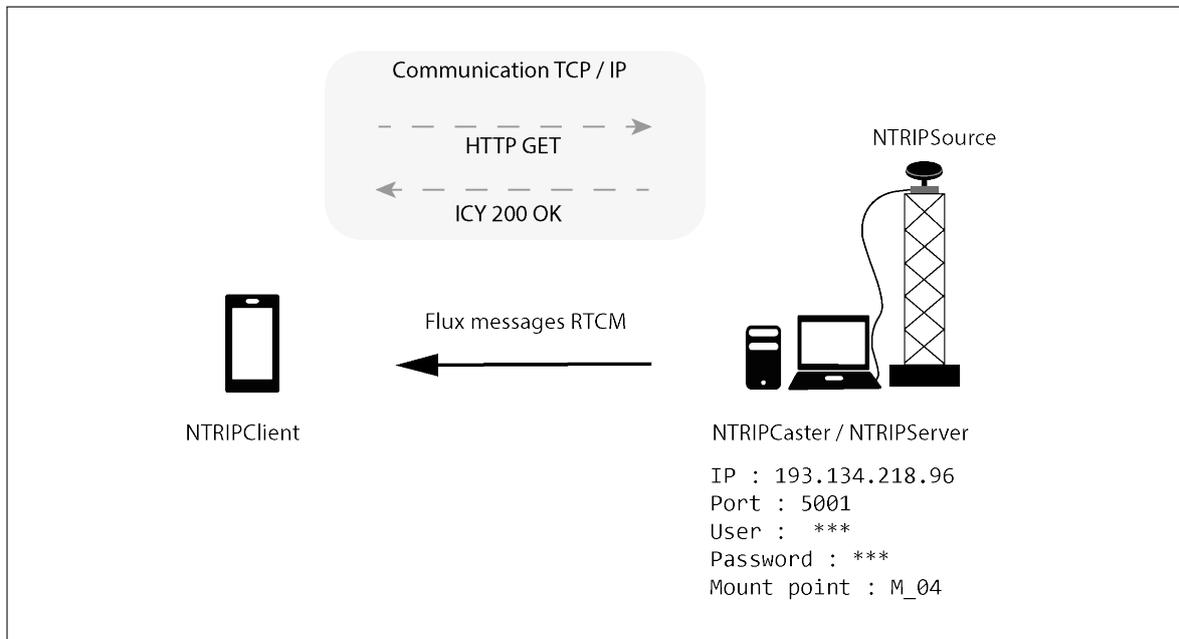


FIGURE 3.2 – Schéma de la transmission des données de la station de base

3.5.2 Format et contenu des données de la station de base

Les données brutes de la station de base sont transmises sous forme de messages RTCM. Le format RTCM est un format binaire composé d'une suite d'octets, chacun composé de 8 bits. Pour interpréter ces données brutes, il est nécessaire de déchiffrer cette suite d'octets selon les spécifications du format RTCM décrites dans le standard RTCM (Radio Technical Commission for Maritime Services (RTCM), 2013).

La structure pour la transmission de messages RTCM est décrite dans le tableau 3.4.

TABLEAU 3.4 – Structure de la transmission de messages RTCM

Préambule	Reservé	Longueur du message	Message	CRC
8 bits 11010011	6 bits -	10 bits Longueur du message en octet	longueur variable 0-1023 octets	24 bits QualComm definition CRC-24Q

Le décodage des messages RTCM est réalisé à partir des fichiers sources Java issu du projet goGPS conçu pour le traitement des observations GNSS (https://github.com/goGPS-Project/goGPS_Java, consulté le 9 février 2024). Les classes `Decode1006Msg`, `Decode1004Msg` ont été simplifiés et légèrement adaptée.

La lecture du flux de données entrant donne comme résultat des octets. Pour déchiffrer complètement les messages RTCM, il est parfois nécessaire de convertir les octets reçus en bits.

La détection des messages RTCM s'effectue de façon séquentielle selon les étapes suivantes :

- Lecture d'un octet en décimal.
- Vérifier que l'octet est égal au préambule (la suite de bits du préambule correspond à la valeur décimale 211).
- Lecture des deux octets, soit des 16 prochains bits.
- Conversion en octets des bits 7 à 16 pour obtenir la longueur du message.
- Lecture du message complet et stockage des octets dans un tableau et de leur équivalent en bits dans un second tableau.
- Lecture et décodage des deux premiers octets du message pour déterminer le type de message.
- Décodage du message spécifique.

Message 1004

Le message RTCM 1004 fournit les données brutes de la constellation GPS (signaux L1 et L2). Il se compose de deux parties. La première partie est un en-tête contenant notamment le temps en semaine GPS (TOW) et le nombre de satellites observés à cet instant. La deuxième partie décrit les observations pour chacun des satellites tel que le type de code, l'identifiant du satellite, la mesure de pseudo-distance, de phase et le C/N_0 . Ainsi, la longueur du message varie en fonction du nombre de satellites.

Le TOW étant une valeur relative, il est nécessaire de déterminer la semaine GPS en cours pour calculer le temps GPS total de l'époque de mesure. Cela peut être réalisé via l'accès au temps du smartphone (*SystemTime*). Celui-ci est exprimé en milliseconde UNIX. Il est donc nécessaire dans un premier temps de convertir ce temps en temps GPS (section 2.7).

Message 1006

Le message 1006 contient les informations sur l'antenne de la station de base, dont notamment ses coordonnées **ECEF** qui sont nécessaires pour le calcul de la position en DGPS.

Les coordonnées du point de montage M04 de la station de base du réseau de la HEIG-VD utilisé sont les suivantes :

$$\mathbf{x}_{M04}^{\text{CHTRF}} = \begin{pmatrix} 4346427.0535 \\ 507507.1239 \\ 4625379.3745 \end{pmatrix} \quad (3.8)$$

3.5.3 Validation des messages RTCM

Les messages RTCM envoyés par la station de base sont vérifiés en comparant avec le logiciel SNIP qui permet de se connecter à un serveur NTRIP et de lire le contenu des messages RTCM (*SNIP NTRIP Caster*, 2024).

3.6 Traitement GNSS différentiel et asynchrone

Dès lors que le calcul de la position s'effectue par positionnement différentiel, le système de mesure comporte plusieurs récepteurs. Pour former les doubles différences, il faut prendre en compte une possible désynchronisation des horloges des récepteurs. Généralement, les récepteurs géodésiques se fixent directement sur un signal synchronisé à la seconde du temps GPS, tandis que les récepteurs peu coûteux embarqués dans les smartphones ne font pas ces corrections directement.

La synchronisation des observations de la station de base est réalisée selon les étapes suivantes :

1. Dès que le smartphone reçoit des mesures, l'époque de mesure en temps GPST est calculée.
2. L'application récupère la liste des observations transmises par la station de base qui sont déchiffrées en continu et récupère l'observation la plus proche ainsi que la deuxième observation la plus proche.
3. Ensuite, l'observation de pseudo-distance de la station de base à l'époque de mesure est calculée par une interpolation linéaire avec les observations les plus proche.

3.7 Formation des doubles différences

Le calcul des doubles différences est effectué à partir des pseudo-distances du smartphone et de la station de base interpolées à l'époque de mesure du smartphone lorsque l'époque de mesure n'est pas la même.

Pour former les simples différences, il est nécessaire de vérifier que, pour la même époque, le smartphone et la station de base ont mesuré le même satellite et que le signal a été correctement décodé.

Ainsi, les étapes suivantes sont entreprises pour former les simples différences à chaque époque :

- Parcourir toutes les observations provenant du smartphone, c'est-à-dire de chaque satellite utilisé dans le calcul de la position.
- Vérifier que le signal est inclus dans les observations de la station de base.
- Vérifier que le smartphone a mesuré le même satellite.
- Vérifier que le smartphone a décodé le même signal.
- Calculer les simples différences entre la station de base et le smartphone en soustrayant les pseudo-distances.
- Pour former les doubles différences la stratégie utilisée est d'utiliser un satellite de référence. Le satellite le plus au zénith est utilisé comme satellite de référence.

3.8 Estimation de la position

Le calcul principal de l'application est une estimation par la méthode des moindres carrés avec l'estimateur de Biber. Cependant, deux autres calculs ont été implémentés : un calcul par la méthode des moindres carrés classique et un calcul contraint.

Le tableau 3.5 présente les paramètres utilisés pour ces calculs.

TABLEAU 3.5 – Paramètres de calcul

Paramètre	Valeur
Écart-type à priori sur les observations de double différence	1.0 m
Valeur du seuil sur les résidus normés pour définir la matrice des poids W	2.5
Fréquence	L1
Masque d'élévation	15°
Constellation	GPS
Rapport signal-bruit	18 dB-Hz

3.9 Structure de l'application

Cette section décrit comment concrètement les différentes étapes du calcul différentiel sont implémentées et comment l'application est structurée.

Une première partie décrit les bases du développement d'application Android et la structure initiale d'une application. La seconde partie présente la logique de l'application avec les processus centraux de l'application schématisés à l'aide de diagrammes et la description des différentes classes utilisées pour le calcul de la position. L'interface utilisateur est présentée à la suite. Finalement, les quelques fonctionnalités annexes au processus central du calcul de la position sont développées à la fin de cette section.

3.9.1 Bases du développement d'application Android

Un projet Android Studio contient un ou plusieurs modules contenant les fichiers de code source et les fichiers ressources.

La structure des modules d'application est divisée en une partie logique qui contient le code source Java, une partie visuelle pour gérer l'interface utilisateur et une partie pour configurer l'application.

Selon cette division, chaque module d'application contient les dossiers suivantes :

- **manifests** : contient le fichier de configuration pour l'application : *AndroidManifest.xml*.
- **java** : contient le code source Java (les différentes classes).
- **res** : contient les fichiers ressources et les différentes mise en page (*layout*) au format XML.

L'unité de base sur laquelle une application Android tourne est appelée une activité. C'est une classe qui s'initialise au moment du lancement de l'application et qui appelle des fonctions de rappel spécifiques, appelées *callback*. Les activités sont déclarées dans le fichier *AndroidManifest*. Par exemple, l'icône de l'application ainsi que le thème graphique sont des éléments également déclarés dans ce fichier.

Autorisations nécessaires

C'est dans le fichier *AndroidManifest* que les permissions utilisées par l'application sont déclarées. Dans certains cas, l'accès à ces ressources doit être explicitement demandé à l'utilisateur.

L'application prototype réalisée dans le cadre de ce travail nécessite les permissions d'accès à la position et à internet suivantes :

```

1 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
2 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
3 <uses-permission android:name="android.permission.INTERNET" />

```

Pour pouvoir accéder au service de positionnement du smartphone, il est nécessaire de demander l'autorisation à l'utilisateur de façon explicite. Ceci est réalisé simplement via la classe `LocationManager`.

Compilation

La compilation de l'application s'effectue avec le moteur de production Gradle, qui permet d'automatiser les actions de compilation, la gestion des dépendances et les configurations de build. Gradle utilise des fichiers de configuration, généralement nommés *build.gradle*, où les dépendances entre modules sont définies et où il est possible d'ajouter d'autres modules ou bibliothèques. Ces fichiers spécifient également les configurations de compilation, comme le niveau de SDK et les paramètres de déploiement, facilitant ainsi la gestion cohérente et efficace du projet.

Cycle de vie d'une application Android

Lorsque l'application est lancée, la classe de l'activité principale s'enclenche et les fonctions de rappel suivantes sont appelées selon le cycle de vie de l'application (voir figure 3.3) :

- `onCreate(Bundle savedInstanceState)` : cette méthode est appelée lorsque l'activité est créée pour la première fois. C'est l'endroit où sont effectués généralement les initialisations de base, comme les éléments de l'interface utilisateur, la récupération de données initiales, la configuration des gestionnaires d'événements, etc. Le paramètre `savedInstanceState` est utilisé pour restaurer l'état précédent de l'activité si elle a été détruite et recréée en raison d'une rotation de l'écran ou d'un autre événement.
- `onStart()` : cette méthode est appelée juste avant que l'activité ne devienne visible à l'utilisateur, mais elle n'est pas encore en premier plan. Des opérations d'initialisation supplémentaires peuvent être effectuées ici.
- `onResume()` : cette méthode est appelée lorsque l'activité devient visible à l'utilisateur et entre en premier plan. C'est l'endroit où les opérations mises en pause dans `onPause()` sont reprises.
- `onPause()` : cette méthode est appelée lorsque l'activité n'est plus visible à l'utilisateur et passe en arrière-plan. C'est l'endroit où les opérations en cours, comme la mise à jour de l'interface utilisateur, doivent être arrêtées et où les ressources doivent être libérées.
- `onStop()` : cette méthode est aussi appelée lorsque l'activité n'est plus visible à l'utilisateur et mise en arrière-plan. Elle est souvent utilisée pour effectuer des opérations de nettoyage plus importantes.
- `onDestroy()` : cette méthode est appelée lorsque l'activité est en cours de destruction. C'est le dernier appel avant que l'activité ne soit complètement supprimée. C'est l'endroit où sont effectués des nettoyages finaux, comme la désinscription des observateurs, etc.

Le code de l'activité est exécuté par le thread principal, appelé également UI thread. Il est responsable des interactions avec l'utilisateur et de l'affichage, ce qui le place au coeur de toute application Java sous Android. Il est implémenté dans une boucle infinie qui lui permet de surveiller en continu les actions et d'effectuer les tâches correspondantes, par exemple l'affichage d'un texte spécifique lorsque l'utilisateur clique sur un bouton.

