

# LOCALISATION GPS CENTIMÉTRIQUE À FAIBLE COÛT

Département : Technologies industrielles (TIN)  
Unité d'enseignement : Travail de Bachelor

Auteurs : Aurélien Berthelot  
Orientation : Électronique Embarquée et Mécatronique  
Professeur : Bertrand Hochet  
Date : 7 juin 2020

## Travail de bachelor 2019 -2020

### localisation GPS centimétrique à faible coût

#### Enoncé :

Depuis peu, sont apparus sur le marché des modules GPS capables de calculer leur position à 2.5 cm près. Ces modules fonctionnent par paire et communiquent entre eux par radio. Ils embarquent un algorithme appelé RTK (Real Time Kinematic) permettant de corriger les erreurs de positionnement dues par exemple aux incertitudes sur les trajets des signaux des satellites à travers la ionosphère.

Le projet consiste à réaliser et tester un dispositif capable de calculer sa position avec une précision meilleur que 5 cm.

# Table des matières

|   |           |
|---|-----------|
| <b>1 Introduction</b>   | <b>7</b>  |
| 1.1 But du projet . . . . .   | 7         |
| 1.2 Planning . . . . .  | 8         |
| <b>2 Fonctionnement et évolutions du GPS</b>  | <b>9</b>  |
| 2.1 Les Débuts du GPS . . . . .   | 9         |
| 2.2 Les améliorations du GPS . . . . .  | 9         |
| 2.2.1 Le LBAS . . . . .   | 9         |
| 2.2.2 Le SBAS . . . . .   | 11        |
| 2.3 Le GPS aujourd’hui . . . . .  | 12        |
| 2.4 Les signaux GPS . . . . .   | 12        |
| 2.5 Le RTK . . . . .  | 15        |
| 2.6 Les types d’erreurs GPS . . . . .   | 19        |
| <b>3 Présentation des modules u-blox</b>  | <b>20</b> |
| 3.1 Recommandations d’utilisation . . . . .   | 21        |
| 3.2 Mise en place d’un GPS avec algorithme RTK depuis u-center . . . . .  | 22        |
| 3.2.1 Configuration de la base avec u-center . . . . .  | 23        |
| 3.2.2 Configuration du rover avec u-center . . . . .  | 28        |
| 3.3 Mise en place d’un GPS avec algorithme RTK depuis un microcontrôleur . . . . .  | 30        |
| 3.3.1 Trouver les commandes UBX . . . . .   | 30        |
| 3.3.2 Vérifications des commandes UBX . . . . .   | 32        |
| <b>4 Mesures</b>  | <b>35</b> |
| 4.1 Récuperation et mise en forme des coordonnées GPS . . . . .   | 35        |
| 4.1.1 Récupérer les informations GPS depuis U-center . . . . .  | 35        |
| 4.1.2 Conversion de coordonnées du format DM en format DD . . . . .   | 38        |
| 4.1.3 Calcul de distance orthodromique entre deux points . . . . .  | 38        |
| 4.2 Mesure de stabilité . . . . .   | 39        |
| 4.2.1 Déroulement de la mesure . . . . .  | 39        |
| 4.2.2 Analyse de la mesure . . . . .  | 39        |
| 4.3 Mesure de distance . . . . .  | 44        |
| 4.3.1 Déroulement de la mesure . . . . .  | 44        |
| 4.3.2 Analyses des mesures . . . . .  | 44        |
| 4.4 Étude de la précision des mesures en fonction de la distance entre le rover et la base . . . . .                              | 47        |
| 4.4.1 Liste du matériel . . . . .   | 47        |
| 4.4.2 Déroulement de la mesure . . . . .  | 48        |
| 4.4.3 Analyses des mesures . . . . .  | 48        |
| 4.5 Étude de la précision des mesures en fonction de la distance entre le rover et la base dans un environnement dégagé . . . . . | 55        |
| 4.5.1 Liste du matériel . . . . .   | 55        |
| 4.5.2 Déroulement de la mesure . . . . .  | 55        |
| 4.5.3 Analyses des mesures . . . . .  | 55        |
| 4.6 Correction de la formule de calcul de distance . . . . .  | 61        |

---

|  |           |
|--|-----------|
| <b>5 Hardware</b>  | <b>66</b> |
| 5.1 Schématique . . . . .  | 66        |
| 5.1.1 Microcontrôleur et bluetooth . . . . .                         | 66        |
| 5.1.2 GPS . . . . .  | 67        |
| 5.1.3 Communication radio . . . . .                                  | 69        |
| 5.1.4 Alimentation . . . . .   | 69        |
| 5.1.5 Charge par induction . . . . .                                 | 71        |
| 5.1.6 ON/OFF et reset . . . . .                                      | 72        |
| 5.1.7 NFC . . . . .  | 74        |
| 5.1.8 SD ou MMC . . . . .  | 74        |
| 5.1.9 Connecteurs de programmation et leds . . . . .                 | 75        |
| 5.2 Layout . . . . .   | 76        |
| 5.2.1 Disposition des composants . . . . .                           | 76        |
| 5.2.2 Routage des antennes . . . . .                                 | 79        |
| 5.2.3 Routage des alimentations . . . . .                            | 80        |
| 5.2.4 Réalisation de l'antenne pour la communication radio . . . . . | 81        |
| <b>6 Continuité et améliorations du projet</b>                       | <b>86</b> |
| <b>7 Conclusion</b>  | <b>87</b> |
| <b>8 Signatures</b>  | <b>87</b> |
| <b>9 Liste des abréviations</b>                                      | <b>88</b> |
| <b>10 Annexes</b>  | <b>91</b> |

## Table des figures

|    |  |    |
|----|--|----|
| 1  | Principe de fonctionnement du LBAS . . . . .   | 10 |
| 2  | Principe de fonctionnement du SBAS . . . . .   | 11 |
| 3  | Couverture des satellites géostationnaire pour l'utilisation du SBAS .   | 12 |
| 4  | Modulation des signaux GPS . . . . .   | 14 |
| 5  | Modulation BPSK . . . . .  | 14 |
| 6  | Principe de fonctionnement du RTK . . . . .  | 15 |
| 7  | Comparaison de la précision des signaux GPS sans, puis avec RTK .  | 17 |
| 8  | Illustration de l'ambiguïté de mesure de phase . . . . .   | 18 |
| 9  | Photo du module GPS avec RTK . . . . .   | 20 |
| 10 | Interfaces de u-center . . . . .   | 21 |
| 11 | Interfaces de message view . . . . .   | 23 |
| 12 | Configuration du point de référence de la base . . . . .   | 24 |
| 13 | Informations disponibles pendant que la base affine sa position . . . .  | 25 |
| 14 | Activation des messages UBX . . . . .  | 26 |
| 15 | Configuration du port radio de la base . . . . .   | 27 |
| 16 | Configuration des messages RTCM de la base . . . . .   | 28 |
| 17 | Configuration du port radio du rover . . . . .   | 29 |
| 18 | Structure d'un message UBX . . . . .   | 30 |
| 19 | Affichage des commandes UBX dans U-center . . . . .  | 31 |
| 20 | Module avec configuration d'usine . . . . .  | 32 |
| 21 | Paramétrage du port COM sur Coolterm . . . . .   | 33 |
| 22 | Envoi d'un message pour configurer un module en rover RTK . . . .  | 34 |
| 23 | Rover configuré grâce à une commande UBX . . . . .   | 34 |
| 24 | Composition d'un fichier UBX . . . . .   | 36 |
| 25 | Messsages GLL . . . . .  | 37 |
| 26 | Détail d'une trame GLL . . . . .   | 37 |
| 27 | Donnée GPS en colonnes : Latitude, Longitude, Heure UTC . . . .  | 38 |
| 28 | Cordonnées pour démontrer la distance orthodromique entre deux points . . . . .                                    | 38 |
| 29 | Disposition des 28800 points GPS . . . . .   | 40 |
| 30 | Dispersion des 28800 points GPS . . . . .  | 41 |
| 31 | Dispersion des 28800 points GPS, divisé par heure . . . . .  | 42 |
| 32 | Tableau récapitulatif de la précision des coordonnée GPS . . . . .   | 43 |
| 33 | Disposition des 2 x 3600 points de mesure distancés de 50 cm . . . .   | 45 |
| 34 | Histogramme des distances entre les 3600 points de départ et les 3600 points d'arrivée . . . . .                   | 46 |
| 35 | Tableau récapitulatif de la précision des distances par coordonnée GPS   | 46 |
| 36 | Disposition des points de mesures. Distance depuis la base : 1, 2, 5, 10, 15 et 20 mètres . . . . .                | 49 |
| 37 | Répartition des points de mesures . . . . .  | 50 |
| 38 | Dispersion des mesures par rapport à la moyenne . . . . .  | 51 |
| 39 | Tableau récapitulatif de la précision des distances par coordonnée GPS   | 52 |
| 40 | Distances obtenues à partir des points se trouvant à 1 m vers les points se trouvant à 2,5,10,15 et 20 m . . . . . | 53 |

|    |  |    |
|----|--|----|
| 41 | Tableau récapitulatif de la précision des distances par coordonnée GPS                                   | 54 |
| 42 | Disposition des points de mesures. Distance depuis la base : 1, 2, 5, 10, 15 et 20 mètres                | 56 |
| 43 | Répartition des points de mesures  | 57 |
| 44 | Dispersion des mesures par rapport à la moyenne  | 58 |
| 45 | Tableau récapitulatif de la précision des distances par coordonnée GPS                                   | 58 |
| 46 | Distances obtenues à partir des points se trouvant à 1 m vers les points se trouvant à 2,5,10,15 et 20 m | 60 |
| 47 | Tableau récapitulatif de la précision des distances par coordonnée GPS                                   | 61 |
| 48 | Comparaison des distances calculables dans matlab avant et après la correction de formule                | 63 |
| 49 | Tableau de comparaison des distances en utilisant la formule erronée et corrigée                         | 64 |
| 50 | Schématique du microcontrôleur et du bluetooth   | 67 |
| 51 | Schématique du GPS   | 68 |
| 52 | Schématique du module radio  | 69 |
| 53 | Schématique de l'alimentation  | 70 |
| 54 | Schématique de la charge par induction   | 71 |
| 55 | Schématique pour éteindre le système   | 72 |
| 56 | Schématique pour allumer ou reset le système   | 73 |
| 57 | Schématique du NFC   | 74 |
| 58 | Schématique du connecteur micro SD   | 75 |
| 59 | Schématique des connecteurs  | 75 |
| 60 | Schématique des leds   | 76 |
| 61 | Layer stack du PCB   | 76 |
| 62 | Routage de la couche top   | 77 |
| 63 | Routage de la couche bottom  | 77 |
| 64 | Routage des couches top et bottom sans le plan de masse ni les plans d'alimentation                      | 78 |
| 65 | Configuration de Linecalc pour l'antenne 868 MHz   | 79 |
| 66 | Routage des plans d'alimentations sur la couche 3  | 80 |
| 67 | Plan de GND sur la couche 2  | 81 |
| 68 | Exemple d'antenne PIFA   | 81 |
| 69 | Substrat utilisé pour l'antenne RF   | 82 |
| 70 | Antenne PIFA 868 MHz réalisée sur ADS  | 83 |
| 71 | Résultats de simulation de l'antenne PIFA sur ADS  | 83 |
| 72 | Antenne PIFA redessinée sur Altium   | 84 |
| 73 | Antenne positionnée à coté du PCB du module  | 85 |

# 1 Introduction

## 1.1 But du projet

Le but de ce projet est de réaliser un système capable d'obtenir sa position GPS avec une précision de 5 cm. Ce système sera contenu dans un boîtier qui embarquera une batterie afin d'alimenter tout l'électronique nécessaire. Il pourra notamment être utilisé lors de courses d'aviron afin de connaître la position exacte de chaque bateau. Comme il serait fixé sur le bateau, il doit être complètement étanche. Un tel système permettrait de détecter les faux départs ou l'ordre d'arrivée d'une course. En plus de servir de GPS, ce système serait capable de récupérer des informations de différents capteurs qui pourraient se trouver sur le bateau tel que des capteurs de puissance afin de suivre l'évolution de l'effort fourni par les rameurs où des capteurs de vitesse.

Ce projet étant une évolution d'un ancien travail de bachelor qui consistait à faire un GPS avec une résolution proche de 1 mètre dans de bonnes conditions. Il va me falloir étudier son travail afin de définir quelles parties peuvent être conservées et quelles parties doivent être modifiées ou remplacées. Il faudra également ajouter de nouvelles fonctionnalités pour compléter le cahier des charges. Un planning est disponible à la page suivante.

## 1.2 Planning

| Planning du travail de bachelier                       | début : 23.09.2019 |
|--|--------------------|
| Planning prévu   | Planning réel      |
| liste des tâches                                       | semaine du :       |
|  | Septembre          |
|  | 23 30              |
|  | 7 14 21 28         |
|  | 4 11 18 25         |
| Analyse du projet précédent                            | Décembre           |
| Mise en place du RPK avec les modules de développement | 2 9 16 23 30       |
| Schematique altium                                     | 6 13 20 27         |
| Rouage PCB + antenne                                   | 3 10 17 24         |
| Montage + vérification du pcb                          | 2 9 16 23          |
| Programmation du microcontrôleur et test du système    | Mars               |
| Mesures et traitements des données                     | 30                 |
| Rédaction du rapport + recherche d'informations        | 6 13 20 27         |
|  | 4 11 18 25         |
|  | 1 Juin             |

## 2 Fonctionnement et évolutions du GPS

### 2.1 Les Débuts du GPS

Le premier système de positionnement par satellite fut le TRANSIT, mis en place par les Etats-Unis en 1964. La précision au début de son utilisation étant d'environ 1 km. Celui-ci était réservé exclusivement à un usage militaire, mais TRANSIT possédait de gros défauts, il fallait attendre parfois plus de 24 heures afin de pouvoir connaître sa position car seulement 6 satellites composaient cette constellation. Au vu des utilisations possibles d'un tel système, les Etats-Unis décidèrent de mettre au point un second système de positionnement par satellites, plus performant. [1–3]

Pour connaître sa position grâce à des satellites, il faut utiliser la triangulation. Pour cela, il faut disposer de 3 satellites pour avoir une position en 2D sur une map, et de 4 pour disposer d'une position en 3D, ce qui inclut cette fois ci l'altitude en plus. Les Etats-Unis ont trouvé que pour disposer d'un service de positionnement 3D en tout temps, il faut une constellation de 24 satellites minimum. Ainsi entre 1978 et 1997, 39 satellites ont été envoyés afin de couvrir la surface de la terre.<sup>1</sup> Le GPS<sup>2</sup> sera utilisé notamment pendant la guerre du Golfe en 1990 - 1991, avant d'être déclaré pleinement opérationnel en 1994. [1, 4–8]

Dès sa mise en fonctionnement, le système GPS Américain est utilisable par les civils, mais le signal y est volontairement dégradé, la précision étant réservée exclusivement à un usage militaire. L'armée obtenait une précision d'environ 10 m, les civils devaient se contenter d'une précision de 100 m. Cette dégradation de signal, portant le nom de SA (Selective Availability), était contraignante pour beaucoup d'applications civiles. C'est pourquoi dès 1980, l'USCG (United States Coast Guard) et la FAA (Federal Aviation Administration) cherchèrent des solutions afin de contourner la limitation imposée par l'armée Américaine. [1, 9]

### 2.2 Les améliorations du GPS

#### 2.2.1 Le LBAS

Le LBAS (Local Based Augmentation System) est un concept étudié rapidement après la mise à disposition du GPS par les Etats-Unis pour les civils.

---

1. Les premiers satellites n'ayant une durée de vie que de 4,5 ans, une partie des envois n'avaient pas pour but d'ajouter des satellites à la constellation mais d'en remplacer certains par d'autres avec une durée de vie plus importante.

2. GPS désigne ici uniquement les satellites et le système mis en place par les Etats-Unis (Global Positioning System). Par la suite, d'autres pays mettront en orbites des systèmes équivalents afin de ne pas être dépendants du système Américain tels que la Russie (Glonass), la Chine (Beidou-2) et l'Europe (Galileo). Le terme précis pour désigner la totalité de ces constellations est GNSS (Global Navigation Satellite System), mais par abus de langage, le sigle GPS est aujourd'hui largement utilisé pour désigner un ou plusieurs systèmes, quelque soit le pays qui l'a mis en orbite. Dans la suite de ce document, le terme GPS ne désignera pas spécifiquement le système Américain mais n'importe quel système en orbite capable d'indiquer une position, sauf mention contraire.

Cette technique de GPS différentiel (DGPS ou DGNSS) avait pour but de résoudre le problème de précision causé par le SA. Bien que le SA soit un système de chiffrement, il n'était pas changé fréquemment et s'appliquait à une grande zone géographique. L'idée est donc venue de faire fonctionner les GPS par paire. On positionne un GPS fixe à un point bien précis, avec des coordonnées connues, puis on compare avec les coordonnées que l'on obtient avec les satellites et le SA. Il faut ensuite calculer l'écart entre la position connue et celle indiquée par les satellites, puis transmettre cette correction par radio au second GPS afin de contrer le chiffrement SA. Cette solution permettait d'obtenir une précision de 15 mètre, soit presque la même précision qu'un signal GPS sans le chiffrement SA. La figure 1 montre le fonctionnement d'un système LBAS. [9–12]

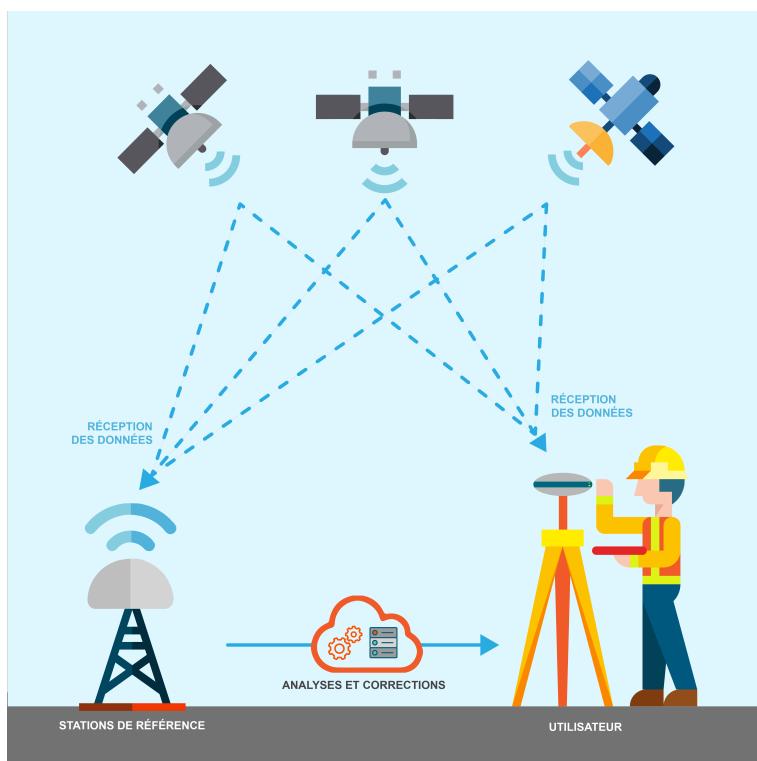


FIGURE 1 – Principe de fonctionnement du LBAS, source : [www.reseau-teria.com](http://www.reseau-teria.com) [12]

L'USCG fut l'une des agences les plus actives lors du développement du LBAS, ainsi dès 1996, ils mirent en place un réseau pour couvrir la plupart des ports d'escales avec des balises terrestres. Ils envisagèrent de couvrir la totalité des Etats-Unis, mais cette technologie possède également des défauts, car si proche de la base, les corrections à appliquer aux GPS alentours sont identiques, la qualité de correction diminue avec la distance. Ce point posait de gros problèmes car la majorité des bases terrestres étaient à proximité des villes, ce qui impliquait que les zones peu peuplées ne seraient pas en mesure de bénéficier de la même précision que les endroits plus habités. [9, 10, 12]

### 2.2.2 Le SBAS

En parallèle du développement du LBAS, d'autres agences comme la FAA cherchaient d'autres solutions pour se passer du désagrément du SA. Ils mirent en place le SBAS (Satellite Based Augmentation System). Comme le LBAS, le SBAS a besoin de bases au sol pour fonctionner, mais contrairement au LBAS, ce ne sont pas directement les bases qui communiquent les facteurs de correction aux utilisateurs. Les facteurs de corrections calculé par les différentes bases sont envoyés vers des satellites géostationnaires. Ces satellites renvoient les informations d'erreurs de tous les satellites à tous les récepteurs GPS compatible SBAS. Le SBAS n'est donc pas dépendant de la distance qui sépare les bases et les utilisateurs GPS. La précision s'effectuant sur des zones beaucoup plus larges, tout le monde peut en profiter, tant que l'on se trouve dans la zone de couverture des satellites géostationnaires. La figure 2 illustre le fonctionnement du SBAS. [5, 9, 10, 12]

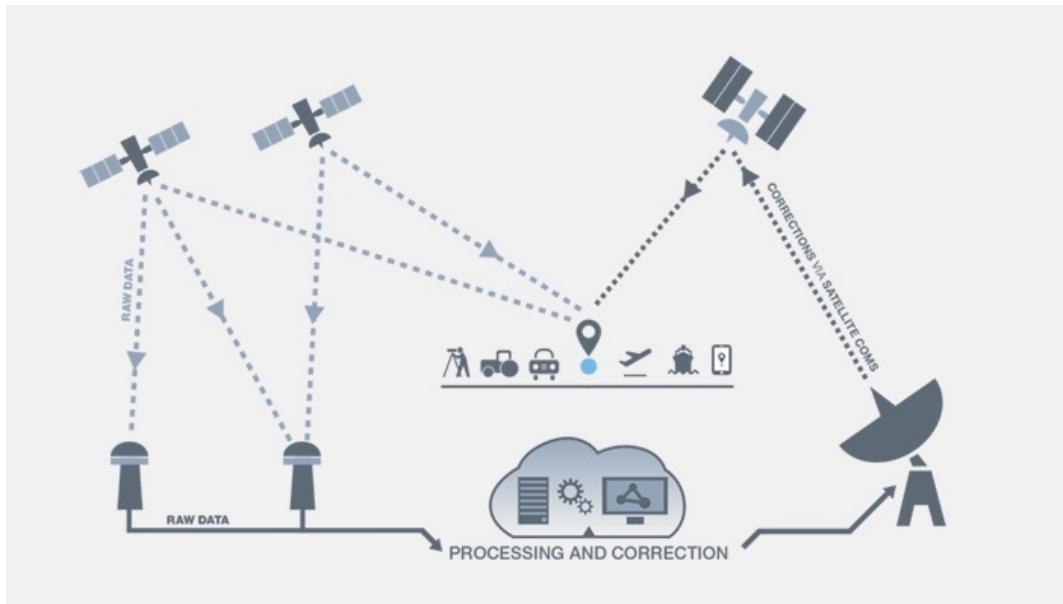


FIGURE 2 – Principe de fonctionnement du SBAS, source : [www.geospatialworld.net](http://www.geospatialworld.net) [13]

Comme le SBAS utilise des satellites géostationnaires, ce sont donc des types de satellites différents de ceux qui composent les constellations en orbite autour de la terre. Ainsi, chaque pays ou groupe de pays souhaitant profiter de cette technologie doit positionner des satellites au dessus de lui. Même si chaque pays nomme cette technologie différemment, le principe fonctionnement reste le même. Les Etats-Unis disposent donc du WAAS, la Russie du SDMC et l'Europe de EGNOS. D'autres systèmes existent ou sont en préparation pour couvrir d'autres parties de la planète. La carte à la figure 3 montre la couverture approximative des systèmes existants. [5]

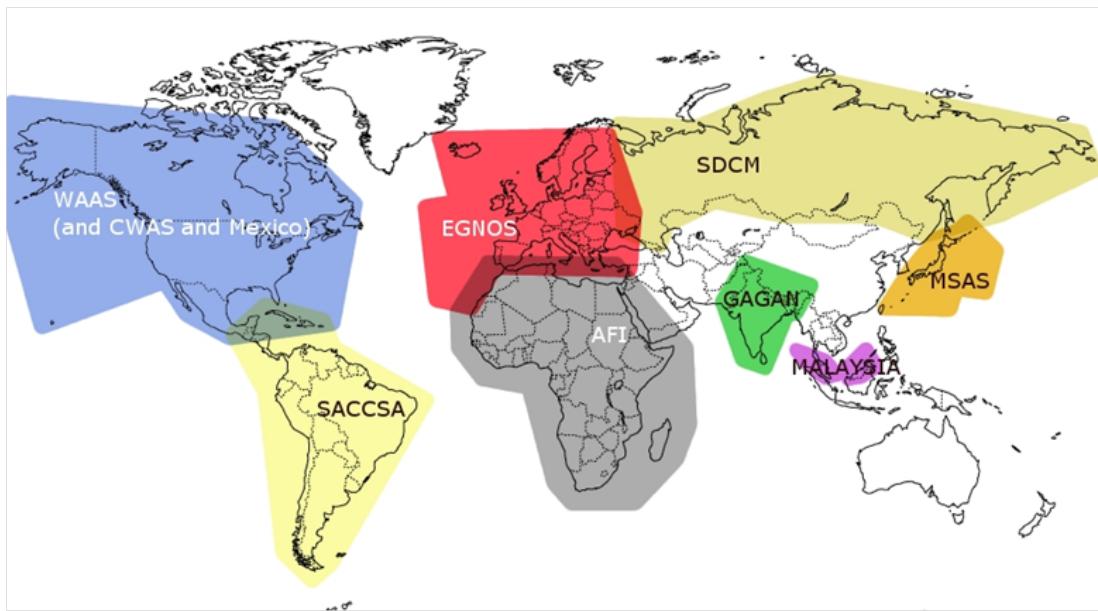


FIGURE 3 – Couverture des satellites géostationnaire pour l'utilisation du SBAS,  
source : gssc.esa.int [14]

### 2.3 Le GPS aujourd’hui

Suite à la découverte et la mise en place du SBAS et du LBAS, les Etats-Unis conclurent que le chiffrement SA n'était plus daucune utilité car ces technologies fournissaient une meilleure précision que le GPS avec le chiffrement SA désactivé. Le chiffrement SA a donc été désactivé en 2000 afin que tout le monde puisse profiter du GPS sans restriction. [9]

Le SBAS permet aujourd’hui une précision de 7,6 mètres à 1 mètres selon le nombre et la qualité des signaux satellites reçus, ainsi que la position qui détermine le SBAS disponible. Le LBAS arrive à une précision très similaire. [12, 15, 16]

### 2.4 Les signaux GPS

Les signaux GNSS émis par les satellites sont une combinaison de plusieurs signaux avec des fonctions bien différentes. Pour émettre une information sur terre, les satellites ont d'abord besoin d'une porteuse. Une porteuse est un signal sinusoïdale de fréquence très précise. Comme le principe de triangulation utilise le temps que mettent les signaux à parcourir le chemin entre le satellite et l'utilisateur pour déterminer une position, cette base de temps est très importante. C'est pourquoi chaque satellite embarque sa propre horloge atomique. La fréquence des horloges atomiques des satellites est de 10.23 MHz, elle est appelée fréquence fondamentale ( $f_o$ ). A partir de cette fréquence, la majorité des satellites génèrent plusieurs porteuses, les plus connus sont nommée L1 et L2 et sont respectivement à 1.57542 GHz et 1.22760 GHz. Ces fréquences ont été choisies afin de profiter de la précision d'horloge dispo-

nible dans les satellites et correspondent à des multiples entiers de  $f_o$ , soit  $154 \cdot f_o$  pour L1 et  $120 \cdot f_o$  pour L2. La fréquence L2 était historiquement réservée à l'usage militaire, mais des signaux ont été rajoutés à cette fréquence pour une utilisation civile (L2C) à partir de 2014. A cette même période, une fréquence L5 avec une partie des signaux pour l'usage civil (L5C) a également commencé à être diffusée. Ces nouvelles fréquences ont permis le développement de GPS civil bi-fréquences et tri-fréquences [17–19]

Un premier code généré depuis le satellite est le code C/A (coarse acquisition). Ce code contient des informations pour identifier le satellite. C'est un message binaire qui n'est diffusé que sur la fréquence L1 et n'est pas chiffré. Sa fréquence est de 1.023 MHz.

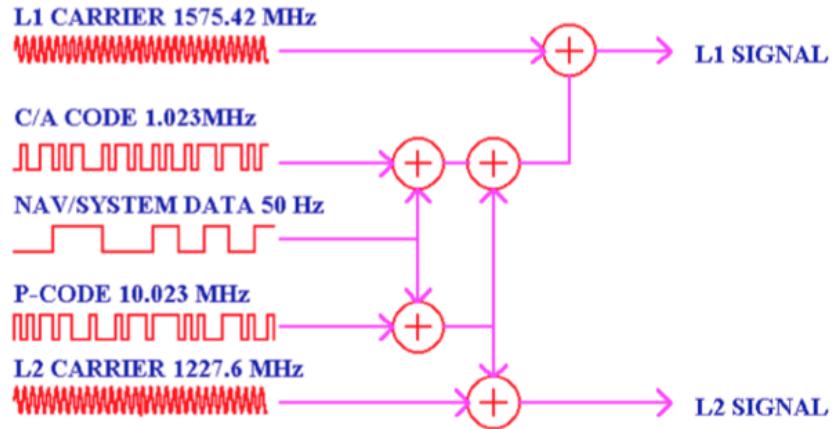
Un deuxième code est envoyé depuis les satellites, il s'agit du code Y(P). Ce code est chiffré et est réservé à un usage militaire. Il est transmis par les fréquences L1 et L2.

Un troisième code est généré à bord des satellites, il s'agit des messages de navigations. Ces messages sont diffusés sur les fréquences L1, L2 et L5 et permettent d'obtenir toutes les informations provenant du satellite tels que : [19]

- La synchronisation de l'horloge interne du récepteur
  - Le calcul des coordonnées du point de station
  - Les corrections de temps
  - Les informations de position et les données orbitales du satellite
  - ...
- le message de navigation a une fréquence de 50 Hz.

C'est grâce aux messages de navigation que les récepteurs GPS peuvent calculer leur position.

La figure 4 montre la modulation des différents signaux pour obtenir les signaux L1 et L2.



## GPS SATELLITE SIGNALS

PH DANA 10/92

FIGURE 4 – Modulation des signaux GPS, source : geologie.ens.fr [18]

Dans cette figure, la fréquence L1 contient le code C/A ainsi que les messages de navigation chiffré et non chiffré, alors que la fréquence L2 ne contient que les messages chiffré par le code Y(P). Les informations civil sur la fréquences L2 ne sont pas représentées ici.

La modulation des signaux s'effectue par modulation BPSK (Binary phase-shift keying) comme sur la figure 5 [19]

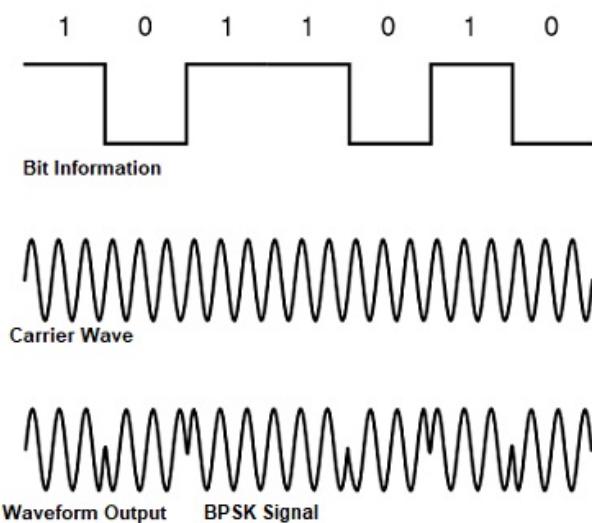


FIGURE 5 – Modulation BPSK, source : electronicsandcommunications.com [20]

Comme tous les signaux ont des fréquences qui sont des sous-multiples de la fréquence de l'horloge atomique, le changement de phase surviendra toujours lorsque la porteuse passe par 0.

Il est important de noter que lors de l'utilisation d'un système GPS, aucun signal n'est émis de l'utilisateur vers les satellites. C'est le récepteur qui calcule sa position à partir des informations de position de chaque satellite contenu dans les messages de navigation. De ce fait, la précision n'est pas dépendante du nombre d'utilisateurs et le système ne peut pas être surchargé par le nombre d'usagers.

## 2.5 Le RTK

Le RTK (Real Time Kinematic) est une technique de géolocalisation basée sur le principe LBAS. C'est un système de type DGNSS, il faut donc au minimum 2 GPS pour que le RTK fonctionne, la base et le rover. La base communiquant des facteurs de correction au rover. La figure 6 montre le fonctionnement du RTK.

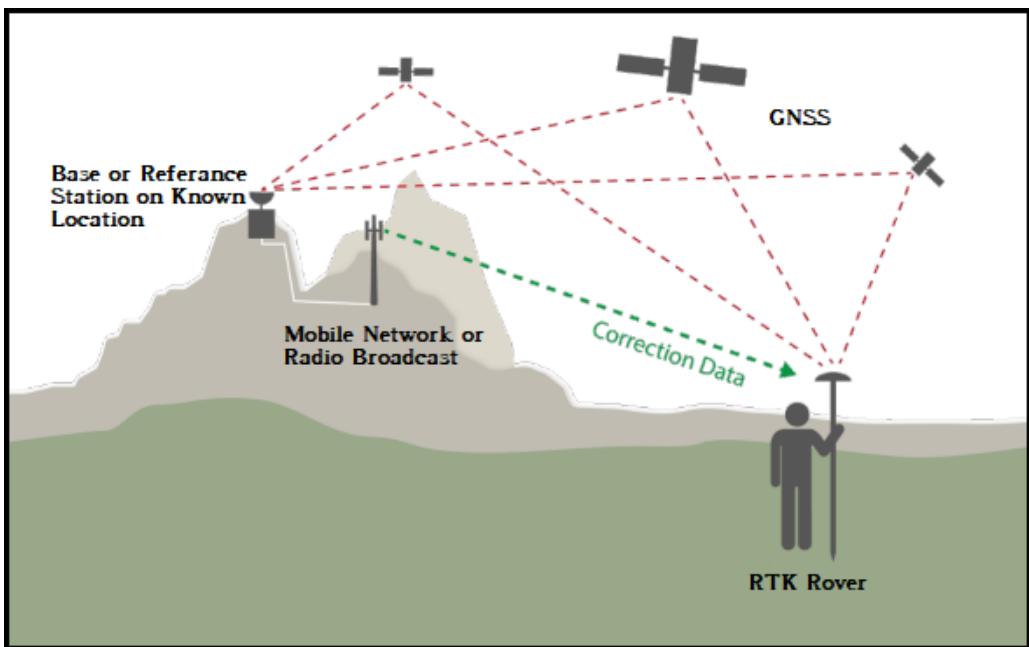


FIGURE 6 – Principe de fonctionnement du RTK, source : docs.datagnss.com [21]

Un des points importants avec l'utilisation du RTK, c'est qu'il a été étudié afin d'utiliser une base mise en place par l'utilisateur. Ainsi il n'est pas nécessaire de se trouver dans la couverture d'un réseau LBAS existant pour fonctionner, une couverture satellite avec n'importe quel constellation suffit. Si un système SBAS est disponible, les GPS RTK peuvent également en profiter. Le RTK est un moyen de gagner en précision mais n'est pas un système de positionnement indépendant des technologies GNSS.

Ce qui différencie le RTK des autres DGNSS, ce sont les signaux qu'il utilise. Alors que les GPS standard utilisent les messages de navigation des satellites afin de calculer leur position, les GPS RTK utilisent également la porteuse comme source d'information.

La porteuse n'était au départ que dédiée au transport des autres signaux qui eux contenaient des informations. Mais la porteuse est un signal sinusoïdal et le GPS RTK utilise cette nouvelle information pour accroître sa précision. Les fréquences des porteuses sont connues, et chaque fréquence a donc une longueur d'onde, elle est de 19,03 cm pour L1, 24,40 cm pour L2 et 25,40 cm pour L3.

Il ne faut pas oublier qu'un récepteur GPS cherche à définir de manière la plus précise possible la distance qui le sépare d'un satellite, et qu'il utilise le temps que met le signal pour lui parvenir depuis le satellite pour trouver cette distance. Un GPS standard utilise les flancs des messages de navigation comme repère temporel, mais les temps de montée et de descente de signaux carré sont imprécis. C'est pourquoi le RTK utilise la porteuse comme repère temporel. La figure 7 montre les différences entre les mesures d'un GPS sans, puis avec RTK.

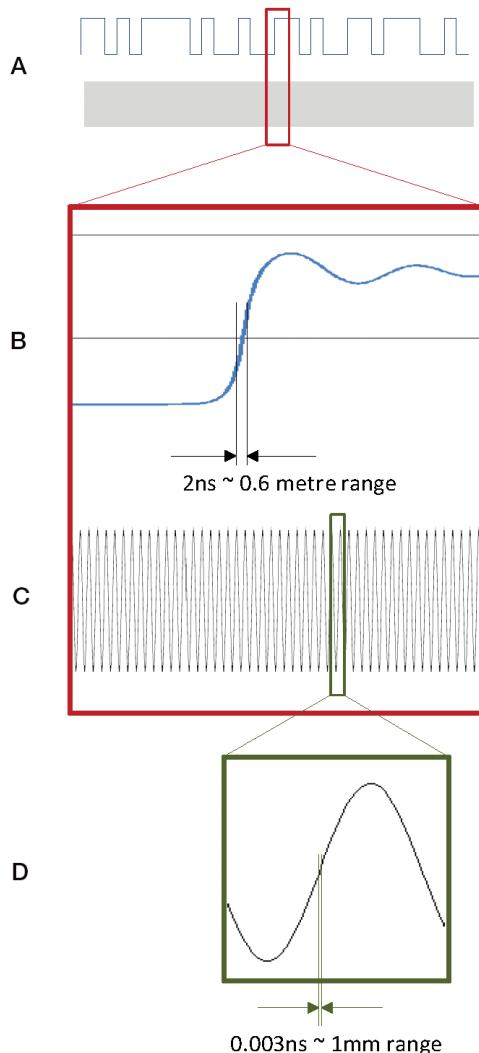


FIGURE 7 – Comparaison de la précision des signaux GPS sans, puis avec RTK,  
source : novatel.com [22]

Dans la figure 7, le signal A représente un message de navigation à 50 Hz et une porteuse proche de 1,5 GHz en dessous. Le signal B montre l'incertitude qui se produit lors de la mesure des flancs du message de navigation. L'incertitude est de 2 ns et produit une erreur de 60 cm. Le signal D nous montre que l'incertitude de mesure sur la phase est de 3 ps, ce qui correspond à environ 1 mm.

Cette image fait abstraction de toutes autres formes de perturbation ou d'incertitudes, mais elle permet de comprendre la différence de fonctionnement entre un GPS avec sans RTK. Dans les faits, la préécision du GPS sans RTK varie de 7.6 m à 1 m comme décrit dans le chapitre 2.3. La technologie RTK permet quand à elle d'optenir une précision de 2 cm. [23]

Utiliser la phase de la porteuse pour connaitre sa positon permet donc un gain important en précision. Mais pour que cette mesure fonctionne, il faut pouvoir lever

l'ambiguïté de mesure de phase initiale. La figure 8 illustre le problème.

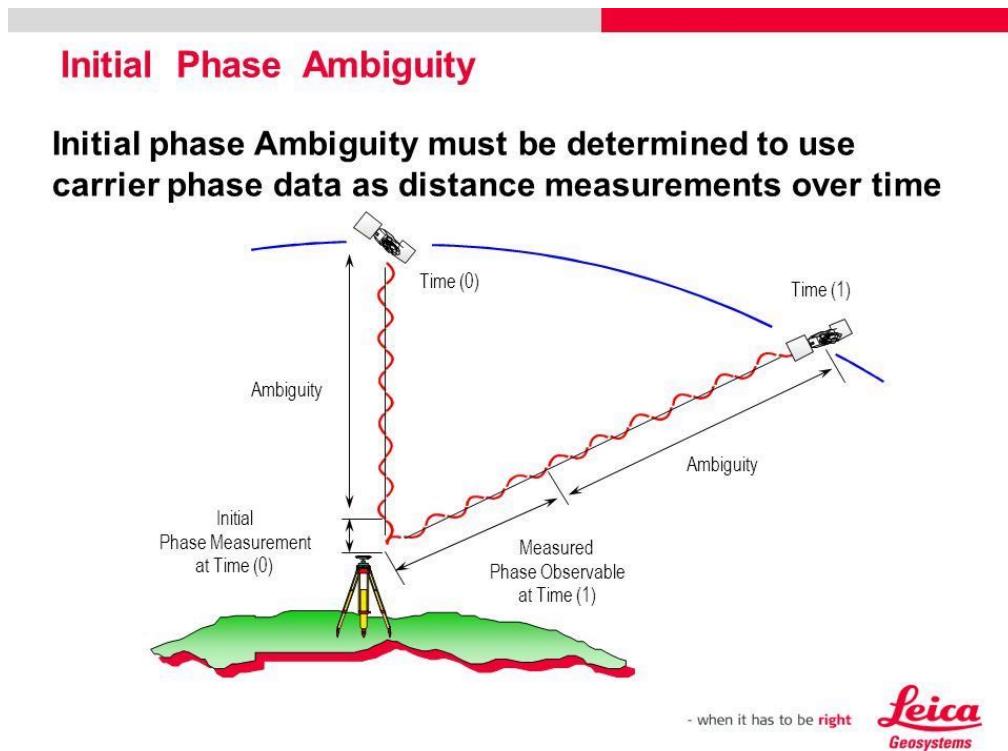


FIGURE 8 – Illustration de l'ambiguïté de mesure de phase, source : slideplayer.com [24]

Le récepteur GPS RTK se base sur la phase pour établir la distance entre lui et le satellite. La phase de la porteuse peut-être découpée en deux parties, un certain nombres d'oscillations complètes et une dernière oscillation partielle. Dès qu'il reçoit une porteuse, il en mesure la phase et obtient donc le déphasage qu'il y a entre lui et le satellite. Comme le GPS connaît la longueur d'onde de la porteuse, il peut déterminer la distance qu'a parcourue l'oscillation partielle. Sur la figure 7 par exemple, l'oscillation partielle de la porteuse au temps 0 semble être une demi-oscillation, ce qui correspond à une demi-longueur d'onde, soit 9,5 cm pour L1. Mais le GPS ignore le nombre d'oscillations entières qui se sont produites depuis l'émission du signal, il ne peut donc pas définir la distance qui le sépare du satellite.