Le thread principal exécute les méthodes `onCreate()`, `onStart()` et `onResume()` avant de rentrer dans la boucle qui gère les actions de l'utilisateur. Puis à l'inverse, lorsque l'application se ferme, il exécute les méthodes `onPause()`, `onStop()` et `onDestroy()` après être sorti de cette boucle.

Le thread ne doit pas être surchargé par des instructions consommatrices de temps, car il est bloqué à l'exécution de cette tâche et ne surveille plus les autres instructions. La solution pour éviter ce problème est d'exécuter les instructions consommatrices de temps dans des threads dédiés. Ainsi durant l'exécution des instructions, l'UI thread pourra continuer de surveiller les actions de l'utilisateur.

De plus, pour assurer la cohérence de ce qui est affiché, l'UI thread doit être le seul thread à effectuer des modifications sur l'affichage.

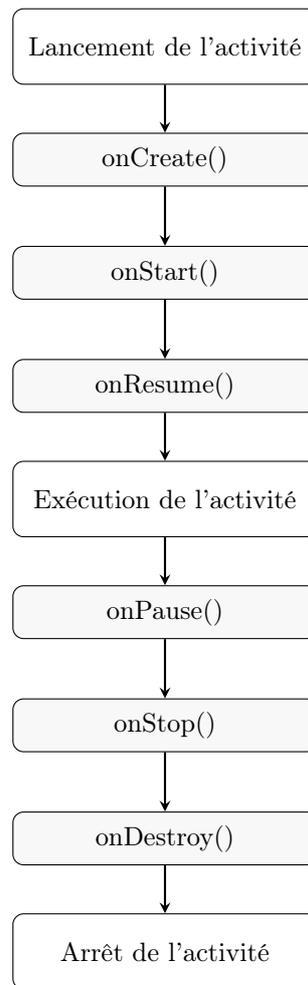


FIGURE 3.3 – Cycle de vie d'une activité

3.9.2 Logique de l'application

Le processus ainsi que les différentes étapes de l'application sont résumés dans la figure 3.4. La majorité des processus sont effectués dans la méthode `onCreate()` appelée dès que l'application est lancée.

La première étape consiste à vérifier les permissions liées à la position qui doivent être explicitement demandées par l'utilisateur. Si celles-ci n'ont pas encore été acceptées, l'application les demande via une fenêtre pop-up.

Puis, la classe `BaseStation` est initialisée et l'enregistrement des données activé. L'enregistrement des données s'effectue dans un thread dédié. Lors de sa construction, la station de base initialise un récepteur (classe `Receiver`) qui s'occupe de stocker les messages reçus. Dès qu'un message d'observations brutes GPS est déchiffré, il est ajouté à la liste des observations du récepteur. Quant aux coordonnées de la station de base déchiffrées à partir du message 1006, elles sont stockées dans l'antenne associée à la station de base (classe `StationaryAntenna`).

L'étape suivante est l'initialisation des classes abonnées aux données brutes. La structure du code pour l'accès aux données brutes du smartphone est inspirée de l'application open source `GNSSLogger` et utilise le concept d'observateur (en anglais : *observer pattern*). Le fonctionnement de ce patron est le suivant : une classe intéressée à recevoir des informations sur des événements spécifiques, ici les mesures brutes GNSS, peut s'y abonner. Elle sera ainsi notifiée lorsque l'un des événements se produit. En Java, cela se concrétise par la création d'une interface qui écoute les événements et notifie toutes les classes souscrites. Pour s'abonner aux événements, le mot-clé Java utilisé est **implements**.

La classe dédiée à la récupération des données brutes s'appelle `MeasurementProvider`. C'est dans celle-ci que le `LocationManager`, la classe de base de l'API `android.location` permettant d'accéder aux données brutes du récepteur, est implémentée. Cet élément essentiel offre l'accès aux services de localisation du système, permettant ainsi aux applications de recevoir des mises à jour régulières sur la position de l'appareil, les spécifications relatives à l'antenne et les caractéristiques du récepteur.

L'interface permettant de notifier les abonnées est le `MeasurementListener`. Le schéma 3.5 présente les classes abonnées au `MeasurementListener`.

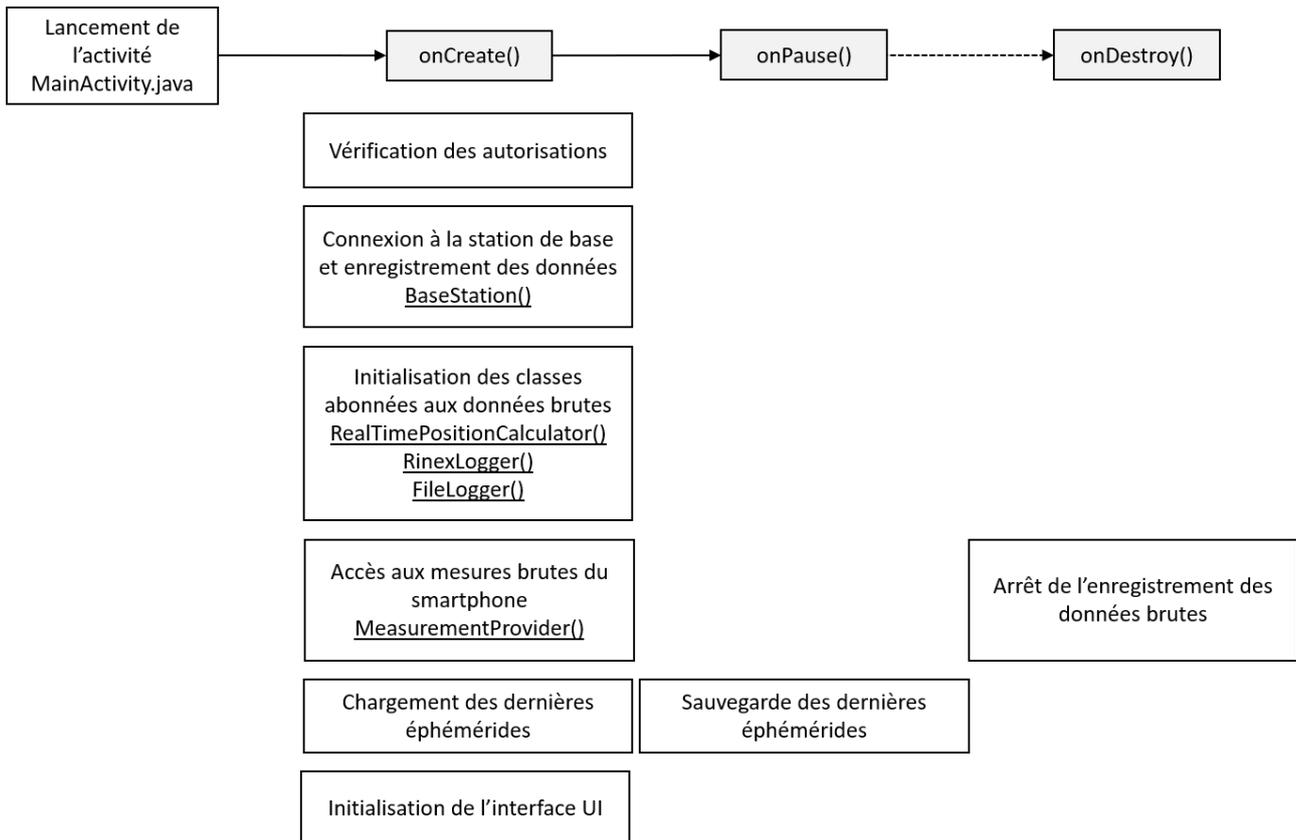


FIGURE 3.4 – Processus général du fonctionnement de l'application

Dès que l'utilisateur quitte l'application, la méthode `onPause()` est appelée et les éphémérides sont sauvegardées dans un fichier texte dans la stockage interne de l'application.

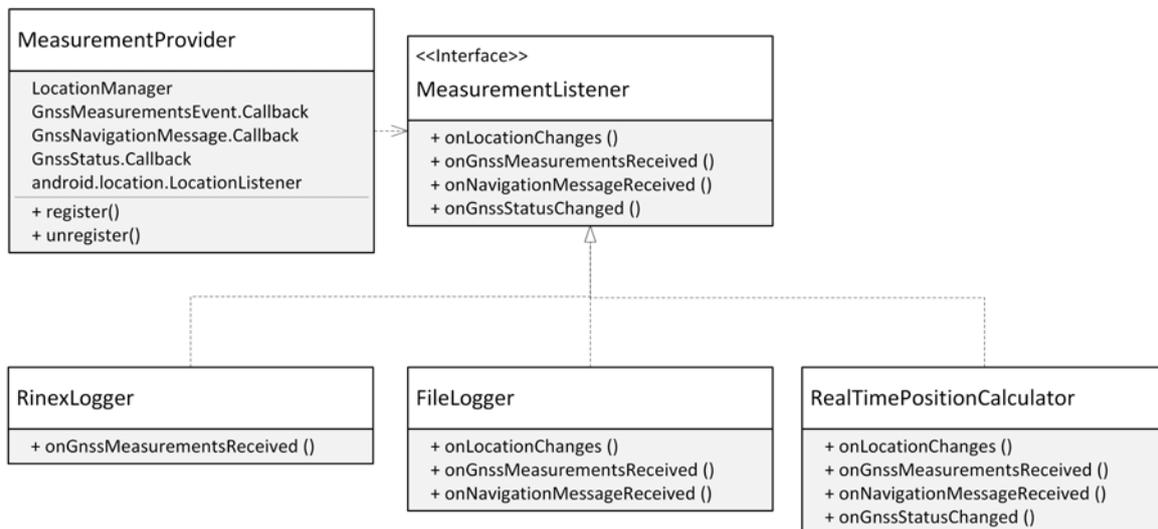


FIGURE 3.5 – Diagramme de l'accès aux données brutes

3.9.3 Interface utilisateur

L'interface utilisateur est structurée en fragments. Ce sont des composants modulaires qui représentent des parties distinctes de l'interface utilisateur et simplifient considérablement la gestion des changements de l'interface utilisateur.

Un menu de navigation permet aux utilisateurs de passer facilement d'un fragment à un autre. L'application utilise trois fragments pour structurer son interface utilisateur. Seuls les deux premiers fragments ont été développés. Chaque fragment est responsable d'une section spécifique de l'interface utilisateur et est associé à une ressource de mise en page. L'application contient les fichiers de mise en page suivants :

- activity_main.xml
- fragment_main.xml
- fragment_status.xml
- fragment_base.xml (non développé)

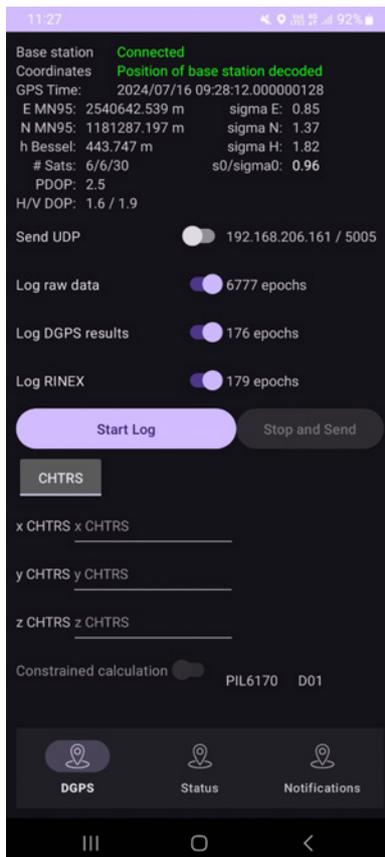


FIGURE 3.6 – Capture d'écran de la mise en page principale de l'application

La figure 3.6 montre l'écran d'accueil et principal de l'application.

Les éléments graphiques des mises en page sont décrits par des balises qui correspondent à des objets appelés *vues*. C'est depuis ces vues qu'il est possible de récupérer les actions de l'utilisateur ou de mettre à jour les éléments graphiques de l'application.

L'application utilise une librairie de liaison de données (*databinding*). Comme son nom l'indique, cela permet de créer une liaison directe entre les éléments de l'interface utilisateur définis dans les fichiers XML et les objets de données dans le code Java. Cela simplifie le processus de mise à jour de l'interface utilisateur en réponse aux changements, l'accès aux éléments de l'interface utilisateur et aux sources de données, et réduit le code nécessaire pour manipuler les vues.

Comme il est généralement coutume en programmation, le traitement de l'interface utilisateur est séparé de la logique dite métier de l'application. Cela se concrétise par l'implémentation de classes dédiées uniquement à réagir avec l'interface utilisateur :

- HomeFragment
- StatusFragment
- BaseFragment (non développé)

Chacune de ces classes hérite de la classe `Fragment`. Cette déclaration s'effectue avec le mot-clé *extends* au début de la classe.

Pour mettre à jour l'interface utilisateur depuis les classes Java ou pour récupérer les actions de l'utilisateur, une approche structurée est adoptée en utilisant des classes internes pour chaque fragment. Chaque fragment a une classe interne dédiée qui gère les interactions spécifiques à ce fragment.

Chaque fragment contient une classe interne qui étend les fonctionnalités du fragment et gère les interactions utilisateur spécifiques à ce fragment.

La classe interne est initialisée et configurée dans les classes Java qui doivent interagir avec l'interface du fragment. Cela permet une interaction directe avec les éléments de l'interface utilisateur du fragment concerné. La classe interne capte les actions de l'utilisateur, telles que les clics sur les boutons ou les entrées de texte, et met à jour l'interface utilisateur en conséquence. Elle permet également de déclencher des actions spécifiques en réponse aux interactions utilisateur.