Ce problème de nombre de période entière inconnu s'appelle ambiguïté de phase. Lever l'ambiguïté n'est pas une action immédiate, elle demande d'observer un maximum de satellites pendant un certain temps. Comme les satellites GNSS orbitent autour de la terre, la distance entre l'utilisateur et chaque satellite varie constamment, ce qui permet d'obtenir continuellement de nouvelles informations. Par la suite, l'utilisation d'algorithmes et de statistiques permettent, avec les facteurs de corrections de la base, de lever l'ambiguïté. Une fois l'ambiguïté levée, le système fonctionne avec sa précision maximum. Il est possible que le GPS perde la mesure de phase pour divers raisons, si cela arrive, il devra essayer de lever l'ambiguïté à

nouveau en utilisant le même procédé.

On parle généralement de float RTK lorsque l'ambiguïté n'est pas levée et de fix RTK lorsque l'ambiguïté est levée.

## 2.6 Les types d'erreurs GPS

Les erreurs GPS peuvent être diverses, certaines ont déjà été détaillées dans les chapitres précédents. Ce chapitre a pour but de lister les principales erreurs et s'il est possible de les corriger.

La première perturbation est une perturbation volontaire. Il s'agit du chiffrement SA que les Etats-Unis ont utilisé, limitant la précision du GPS Américain à environ 100 m pour l'usage civil. Des technologies tels que le SBAS ou le LBAS permettent de réduire fortement l'erreur et de la ramener à environ 15 m, jusqu'à ce que le SA soit désactivé en 2000.

Une autre forme de perturbation est la perturbation que les signaux subissent en entrant dans l'atmosphère, principalement dans la ionosphère et la troposphère. Ces sources d'erreurs peuvent être partiellement voir totalement corrigée grâce aux bases terrestre et aux système SBAS ou LBAS. L'utilisation de GPS bi-fréquence ou tri-fréquence permet également de corriger ces effets. [4, 10]

Les erreurs d'orbites des satellites causent également des imprécisions. [4]

Les horloges atomiques des satellites ont elles aussi quelques imprécisions. Ces problèmes peuvent être résolus si au moins 4 satellites sont disponibles, 3 pour la position et un pour fournir la base de temps. [10]

Les erreurs de multi-trajet sont également sources de problèmes, principalement en ville où le signal rebondit sur les bâtiments avant d'arriver au récepteur. [25]

### 3 Présentation des modules u-blox

Cette section va détailler la mise en place d'un GPS avec RTK. Cette démonstration sera faite avec des modules de développement de u-blox. Ces cartes sont des C94-M8P-3-11. Les deux cartes sont identiques et sont construites autour du NEO-M8P. Elles embarquent tout le nécessaire pour utiliser le RTK, à condition d'en posséder 2, une utilisée comme base, et l'autre comme rover. La figure 9 est une photo du module.



FIGURE 9 – Photo du module GPS avec RTK

Ce module dispose de beaucoup de connecteurs, mais ils ne seront pas tous utiles pour notre application. Les connecteurs utiles sont l'USB et les deux connecteurs pour les antennes. L'USB sert à programmer et alimenter le module ainsi que pour suivre en directe différents paramètres de la carte grâce à l'application u-center fournis par u-blox. Le connecteur UHF permet d'y fixer une antenne bâton pour que les modules puissent communiquer ensemble et ainsi que la base puisse envoyer les facteurs de corrections au rover pour obtenir un GPS avec RTK. Le connecteur GNSS accueille quant à lui une antenne GPS déportée.

U-center est le logiciel disponible sur le site de u-blox et qui permet de configurer les modules. Pour ce connecter à un module, il suffit de le brancher en USB à l'ordinateur puis de le choisir dans la liste des ports COM. L'interface de U-center permet d'afficher beaucoup d'informations. La figure 10 montre les plus importantes.

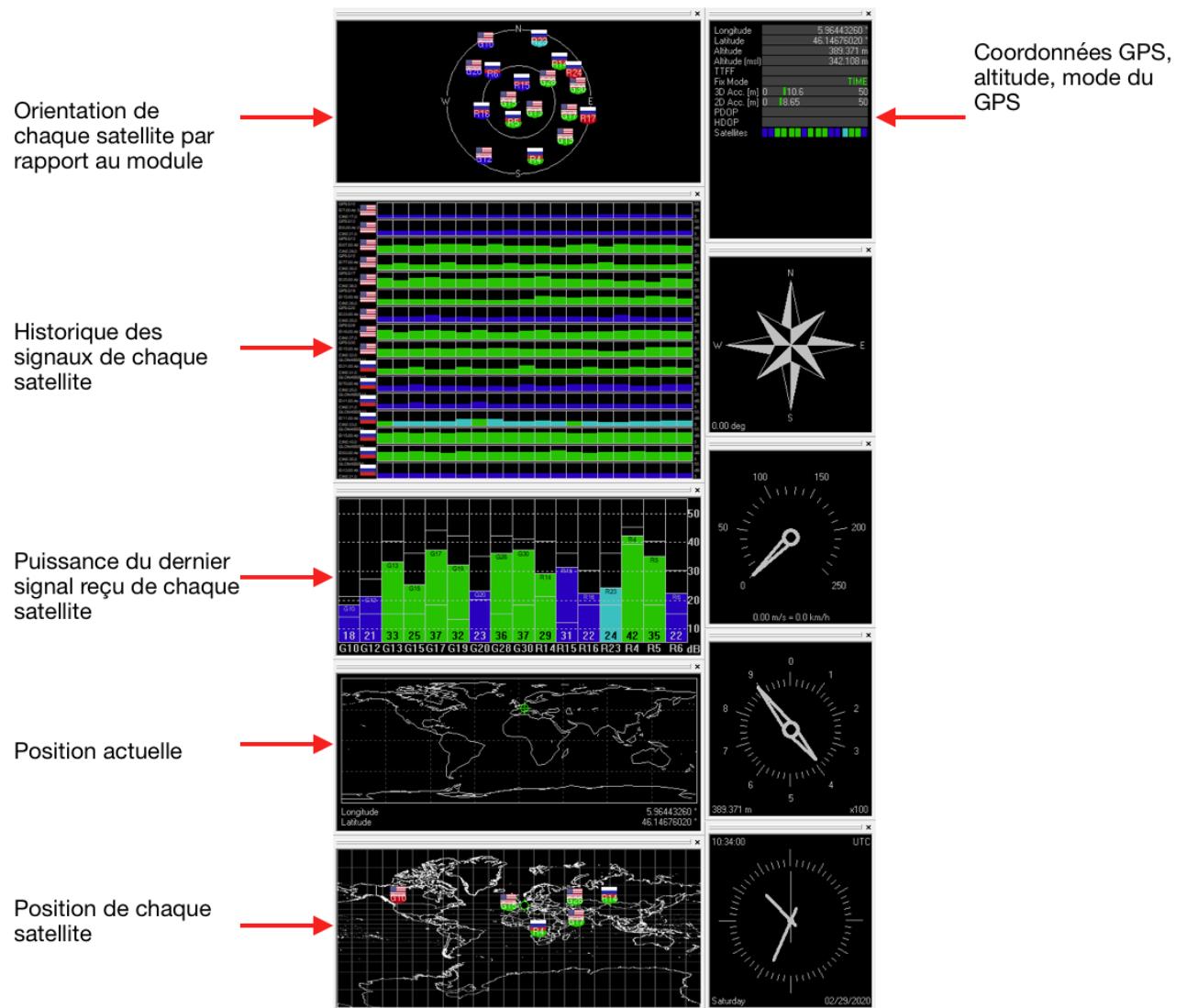


FIGURE 10 – Interfaces de u-center

Parmi tous ces outils, les plus importants sont : l'historique des signaux, la puissances des signaux et le mode du GPS. Les signaux verts et bleus clairs sont les signaux exploitables pour la localisation GPS, les bleus foncés sont reçus mais non exploitables.

### 3.1 Recommandations d'utilisation

Cette section donne quelques informations à propos de l'utilisation des NEO-M8P car comme cette puce exploite le RTK, un certain nombre de choses peuvent changer par rapport aux autres technologies GNSS.

La première chose à connaître c'est le TTFF (Time to first fix). Il s'agit temps que va mettre le GPS à obtenir sa première position à partir du moment où il est allumé. Bien que ce temps varie en fonction du temps pendant lequel le GPS n'a pas été allumé, j'ai constaté lors de son utilisation qu'il trouvait sa première position au bout de 3 à 10 secondes environ, soit un temps plutôt normal pour un GPS. La datasheet indique que ce temps peut aller jusqu'à 29 secondes [26].

La seconde valeur intéressante, c'est le temps de convergence du GPS. Ce temps correspond au temps nécessaire au GPS pour trouver sa position avec sa résolution maximum. Dans le cas de la base, j'ai souvent utilisé une précision de 1 mètres et cette précision est atteinte en 15 à 20 minutes. Il arrive parfois que la base n'arrive pas à se fixer avec une telle précision. Dans ce cas, il faut revoir à la baisse la précision souhaitée, ce qui revient à fixer la valeur de la précision attendue à 1,5 ou 2 mètres.

Concernant le rover, le temps de convergence est très variable. J'ai constaté des temps de convergence compris entre quelques minutes et plus d'une heure. Ce temps est le temps nécessaire pour lever l'ambiguïté de mesure de phase initiale. Pendant ce temps, le rover se trouve généralement en float, même s'il arrive qu'il passe parfois en fix avant de retourner en float une ou deux minutes plus tard. Ce temps d'attente est peu inhabituel pour l'utilisation d'un GPS et il faut en être conscient. Ce temps de convergence implique de devoir fixer la base puis allumer le rover bien avant d'en avoir réellement besoin. J'ai également observé que parfois, lorsque le rover met plus de 45 minutes pour se fixer, une solution peut être de vider la base de données depuis u-center, ce qui permet de relancer la recherche et permet parfois de fixer le rover peu de temps après.

Le dernier point important concernant ce module est la distance entre les émetteurs radio. Bien qu'une distance trop grande entre eux les empêche de s'envoyer les facteurs de correction RTK, une distance trop petite cause une saturation du signal dans le module radio du rover et l'empêche de fonctionner correctement. Afin de se prémunir de ce problème, je recommande une distance minimum de 2 mètres entre les deux émetteurs radio. Il est fort probable que la communication radio fonctionne à plus petite distance, mais si le rover est utilisé dans un espace suffisamment dégagé, positionner la base à au moins 2 mètres ne devrait pas poser de problème.

### 3.2 Mise en place d'un GPS avec algorithme RTK depuis u-center

Cette section montre comment configurer un module en base ou en rover depuis l'application u-center. Comme cette configuration a besoin de signaux GPS, il est préférable d'être en extérieur avec un espace bien dégagé autour du module pour avoir la meilleure réception possible.

Avant de commencer à configurer le module, il faut télécharger puis installer U-center disponible sur le site de u-blox. Il faut également monter les antennes GPS

et radio sur le module. Une fois l'application ouverte, on peut brancher le module à l'ordinateur avec le câble USB puis se connecter dessus. Automatiquement le module doit commencer à recevoir des informations de satellites comme sur la figure 10.

### 3.2.1 Configuration de la base avec u-center

Pour configurer les module, il faut utiliser l'outil "message view" qui se trouve dans le menu "view". La figure 11 montre à quoi ressemble l'outil de configuration.

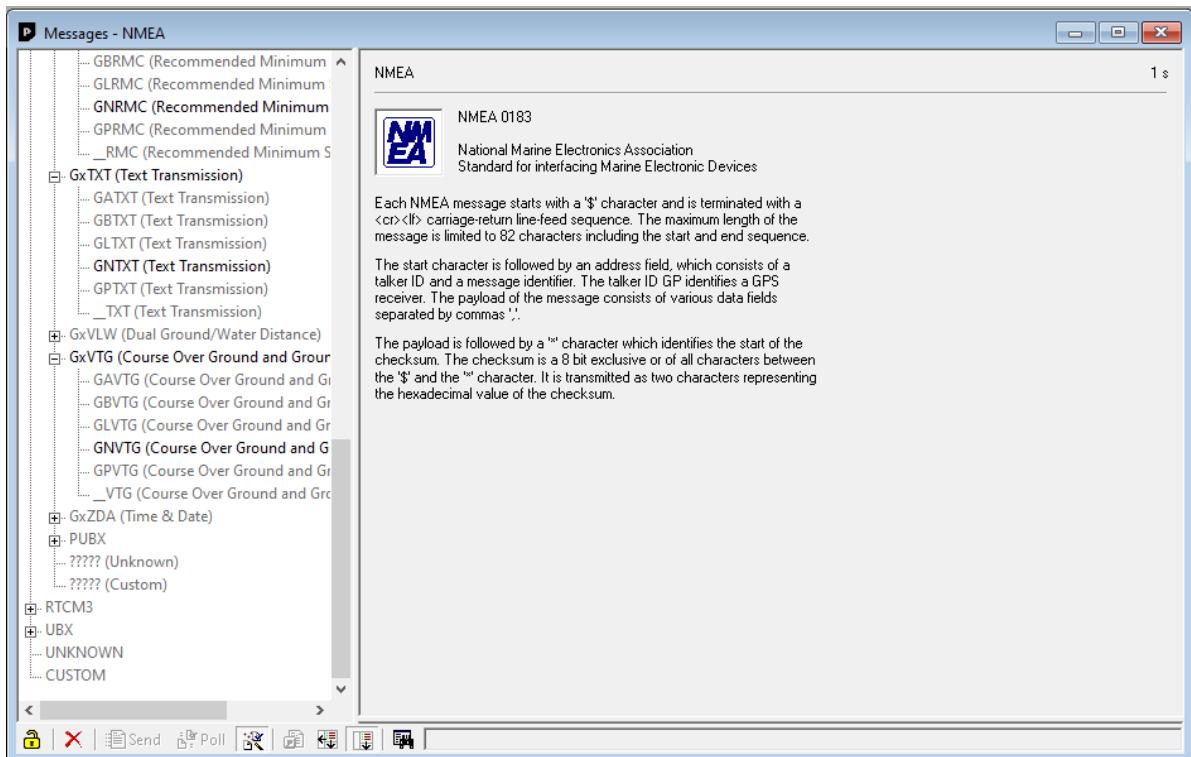


FIGURE 11 – Interfaces de message view

La totalité de la configuration va se dérouler dans la partie "UBX" que l'on voit sur la figure 11. UBX est le protocole de communication utilisé pour configurer les chips u-blox. Les boutons "send" et "poll" en bas de la fenêtre permettent d'envoyer une configuration ou de lire la configuration actuelle du module.

Pour commencer la configuration de la base, il faut se rendre dans le menu : UBX -> CFG -> TMODE3. Ce menu permet de sélectionner la méthode avec laquelle la base va trouver sa position. Le mode 1 lui permet de trouver sa position toute seule en analysant sa position grâce aux satellites alors que le mode 2 permet d'entrer directement les coordonnées du lieu où se trouve la base.

Pour cette démonstration, je choisis le mode 1 et je lui laisse un temps d'observation de 5 minutes pour essayer d'atteindre une précision de 10 mètres. Plus longtemps

la base reste en état d'observation et plus sa position doit se préciser. La figure 12 montre cette configuration.

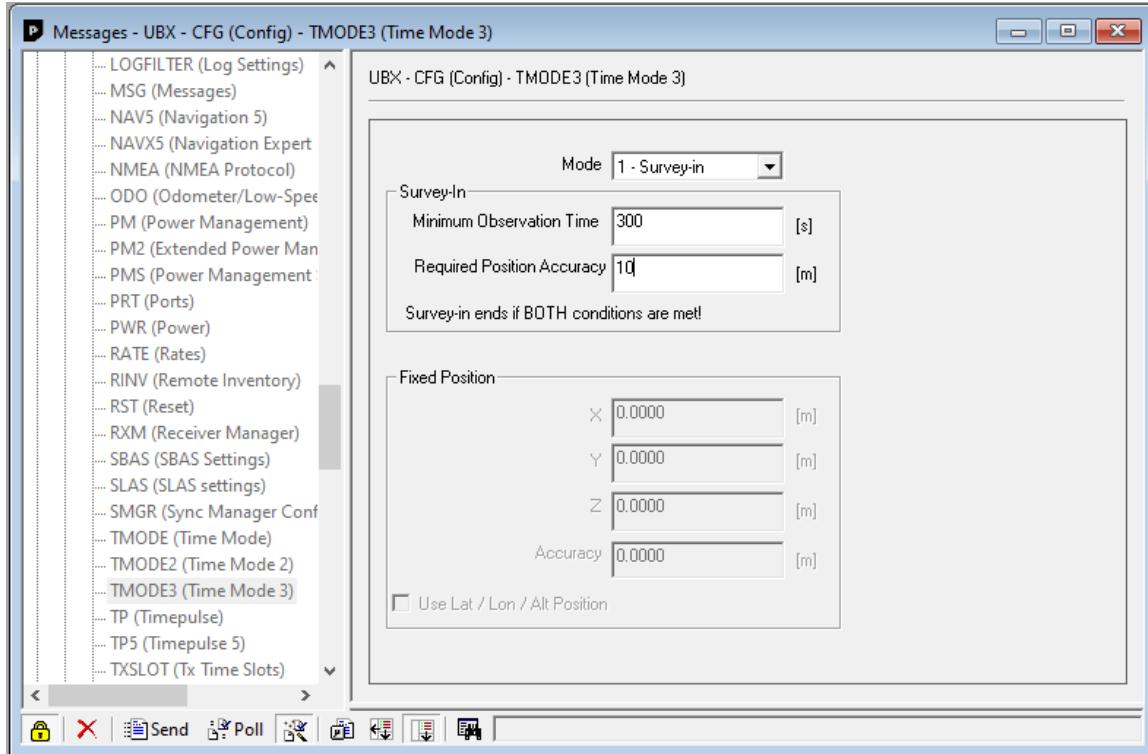


FIGURE 12 – Configuration du point de référence de la base

Si la configuration de la base a pour but de faire des mesures par la suite avec le rover, il est recommandé de fixer la base plus précisément, comme avec les valeurs indiquées dans le chapitre 3.1.

Une fois que la configuration est faite, ne pas oublier de presser le bouton "send" afin d'envoyer la nouvelle configuration au module.

Pendant que la base affine sa position, il est possible de voir sa précision actuelle ainsi que le temps écoulé depuis le début de l'opération. Cette fonction se trouve dans le menu : UBX -> NAV -> SVIN. La figure 13 montre les informations disponibles.

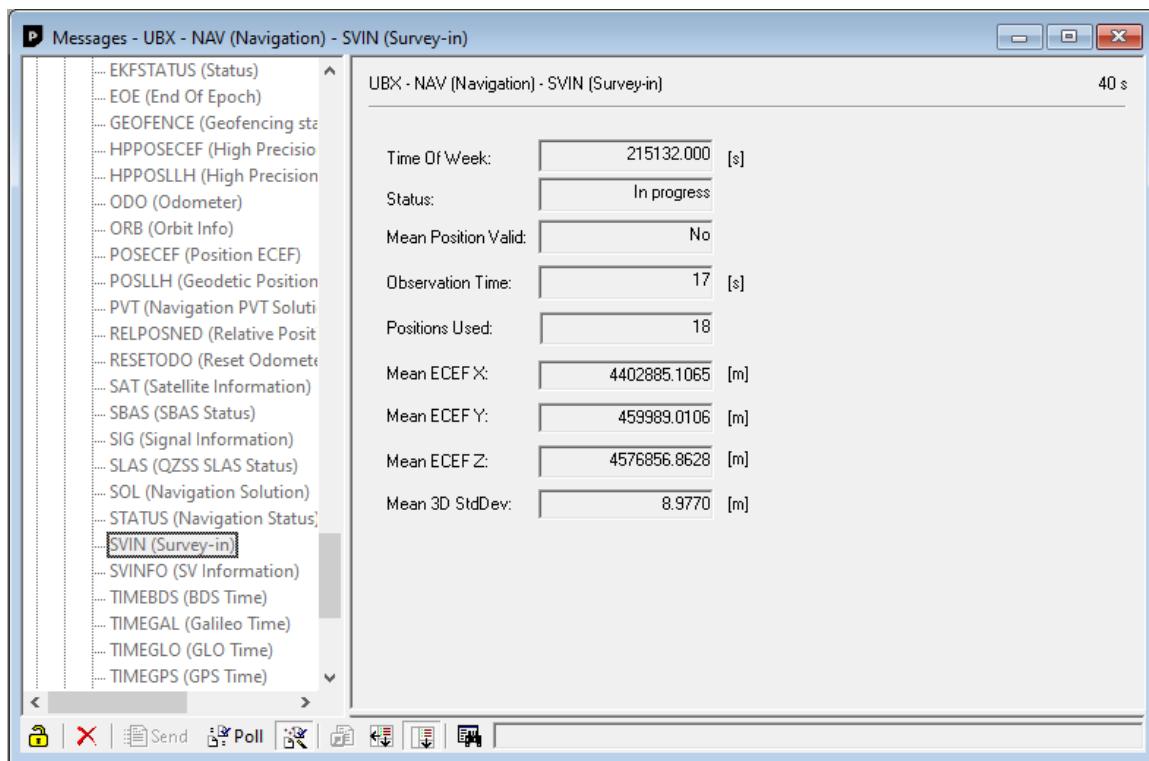


FIGURE 13 – Informations disponibles pendant que la base affine sa position

La première ligne à regarder est la ligne "observation time", cette valeur doit s'incrémenter chaque seconde. Si cette valeur reste fixe, c'est que les messages UBX ne sont pas activés. Il est impératif de les activer afin que la configuration se fasse correctement. Pour les activer, il faut remonter la liste des menus et retourner sur le menu principal "UBX" puis faire un clic droit et sélectionner "Enable Child Messages" comme le montre la figure 14.

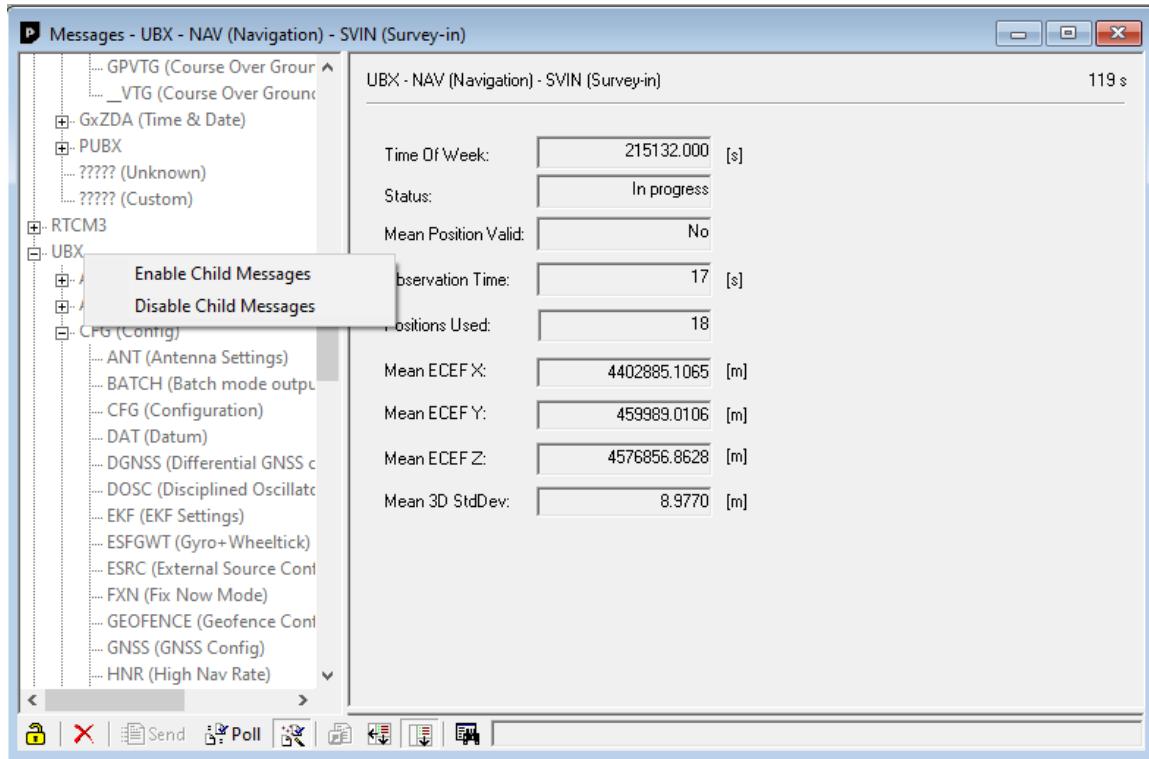


FIGURE 14 – Activation des messages UBX

Maintenant que les messages sont activés, la case "Observation Time" présente dans le menu "SVIN" s'incrémentera chaque seconde. Une fois que le temps d'observation est dépassé et que la précision indiquée dans la case "Mean 3D StdDev" est meilleure que la précision demandée, la case "Status" doit indiquer "Successfully finished". Un autre moyen de voir que la base ne cherche plus sa précision est de regarder le mode du GPS, il doit indiquer "TIME" comme sur la figure 10.

Une fois que la base a trouvé sa position GPS, il faut paramétriser le port de communication avec lequel elle va transmettre les facteurs de correction au rover pour l'utilisation du RTK. La configuration du port se fait dans le menu : UBX -> CFG -> PRT. La figure 15 montre la configuration du port.

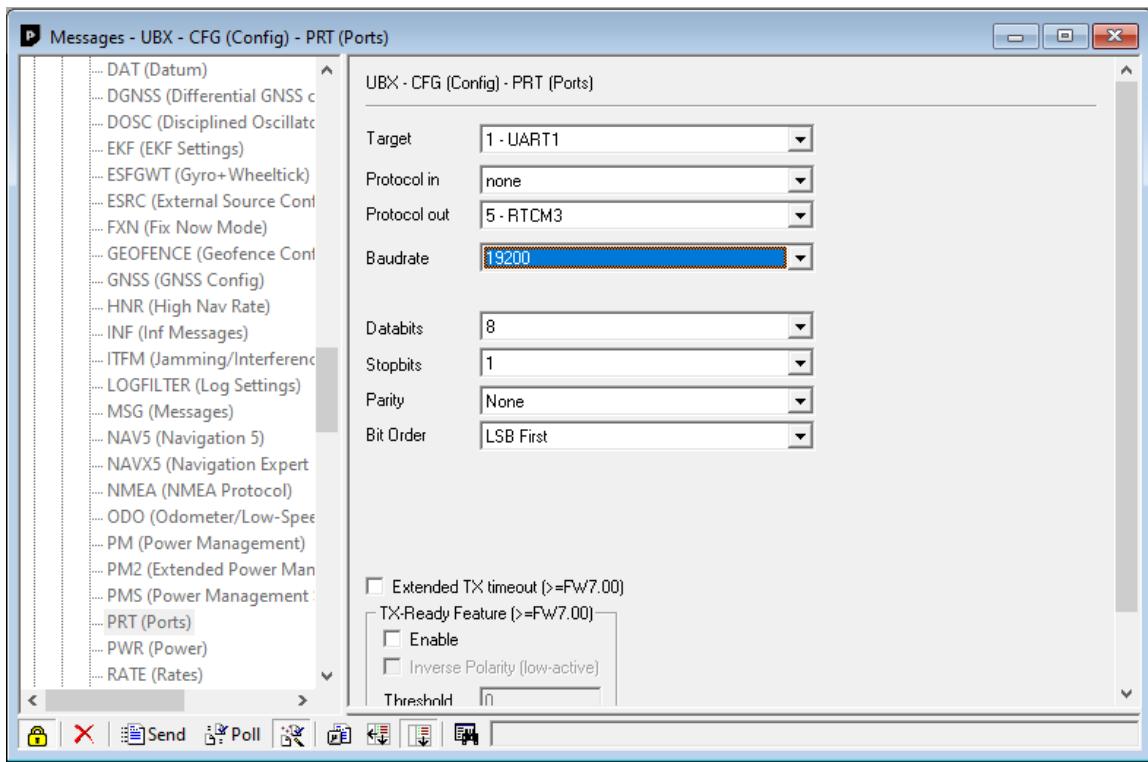


FIGURE 15 – Configuration du port radio de la base

L'envoi des facteurs de correction se fait depuis le port UART1 jusqu'à l'émetteur radio. La base ne reçoit rien par la radio, mais elle envoie des messages RTCM3 avec un baudrate de 19200. Une fois que le port UART1 est configuré, ne pas oublier d'envoyer la configuration avec le bouton "send".

Le protocole RTCM est un protocole de communication spécifiquement utilisé pour transmettre des facteurs de correction pour corriger le positionnement des utilisateurs de service DGNSS.

Le protocole RTCM contient beaucoup de messages différents mais dans notre cas, nous allons en utiliser 4. Les messages portent chacun un numéro distinct ce qui permet de les identifier rapidement. Les 4 messages utiles sont les suivants :

- 1005 : Stationary RTK Reference Station ARP (antenna reference point)
- 1077 : GPS MSM7
- 1087 : GLONASS MSM7
- 1230 : GLONASS L1 and L2 Code-Phase Biases

Ces messages sont à configurer dans le menu : UBX -> CFG -> MSG. Il faut ensuite sélectionner le premier message, puis cocher la case UART1 comme sur la figure 16.

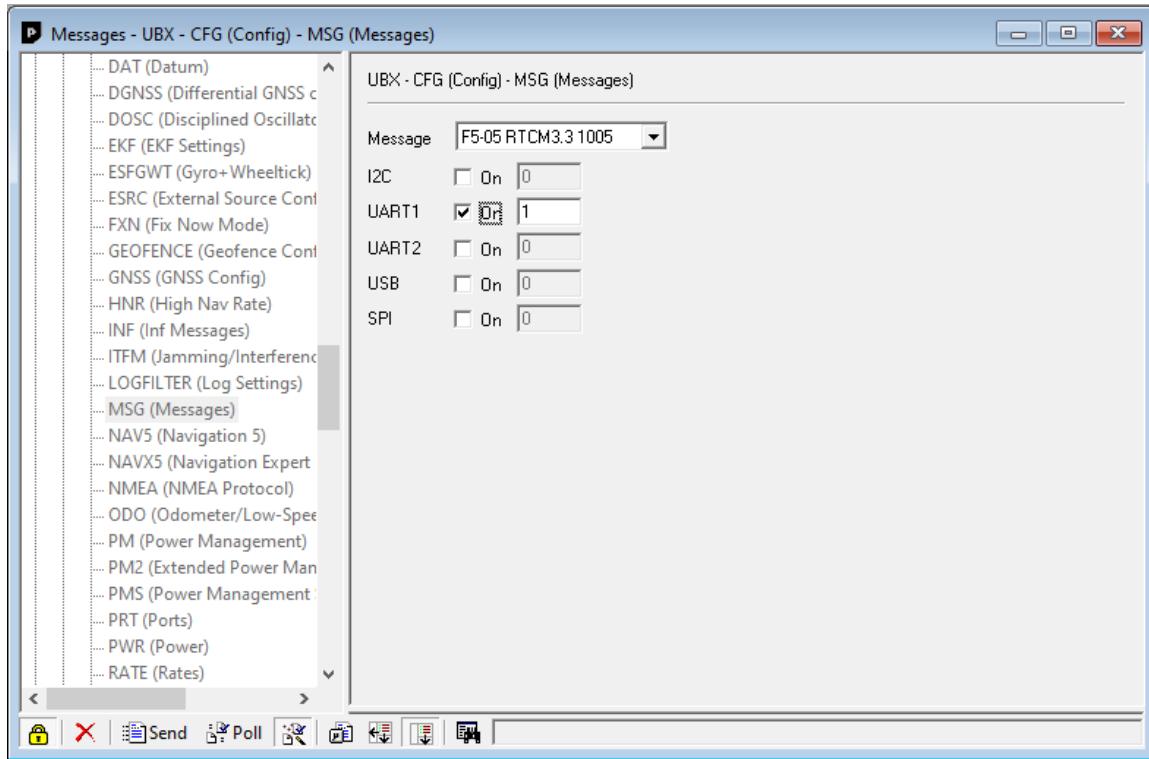


FIGURE 16 – Configuration des messages RTCM de la base

Il faut presser le bouton "send" après chaque message configuré. Pour le message 1230, on peut changer l'intervalle de temps entre deux messages en changeant la valeur en face de UART1. En passant cette valeur de 1 à 10 cela permet d'actualiser les facteurs transmis par les messages 1230 toute les 10 secondes, ce qui est largement suffisant.

La base est maintenant configurée.

### 3.2.2 Configuration du rover avec u-center

Pour configurer le second module comme rover, il faut aller dans le menu : UBX -> CFG -> PRT et configurer le port UART1 pour qu'il puisse recevoir les messages envoyés par la base, comme le montre la configuration à la figure 17

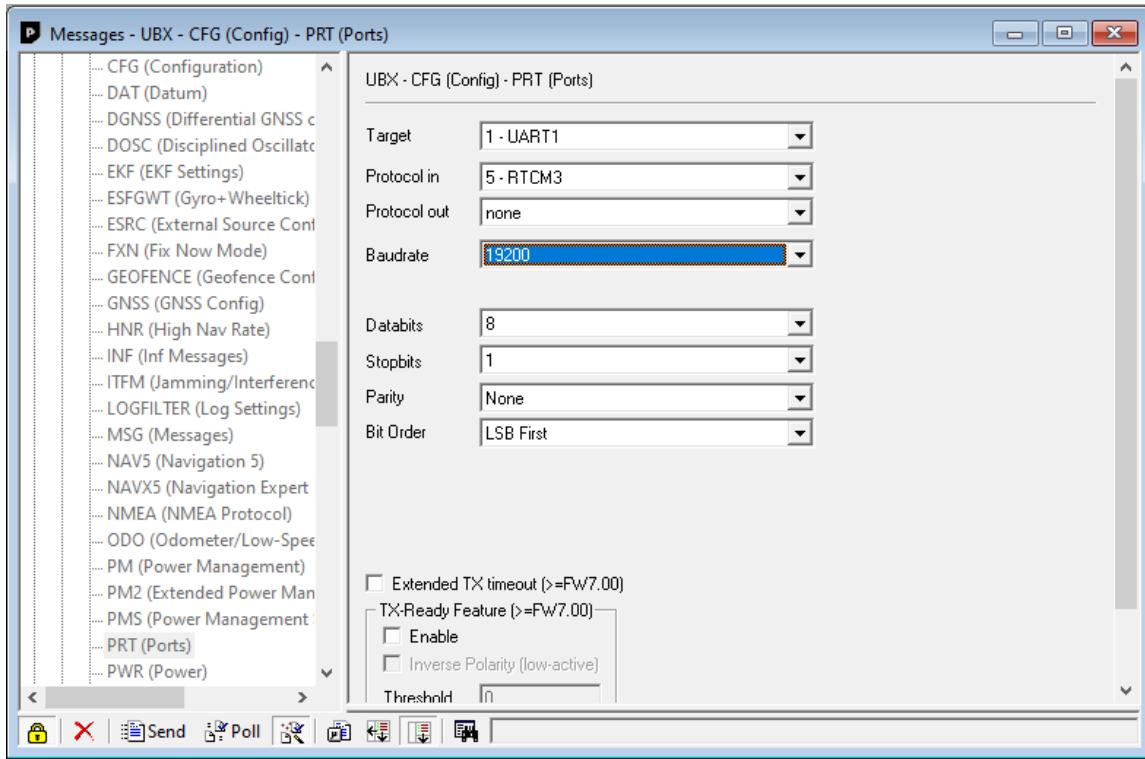


FIGURE 17 – Configuration du port radio du rover

La configuration du port UART1 doit permettre de recevoir les messages RTCM de la base et ne renvoie rien à la base. La communication s'effectue toujours avec un baudrate de 19200.

Dès que cette opération est faite, le rover va essayer de préciser sa position grâce aux facteurs envoyés depuis la base. Le rover pourra se trouver dans différents modes selon la qualité des messages radio reçus.

- 3D/DGNSS : Le mode RTK n'est pas utilisé car la réception radio ou GPS est trop mauvaise. Dans ce cas, la précision de positionnement sera au mieux de 2,5 m [26]
- 3D/DGNSS/FLOAT : Le mode RTK est utilisé, mais l'algorithme utilisé pour le RTK ne parvient pas à situer précisément le rover. Dans ce cas, la précision ne peut pas être garantie. Elle est probablement inférieur a 2,5 m mais il faut attendre que le GPS lève l'incertitude pour savoir s'il est possible d'avoir une précision de 2,5 cm ou s'il faut se contenter de 2,5 m.
- 3D/DGNSS/FIX : Le mode RTK est utilisé et fonctionne correctement. La position obtenue a une précision de 2,5 cm. [26]

La configuration du rover est terminée, si les conditions lui permettent, il devrait être capable d'indiquer sa position avec une précision de 2,5 cm.

### 3.3 Mise en place d'un GPS avec algorithme RTK depuis un microcontrôleur

Pour mettre en place une configuration rover sur le PCB développé dans le cadre de ce travail de bachelor, il est impossible de disposer du logiciel U-center à chaque démarrage. C'est le microcontrôleur qui va devoir initialiser le GPS. Pour ceci, il faut utiliser le protocole UBX, U-center n'étant qu'une interface qui permet d'utiliser le protocole UBX et d'afficher les données. La structure d'un message UBX est affichée à la figure 18

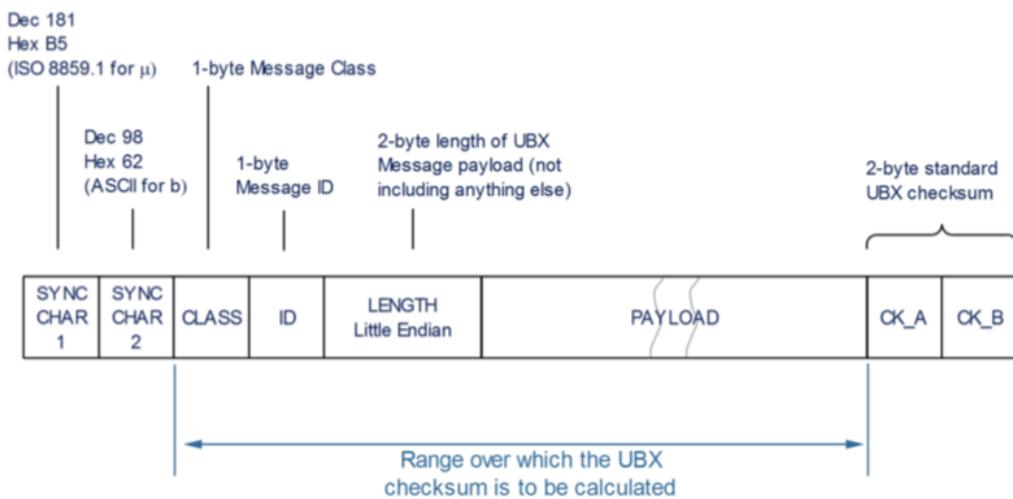


FIGURE 18 – Structure d'un message UBX, source : u-blox.com [27]

La structure des messages permet de reconnaître la forme générale et de comprendre rapidement quelle est la fonction globale de chaque message ainsi que de savoir où commence un nouveau message. Il n'est pourtant pas recommandé d'écrire les messages UBX sois-même.

#### 3.3.1 Trouver les commandes UBX

La méthode recommandée par u-blox pour envoyer ses propres messages UBX et configurer un chip est de les récupérer directement dans l'application U-center. U-center permet d'afficher chaque commande UBX comme elle sera envoyée dès que le bouton "send" en bas de la fenêtre sera pressé. La figure 19 montre comment l'afficher et comment il est représenté.

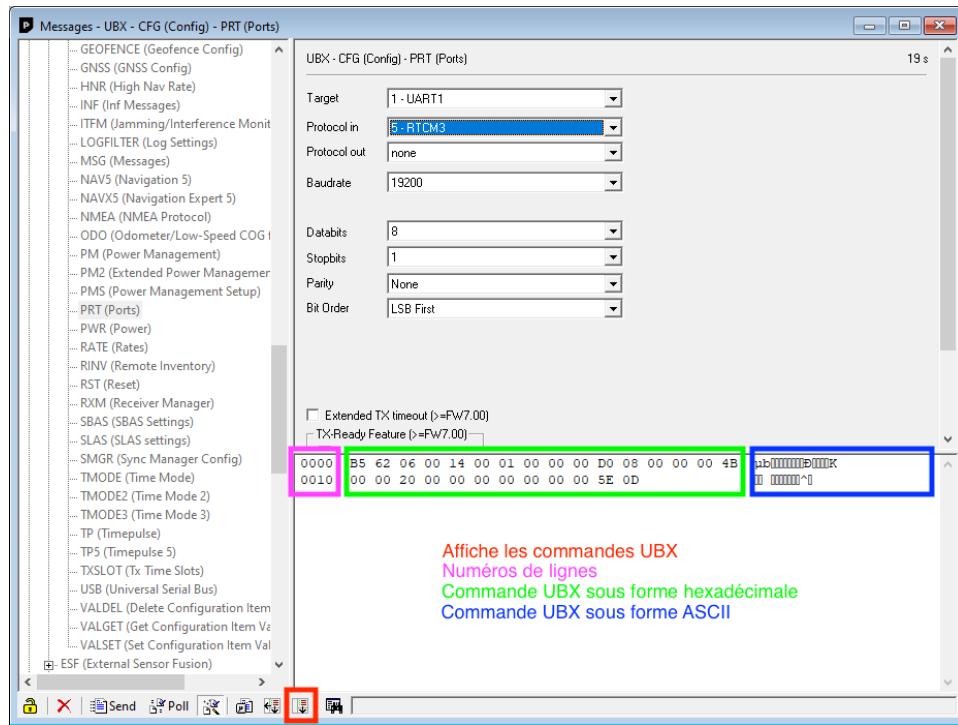


FIGURE 19 – Affichage des commandes UBX dans U-center

L'exemple de la figure 19 montre le message pour configurer le port UART1 du rover. Cette étape étant la seule nécessaire à la configuration du rover, l'envoi d'une seule commande suffit pour configurer entièrement le rover.

D'autres commandes peuvent être nécessaires comme la commande pour sauver une configuration dans le rover. Ainsi la configuration du rover sera automatiquement prête au démarrage du rover. Il faudra du coup prévoir les commandes pour remettre le chip en configuration d'usine si jamais un problème survient, commande que l'on pourrait déclencher lors du reset du microcontrôleur.