3.9.4 Fonctionnalités

Enregistreurs

Comme illustré sur la figure 3.6, l'application permet d'enregistrer différents fichiers. Pour cela, l'utilisateur coche les fichiers qui l'intéressent avant d'appuyer sur le bouton « start log », qui va déclencher l'écriture des fichiers. Une fois l'acquisition terminée, il appuie sur le bouton « stop log » qui ferme l'écriture des fichiers et libère les ressources.

L'application compte trois classes responsables d'enregistrer des données :

- `FileLogger` : enregistre les données brutes.
- `RinexLogger` : enregistre un fichier RINEX contenant les observations de pseudo-distance calculées.
- `RealTimePositionCalculator` : enregistre plusieurs fichiers de résultat des calculs.

Chacune de ces classes a directement accès aux données brutes du récepteur puisqu'elles implémentent le `MeasurementListener`.

Puisque l'écriture est déclenchée par des événements pouvant entrer en concurrence, une écriture propre et cohérente des fichiers est garantie en protégeant l'accès aux fichiers d'écriture à l'aide du mot-clé *synchronized* combiné avec l'utilisation d'un verrou unique pour chacun des fichiers. Ce concept d'écriture des fichiers est repris de l'application open source `GNSSLogger`.

Données brutes `FileLogger` a été presque entièrement récupéré tel quel de `GNSSLogger`.

RINEX L'enregistreur RINEX est conçu pour écrire des fichiers RINEX de version 3.03.

Le concept d'écriture du fichier RINEX est identique aux enregistreurs de l'application `GNSSLogger`, mais la fonction pour l'écriture des données a été entièrement codée à partir des données brutes du smartphone.

Dès que le récepteur reçoit une mesure, l'écriture des données RINEX se déroule selon les étapes suivantes permettant de garantir la précision et la validité des données enregistrées.

- Vérification du décodage du temps GPS : la première étape consiste à vérifier que le temps GPS (*Full-BiasNanos*) est entièrement décodé. Si ce n'est pas le cas, celui-ci ne pourra être déterminé, et le processus s'arrête.
- Verrouillage et contrôle du fichier : une fois la première étape vérifiée, le verrou pour l'écriture du fichier est pris afin de garantir l'intégrité des écritures simultanées.
- Calcul du temps GPS.
- Écriture de l'en-tête (uniquement si c'est le premier jeu de données reçu) : cet en-tête contient des informations essentielles sur le format et le contenu du fichier.
- Traitement des mesures : chaque mesure est ensuite parcourue :
 - si le satellite associé à la mesure n'est pas un satellite GPS, l'itération en cours est interrompue, et la boucle passe directement à la mesure suivante.
 - Le rapport signal sur bruit (carrier noise ratio) est vérifié pour être supérieur à un seuil de 18 dB-Hz. Cette vérification assure que seules les mesures de qualité suffisante sont prises en compte. Il est également vérifié que le Time of Week (TOW) est entièrement décodé, car sans cela, les pseudo-distances ne peuvent pas être calculées.
 - Initialisation des biais d'horloge : si les biais d'horloge n'ont pas déjà été initialisés, ils le sont à cette étape pour corriger les décalages temporels dans les mesures.
 - Calcul des pseudo-distances, du Doppler et des carriers : conformément à la section 3.3, les pseudo-distances, le Doppler et les carriers sont calculés pour chaque mesure valide.

- Formatage des mesures : chaque mesure validée est directement convertie en chaîne de caractères formatée selon les spécifications RINEX. Le nombre total de mesures est incrémenté à chaque fois qu'une mesure est écrite.
- Création du champ d'époque : une fois toutes les mesures traitées, un champ d'époque au format RINEX est créé, incluant le nombre total de mesures pour cette époque.
- Écriture dans le fichier : enfin, le champ d'époque et toutes les mesures de l'époque sont écrits dans le fichier RINEX après l'en-tête.

Ce processus méthodique assure que les données enregistrées dans le fichier RINEX sont précises, complètes et prêtes à être utilisées pour des analyses ultérieures.

Résultats des calculs Les fichiers de résultats de calcul n'ont pas une structure normée ni identique pour l'estimation standard, par moindres carrés et avec contraintes. Le contenu et la structure des fichiers ont été réalisés selon les besoins directs pour effectuer des tests et traiter les données.

Les fichiers de résultats se distinguent entre des fichiers détaillés et des fichiers avec séparateur classiques contenant seulement les données essentielles post-traitées.

- Résultats du calcul robuste (estimateur de Biber) détaillé avec les observations de double différence et de pseudo-distance, et les résidus.
- Résultats des positions du calcul robuste avec uniquement l'époque de mesure, les coordonnées estimées, leur écart-type et la différence de temps entre les mesures du smartphone et de la station de base.
- Résultats des positions du calcul par moindres carrés.
- Résultats du calcul contraint détaillé.
- Résultats des résidus sur les doubles différences avec l'époque de mesure, la différence de temps entre les mesures du smartphone et de la station de base, le nombre de satellites utilisés pour le calcul et les résidus sur les observations de double différence.

Calcul contraint sur un point connu

L'utilisateur a la possibilité d'indiquer les coordonnées de sa position s'il les connaît. Il peut alors cocher l'option de calcul contraint et il verra apparaître à droite des coordonnées les écarts à la position vraie pour chaque époque de mesure.

Cette fonctionnalité a été développée initialement à des fins de développement. C'est pourquoi l'application dispose également de deux boutons intitulés « D01 » et « PIL6170 » qui correspondent aux coordonnées de points fréquemment utilisés lors du développement de l'application et de l'implémentation des estimations de la position. Les résultats du calcul contraint sur les coordonnées sont également enregistrés lorsque l'option « log DGPS results » est activée.

Chapitre 4

Tests et évaluation de l'application

4.1 Validation du calcul des pseudo-distances

Les données brutes du smartphone ne permettant pas d'obtenir directement les observations de pseudo-distance, il est primordial de vérifier que le calcul des pseudo-distances soit correct. Pour cela, un enregistrement des données au format Rinex est réalisé. L'enregistrement des données au format Rinex offre une grande flexibilité puisque c'est un format standard et qu'il peut être lu par une grande partie des logiciels.

Le logiciel RTKLIB est utilisé pour contrôler qu'il n'y a pas d'erreur grossière ou de cohérence dans les pseudo-distances. Pour cela, quelques acquisitions ont été réalisées sur un point connu en coordonnées. Ensuite, les positions du smartphone ont été calculées à partir des pseudo-distances contenues dans le fichier RINEX exporté avant d'être comparées aux coordonnées des points connus. Cette démarche a également permis de valider que le format du fichier RINEX est correct.

Cette démarche a permis de voir que RTKLIB pour un calcul DGPS ne donne pas de solution lorsque les mesures de phase ont la valeur de zéro ou lorsqu'elles ne sont pas renseignées. Pour éviter cela, la phase vaut par défaut 1.000 dans les fichiers RINEX exportés par l'application.

4.2 Évaluation de la précision du positionnement

Les tests visant à évaluer la précision du positionnement DGPS obtenu avec l'application se sont déroulés en deux phases :

1. Mesures statiques sur un point de référence géodésique connu en coordonnées.
2. Mesures cinématiques avec suivi de la position par tachéométrie.

Les tests sont réalisés à la HEIG-VD dans le périmètre du réseau de points connus. La zone de test est fortement dégagée puisqu'elle se trouve dans un champ sans végétation ni bâtiment à proximité.

Le réseau de la HEIG-VD dispose d'un point de montage NTRIPCaster avec lequel les tests de l'application ont été effectués.

TABLEAU 4.1 – Paramètres de connexion à la station de base

Adresse IP :	193.134.218.96
Port :	5001
Nom du point de montage :	M_04
Nom de client :	****
Mot de passe :	****

La ligne de base entre la station de base et le smartphone est d'environ 150 mètres et permet d'éliminer les retards dus à la troposphère et à la ionosphère.

Pour les deux tests, la position vraie du smartphone est connue. Pour les mesures statiques, le point de référence géodésique est un pilier en béton faisant partie du réseau de points de la HEIG-VD (figure 4.1). Les coordonnées de ces points sont connues dans le cadre de référence **MN95** et avec l'altitude en **RAN95**. Pour les mesures cinématiques, la position vraie du smartphone est obtenue par mesures terrestres en suivant un prisme fixé sur le smartphone. La connaissance de la position vraie du smartphone pour les tests à l'avantage de permettre de comparer directement la position réelle mesurée l_i à la position vraie de celui-ci \check{l} . La différence entre ces deux grandeurs s'appelle l'écart vrai noté $\check{\eta}$:

$$\check{\eta} = l_i - \check{l} \quad (4.1)$$



FIGURE 4.1 – Pilier géodésique utilisé pour effectuer les mesures statiques

Dès lors que la mesure vraie ou qu'une valeur de référence est connue, il est possible d'évaluer la précision du positionnement en calculant les valeurs suivantes :

- la moyenne des erreurs vraies, ou le **biais** qui mesure la tendance systématique des valeurs observées à s'éloigner (ou à se rapprocher) de la valeur vraie. Il indique si, en moyenne, les observations sont surestimées ou sous-estimées par rapport à la valeur vraie. Un biais positif signifie que les observations sont en moyenne supérieures à la valeur vraie, tandis qu'un biais négatif signifie que les observations sont en moyenne inférieures à la valeur vraie. Un biais nul indique que les observations ont tendance à être centrées autour de la valeur vraie, en moyenne.

$$\text{Biais} = \frac{1}{n} \sum_{i=1}^n \check{\eta} \quad (4.2)$$

- l'**écart-type** qui indique la variabilité des erreurs autour de leur moyenne arithmétique μ .

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (l_i - \mu)^2} \quad (4.3)$$

- l'écart quadratique moyen, ou **RMSE** de l'anglais *Root Mean Square Error*. Cet indicateur permet d'analyser si les valeurs calculées sont en moyenne proches ou éloignées des valeurs réelles. Une petite valeur de RMSE indique que les observations sont proches des valeurs vraies tandis qu'une grande valeur indique des observations plus éloignées des valeurs vraies. Le RMSE ne permet pas de distinguer les erreurs systématiques (biais) des erreurs aléatoires, c'est pourquoi sa valeur numérique est toujours plus grande que l'écart-type σ .

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \check{\eta}^2} \quad (4.4)$$

- la **valeur minimale** et **maximale** des écarts vrais $\check{\eta}$.

De plus, puisque l'application permet également d'enregistrer les données au format RINEX, les résultats DGPS obtenus via le smartphone seront comparés aux résultats des calculs DGPS et SPP effectués avec le logiciel RTKLIB.

Ainsi, pour chacun des tests, la position est calculée selon les quatre méthodes suivantes :

- Position DGPS de l'application prototype.
- Localisation du smartphone.
- RTKLIB DGPS.
- RTKLIB Single.

Durant les sessions de mesures, l'application smartphone enregistre quatre fichiers différents :

1. un fichier contenant les résultats du calcul DGPS de l'application contenant le temps GPS et les coordonnées de la position estimée dans le cadre de référence MN95 avec l'altitude sur l'ellipsoïde de Bessel.
2. un fichier contenant les résultats du calcul DGPS avec plus d'informations sur le calcul comme le nombre d'observations, le quotient, les résidus sur les doubles différences, etc.
3. un fichier contenant les données brutes de localisation, de navigation et des mesures GPS.
4. un fichier au format RINEX avec les observations brutes.

Service de localisation du smartphone

Le smartphone dispose de son propre service de localisation. La localisation est mise à jour dès que le smartphone détecte un changement dans sa position. Elle est fournie sous la forme de latitude, longitude en WGS84 et d'altitude sur l'ellipsoïde de référence GRS80. L'époque de cette localisation est donnée en millisecondes dans le système de temps UNIX.

Pour le traitement des résultats, les données du service de localisation sont récupérées dans le fichier de données brutes. Les coordonnées sont transformées dans le cadre de référence **MN95** selon le processus décrit dans la section 2.6 et le temps UNIX est converti en temps GPS (voir section 2.7).

Calcul de la position DGPS et Single avec RTKLIB

Le calcul de la position dans RTKLIB est effectué avec des paramètres de calcul similaires à ceux utilisés dans l'algorithme du smartphone afin de pouvoir comparer les résultats. Le tableau 4.2 présente les paramètres utilisées.

TABLEAU 4.2 – Paramètres de calcul RTKLIB

Méthode	Single	DGPS
Fréquence	L1	L1
Masque d'élévation	15°	15°
Éphémérides	radiodiffusés	radiodiffusés
Corrections ionosphériques	radiodiffusés	désactivé
Corrections troposphériques	Saastamoinen	désactivé
Constellation	GPS	GPS

Les calculs sont effectués en post-traitement avec des données en entrée qui sont strictement identiques aux données utilisées par le smartphone pour calculer sa position en temps réel. Les observations brutes calculées par le smartphone se trouvent dans le fichier RINEX exporté lors du calcul et les observations brutes de la station de base ainsi que les éphémérides radiodiffusées sont systématiquement sauvegardées chaque heure sur le serveur de la HEIG-VD et peuvent être récupérées.

4.2.1 Mesures statiques sur un point connu

Pour les mesures statiques le smartphone a été simplement posé sur le un pilier géodésique du réseau interne à la HEIG-VD connu en coordonnées (figure 4.1) :

$E_{PIL6170}^{MN95}$	$N_{PIL6170}^{MN95}$	$H_{PIL6170}^{RAN95}$
2540621.753 m	1181302.435 m	448.6834 m

Tests sur le bruit de mesure

Deux premières acquisitions ont été réalisées sur le point de référence avec deux supports différents pour le smartphone dans le but de déterminer quel est le meilleur dispositif à partir des bruits des mesures. La première acquisition est réalisée avec le smartphone fixé sur un support imprimé en 3D compatible avec les porte-voyants classiques Leica (figure 4.2) tandis que la deuxième session de mesures est réalisée avec le smartphone posé sur une plaque métallique.



FIGURE 4.2 – Support pour le smartphone pour porte-voyant

Le but de ce second support est de réduire le bruit des mesures dû à la diminution des effets de multi-trajet des signaux GPS. La plaque métallique placée sous le récepteur agit comme un réflecteur. Elle aide à assurer que les signaux reçus soient principalement des signaux directs venant des satellites et que les signaux réfléchis provenant du sol soient bloqués.

Les mesures ont été réalisées pendant environ une minute et le calcul a été contraint sur les points connus. Cela permet d'obtenir les résidus vrais sur les observations de double différence et de voir dans quel cas de figure les résidus de double différence sont les plus petits.

Le tableau 4.4 présente les résultats sur les résidus de double différence. Ceux-ci sont de l'ordre de plusieurs mètres. Lorsque le support pour porte-voyant est utilisé, la moyenne des résidus est supérieure au mètre, tandis qu'elle est inférieure au mètre avec l'utilisation de la plaque en métal. L'écart-type des résidus est beaucoup plus élevé pour les mesures sur le porte-voyant ainsi que la valeur maximale obtenue.

Ces résultats doivent cependant être interprétés avec prudence. En effet, les problèmes dus au multi-trajet sont corrélés avec la position des satellites. Plus ils sont proches de l'horizon, plus les effets du multi-trajet sont grands. La configuration des satellites n'est pas prise en compte dans cette analyse. Ainsi, il est possible que lors de la mesure avec la plaque, la configuration des satellites était telle que les résultats globaux sont meilleurs.

Pour pouvoir analyser correctement l'effet de la plaque par rapport au support du porte-voyant, il faudrait traiter les observations de pseudo-distance qui sont directement issues des signaux et voir dans quelle mesure celles-ci sont corrélées à l'élévation des satellites et au rapport signal-bruit.

De plus, il existe une forte corrélation entre la qualité des mesures et le rapport signal-bruit. Il faudrait donc voir dans quelle mesure les résidus sur les doubles différences sont causés par une faible force des signaux, d'autant que la qualité des récepteurs embarqués dans les smartphones est souvent moindre.

TABLEAU 4.4 – Résidus sur les doubles différences en mètre

Support	Moyenne	Médiane	Écart-type	RMSE	Minimum	Maximum
Porte-voyant	-1.32	0.13	7.92	8.03	0.01	33.79
Plaque en métal	-0.52	-0.74	3.12	3.16	0.00	20.08

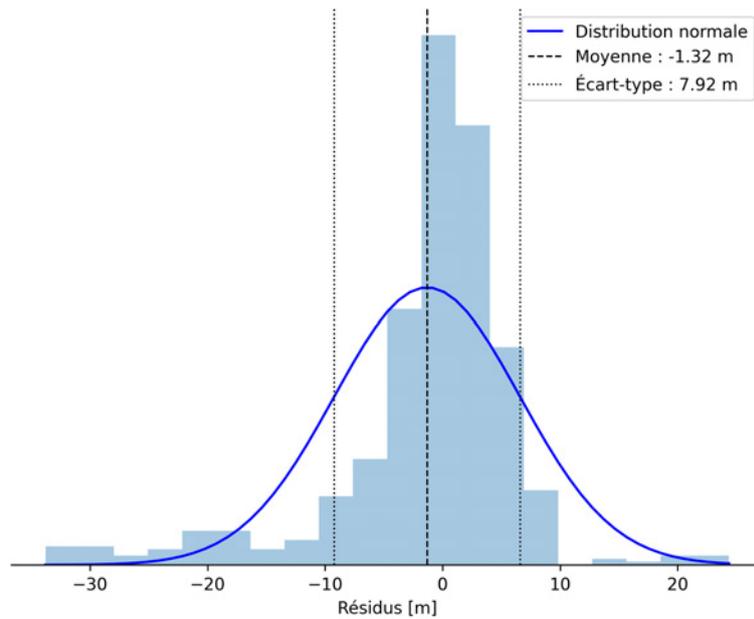


FIGURE 4.3 – Histogramme des résidus sur les doubles différences avec mesures sur le porte-voyant

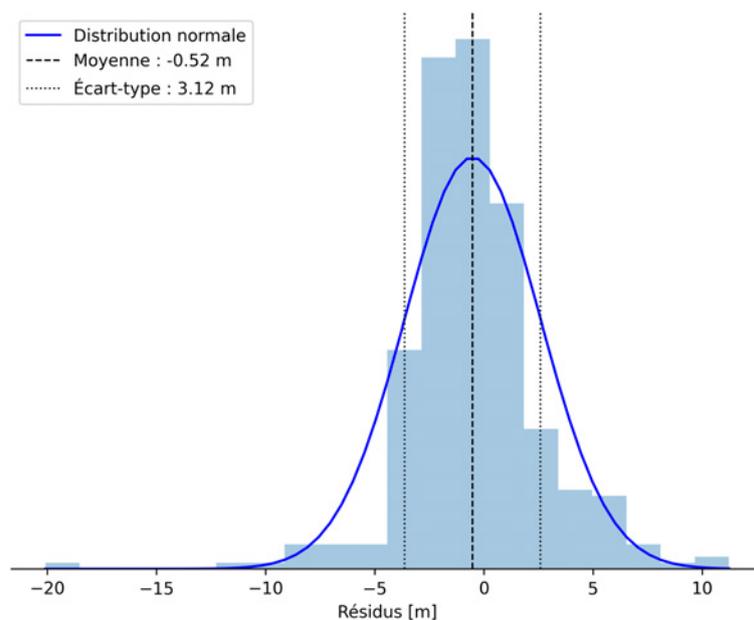


FIGURE 4.4 – Histogramme des résidus sur les doubles différences avec mesures sur la plaque en métal

Pour résumer, les résultats des tests semblent montrer, comme attendu, que l'utilisation d'une plaque en métal lors de l'acquisition améliore la précision des observations de double différence. C'est pourquoi, la session de mesures statiques est effectuée avec le smartphone posé sur la plaque métallique.

Session de mesures statiques

Les mesures ont été réalisées durant un peu plus d'une demi-heure. Le tableau 4.5 présente en détail les informations de la session de mesures effectuées.

Initialement, il était prévu de réaliser une heure de mesure, mais l'application s'est interrompue et a arrêté de mesurer après environ 40 minutes. Cela est probablement dû à une coupure du réseau internet et donc à la déconnexion de la station de base. Les 30 dernières mesures ayant encore convergé en une position ont également été retirées du jeu de données, car la position s'éloignait de plus en plus de la position vraie. Cela est dû à l'extrapolation des mesures de pseudo-distances de la station de base qui se dégrade avec l'augmentation de la différence de temps avec les dernières mesures reçues par le smartphone. Ainsi, le jeu de données tronqué se compose de 2501 époques de mesures dont seulement 23 époques qui n'ont pas convergé en une solution.

TABLEAU 4.5 – Informations et paramètres de calcul de la session de mesures statiques

Paramètre	Valeur
Date	14.07.2024
Début	12:52:43 GPST
Fin	13:34:22 GPST
Nb époques calculées	2501
Nb époques avec solution	2478
Fréquence	L1
Masque d'élévation	15°
Éphémérides	radiodiffusés
Corrections ionosphériques	aucune
Corrections troposphériques	aucune
Constellation	GPS

TABLEAU 4.6 – Résultats des mesures statiques : indicateurs statistiques [m]

Méthode		Biais	Écart-type	RMSE	Minimum	Maximum
Smartphone DGPS	ΔE	-0.67	1.16	1.34	0.00	5.38
RTKLIB DGPS	ΔE	1.18	1.42	1.85	0.00	5.11
Smartphone location	ΔE	0.83	0.00	0.83	0.83	0.83
RTKLIB Single	ΔE	2.47	1.47	2.88	0.00	7.45
Smartphone DGPS	ΔN	-0.58	1.65	1.75	0.00	6.47
RTKLIB DGPS	ΔN	-0.57	1.48	1.59	0.00	8.52
Smartphone location	ΔN	1.31	0.00	1.31	1.31	1.31
RTKLIB Single	ΔN	1.71	1.61	2.35	0.00	9.56
Smartphone DGPS	ΔH	-1.71	3.53	3.92	0.00	13.91
RTKLIB DGPS	ΔH	-1.39	3.25	3.54	0.00	11.45
Smartphone location	ΔH	2.06	0.00	2.06	2.06	2.06
RTKLIB Single	ΔH	0.82	3.47	3.56	0.00	12.22

Le tableau 4.6 présente les différences entre les positions calculées et la position connue du pilier géodésique. Les résultats sont séparés selon les quatre méthodes décrites plus haut dans ce chapitre, à savoir le calcul DGPS de l'application et de RTKLIB, la localisation interne du smartphone et le calcul RTKLIB en position simple sur le code. Les résultats des positions en planimétrie se trouvent sur la figure 4.5 et les séries temporelles des écarts vrais se trouvent sur la figure 4.6.

La figure 4.7 présente les séries temporelles des écarts vrais obtenus avec l'application sur une période plus courte avec l'indicateur de la précision. Finalement, la figure 4.8 montre la distribution des écarts vrais pour la coordonnée est. Ceux-ci sont de l'ordre du mètre et suivent une distribution normale.

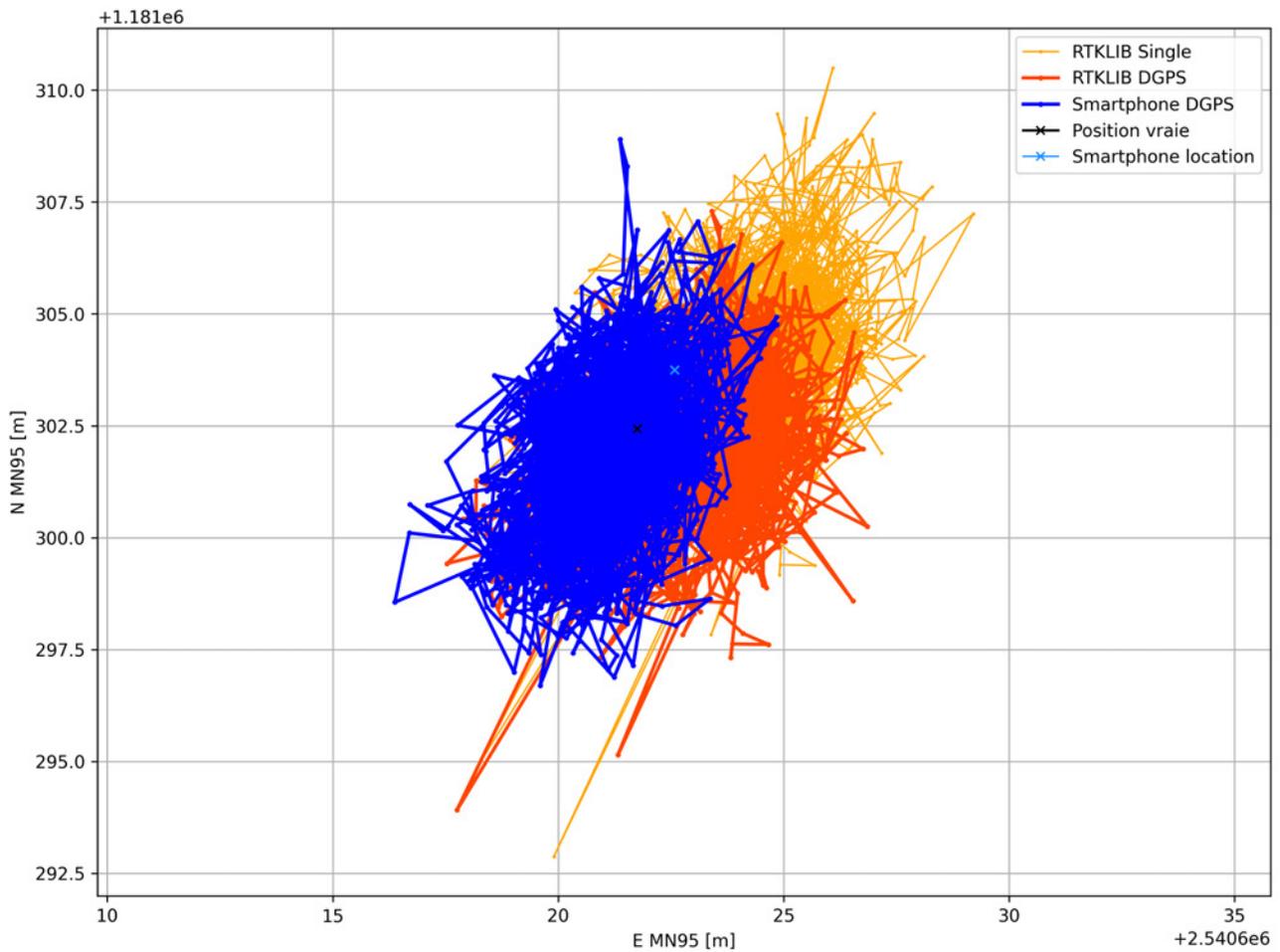


FIGURE 4.5 – Résultats des mesures statiques en planimétrie

Les résultats permettent de tirer les observations suivantes :

- La dispersion des positions autour de leur moyenne, observée via la grandeur statistique de l'écart-type, est très similaire entre les calculs DGPS du smartphone et les calculs DGPS et Single de RTKLIB. Ce résultat était attendu puisque les mesures utilisées dans les deux calculs sont identiques et la qualité de ces mesures impacte directement la qualité des positions calculées. De plus, les séries temporelles montrent des pics qui se suivent entre les différents calculs (figure 4.6). Cela confirme le constat que les positions obtenues sont directement liées à la qualité des mesures brutes, qui est moindre sur les antennes des smartphones.
- Globalement, la dispersion ainsi que les maximums sont plus grands sur les altitudes que sur les composantes planimétriques. C'est généralement le cas des résultats obtenus par positionnement par satellites. En effet, la configuration géométrique est défavorable pour l'altitude.
- Le biais entre les deux calculs DGPS du smartphone et de RTKLIB est similaire sur la coordonnée nord et sur l'altitude, mais diffère sur la coordonnée est. Cette différence est visible sur la série temporelle (figure 4.6) où les écarts sur la coordonnée est augmentent vers la fin de la série de mesures. Il est difficile de trouver la cause de cet effet puisque seule la coordonnée est concernée.
- Le service de localisation intégré du smartphone permet d'obtenir une position à 1-2 mètres de la position vraie. Cependant, puisque le smartphone n'a pas détecté de mouvement, la position est restée identique durant l'acquisition et il n'y a donc qu'une seule mesure.
- Concernant le biais, il est attendu que les positions calculées par la méthode de positionnement différentiel soit meilleure puisque celle-ci permet d'éliminer les erreurs d'horloge et les erreurs dues à l'atmosphère. Cela ne ressort pas de façon évidente dans les résultats obtenus. La position absolue est meilleure sur les coordonnées planimétriques avec les calculs DGPS qu'avec un calcul seul sur le code, mais moins bonne pour l'altimétrie.