La liste ci-dessous contient les messages utiles pour la réalisation du GPS. Toutes les commandes ont été obtenues à partir du logiciel U-center :

Configurer un module en rover :

B5 62 06 00 14 00 01 00 00 00 D0 08 00 00 00 4B 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 5E 0D

Sauvegarder une configuration :

B5 62 06 09 0D 00 00 00 00 FF FF 00 00 00 00 00 00 17 31 BF

Charger les paramètres d'usine :

B5 62 06 09 0D 00 FF FF 00 00 00 00 00 FF FF 00 00 17 2F AE

### 3.3.2 Vérifications des commandes UBX

Afin de vérifier les commandes UBX et de s'assurer qu'elles fonctionnent, on peut les tester avec un autre logiciel. Le but étant d'essayer d'envoyer la commande et de vérifier si le comportement que l'on obtient est celui que l'on attendait. Pour faire ce contrôle, on peut utiliser n'importe quel logiciel capable d'envoyer des caractères vers un port COM précis. Je vais utiliser Coolterm pour faire ces vérifications.

Dans cet exemple, la base est sur le COM3 et le rover sur le COM4. La base est complètement configurée alors que le rover pas. La capture 20 montre que je suis connecté au rover et qu'il est en mode GPS non RTK car il affiche uniquement le mode 3D qui correspond à la configuration d'usine du module. Il ne communique donc pas avec la base.

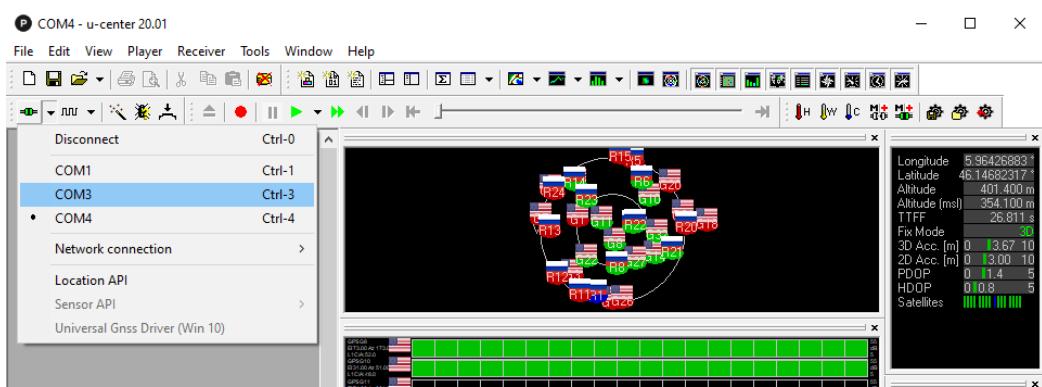


FIGURE 20 – Module avec configuration d'usine

La suite de cet exemple va permettre de s'assurer que la commande UBX du chapitre 3.3.1 configure bien le module en rover. On va donc se connecter sur un autre port COM ou se déconnecter complètement des ports COM sur U-center, le but étant de libérer le port COM4. On peut maintenant ouvrir l'application Coolterm et cliquer sur Options. La figure 21 montre les paramètres à utiliser pour se connecter sur un module.

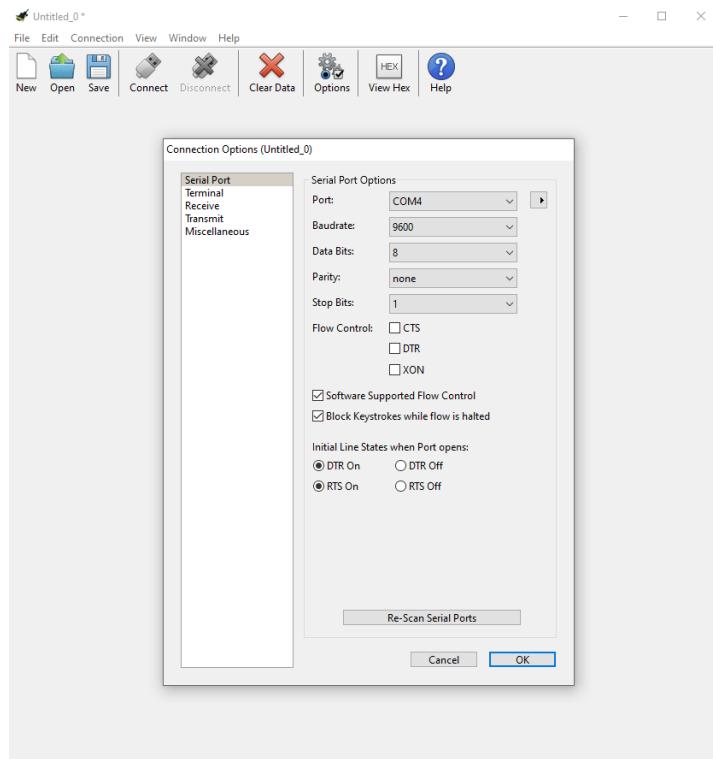


FIGURE 21 – Paramétrage du port COM sur Coolterm

Les paramètres importants étant de sélectionner le bon port COM, un baudrate de 9600, 8 bits de données, pas de parité et un bit de stop. On valide les paramètres puis on clique sur le bouton Connect dans la barre des menus. Les informations reçues par le module doivent défiler dans la fenêtre. En allant dans le menu Connections puis en ouvrant l'outil Send String, il est possible d'envoyer un message au rover. Le message à la figure 22 permet de configurer le module en rover RTK. Ne pas oublier de sélectionner le format du message qui est hexadécimale.

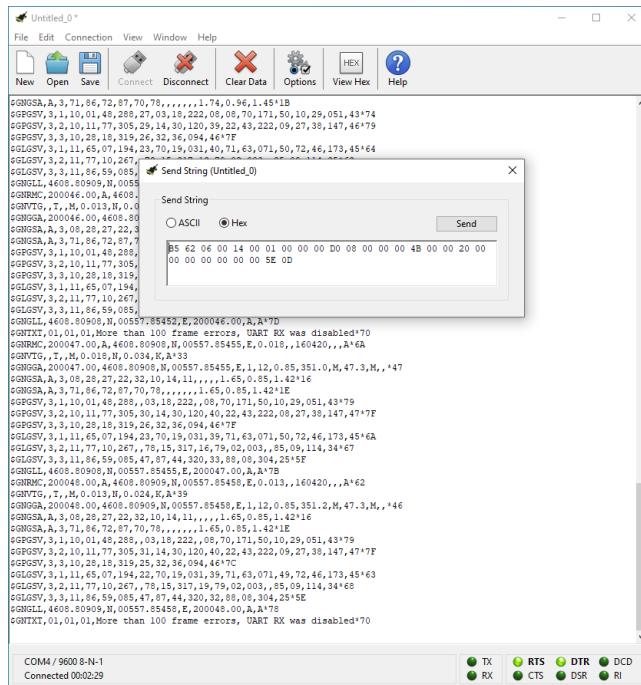


FIGURE 22 – Envoi d'un message pour configurer un module en rover RTK

Après avoir envoyé le message, on peut se déconnecter du rover sur Coolterm et se reconnecter sur le rover sur U-center. U-center indique maintenant que le rover est en mode 3D/DGNSS/FLOAT comme sur la figure 23, il est donc correctement configuré en mode RTK.

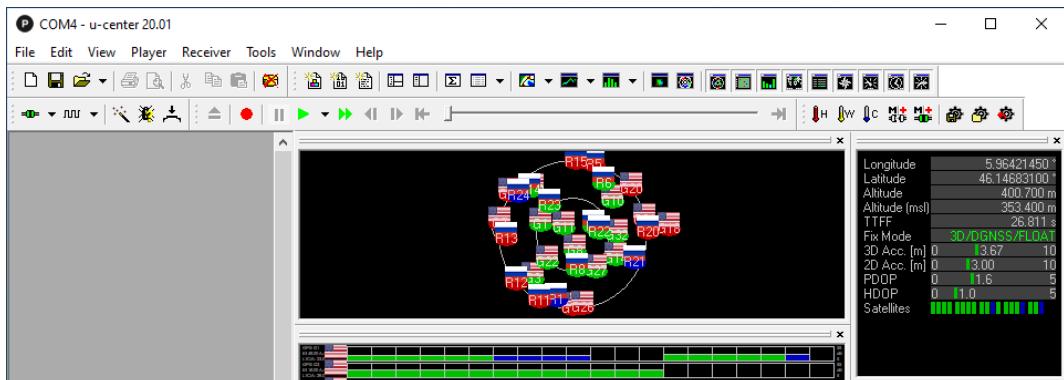


FIGURE 23 – Rover configuré grâce à une commande UBX

## 4 Mesures

Cette section contient des mesures effectuées avec les modules de développement de u-blox, les C94-M8P-3. A cause du confinement français imposé pour contenir le COVID-19, une première série de mesures a du être réalisée dans mon jardin, où les obstacles aux signaux GPS sont nombreux (arbres et maison) ainsi que la ligne à haute tension qui passe juste au dessus. Cette première série ne sera donc pas forcément la plus précise possible mais permettra d'avoir une idée des performances qu'il est possible d'avoir lorsque les conditions de réception GPS ne sont pas optimales.

Une deuxième série de mesures a ensuite été faite une fois que le déconfinement a été effectif afin de comparer la première et la seconde série. La seconde série de mesures s'est déroulée dans un espace bien plus dégagé que la première.

### 4.1 Récuperation et mise en forme des coordonnées GPS

#### 4.1.1 Récupérer les informations GPS depuis U-center

U-center permet d'enregistrer les données reçues par le rover. L'enregistrement peut être lancé à tout moment grâce au bouton contenant un rond rouge visible à la figure 23. Le bouton eject qui se trouve juste à côté permet de stopper l'enregistrement. Lors du lancement de l'enregistrement, U-center demande s'il doit enregistrer la configuration du rover dans le fichier de sauvegarde. Il est préférable de cocher non. Dans le cas contraire, des commandes UBX sous forme ascii apparaîtront entre les messages GPS contenant les coordonnées et il faudra les enlever à la main pour pouvoir récupérer les données GPS. L'enregistrement des données se fait dans un fichier texte au format UBX. Ce fichier peut être lu dans U-center pour revoir le déroulement des événements, ce qui peut être utile si une partie des mesures ne correspond pas à ce qui était attendu pour cause de perte de signal par exemple. L'interface de U-center affichant la puissance de chaque satellite à tout instant. La figure 24 montre ce que contient un fichier UBX lorsque l'on l'ouvre dans un éditeur de texte.

```

1 $GNRMC,120424.00,A,4608.81479,N,00557.85447,E,0.010,,070520,,,D*6F
2 $GNVTG,,T,,M,0.010,N,0.019,K,D*31
3 $GNGGA,120424.00,4608.81479,N,00557.85447,E,4,12,0.79,345.8,M,47.3,M,1.0,0000*6A
4 $GNGSA,A,3,02,21,25,26,29,31,18,14,12,,,1.45,0.79,1.21*14
5 $GNGSA,A,3,77,87,78,79,88,81,,,,,1.45,0.79,1.21*17
6 $GPGSV,4,1,13,02,14,039,43,04,04,335,,05,08,075,26,12,17,102,32*7D
7 $GPGSV,4,2,13,14,14,241,29,16,00,288,,18,34,168,45,21,35,175,43*75
8 $GPGSV,4,3,13,25,47,100,47,26,30,295,44,29,78,041,49,31,59,276,48*7E
9 $GPGSV,4,4,13,32,02,222,*4A
10 $GLGSV,2,1,08,70,05,329,,71,04,015,29,77,21,136,14,78,76,121,48*6B
11 $GLGSV,2,2,08,79,46,322,50,81,47,210,51,87,33,036,49,88,75,052,50*6D
12 $GNGLL,4608.81479,N,00557.85447,E,120424.00,A,D*77
13 $GNRMC,120425.00,A,4608.81479,N,00557.85447,E,0.019,,070520,,,D*67
14 $GNVTG,,T,,M,0.019,N,0.035,K,D*36
15 $GNGGA,120425.00,4608.81479,N,00557.85447,E,4,12,0.71,345.8,M,47.3,M,1.0,0000*63
16 $GNGSA,A,3,02,21,25,26,29,31,18,14,12,,,1.25,0.71,1.04*1D
17 $GNGSA,A,3,77,87,78,79,88,81,,,,,1.25,0.71,1.04*1E
18 $GPGSV,4,1,13,02,14,039,42,04,04,335,,05,08,075,25,12,17,102,31*7C
19 $GPGSV,4,2,13,14,14,241,29,16,00,288,,18,34,168,45,21,35,175,43*75
20 $GPGSV,4,3,13,25,47,100,47,26,30,295,43,29,78,041,49,31,59,276,48*79
21 $GPGSV,4,4,13,32,02,222,*4A
22 $GLGSV,2,1,08,70,05,329,,71,04,015,29,77,21,136,19,78,76,121,48*66
23 $GLGSV,2,2,08,79,46,322,50,81,47,210,51,87,33,036,49,88,75,052,50*6D
24 $GNGLL,4608.81479,N,00557.85447,E,120425.00,A,D*76
25 $GNRMC,120426.00,A,4608.81479,N,00557.85447,E,0.008,,070520,,,D*64

```

FIGURE 24 – Composition d'un fichier UBX

On remarque que chaque ligne commence par le symbole \$ suivi de 5 lettres, ce sont les identifiants des messages NMEA. Ces messages sont conçus pour transporter les données liées au GPS. Les 2 premières lettres servent à identifier le récepteurs et les 3 autres à identifier le type de message. L'ordre de chaque type de message est cyclique. Dans notre cas, le message qui nous intéresse est le message GLL car il contient la position GPS ainsi que l'heure d'acquisition. Grâce au logiciel de traitement de text TextMate et sa fonction de recherche dans un document, il est possible de récupérer toutes les lignes contenant la chaîne de caractère recherchée. En copiant les lignes et en les collant dans un nouveau document texte, tous les messages GLL sont récupérés sans modifier le fichier UBX. La figure 25 montre tous les messages GLL récupérés.

```

2706 $GNGLL,4608.81331,N,00557.85593,E,124929.00,A,A*75
2707 $GNGLL,4608.81330,N,00557.85592,E,124930.00,A,A*7D
2708 $GNGLL,4608.81329,N,00557.85589,E,124931.00,A,A*7E
2709 $GNGLL,4608.81328,N,00557.85586,E,124932.00,A,A*73
2710 $GNGLL,4608.81327,N,00557.85584,E,124933.00,A,A*7F
2711 $GNGLL,4608.81325,N,00557.85580,E,124934.00,A,A*7E
2712 $GNGLL,4608.81324,N,00557.85577,E,124935.00,A,A*76
2713 $GNGLL,4608.81324,N,00557.85574,E,124936.00,A,A*76
2714 $GNGLL,4608.81323,N,00557.85573,E,124937.00,A,A*77
2715 $GNGLL,4608.81323,N,00557.85572,E,124938.00,A,A*79
2716 $GNGLL,4608.81323,N,00557.85571,E,124939.00,A,A*7B
2717 $GNGLL,,,,,124940.00,V,N*5E
2718 $GNGLL,4608.81306,N,00557.85557,E,124941.00,A,D*72
2719 $GNGLL,4608.81305,N,00557.85553,E,124942.00,A,D*76
2720 $GNGLL,4608.81311,N,00557.85552,E,124943.00,A,D*73
2721 $GNGLL,4608.81310,N,00557.85551,E,124944.00,A,D*76
2722 $GNGLL,4608.81312,N,00557.85544,E,124945.00,A,D*71
2723 $GNGLL,4608.81312,N,00557.85539,E,124946.00,A,D*78
2724 $GNGLL,4608.81313,N,00557.85530,E,124947.00,A,D*71
2725 $GNGLL,4608.81311,N,00557.85521,E,124948.00,A,D*7C
2726 $GNGLL,4608.81308,N,00557.85512,E,124949.00,A,D*75
2727 $GNGLL,4608.81311,N,00557.85505,E,124950.00,A,D*73

```

FIGURE 25 – Messages GLL

Sur la figure 25, on remarque que le message à la ligne 2717 n'a pas été correctement reçu, il faut donc supprimer cette ligne. Avant de pouvoir extraire toutes les données nécessaires de ces messages, il faut savoir exactement comment est constitué un message GLL [28]. Prenons comme exemple le message à la ligne 2706.

| Message GLL   | Définition  |
|---------------|---|
| \$            | Caractère de start  |
| GP            | Identification du récepteur qui représente ici GPS                    |
| GLL           | Identification pour Positionnement Géographique Longitude et Latitude |
| 4608.81331,N  | Latitude 46 degrés 8.81331 minutes Nord                               |
| 00557.85593,E | Longitude 5 degrés 57.85593 minutes Est                               |
| 124929        | reçu à 12 :49 :29 UTC   |

FIGURE 26 – Détail d'une trame GLL

Maintenant que la construction d'une trame GLL est connue, il est possible d'extraire les données importantes pour les utiliser dans matlab. J'effectue cette tâche grâce au mode édition en colonne de TextMate qui permet d'éditer toutes les lignes d'un document simultanément. Cette manipulation permet de ne conserver que les informations essentielles, c'est à dire les valeurs latitude, longitude et heure. La figure 27 montre à quoi les données ressemblent une fois qu'elles ont été éditées.

|      |                                      |
|------|--------------------------------------|
| 2719 | <b>4608.81311 00557.85552 124943</b> |
| 2720 | <b>4608.81310 00557.85551 124944</b> |
| 2721 | <b>4608.81312 00557.85544 124945</b> |
| 2722 | <b>4608.81312 00557.85539 124946</b> |
| 2723 | <b>4608.81313 00557.85530 124947</b> |
| 2724 | <b>4608.81311 00557.85521 124948</b> |
| 2725 | <b>4608.81308 00557.85512 124949</b> |
| 2726 | <b>4608.81311 00557.85505 124950</b> |
| 2727 | <b>4608.81319 00557.85498 124951</b> |

FIGURE 27 – Donnée GPS en colonnes : Latitude, Longitude, Heure UTC

Ces données peuvent maintenant être utilisées dans matlab en utilisant l'outil d'importation.

#### 4.1.2 Conversion de coordonnées du format DM en format DD

Maintenant que les données sont importées dans matlab, il faut convertir ces données. En effet les coordonnées GPS récupérée au chapitre 4.1.1 sont au format DM (Degrés Minutes), or il est impossible de calculer des distances entre des points sous cette forme. Il faut les convertir en DD (Degrés Décimales). Le même calcul doit être appliqué sur la latitude et sur la longitude. La conversion n'est pas très compliquée, il suffit de diviser la partie minutes par 60. L'équation 1 montre le résultat avec la latitude de l'exemple à la figure 26. La partie noté d'un [ ' ] désigne les minutes.

$$4608.81331 = 46^{\circ}08.81331' = 46^{\circ} + (8.81331/60)^{\circ} = 46.1468885^{\circ} \quad (1)$$

#### 4.1.3 Calcul de distance orthodromique entre deux points

La distance orthodromique entre deux points GPS est la distance la plus courte qui relie ces deux points<sup>3</sup> [29]. Dès qu'il est question de positionnement GPS, on peut faire intervenir une distance par rapport à un autre point, que ce soit le point de la mesure précédente ou un point de référence connu. Le tableau 28 contient un point de départ et un point d'arrivée qui vont servir d'exemple afin de calculer la distance entre ces points.

|                | <b>Latitude</b>      | <b>Longitude</b>      |
|----------------|----------------------|-----------------------|
| <b>Départ</b>  | $Lat_d = 46.146898$  | $Lon_d = 5.964253$    |
| <b>Arrivée</b> | $Lat_a = 46.1469025$ | $Lon_a = 5.964252333$ |

FIGURE 28 – Cordonnées pour démontrer la distance orthodromique entre deux points

Ces deux points ont été obtenus par mesure avec les GPS RTK. Une mesure a été faite pour obtenir le point de départ, ensuite l'antenne GPS a été déplacée de 50

3. Cette distance est souvent appelée "distance à vol d'oiseau"

cm, puis les coordonnées du second point ont été prises.

L'équation 2 permet d'obtenir la distance orthodromique entre ces deux points. [29]

$$\begin{aligned}
 D &= 60 \cdot \arccos(\sin(Lat_d) \cdot \sin(Lat_a) + \cos(Lat_d) \cdot \cos(Lat_a) \cdot \cos(Lon_a - Lon_d)) \cdot 1851.85 \\
 &= 60 \cdot \arccos(0.686937160874035 + 0.313062839115771) \cdot 1851.85 \\
 &= 60 \cdot 4.515322316762228 \cdot 10^{-6} \cdot 1851.85 = 0.5017[m]
 \end{aligned}$$

(2)

L'équation calcule la longueur de l'arc entre le point d'arrivé et le point de départ grâce à l'angle formé entre le point de départ et le point d'arrivée. L'angle est ensuite multiplié par 60 afin d'obtenir une distance en mille marin puis remultiplié par 1851,85 pour obtenir la distance en mètres.<sup>4</sup>. Grâce à cette équation, on trouve bien la distance attendue, la différence étant due aux incertitudes de positionnement ainsi qu'à la précision du GPS RTK.