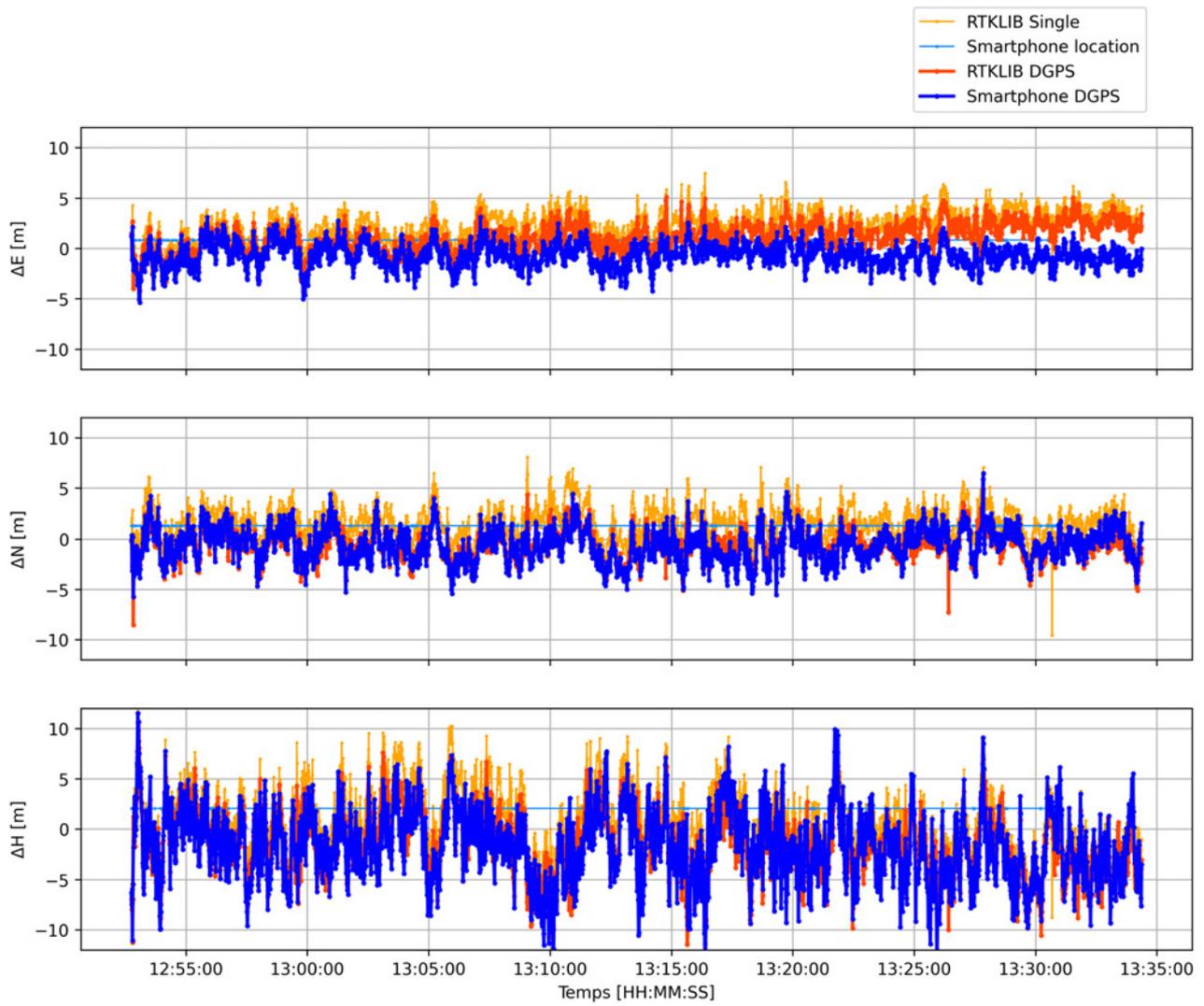


FIGURE 4.6 – Séries temporelles des différences avec la position vraie

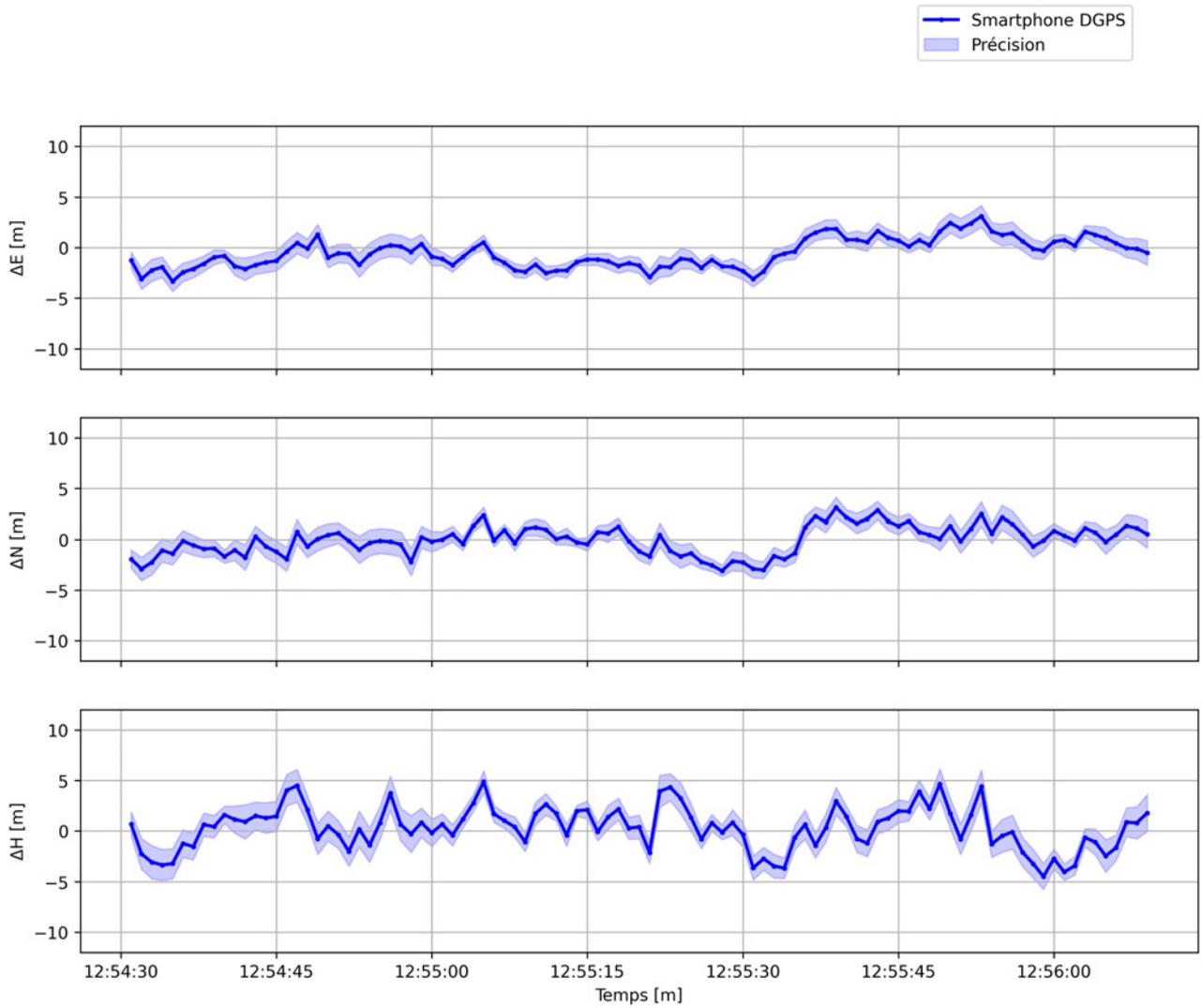


FIGURE 4.7 – Séries temporelles des écarts vrais avec la position DGPS du smartphone

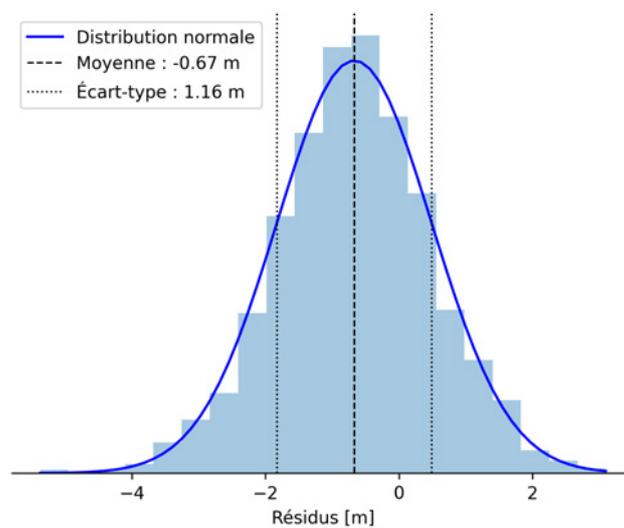


FIGURE 4.8 – Histogramme des écarts vrais de la coordonnée est du calcul smartphone DGPS

4.2.2 Mesures cinématiques avec suivi de la position par tachéométrie

Les mesures cinématiques sont réalisées à pied aux alentours du pilier géodésique et en zone dégagée.

Le suivi de la position du smartphone est effectué avec une station totale¹ et le logiciel Q_Kinematic_TS installé sur un ordinateur portable. Ce logiciel a été développé par le Prof. Dr Sébastien GUILLAUME pour les besoins académiques internes. Ce logiciel permet d'obtenir les coordonnées mesurées par la station de base avec l'époque de mesure en temps UTC ainsi que l'horloge interne du portable. Pour pouvoir obtenir l'époque de mesure en UTC, l'ordinateur est branché à un récepteur GNSS auprès duquel il récupère le temps GPS et calcule à une fréquence de 20 Hz la différence entre son horloge interne et le temps GPS. De cette manière l'ordinateur peut déterminer pour chaque mesure de la station totale l'époque de mesure en temps UTC. La station totale suit le prisme et transmet par Bluetooth à l'ordinateur les positions mesurées à une fréquence de 20 Hz.

Le suivi est réalisé avec un mini prisme Leica GRZ101 qui est simplement fixé au milieu de l'écran du smartphone. Le smartphone est tenu à l'horizontale durant les mesures (voir figure 4.9).



FIGURE 4.9 – Photographie d'une session de mesures cinématiques.

Les mesures cinématiques ont été effectuées en trois sessions de quelques minutes. Les paramètres de calcul ainsi que les conditions d'acquisition sont identiques pour toutes les sessions. L'objectif d'effectuer plusieurs sessions était de vérifier la robustesse de l'application et la cohérence des résultats. Les résultats de la première session sont présentés de manière détaillée à la suite, car c'est l'acquisition avec le plus d'époques de mesure, tandis que les résultats détaillés des autres sessions se trouvent dans l'annexe 4.3.

Session n° 1

La session de mesures numéro 1 dure environ 2 minutes 30 avec 156 époques de mesure dont trois époques n'ayant pas convergé.

La trace parcourue à pied est visible sur la figure 4.10 avec comme symbole un trait plein noir. Elle est mesurée avec la station totale et est considérée comme la trajectoire vraie. La figure montre que les quatre méthodes utilisées ne donnent pas de positions aberrantes et les traces suivent relativement bien la trace mesurée avec la station totale. Elle montre également des positions qui semblent fortement bruitées excepté pour les positions obtenues via la localisation interne du smartphone qui est plus lisse. Cela s'explique par l'utilisation d'autres sources de données ou capteurs, mais également par un algorithme de calcul plus performant qui permet de limiter les répercussions du bruit directement dans la position.

Concernant la dispersion des observations, les constats sont identiques que pour les mesures statiques, à savoir : une dispersion similaire entre le calcul DGPS de l'application et les deux calculs RTKLIB mais plus faible pour les coordonnées planimétriques que pour l'altitude.