## 4.2 Mesure de stabilité

### 4.2.1 Déroulement de la mesure

Cette mesure consiste à mettre en place le GPS RTK sur une surface plane et dégagée, attendre qu'il lève l'ambiguïté de phase, puis à lancer une mesure la plus longue possible. Cette mesure à durée 8 heures et permet de vérifier que le GPS n'a pas de perte de signal ou de dérive de position dans le temps.

Pour cette mesure, les deux modules sont connectés au même ordinateur. La distance entre les antennes RF est d'environ 3 mètres. Cette distance se trouve également entre les deux antennes GPS. La distance entre les GPS n'est pas très importante ici, tant qu'ils sont suffisamment proches pour que les corrections de la base soit également valable pour le rover. La base s'est fixée avec une précision de 1 mètre après 15 minutes de recherche.

### 4.2.2 Analyse de la mesure

La mesure a durée 8 heures et comme une mesure est effectuée chaque seconde, le nombre de points GPS totaux obtenu est 28800. La figure 29 obtenue sur matlab offre un premier aperçu des mesures.

---

4.  $1/60$  degrés = distance de 1 mille = 1851.85 mètres

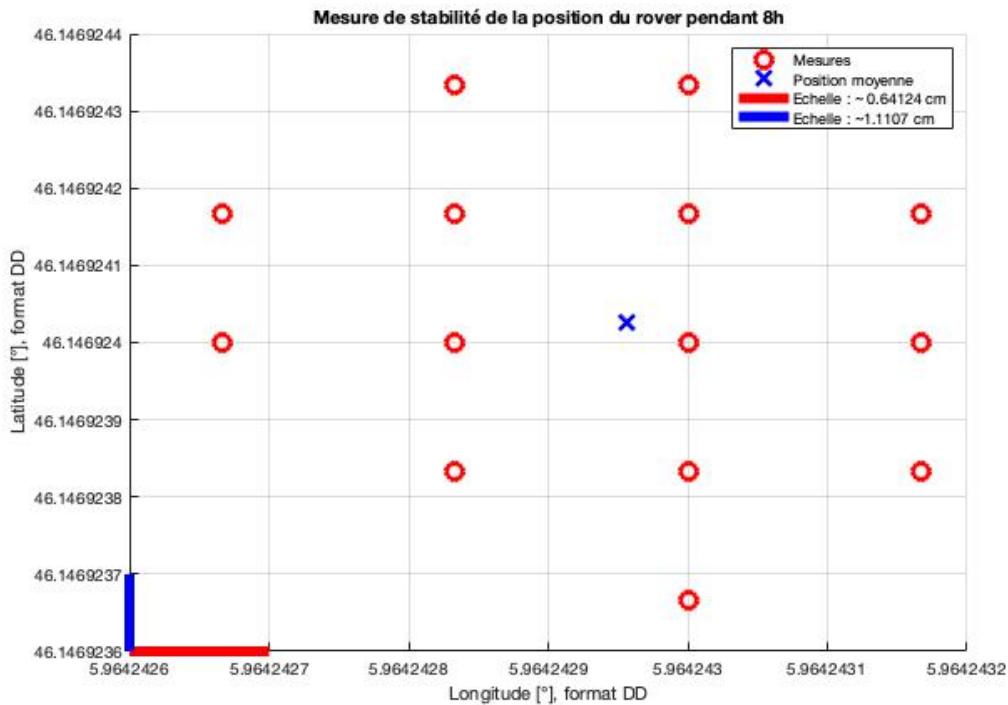


FIGURE 29 – Disposition des 28800 points GPS

La première chose que l'on peut constater, c'est qu'en 8 heures de mesures sans bouger, le GPS n'a pas perdu le signal car ce graphique est constitué de 28800 points comme on s'y attendait. La deuxième chose importante, tous les points se trouvent dans un cercle de quelques centimètres. Les deux segments à l'origine du graphique donnent une indication de l'échelle<sup>5</sup>. La disposition des points est également intéressante, on remarque que les différents points de mesure sont équitablement espacés, ce qui semble indiquer que c'est la résolution maximum atteignable par le système.

Bien que la moyenne des points soit indiquée, on ne peut pas voir la répartition des points sur ce graphique car les 28800 points se trouvent divisés en 14 positions différentes. La figure 30 montre la dispersion des mesures.

5. Cette indication n'est présente que pour faire une approximation de la distance car la distance parcourue pour un degré en longitude n'est pas la même en fonction de la latitude à laquelle on se trouve.

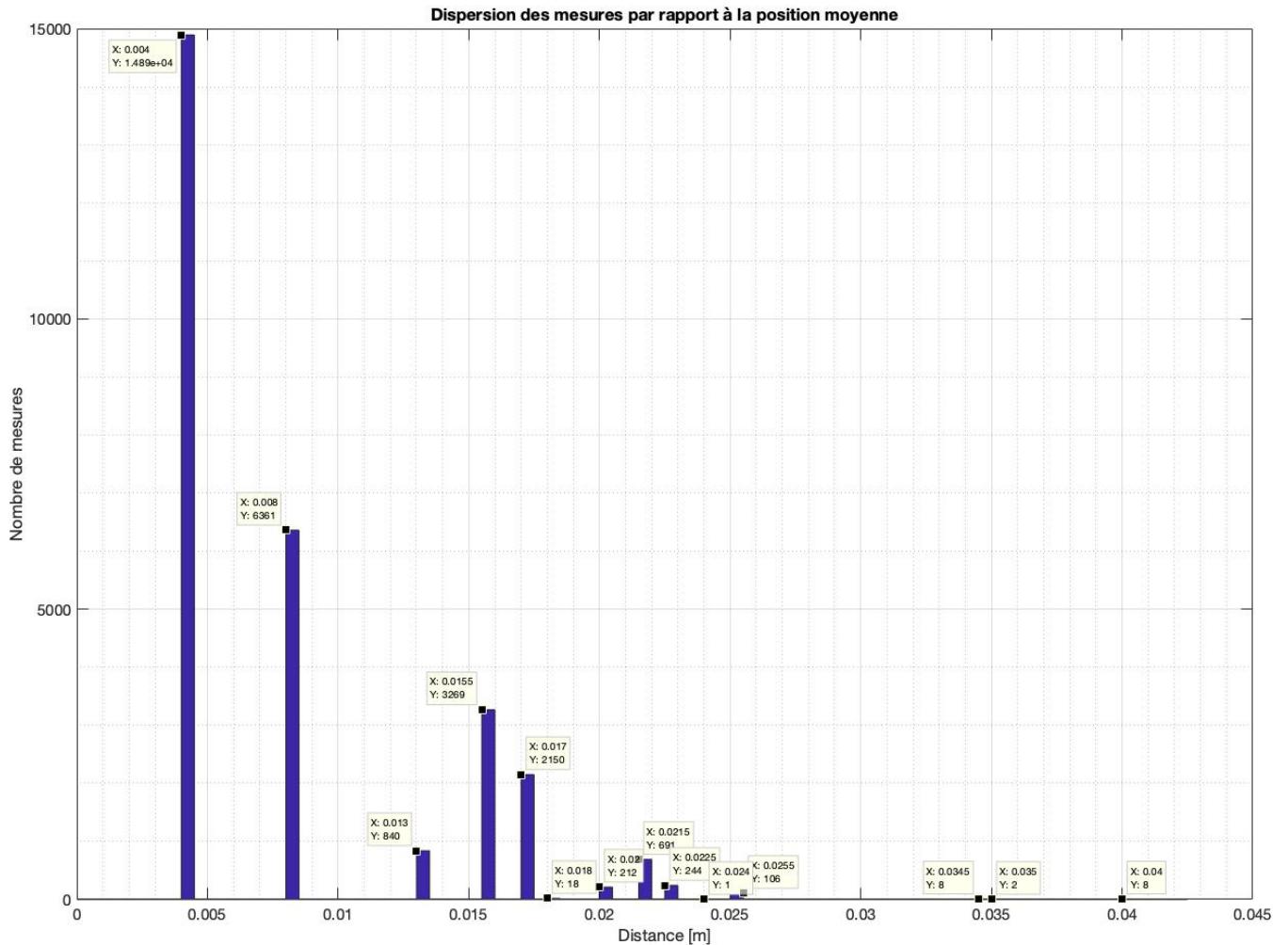


FIGURE 30 – Dispersion des 28800 points GPS

Cette figure nous montre la répartition des points. On y apprend que la majorité des points se trouvent à 4 mm de la position moyenne. L'écart type de distance entre la moyenne et les points est de 5.5 mm. Le point le plus distant de la moyenne se trouve à 4.0 cm.

La précision annoncée par u-blox pour le NEO-M8P est de 0.025 m + 1 ppm CEP [26]. Ce qui signifie que 50%<sup>6</sup> des mesures doivent se trouver dans un cercle de 2,5 cm de rayon et que pour chaque kilomètre de distance entre la base et le rover, le rayon du cercle augmente de 1 mm.

D'après le graphique 30, 51,7% des mesures se trouvent dans un cercle de 4 mm,

6. La mesure CEP peut être utilisée pour indiquer un autre taux de précision comme 90% par exemple, mais ce taux doit alors clairement et défini, en cas d'absence d'information, c'est un taux de 50% qui est sous-entendu.

ce qui est bien meilleur que la précision annoncée par U-blox. Comme je n'ai pas d'informations sur la méthode qu'ils ont utilisée pour calculer la précision qu'ils avancent mais que mes résultats semblent être 6 fois plus précis, je suppose que ma précision est due à la durée de la mesure. Les 8 heures de mesures auraient permis à mon GPS d'affiner sa position avec le temps alors que si la mesure avait duré moins longtemps, la précision aurait été moins bonne. Afin de vérifier cette hypothèse, je retrace le même graphique que celui de la figure 30 mais en différenciant chaque heure de mesure, ce qui va permettre de voir si avec le temps, il y a convergence des valeurs vers la valeur moyenne ou pas.

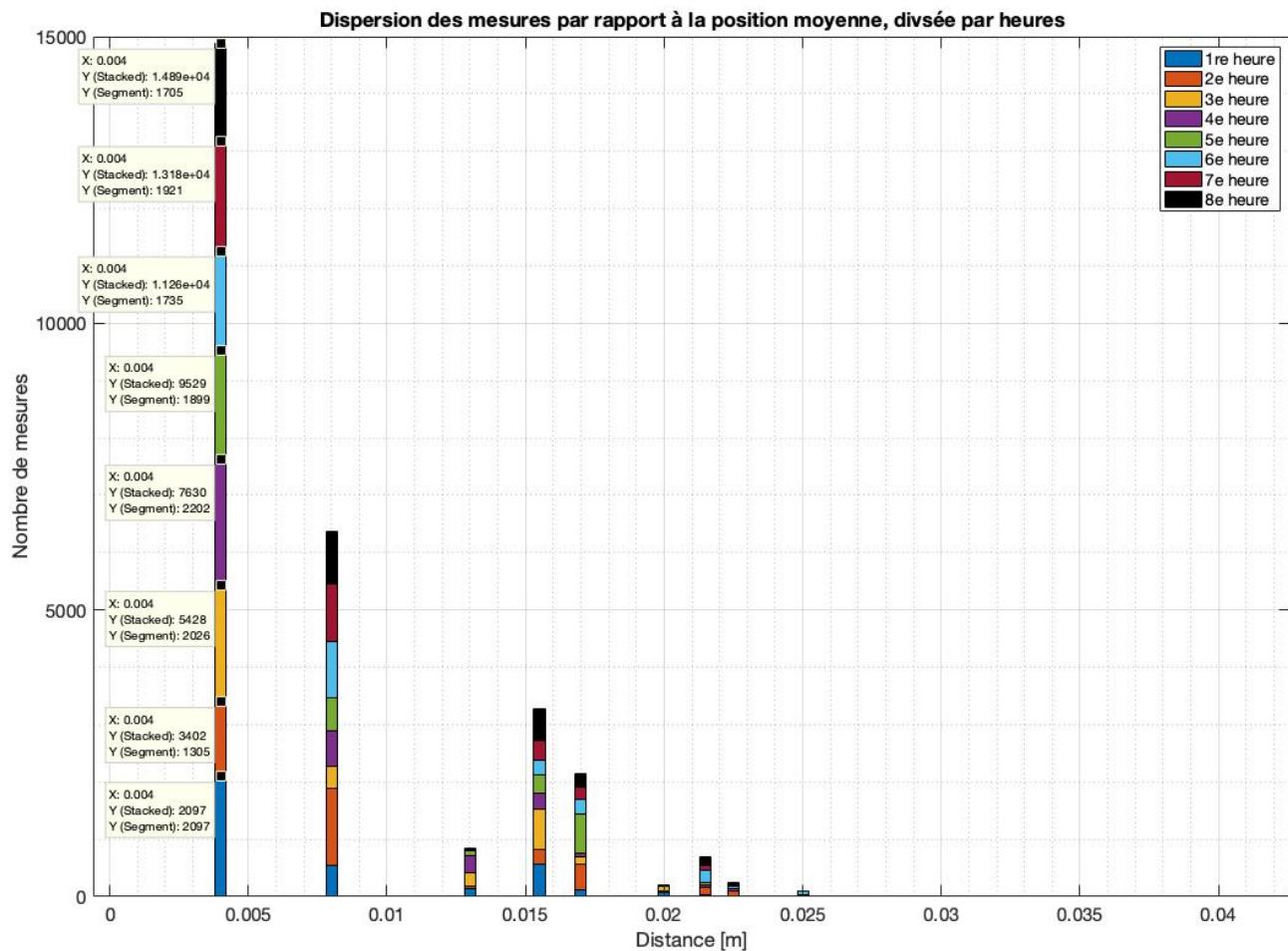


FIGURE 31 – Dispersion des 28800 points GPS, divisé par heure

D'après le graphique 31, les valeurs ne convergent pas avec le temps. Ce qui signifie que faire une mesure de 1 heure ou de 8 heures ne change pas la précision que l'on peut atteindre. Le tableau à la figure 32 permet de voir un peu plus clairement les précisions qui ont été obtenues lors de cette mesure.

|   |         | Cercle de 0.4 cm de rayon |         |             |             |         | Cercle de 2.5 cm de rayon |             |             |  |  |
|---|---------|---------------------------|---------|-------------|-------------|---------|---------------------------|-------------|-------------|--|--|
| h | pts tot | nouv pts                  | pts tot | rapport pts | rapport tot | nouv pt | pts tot                   | rapport pts | rapport tot |  |  |
| 1 | 3600    | 2097                      | 2097    | 58,25%      | 58,25%      | 3600    | 3600                      | 100,00%     | 100,00%     |  |  |
| 2 | 7200    | 1305                      | 3402    | 36,25%      | 47,25%      | 3600    | 7200                      | 100,00%     | 100,00%     |  |  |
| 3 | 10800   | 2026                      | 5428    | 56,28%      | 50,26%      | 3600    | 10800                     | 100,00%     | 100,00%     |  |  |
| 4 | 14400   | 2202                      | 7630    | 61,17%      | 52,99%      | 3600    | 14400                     | 100,00%     | 100,00%     |  |  |
| 5 | 18000   | 1899                      | 9529    | 52,75%      | 52,94%      | 3598    | 17998                     | 99,94%      | 99,98%      |  |  |
| 6 | 21600   | 1735                      | 11264   | 48,19%      | 52,15%      | 3600    | 21598                     | 100,00%     | 99,99%      |  |  |
| 7 | 25200   | 1921                      | 13185   | 53,36%      | 52,32%      | 3600    | 25198                     | 100,00%     | 99,99%      |  |  |
| 8 | 28800   | 1705                      | 14890   | 47,36%      | 51,70%      | 3584    | 28782                     | 99,55%      | 99,93%      |  |  |

FIGURE 32 – Tableau récapitulatif de la précision des coordonnée GPS

Avec les colonnes :

- h : Nombre d'heures écoulées depuis le début de la mesure.
- pts tot : Nombre de points totaux depuis le début de la mesure.
- nouv pts : Nombre de nouveaux points se trouvant dans un cercle de 0.4 / 2.5 cm.
- pts tot : Nombre de points totaux se trouvant dans un cercle de 0.4 / 2.5 cm.
- rapport pts : Rapport entre les points se trouvant dans un cercle de 0.4 / 2.5 cm et les 3600 points de mesure de l'heure actuelle.
- rapport tot : Rapport entre les points se trouvant dans un cercle de 0.4 / 2.5 cm et tous les points obtenus jusqu'à présent.

Le tableau 32 permet de voir que le pourcentage de coordonnée GPS à se trouver dans un cercle de 0.4 cm autour de la moyenne lors de la première heure de mesure est de 58,25 %. Ce rapport descend légèrement pendant les heures de mesures suivantes (sauf ligne 4). Il n'y a donc pas de gain de précision lorsque l'on laisse tourner plusieurs heures le GPS en espérant qu'il accroisse sa précision. Cela permet en revanche d'écartier des valeurs aberrantes qui pourraient survenir occasionnellement, en cas de perte de la mesure de phase par exemple.

Ce tableau met également en évidence que le GPS de U-blox permet d'obtenir une précision de 2,5 cm supérieur à 99% du temps alors que leur documentation laissait entendre que cette valeur ne serait atteinte que pour 50% du temps.

Cette première mesure permet donc de s'assurer qu'il est possible d'obtenir une précision de 2.5 cm comme annoncée par le constructeur, et même mieux en cas de mesure statique. Mais il est important de garder à l'esprit que bien qu'il soit extrêmement stable en mesure statique, cela n'enlève pas une erreur possible d'un décalage constant de position. En effet je ne dispose d'aucun moyen pour m'assurer que la position moyenne que j'ai obtenu corresponde bien à la position de mon GPS. Il se pourrait par exemple que le GPS subisse un décalage de 10 cm vers le nord pour chacune des mesures par exemple. Une telle erreur m'est impossible à détecter car je ne dispose pas de point de référence afin de comparer les coordonnées. Pour cette

mesure et toutes les suivantes, je vais devoir supposer que si les mesures sont suffisamment proches pour obtenir une moyenne cohérente, les coordonnées moyennes indiquent exactement la position de mon antenne GPS.

### 4.3 Mesure de distance

#### 4.3.1 Déroulement de la mesure

Maintenant que la stabilité du GPS et sa capacité à garder l'accroche au signal GPS ont été démontrées, il est possible de commencer à faire des mesures de distances en positionnant l'antenne GPS à différents endroits. La mesure qui a été effectuée est la suivante :

La base a été positionnée sur une table et s'est fixée en 15 min avec une précision de 1 mètre. Le rover se trouve sur la même table. Deux points, distant de 50 cm, ont été marqué sur la table à l'aide d'une règle. On positionne l'antenne GPS du rover sur le premier points (point de départ), on s'assure que la qualité du signal est bonne, on récupère les coordonnées pendant 1 heure, puis on déplace l'antenne GPS sur le second point (point d'arrivée) pour y récupérer les coordonnées pendant 1 heure également.

#### 4.3.2 Analyses des mesures

La figure 33 montre la dispersion des deux points mesurés.

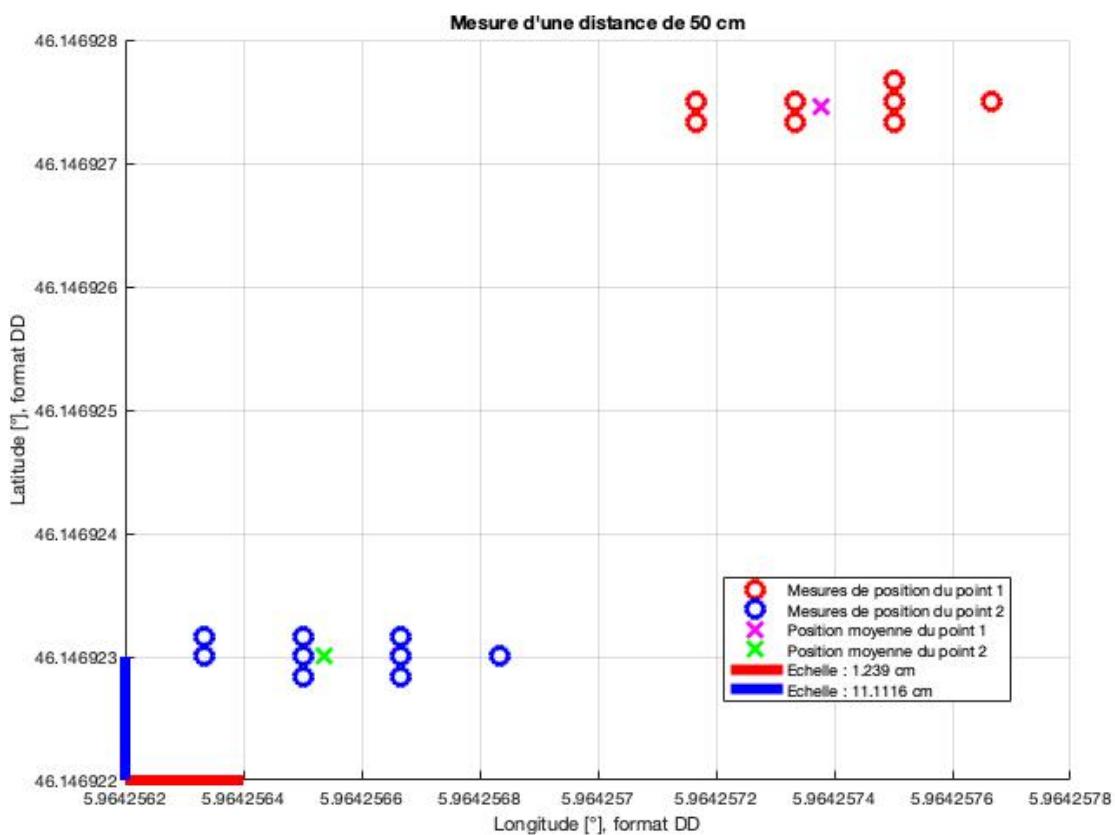


FIGURE 33 – Disposition des 2 x 3600 points de mesure distancés de 50 cm

L'écart type autour de la moyenne du point de départ est de 3,8 mm et de 6,4 mm autour du point d'arrivée. Le point de départ le plus loin de la moyenne de départ se trouve à 2,4 cm et le point d'arrivée le plus loin de la moyenne se trouve à 2,1 cm. Ces valeurs montrent que 100% des mesures se trouvent dans un cercle de 2,5 cm de rayon, alors que u-blox en annonçait 50%, mais le plus intéressant dans cette mesure est d'étudier la distance entre les points de départ et d'arrivée.

La distance entre les positions moyennes est de 49,7 cm. La distance théorique entre les deux points de mesure est de 50 cm, mais l'antenne a été positionnée à la main et il est fort possible que lors de son positionnement une erreur de 1 ou 2 mm ait été commise à chaque positionnement. Mais cette distance de 49,7 cm est une distance obtenue grâce à deux acquisitions d'une heure chacune, or il est possible que s'il faut faire une mesure de distance comme celle-ci avec un but autre que simplement étudier la précision du GPS, on ne dispose pas forcément de deux heures pour obtenir cette distance. La figure 34 représente l'histogramme des distances. Celui-ci contient la distance entre chaque point de départ, vers chaque point d'arrivée, soit 12'960'000 distances.

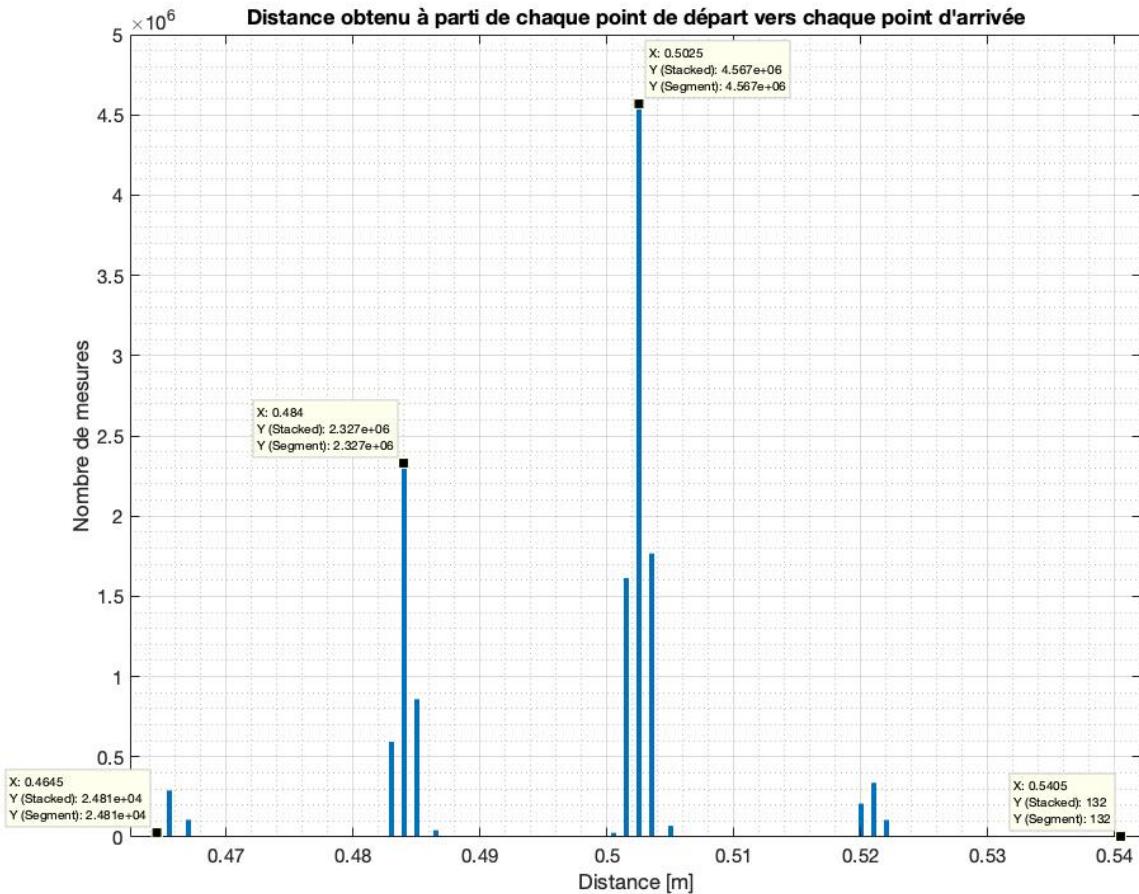


FIGURE 34 – Histogramme des distances entre les 3600 points de départ et les 3600 points d'arrivée

Parmi tous les points, la distance la plus courte obtenue est 46,4 cm et la distance la plus longue est de 54 cm. L'erreur la plus grande est donc de 4 cm. Si l'on se base sur le tableau de la mesure de stabilité 32, plus de 99% des mesures avaient une erreur inférieure à 2,5 cm. On peut en déduire que dans le pire des cas, à condition d'avoir suffisamment de mesures afin d'écartier les 1% restant, l'erreur pourrait atteindre environ 2 fois l'erreur maximum sur une mesure, soit environ 5 cm en tout. Le tableau 35 reprend les valeurs de cet histogramme et met en évidence le pourcentage de mesures qui se situent à des distances fixes de la moyenne afin de pouvoir comparer un peu mieux les précisions que l'on peut attendre.

| 50 cm ± :                               | 3 mm    | 4 mm    | 1 cm    | 2,5 cm  |
|---|---------|---------|---------|---------|
| Pourcentage de points plage de distance | 47,89 % | 61,53 % | 62,07 % | 96,67 % |

FIGURE 35 – Tableau récapitulatif de la précision des distances par coordonnée GPS

Ce tableau montre que près de la moitié des distances calculées ont une erreur égale ou inférieure à 3 mm. Ce qui signifie que si l'on doit connaître une distance avec une

précision de 3 mm, seules 50% des mesures sont correctes, alors que si une précision de 2,5 cm est suffisante, 96,67% des mesures sont utilisables. Si 3 mm d'erreur peuvent sembler très correct sur une distance de 50 cm, 2,5 cm représentent quand même une erreur de 5%, et une telle erreur ne peut pas convenir pour toutes les applications. Mais il ne faut pas oublier que cette erreur n'est pas proportionnel à la distance que l'on mesure. Ainsi, si on essaye de mesurer 1 km avec la même technique, les erreurs devraient être similaire, mais que ce soit 3 mm ou 2,5 cm (+ 1 PPM, donc + 1 mm) sur 1 km, le rapport d'erreur commis est beaucoup plus petit et la précision est donc beaucoup plus importante.

A titre de comparaison, un odomètre de précision (roue qui compte la distance parcourue) a une précision de 2%, soit 2 cm pour 100 m. Mais cette erreur est proportionnelle à la distance, elle serait donc de 20 cm pour 1 km, alors que l'erreur resterait de 2,5 cm avec le GPS RTK, avec un taux de fiabilité proche de 96% s'il reste identique à cette mesure.

Cette mesure a permis de voir que le GPS RTK peut mesurer des distances. La précision n'étant que très peu dépendante de la distance, le GPS RTK serait plutôt recommandé pour mesurer de grandes distances, à partir du moment où l'on juge acceptable 2,5 cm d'erreur et 96% de fiabilité.

#### 4.4 Étude de la précision des mesures en fonction de la distance entre le rover et la base

Cette mesure a pour but de confirmer que la précision des coordonnées GPS du rover ne se dégrade pas trop lorsque l'on éloigne le rover de la base.

Cette mesure a du être faite dans mon jardin pour cause du confinement français dû au COVID-19. Une partie de mon jardin est suffisamment dégagé pour permettre des mesures statiques mais cette mesure a été effectué avec des distances de 1 à 20 m entre la base et le rover. Bien que la base soit restée fixe dans la partie dégagée, le rover lui, a dû être placé dans des endroits peu propices à la réception des signaux GPS car il était situé entre de très grands arbres, la maison, et une ligne haute tension passant juste au dessus. Il faut donc s'attendre à de grosses variations dans les mesures.

##### 4.4.1 Liste du matériel

Le matériel utilisé pour cette mesure est le suivant :

- 2x C94-M8P.
- 2x Ordinateur portable avec u-center.
- 1x Télémètre laser (aucune référence/fabricant/modèle sur le miens ), portée théorique, de 5 cm à 100 m, aucune indication de la précision.
- 1x table fixe (pour y placer la base)
- 1x table motorisée réglable en hauteur pour y placer le rover

- rallonges électriques pour la table motorisée et les ordinateurs

#### 4.4.2 Déroulement de la mesure

Cette mesure est un peu plus compliquée à mettre en place car contrairement aux autres mesures où les déplacements pouvaient être fait sur une table, il faut maintenant déplacer une table et y positionner chaque fois l'antenne GPS du rover. La base est restée sur une autre table sans être déplacée. Il n'est donc plus possible de mesurer la distance entre la base et le rover avec une simple règle. Au début je pensais tendre une corde entre deux points du jardin et y faire des repères aux distances souhaitées en mesurant avec une seconde corde de 2 m. Mais au bout de 5 m de mesure, lorsque je refaisais la même mesure pour m'assurer de la distance, j'y trouvais une différence de près de 3 cm entre les deux points. Cette méthode de mesure n'était donc vraiment pas adaptée. Mais j'ai pu me procurer un télémètre laser, ce qui m'a permis de mesurer les distances de manière beaucoup plus fiable.

Le déroulement de la mesure est donc le suivant : la base est positionnée sur une table, dans un endroit bien dégagé. Elle s'est fixée en environ 20 minutes avec une précision de 1 m. Le rover a ensuite été placé à 1 mètre de distance, distance obtenue grâce au télémètre laser. Une fois que tous les signaux sont stables et que le rover est en RTK FIX, une acquisition de 20 min est lancée, ce qui représente 1200 mesures. Puis le rover est déplacé à une autre distance et la procédure recommence. Les mesures ont été effectuées à 1, 2, 5, 10, 15 et 20 mètres. Afin de compenser le dénivelé du jardin et de garder l'antenne du rover juste au dessous du pointeur laser du télémètre, le rover a été placé sur une table motorisée, ce qui permet d'ajuster la hauteur de la table et ainsi faire correspondre la hauteur où arrive le laser avec la hauteur à laquelle se trouve l'antenne GPS du rover.

#### 4.4.3 Analyses des mesures

Comme pour les mesures précédentes, la première figure montre la disposition des mesures.

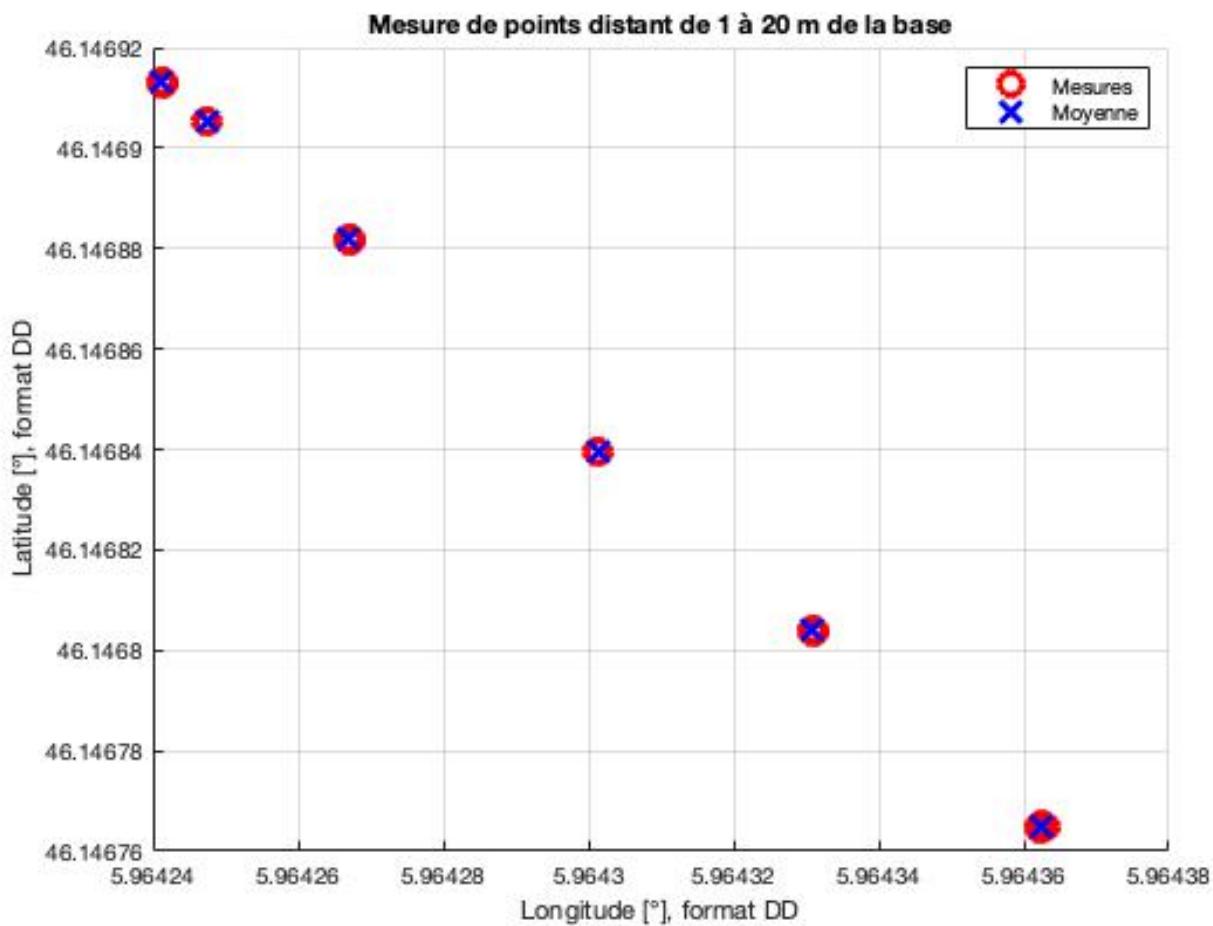


FIGURE 36 – Disposition des points de mesures. Distance depuis la base : 1, 2, 5, 10, 15 et 20 mètres

La figure 36 montre que tous les points sont relativement groupés. Le premier groupe de points en haut à gauche se trouve à une distance de 1 mètre de la base, et celui tout en bas à droite représente la mesure qui se trouve à 20 mètres de la base, donc à 19 mètres de la première mesure.

Comme les points les plus éloignés sont espacés de 19 mètres, on ne peut pas voir la dispersion des points pour chaque mesure. Le graphique 37 est un zoom sur chaque point de mesures. Les moyennes sont à peu près alignées et les axes des graphiques font la même taille afin de pouvoir étudier la dispersion des mesures

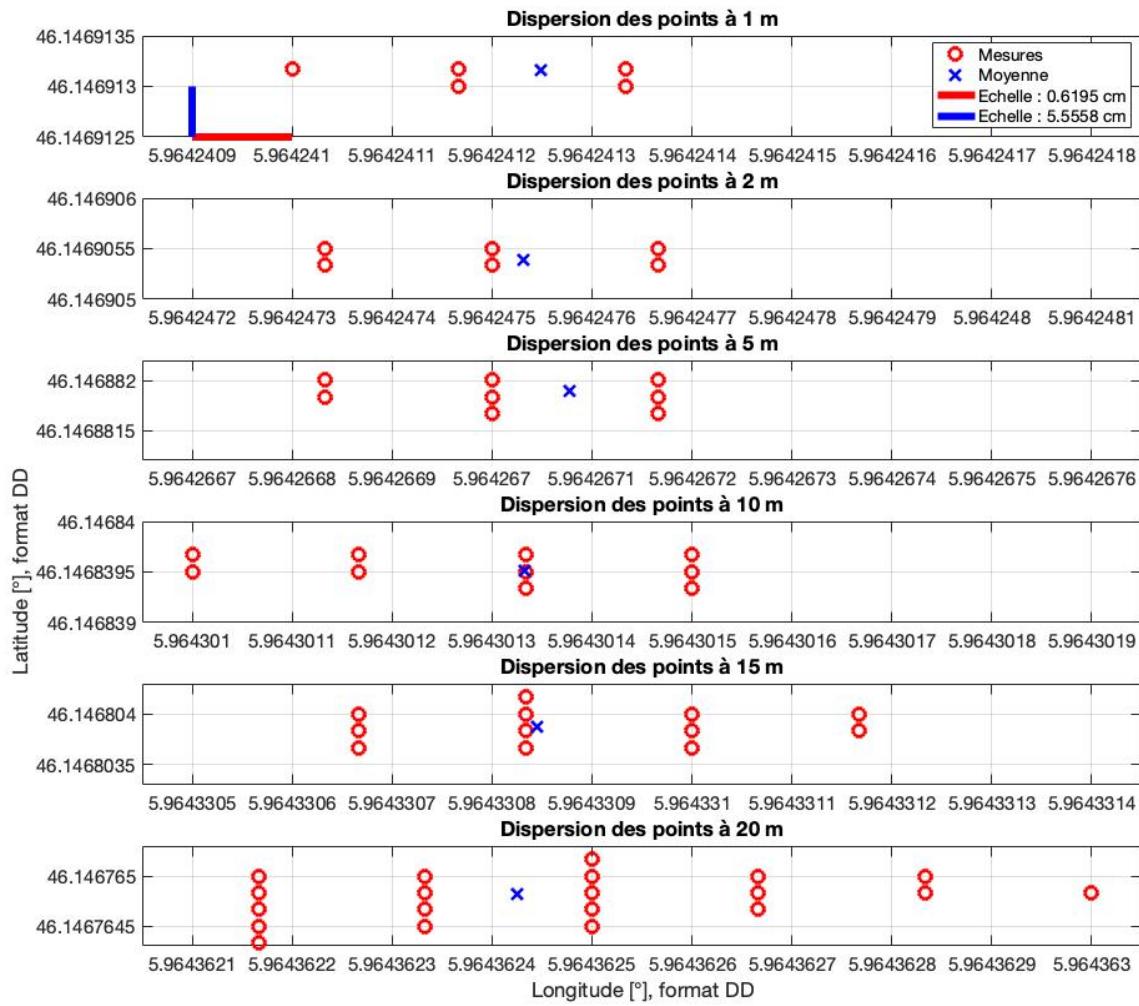


FIGURE 37 – Répartition des points de mesures

La figure 37 montre que la dispersion des mesures augmente sensiblement lorsque le rover s'éloigne de la base, mais cette dispersion n'est pas forcément dû à la distance qui augmente. Comme ces mesures ont été effectuée chez moi, les mesures de 10 15 et 20 mètres ont été faites avec l'antenne GPS se trouvant entourée d'une très haute rangée d'arbres très denses d'un coté et la maison de l'autre. Il est fort possible que tous ces obstacles aient perturbé les signaux GPS, causant notamment des multi-trajets et des blocages de certains signaux GPS. Il est possible que la baisse de précision des mesures soit dû à la diminution de la quantité de signaux valides reçus. L'histogramme à figure 38 montre la dispersion des mesures par rapport à la moyenne.