1. Le modèle utilisé est le MS60 de Leica.

Sur cette session de mesure, il apparaît plus clairement que la position absolue obtenue est meilleure avec les calculs différentiels. Les biais pour les résultats DGPS sont de quelques décimètres (3 cm au minimum et 30 cm au maximum) tandis qu'ils sont supérieurs au mètre pour les résultats avec une position simple sur le code (1.2 m au minimum et 2.3 m au maximum).

TABLEAU 4.7 – Informations sur le calcul - session cinématique n° 1

Paramètre	Valeur
Date	16.07.2024
Début	09:11:23 GPST
Fin	09:13:58 GPST
Nb époques calculées	156
Nb époques avec solution	153

TABLEAU 4.8 – Indicateurs statistiques- session cinématique n° 1

Méthode		Biais	Écart-type	RMSE	Minimum	Maximum
Smartphone DGPS	ΔE	-0.16	0.97	0.98	0.00	3.39
RTKLIB DGPS	ΔE	0.34	0.99	1.05	0.01	3.99
Smartphone location	ΔE	0.84	0.35	0.91	0.08	1.72
RTKLIB Single	ΔE	1.66	0.98	1.92	0.05	5.27
Smartphone DGPS	ΔN	-0.24	1.64	1.66	0.02	6.02
RTKLIB DGPS	ΔN	-0.03	1.29	1.29	0.00	3.97
Smartphone location	ΔN	1.70	0.56	1.79	0.23	3.97
RTKLIB Single	ΔN	2.26	1.34	2.63	0.01	6.18
Smartphone DGPS	ΔH	0.06	2.90	2.90	0.01	8.23
RTKLIB DGPS	ΔH	0.15	2.82	2.82	0.05	6.72
Smartphone location	ΔH	2.32	0.77	2.45	0.54	4.49
RTKLIB Single	ΔH	1.22	2.72	2.98	0.04	7.72



FIGURE 4.10 – Résultats planimétriques - session cinématique n° 1

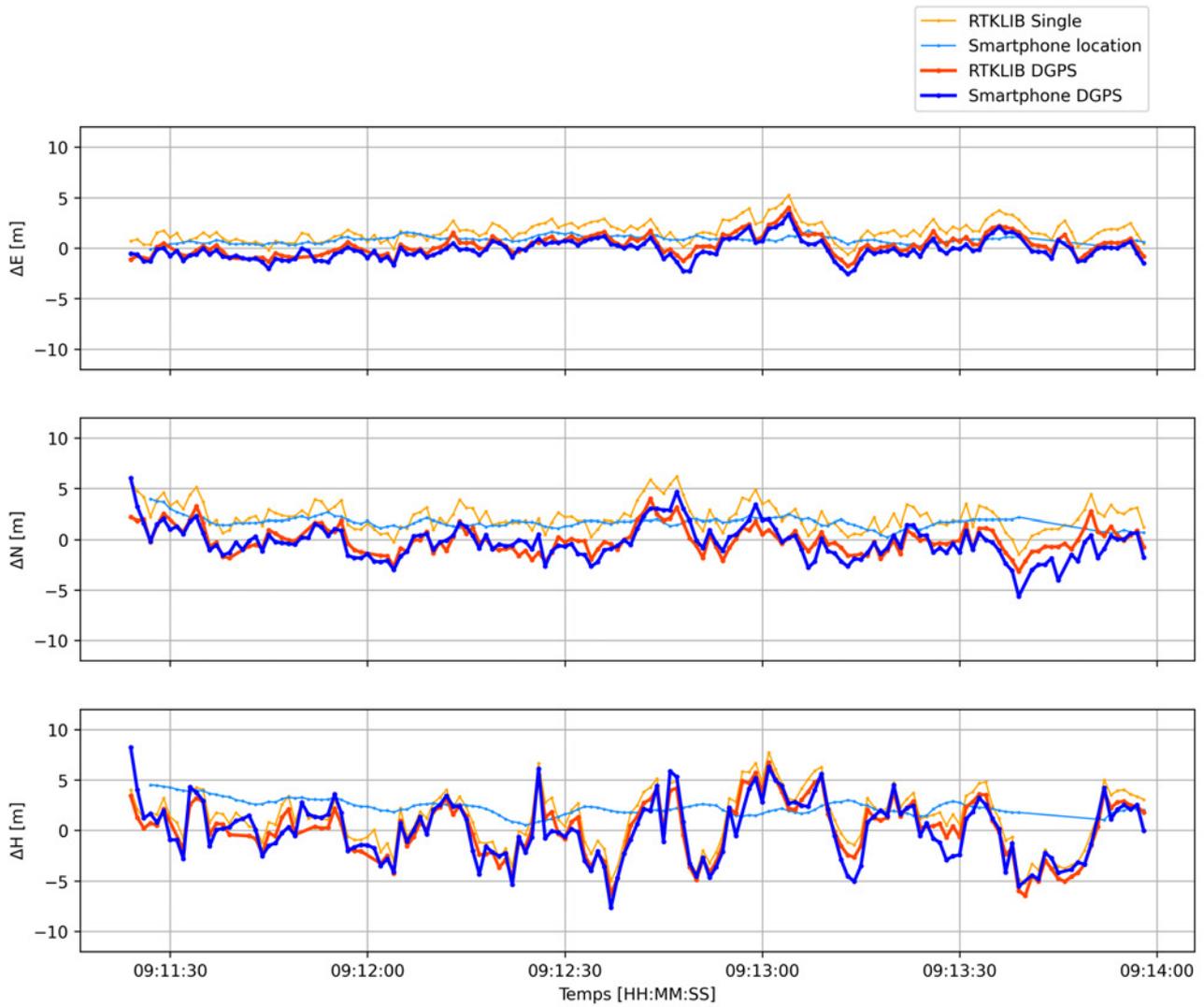


FIGURE 4.11 – Séries temporelles des différences avec la position vraie - session cinématique n° 1

4.2.3 Résumé

Le tableau 4.9 regroupe les indicateurs statistiques des résultats obtenus avec l'application DGPS pour chaque session de mesures et le tableau 4.10 regroupe ceux obtenus avec RTKLIB en position simple sur le code.

Les positions fournies par l'application smartphone sont meilleures que celles calculées avec RTKLIB en position simple sur le code, avec des écarts par rapport aux positions vraies de l'ordre d'un demi-mètre. Les écarts des positions avec RTKLIB sont, quant à eux, de l'ordre de 1 à 3 mètres. Cela montre qu'une application de positionnement différentiel peut apporter à l'utilisateur un gain de précision absolu sur les positions calculées.

TABLEAU 4.9 – Indicateurs statistiques - smartphone DGPS

Mesures		Biais	Écart-type	RMSE	Minimum	Maximum
Statique	ΔE	-0.67	1.16	1.34	0.00	5.38
Session 1	ΔE	-0.16	0.97	0.98	0.00	3.39
Session 2	ΔE	-0.47	1.03	1.13	0.00	3.09
Session 3	ΔE	-0.16	1.23	1.24	0.01	4.14
Statique	ΔN	-0.58	1.65	1.75	0.00	6.47
Session 1	ΔN	-0.24	1.64	1.66	0.02	6.02
Session 2	ΔN	-0.82	1.80	1.98	0.04	6.42
Session 3	ΔN	-0.34	1.85	1.88	0.01	5.90
Statique	ΔH	-1.71	3.53	3.92	0.00	13.91
Session 1	ΔH	0.06	2.90	2.90	0.01	8.23
Session 2	ΔH	-1.09	4.54	4.67	0.04	14.92
Session 3	ΔH	0.31	3.82	3.83	0.03	15.05

TABLEAU 4.10 – Indicateurs statistiques - RTKLIB SPP

Mesures		Biais	Écart-type	RMSE	Minimum	Maximum
Statique	ΔE	2.47	1.47	2.88	0.00	7.45
Session 1	ΔE	1.66	0.98	1.92	0.05	5.27
Session 2	ΔE	3.66	1.05	3.81	1.25	6.09
Session 3	ΔE	1.42	1.24	1.89	0.05	5.50
Statique	ΔN	1.71	1.61	2.35	0.00	9.56
Session 1	ΔN	2.26	1.34	2.63	0.01	6.18
Session 2	ΔN	2.14	2.20	3.07	0.02	8.82
Session 3	ΔN	1.80	2.13	2.79	0.04	10.30
Statique	ΔH	0.82	3.47	3.56	0.00	12.22
Session 1	ΔH	1.22	2.72	2.98	0.04	7.72
Session 2	ΔH	5.07	3.43	6.12	0.18	15.24
Session 3	ΔH	2.10	3.34	3.95	0.05	15.57

4.3 Analyse des performances de l'application prototype

Aucune analyse spécifique des performances et de la stabilité de l'application n'a été réalisée. La priorité a été donnée à l'implémentation du calcul de positionnement différentiel plutôt qu'au développement de l'application, qui sert principalement de support pour traiter les données brutes du récepteur GNSS embarqué. Néanmoins, l'utilisation de l'application lors des tests a démontré son bon fonctionnement : aucune erreur n'a été constatée au niveau de l'interface utilisateur, qui réagissait de manière fluide, et les mesures ont pu être effectuées sans problème.

Cependant, l'arrêt inattendu des mesures lors de l'acquisition sur le pilier indique que les performances de l'application ne sont pas optimisées et que la stabilité du calcul dépend d'autres facteurs.

La performance de l'application en mode veille ou lorsqu'elle fonctionne en arrière-plan n'est pas connue. Les mesures ont été systématiquement réalisées avec le smartphone déverrouillé et l'application active.

Conclusion et perspectives

Ce travail visait à développer une application de positionnement différentiel GPS en temps réel pour smartphones afin d'améliorer la précision de localisation. L'objectif est atteint. Dans le cadre de ce travail, une application prototype a été réalisée permettant d'exploiter les données brutes de l'antenne intégrée au smartphone et de calculer la position en temps réel mais également en post-traitement grâce aux fonctionnalités d'enregistrement des données selon différents formats.

Les résultats ont montré que le positionnement différentiel améliore la précision par rapport au positionnement absolu. Le biais par rapport aux coordonnées vraies est de l'ordre d'un demi-mètre pour les mesures en mode DGPS, tandis qu'il est d'un à trois mètres pour le positionnement simple. Toutefois, il faut souligner que le bruit est similaire et relativement grand (1 à 2 mètres sur les coordonnées planimétriques) pour les résultats DGPS avec le smartphone et RTKLIB, mais aussi pour le calcul simple sur le code réalisé avec RTKLIB. Ainsi, les bruits sur les résultats de la position peuvent être attribués aux bruits des mesures brutes.

Améliorations possibles

Les améliorations possibles dans ce travail peuvent être séparées selon deux catégories : les améliorations au niveau de l'algorithme d'estimation de la position afin d'obtenir une meilleure position et les améliorations directement appliquées au niveau de l'usage de l'application, en améliorant par exemple les fonctionnalités.

Bien que les mesures soient fortement bruitées et sujettes au multi-trajet, le calcul de la position peut être amélioré par différents moyens :

- En prenant en compte l'élévation et la force des signaux pour définir le modèle stochastique des observations de double différence.
- En effectuant le calcul sur plusieurs époques ou en procédant à une estimation séquentielle.
- En utilisant les signaux d'autres constellations.

Voici une liste non exhaustive des futures améliorations possibles au niveau de l'application :

- Connexion à d'autres points de montage. Par défaut, l'application prototype se connecte directement au point de montage de la HEIG-VD. Pour gagner en flexibilité, l'application devrait permettre à l'utilisateur de se connecter à tous les points de montage transmettant les messages utilisés dans le calcul de la position. Il est important que l'application puisse se connecter à une station de base relativement proche puisqu'aucun modèle pour corriger les erreurs dues à l'atmosphère n'est implémenté.
- Fichiers de résultats de calcul. L'enregistrement des résultats a été réalisé pour les besoins spécifiques d'analyse des données. Il serait bien que les résultats des calculs soient plus détaillés et aient une structure plus standardisée permettant à n'importe quel utilisateur de facilement récupérer les positions calculées, les différents indicateurs de précision et les paramètres de calcul.
- Éphémérides : Lors du démarrage de l'application, il faut attendre beaucoup de temps pour que les messages de navigation soient complètement déchiffrés. Une fonctionnalité permettant à l'utilisateur de télécharger un fichier RINEX d'éphémérides serait pertinente.

Pour conclure, l'application prototype réalisée dans le cadre de ce travail a permis de mettre en évidence le potentiel du calcul différentiel sur le code pour améliorer la précision de la position. Bien que celle-ci ne soit pas opérationnelle et suffisamment stable pour une utilisation pratique, elle offre la structure nécessaire pour développer l'algorithme de calcul et essayer d'améliorer d'avantage la précision de la position.

Bibliographie

- EUROPEAN GNSS AGENCY (GSA), 2017. *Using GNSS Raw Measurements on Android Devices*. Aussi disponible à l'adresse : <https://www.euspa.europa.eu/newsroom/news/available-now-white-paper-using-gnss-raw-measurements-android-devices>.
- GUILLAUME, S., 2021. *Eléments de géodésie*. HEIG-VD.
- GUILLAUME, S., 2022a. *Introduction au GNSS*. HEIG-VD.
- GUILLAUME, S., 2022b. *Méthodes d'estimation et réseaux géodésiques*. HEIG-VD.
- LIU, W. ; SHI, X. ; ZHU, F. et al., 2019. Quality analysis of multi-GNSS raw observations and a velocity-aided positioning approach based on smartphones. *Advances in Space Research*. T. 63, n° 8, p. 2358-2377.
- MARTI, U., 1997. Das Geoid in der Schweiz 1997. *Geodätisch-geophysikalische Arbeiten in der Schweiz*. T. 56.
- PESYNA, K. M. ; HEATH, R. W. ; HUMPHREYS, T. E. et al., 2014. Centimeter positioning with a smartphone-quality GNSS antenna. In : *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*, p. 1568-1577.
- RADIO TECHNICAL COMMISSION FOR MARITIME SERVICES (RTCM), 2013. *RTCM 10403.2 : Differential GNSS (Global Navigation Satellite Systems) Services – Version 3*.
- SNIP NTRIP Caster, 2024. Aussi disponible à l'adresse : <https://www.use-snip.com/>.
- ZANGENEHNEJAD, F. ; GAO, Y., 2021. GNSS smartphones positioning : Advances, challenges, opportunities, and future perspectives. *Satellite navigation*. T. 2, p. 1-23.
- ZANGENEHNEJAD, F. ; JIANG, Y. ; GAO, Y., 2023. GNSS Observation Generation from Smartphone Android Location API : Performance of Existing Apps, Issues and Improvement. *Sensors*. T. 23, n° 2. ISSN 1424-8220. Aussi disponible à l'adresse : <https://www.mdpi.com/1424-8220/23/2/777>.
- ZHANG, X. ; TAO, X. ; ZHU, F. et al., 2018. Quality assessment of GNSS observations from an Android N smartphone and positioning performance analysis using time-differenced filtering approach. *Gps Solutions*. T. 22, p. 1-11.

Table des figures

2.1	Structure du code et du message de navigation du signal GPS L1 C/A (European GNSS Agency (GSA), 2017)	6
2.2	Sources d'erreurs sur la mesure de pseudo-distance (European GNSS Agency (GSA), 2017)	8
2.3	Principe du positionnement différentiel (Guillaume, 2022a)	9
2.4	Étapes pour la transformation des coordonnées WGS84 à MN95	15
3.1	Éléments nécessaires minimums pour déterminer la position du smartphone	20
3.2	Schéma de la transmission des données de la station de base	24
3.3	Cycle de vie d'une activité	28
3.4	Processus général du fonctionnement de l'application	29
3.5	Diagramme de l'accès aux données brutes	29
3.6	Capture d'écran de la mise en page principale de l'application	30
4.1	Pilier géodésique utilisé pour effectuer les mesures statiques	34
4.2	Support pour le smartphone pour porte-voyant	36
4.3	Histogramme des résidus sur les doubles différences avec mesures sur le porte-voyant	37
4.4	Histogramme des résidus sur les doubles différences avec mesures sur la plaque en métal	37
4.5	Résultats des mesures statiques en planimétrie	39
4.6	Séries temporelles des différences avec la position vraie	40
4.7	Séries temporelles des écarts vrais avec la position DGPS du smartphone	41
4.8	Histogramme des écarts vrais de la coordonnée est du calcul smartphone DGPS	41
4.9	Photographie d'une session de mesures cinématiques.	42
4.10	Résultats planimétriques - session cinématique n° 1	44
4.11	Séries temporelles des différences avec la position vraie - session cinématique n° 1	45
A.1	Résultats planimétriques - session cinématique n° 2	60
A.2	Série temporelle des différences avec la position vraie - session cinématique n° 2	61
A.3	Résultats planimétriques - session cinématique n° 3	63
A.4	Série temporelle des différences avec la position vraie - session cinématique n° 3	64

Liste des tableaux

1.1	Applications d'enregistrement de données brutes GNSS	4
2.1	Signaux GPS	6
2.2	Statuts de synchronisation	7
2.3	Relations entre les systèmes de temps	17
3.1	Liste des variables de la classe <code>GnssClock</code>	20
3.2	Liste des variables de la classe <code>GnssMeasurement</code>	21
3.3	Liste des variables de la classe <code>GnssNavigationMessage</code>	22
3.4	Structure de la transmission de messages RTCM	24
3.5	Paramètres de calcul	26
4.1	Paramètres de connexion à la station de base	33
4.2	Paramètres de calcul RTKLIB	35
4.4	Résidus sur les doubles différences en mètre	37
4.5	Informations et paramètres de calcul de la session de mesures statiques	38
4.6	Résultats des mesures statiques : indicateurs statistiques [m]	38
4.7	Informations sur le calcul - session cinématique n° 1	43
4.8	Indicateurs statistiques- session cinématique n° 1	43
4.9	Indicateurs statistiques - smartphone DGPS	46
4.10	Indicateurs statistiques - RTKLIB SPP	46
A.1	Informations sur le calcul - session cinématique n° 2	59
A.2	Indicateurs statistiques - session cinématique n° 2	59
A.3	Informations sur le calcul - session cinématique n° 3	62
A.4	Indicateurs statistiques- session cinématique n° 3	62
B.1	Caractéristiques du smartphone utilisé pour les tests	65

Abréviations

API	Application Programming Interface
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPST	Global Positioning System Time
HEIG-VD	Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud
IGS	International GNSS Service
PPP	Precise Point Positioning
PVT	Position Velocity Time
RINEX	Receiver Independent Exchange Format
RTK	Real Time Kinematic
SBAS	Satellite-Based Augmentation System
SPP	Single Point Positioning
TAI	Temps Atomique International
TOW	Time Of Week
TRS	Terrestrial Reference System
UTC	Universal Time Coordinated
WGS84	World Geodetic System 1984
WN	Week Number
EGNOS	European Geostationary Navigation Overlay Service

Constantes

Les constantes utilisées dans ce projet sont tirées de (Guillaume, 2021).

c	Vitesse de la lumière dans le vide : $2.99792458 \cdot 10^8$ [$\frac{\text{m}}{\text{s}}$]
a_{GRS80}	Demi-grand axe de l'ellipsoïde GRS80 : 6378137.000 [m]
f_{GRS80}	Applatissement de l'ellipsoïde GRS80 : 1/298.257222101 [-]
a_{Bessel}	Demi-grand axe de l'ellipsoïde de Bessel 1841 : 6377397.155 [m]
f_{Bessel}	Applatissement de l'ellipsoïde de Bessel 1841 : 1/299.1528128 [-]

Annexe A

Résultats des mesures cinématiques

A.1 Session 2

TABLEAU A.1 – Informations sur le calcul - session cinématique n° 2

Paramètre	Valeur
Date	16.07.2024
Début	09:21:01 GPST
Fin	09:23:38 GPST
Nb époques calculées	159
Nb époques avec solution	157

TABLEAU A.2 – Indicateurs statistiques - session cinématique n° 2

Méthode		Biais	Écart-type	RMSE	Minimum	Maximum
Smartphone DGPS	ΔE	-0.47	1.03	1.13	0.00	3.09
RTKLIB DGPS	ΔE	2.34	1.02	2.55	0.10	4.59
Smartphone location	ΔE	1.01	0.40	1.09	0.05	1.73
RTKLIB Single	ΔE	3.66	1.05	3.81	1.25	6.09
Smartphone DGPS	ΔN	-0.82	1.80	1.98	0.04	6.42
RTKLIB DGPS	ΔN	-0.23	1.98	1.99	0.01	5.99
Smartphone location	ΔN	2.30	0.90	2.46	0.48	4.29
RTKLIB Single	ΔN	2.14	2.20	3.07	0.02	8.82
Smartphone DGPS	ΔH	-1.09	4.54	4.67	0.04	14.92
RTKLIB DGPS	ΔH	4.22	3.36	5.40	0.03	14.88
Smartphone location	ΔH	1.33	1.68	2.14	0.00	5.06
RTKLIB Single	ΔH	5.07	3.43	6.12	0.18	15.24

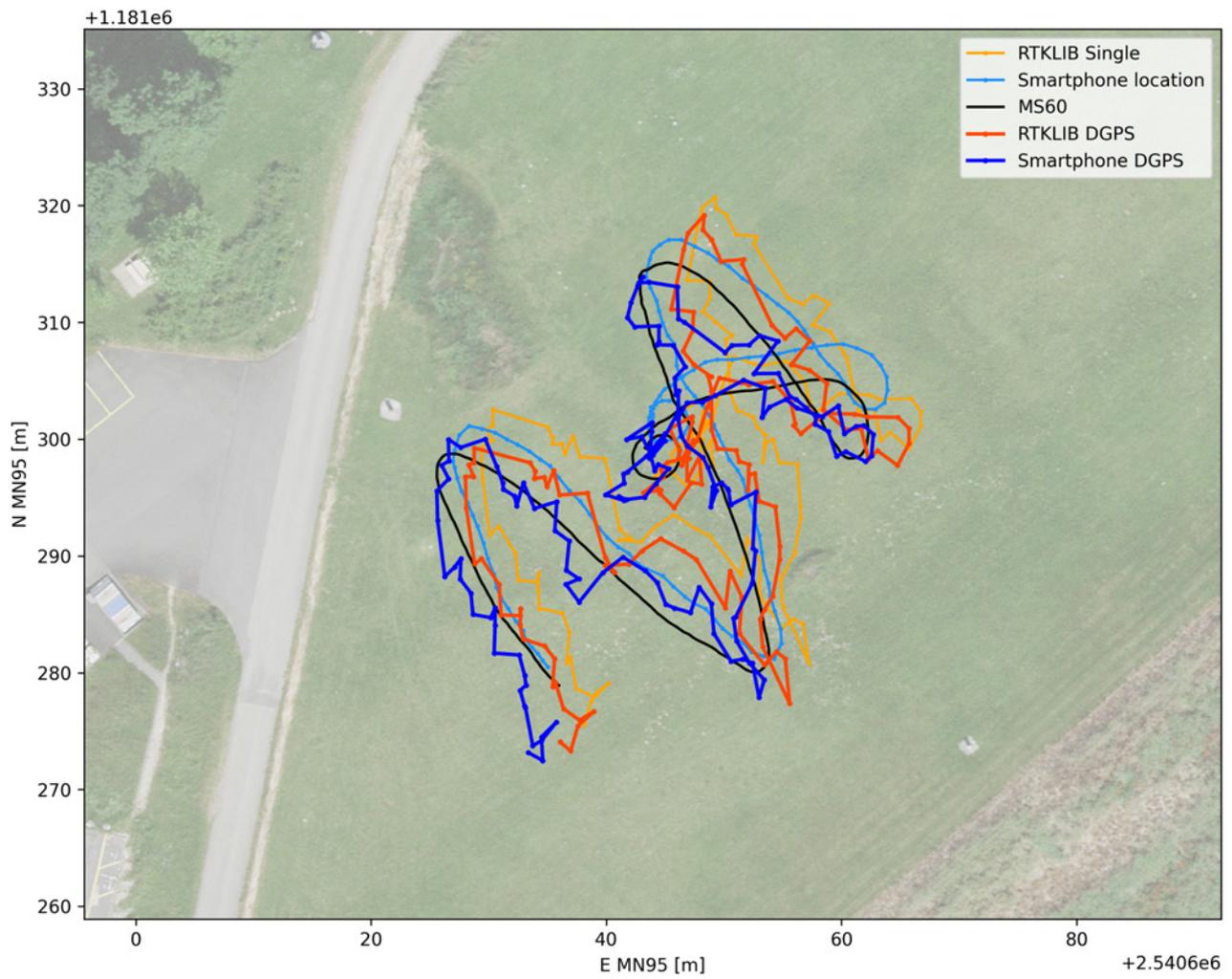


FIGURE A.1 – Résultats planimétriques - session cinématique n°2

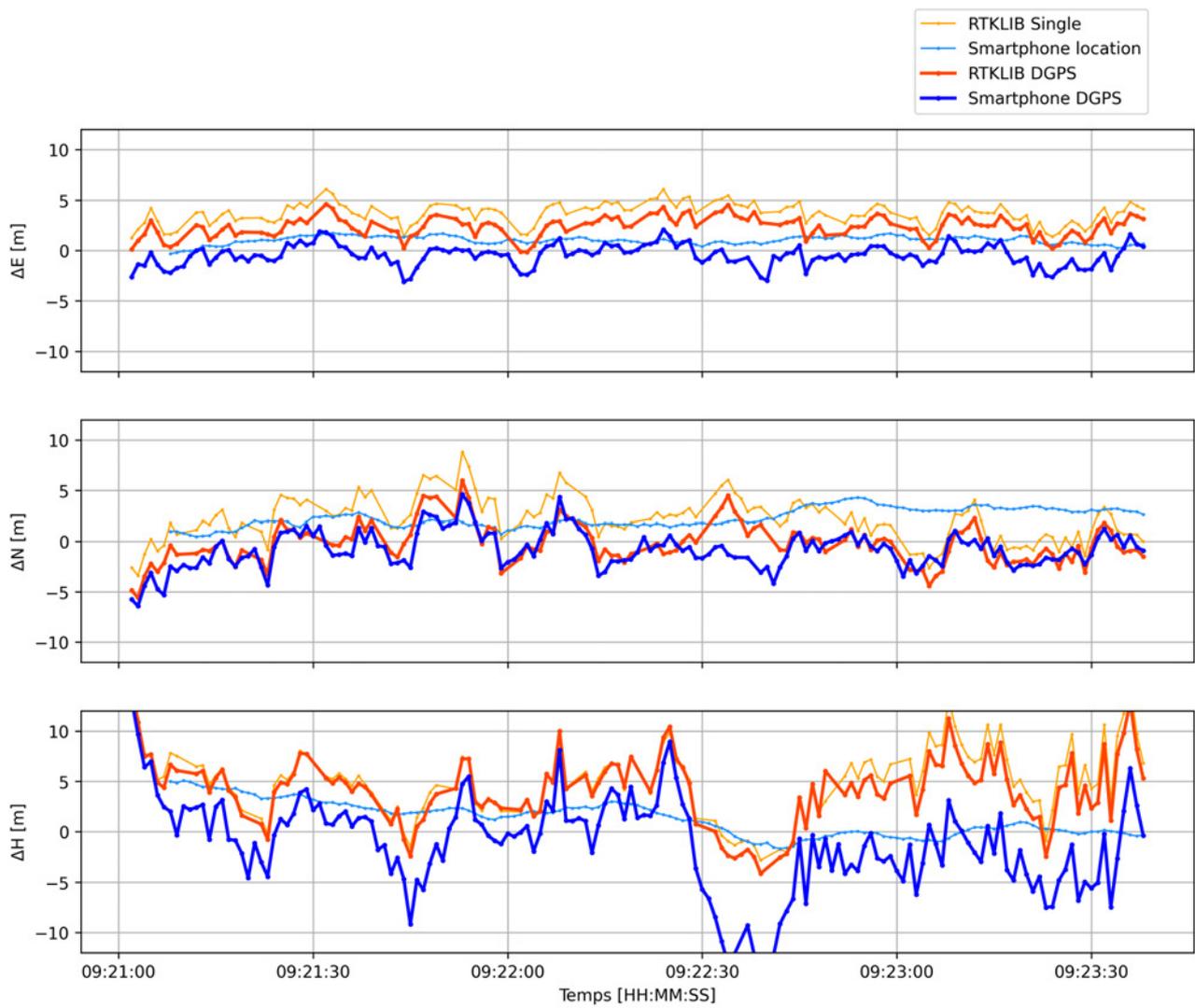


FIGURE A.2 – Série temporelle des différences avec la position vraie - session cinématique n° 2

A.2 Session 3

TABLEAU A.3 – Informations sur le calcul - session cinématique n° 3

Paramètre	Valeur
Date	16.07.2024
Début	09:25:50 GPST
Fin	09:28:05 GPST
Nb époques calculées	136
Nb époques avec solution	133

TABLEAU A.4 – Indicateurs statistiques- session cinématique n° 3

Méthode		Biais	Écart-type	RMSE	Minimum	Maximum
Smartphone DGPS	ΔE	-0.16	1.23	1.24	0.01	4.14
RTKLIB DGPS	ΔE	0.10	1.26	1.26	0.01	4.72
Smartphone location	ΔE	1.22	0.42	1.29	0.11	2.16
RTKLIB Single	ΔE	1.42	1.24	1.89	0.05	5.50
Smartphone DGPS	ΔN	-0.34	1.85	1.88	0.01	5.90
RTKLIB DGPS	ΔN	-0.24	2.13	2.14	0.03	7.92
Smartphone location	ΔN	2.24	0.62	2.33	0.22	4.04
RTKLIB Single	ΔN	1.80	2.13	2.79	0.04	10.30
Smartphone DGPS	ΔH	0.31	3.82	3.83	0.03	15.05
RTKLIB DGPS	ΔH	0.57	3.35	3.40	0.01	14.04
Smartphone location	ΔH	2.37	1.48	2.80	0.03	5.36
RTKLIB Single	ΔH	2.10	3.34	3.95	0.05	15.57

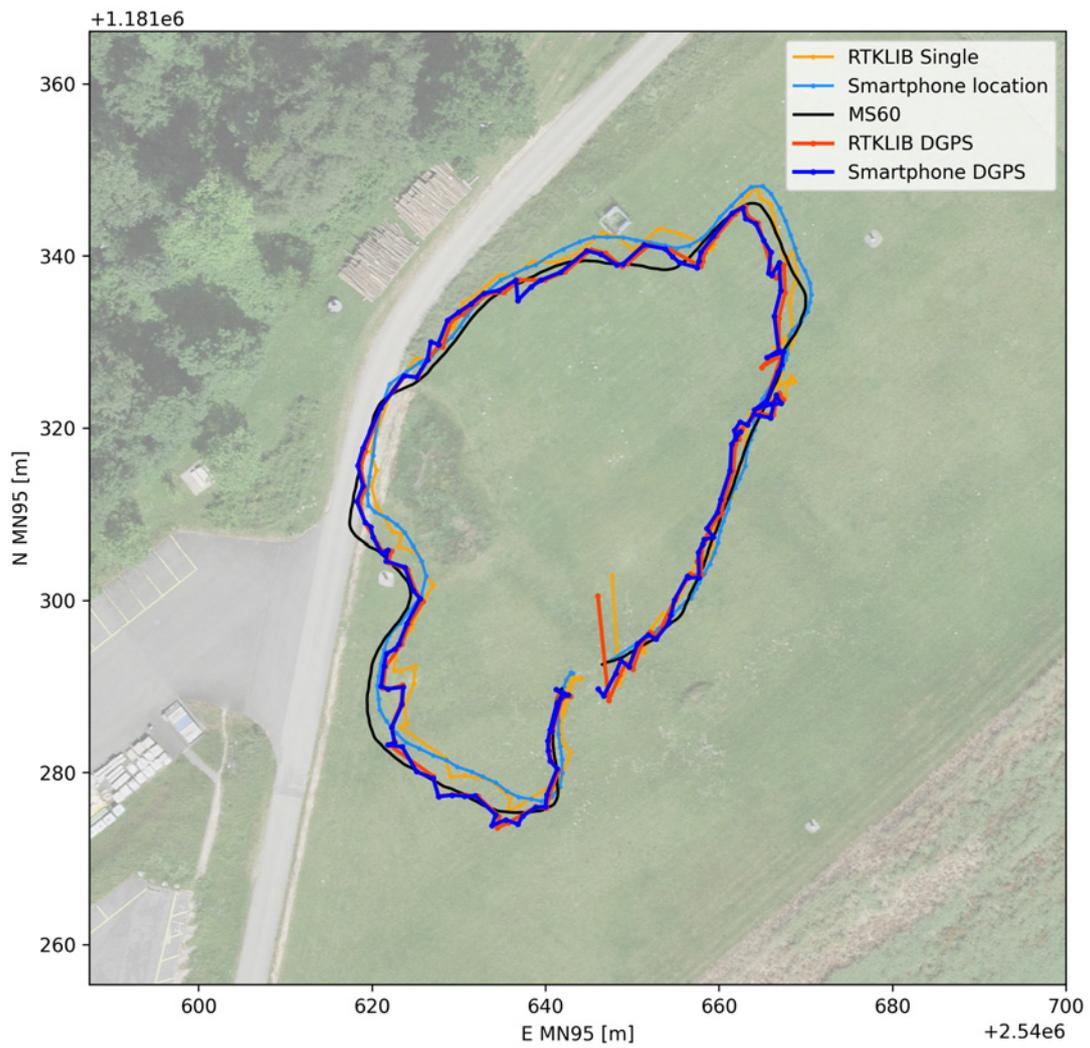


FIGURE A.3 – Résultats planimétriques - session cinématique n° 3

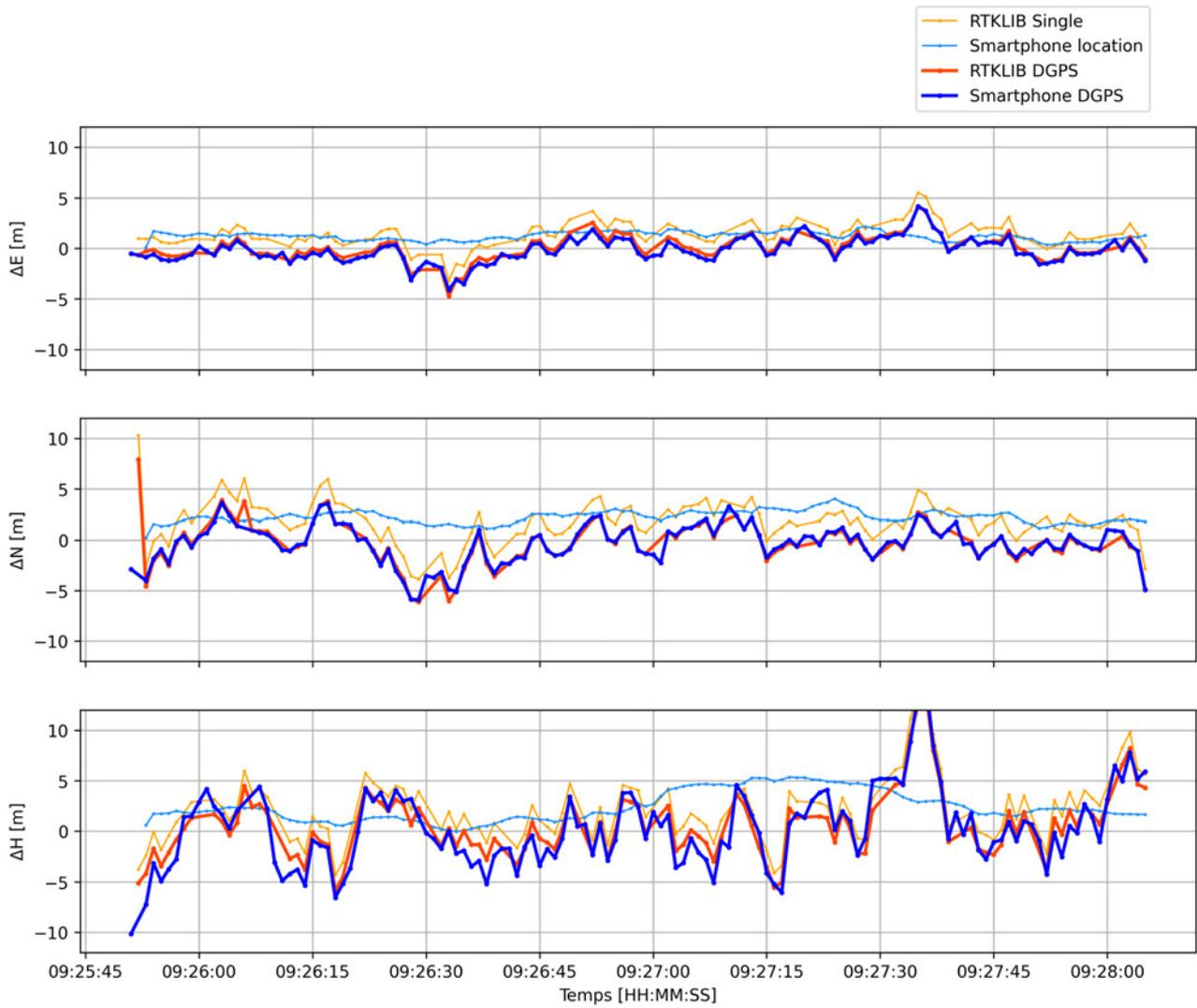


FIGURE A.4 – Série temporelle des différences avec la position vraie - session cinématique n° 3

Annexe B

Caractéristiques du smartphone

Le smartphone utilisé pour réaliser le développement de l'application ainsi que les tests est un Samsung Galaxy A13. Les caractéristiques techniques du smartphone et du récepteur GNSS embarqué sont décrites dans le tableau B.1. Elles sont obtenues à partir de l'application open source GPSTest qui permet de tester les données de positionnement disponibles sur un appareil Android. La liste complète des caractéristiques des appareils se trouve dans une base de données disponible à l'adresse suivante : <https://bit.ly/gpctest-device-database> (consultée le 3 février 2024).

TABLEAU B.1 – Caractéristiques du smartphone utilisé pour les tests

<i>Fabricant</i>	Samsung
<i>Modèle</i>	SM-A137F
<i>Nom</i>	Galaxy A13
<i>Version Android</i>	13
<i>Niveau de l'API</i>	33
<i>Année du matériel GNSS</i>	2020
<i>Nom du modèle du matériel GNSS</i>	MediaTek, MT0000
<i>Double fréquence</i>	non supporté
<i>GNSS prise en charge</i>	GPS, GLONASS, GALILEO, BEIDOU
<i>GNSS CFs¹</i>	B1, E1, L1
<i>SBAS prise en charge</i>	EGNOS
<i>SBAS CFs</i>	L1
<i>Mesures brutes</i>	prise en charge
<i>Phase du porteuse</i>	prise en charge
<i>Messages de navigation</i>	prise en charge
<i>NMEA</i>	prise en charge
<i>Service d'injection de données de satellite</i>	prise en charge
<i>prédit (PSDS)</i>	
<i>Injection de temps</i>	prise en charge
<i>Contrôle automatique du gain</i>	prise en charge
<i>Information sur l'antenne</i>	non supporté

Le Samsung Galaxy A13 ne traite pas les signaux double fréquence donc il ne prend pas en charge les signaux GPS sur la bande L2. Dans le cadre de ce projet, seuls les signaux GPS sur la bande L1 sont nécessaires puisque l'application ne prévoit pas de faire un calcul de positionnement double fréquence.

Le smartphone prend en charge le système d'augmentation de la navigation par satellite (SBAS) EGNOS (*European Geostationary Navigation Overlay Service*) qui désigne le système européen de navigation par recouvrement géostationnaire. Ces services diffusent des corrections pour les signaux GPS aux récepteurs compatibles afin d'améliorer la précision de leur position. Dans ce travail, la position du smartphone est déterminée uniquement à partir des données brutes du smartphone et de la station de base sans aucune correction extérieure.

Les mesures brutes ainsi que les messages de navigation nécessaires au calcul de la position des satellites sont disponibles sur cet appareil.

1. CF de l'anglais *carrier frequency* définit les fréquence de l'onde porteuse supportées.