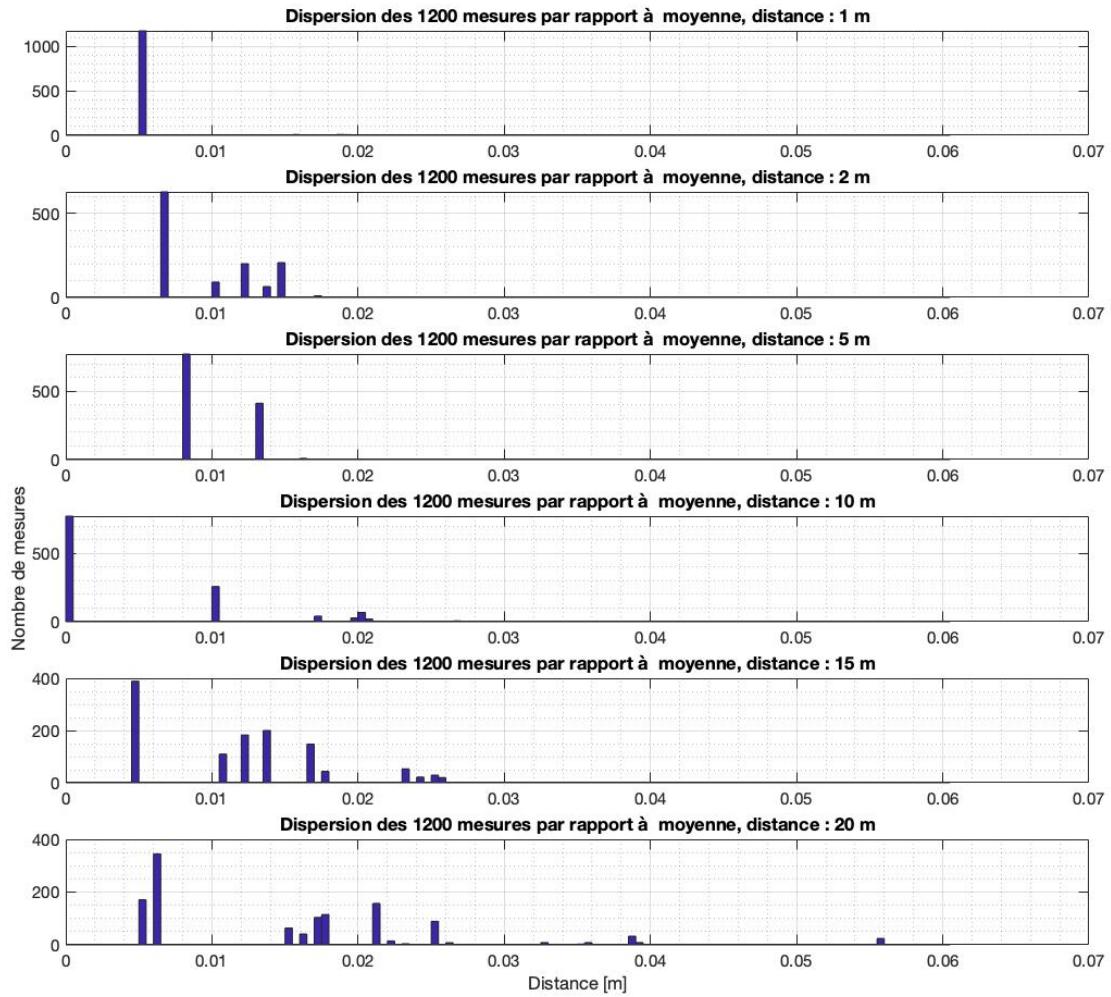


FIGURE 38 – Dispersion des mesures par rapport à la moyenne

Cet histogramme permet de confirmer qu'il y a effectivement plus de dispersion sur les mesures qui sont les plus éloignées de la base. Alors que les mesures les plus lointaines sont à 1,9 cm de la moyenne lorsque le rover se trouve à 1 m de la base, elle sont à plus de 5,5 cm lorsque le rover est à 20 m. Encore une fois, cette perte de précision n'est pas forcément dû à la distance mais peut être à cause de la qualité de réception des signaux GPS. Le tableau 39 fait ressortir les chiffres importants de cet histogramme.

| distance base/rover [m] | écart type [cm] | distance max [cm] | rayon de 2,5 cm |
|-------------------------|-----------------|-------------------|-----------------|
| 1                       | 0,18            | 1,9               | 100 %           |
| 2                       | 0,34            | 1,7               | 100 %           |
| 5                       | 0,25            | 2,6               | 100 %           |
| 10                      | 1,06            | 2,7               | 99,5 %          |
| 15                      | 0,73            | 3,2               | 98,25 %         |
| 20                      | 1,06            | 5,6               | 92,08 %         |

FIGURE 39 – Tableau récapitulatif de la précision des distances par coordonnée GPS

Avec les colonnes :

- distance max : Distance entre la moyenne et le point le plus éloigné de la moyenne.
- rayon de 2,5 : Pourcentage des points se trouvant dans un rayon de 2,5 mm autour de la moyenne.

La colonne distance max permet de mieux voir la dispersion des points au fil des mesures. Cette dispersion est également visible dans la colonne écart type. On remarque également que bien que les mesures s'éloignent un peu de la moyenne, plus de 92% des points se trouvent toujours dans un cercle de 2,5 cm de rayon, ce qui est bien supérieur aux 50 % annoncé par U-blox.

Maintenant que les erreurs autour des moyennes de chaque distance sont connues, il est possible d'étudier la distance entre chaque mesure. La figure 40 montre les distantes que l'on obtient si l'on prend tous les points se trouvant à 1 mètre et que l'on calcule la distance avec chacun des points se trouvant à 2,5,10,15 et 20 mètres, nous devrions trouver des distances de 1,4,9,14 et 19 mètres.

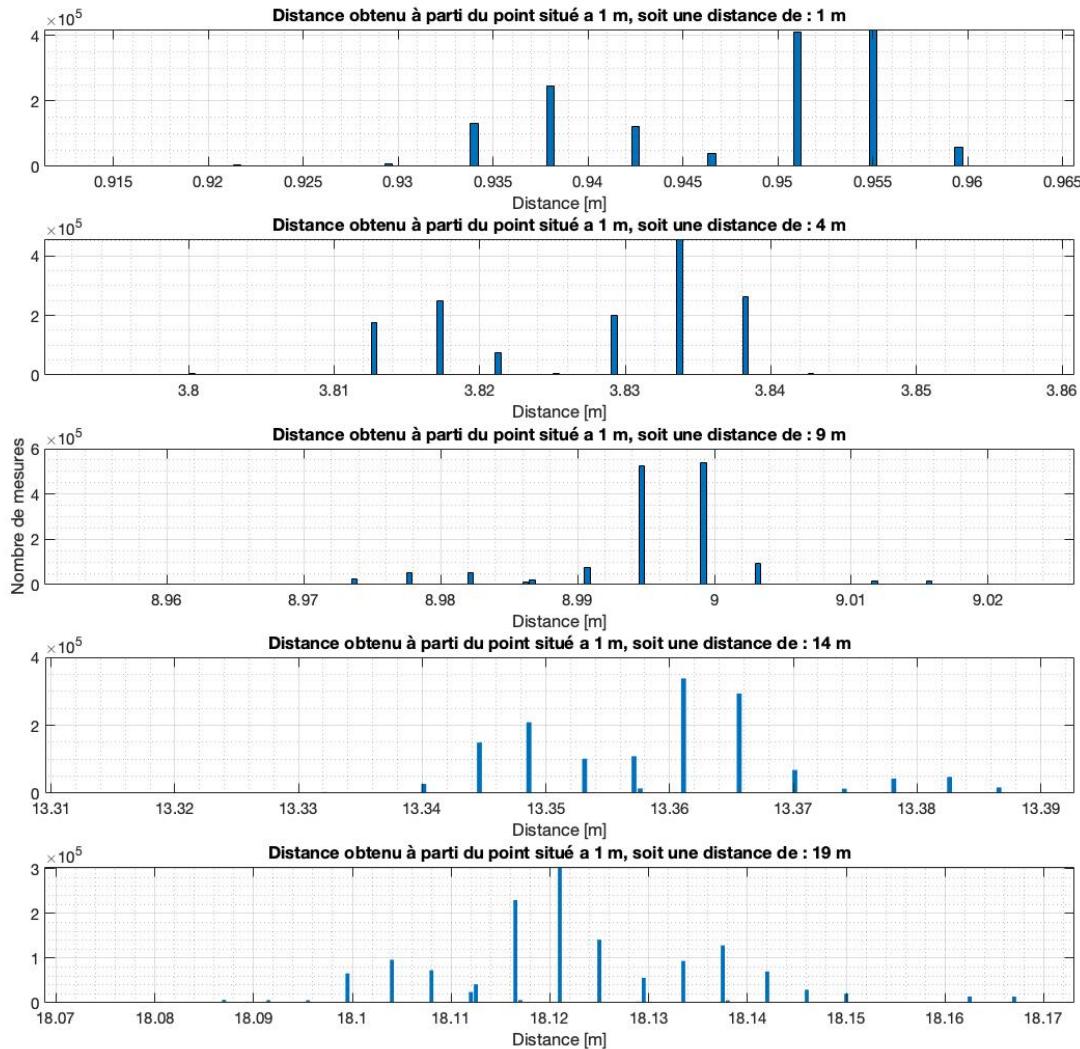


FIGURE 40 – Distances obtenues à partir des points se trouvant à 1 m vers les points se trouvant à 2,5,10,15 et 20 m

Ce graphique montre que les distances attendues entre certaines mesures ne sont pas du tout correctes. Alors que la distance attendue était de 19 m, la distance moyenne constatée est d'environ 18,12 mètres, soit près de 90 cm d'erreur alors que la mesure de 9 m semble tout à fait correcte. Étant donné que l'écart entre la mesure la plus lointaine et la moyenne à 20 m est de 5,5 cm et que le point le plus loin de la moyenne à 1 m se trouve à 1,9 cm, l'écart maximum par rapport à la distance attendu devrait être au pire de  $\pm 7,4$  cm, ce qui n'est pas du tout le cas ici.

A ce stade de la mesure, deux problèmes peuvent être la cause de telles erreurs.

Le premier serait une erreur de mesure du télémètre laser, étant donné qu'il n'y a

aucune référence ni aucun fabriquant qui figure sur l'appareil, il est possible que ça ne soit pas un appareil d'une grande fiabilité ni d'une grande précision. Cependant, après le déconfinement français, j'ai pu récupérer un autre télémètre laser, un würth wdm 8-14 afin de confronter les mesures des deux appareils. Bien que j'ai essayé de mettre en défaut le télémètre sans fabricant, l'écart de mesure entre les deux appareils ne dépassait pas 2 cm, ce qui n'est pas une erreur suffisante pour justifier l'erreur entre les distances.

Le second problème serait un gros décalage lors des mesures GPS. En effet, comme déjà expliqué lors de la mesure de stabilité de 8 heures, les mesures permettent de vérifier la fidélité de celles-ci, mais pas la justesse. Cela signifie que si un offset est présent sur toutes les mesures d'un même point, il m'est impossible de le mesurer avec certitude car j'utilise comme point de référence une autre mesure effectuée avec le même GPS.

C'est selon moi ce qui c'est passé lors de cette mesure, les signaux GPS ont été obstrués et certains ont subi de multi-trajets, ce qui a provoqué des erreurs de positionnement de près d'un mètre. Le tableau 41 montre les erreurs pour chaque distance.

| distance rover/rover [m] | distance moyenne [m] | erreur [m] | erreur [%] |
|--------------------------|----------------------|------------|------------|
| 1                        | 0,95                 | 0,05       | 5 %        |
| 4                        | 3,83                 | 0,17       | 4,25 %     |
| 9                        | 9,00                 | 0          | 0 %        |
| 14                       | 13,36                | 0,64       | 4,57 %     |
| 19                       | 18,12                | 0,88       | 4,63 %     |

FIGURE 41 – Tableau récapitulatif de la précision des distances par coordonnée GPS

Le tableau 41 montre que l'erreur de chaque mesure se situe entre 4 et 5 %, sauf pour celle à 9 mètre qui elle, est parfaite, alors qu'elle a été effectuée juste sous un arbre ... Cette mesure permet de chiffrer l'erreur que l'on peut avoir si on utilise le RTK pour mesurer des distances dans des endroits où le ciel n'est pas complètement dégagé et où il y a des objets hauts et denses à proximité.

Cette mesure montre les difficultés que l'on peut rencontrer si l'on essaye d'utiliser le GPS RTK dans des endroits non adaptés. Mais toutes les erreurs ont été justifiées par l'hypothèse que les signaux GPS ont été obstrués. Il va donc falloir refaire cette mesure mais dans un endroit suffisamment dégagé cette fois-ci afin de s'assurer que les erreurs de distances disparaissent bien.

## 4.5 Étude de la précision des mesures en fonction de la distance entre le rover et la base dans un environnement dégagé

Cette mesure a pour but de confirmer qu'il est possible d'atteindre les précisions annoncées par u-blox si le GPS se trouve dans un endroit dégagé. Comme découvert dans la mesure précédente, il est possible que les mesures soient centrées dans un rayon de quelques centimètres autour de la moyenne mais que la distance entre les moyennes soient complètement fausses. Cette mesure va donc être similaire à la mesure précédente et devrait permettre d'obtenir une diminution significative des erreurs de distances entre les moyennes

### 4.5.1 Liste du matériel

Comme la mesure est effectuée en dehors de chez moi cette fois ci, le matériel utilisé diffère un peu, notamment le télémètre. Celui sans nom de la mesure précédente n'est pas utilisable au soleil car il est impossible de voir le laser. Le nouveau télémétrage Würth contourne ce problème car il embarque une caméra qui indique précisément où la mesure sera effectuée.

Le matériel utilisé pour cette mesure est le suivant :

- 2x C94-M8P.
- 2x Ordinateur portable avec u-center.
- 1x Télémètre laser Würth WDM 8-14 (portée : 200m , précision :  $\pm 1\text{mm}$ )
- 1x Pied d'appareil photo Feisol CT-3301
- 1x Tête pendulaire Benro GH3 pour y fixer une planche maintenant en place le télémètre et l'antenne GPS de la base
- 1x Table pour y placer le rover, la table sera déplacée à chaque fois (faite avec 2 tréteaux et une planche afin de pouvoir ajuster la hauteur de l'antenne GPS sur le laser)
- 1x Groupe électrogène pour recharger les ordinateurs pendant les mesures
- Rallonges électriques pour les ordinateurs

### 4.5.2 Déroulement de la mesure

La première étape est de fixer la position de la base, elle a mis environ 20 minutes pour arriver à une précision de 1 mètre. Il faut ensuite positionner l'antenne du rover à 1 mètre grâce au télémètre, puis enregistrer les coordonnées pendant au moins 20 min. On peut ensuite déplacer le plateau et les tréteaux pour arriver à la nouvelle distance souhaitée. Les distances mesurées sont 1, 2, 5, 10, 15 et 20 mètre.

### 4.5.3 Analyses des mesures

L'analyse des mesures va se dérouler de manière identique à celle effectuée dans le jardin. La figure 42 montre la disposition des mesures

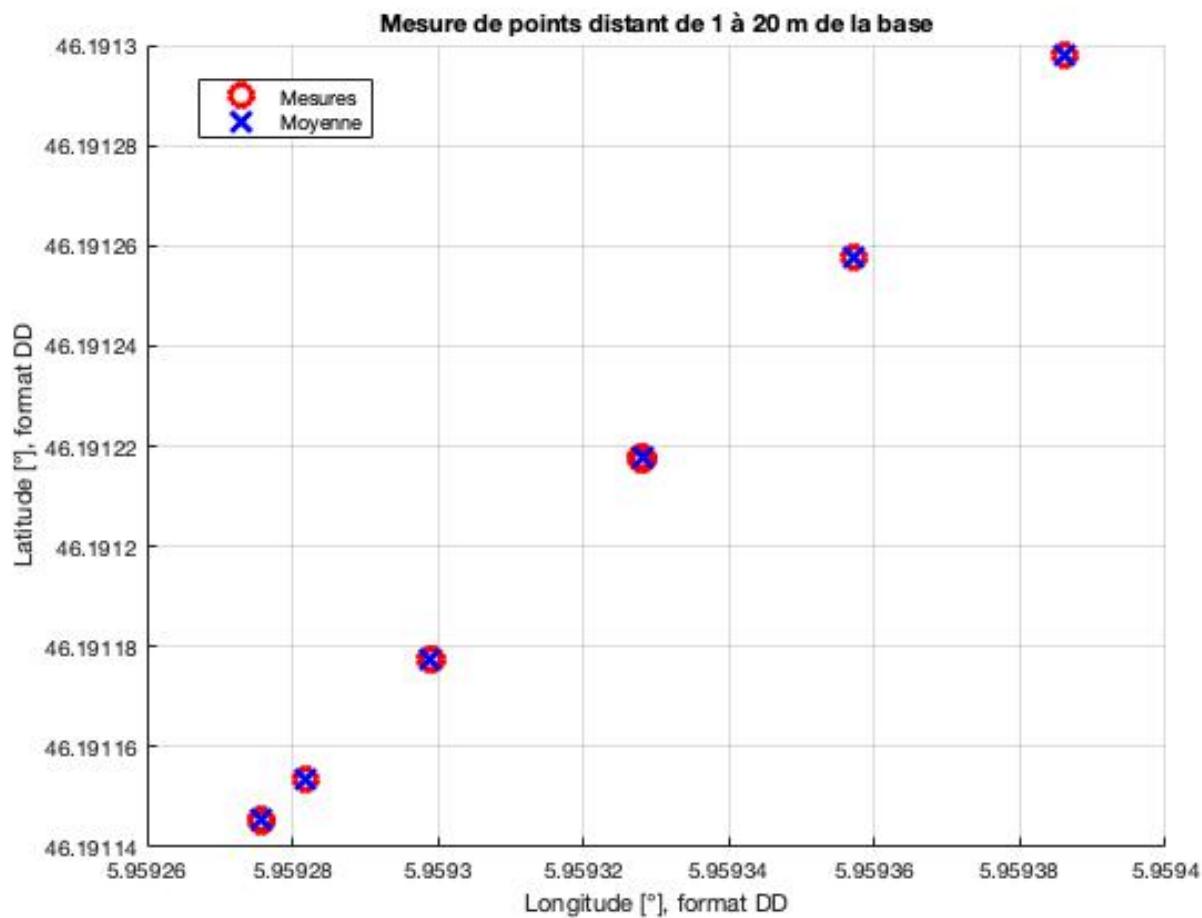


FIGURE 42 – Disposition des points de mesures. Distance depuis la base : 1, 2, 5, 10, 15 et 20 mètres

Comme pour la mesure précédente, ce graphique nous indique que les points de chaque mesure sont bien groupés. Il n'y a donc pas eu de perte de signal ou de perturbation majeure lors des mesures. Le graphique à la figure 43 montre la dispersion des mesures, les tailles de chaque axe sont conservées afin de pouvoir comparer les mesures.

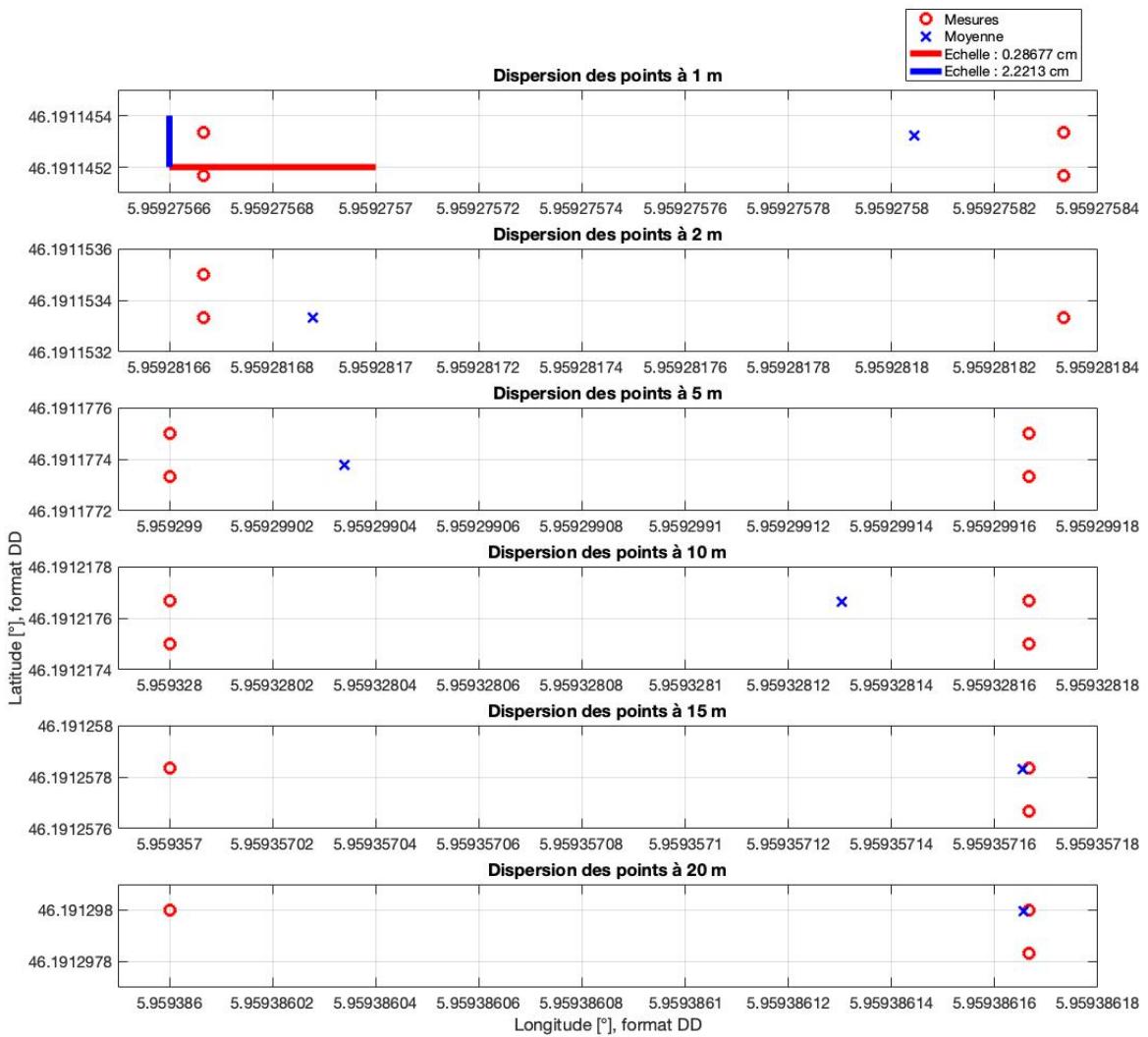


FIGURE 43 – Répartition des points de mesures

Cette figure permet de voir que les mesures semblent effectivement beaucoup moins dispersées que lors de la même mesure dans le jardin. Contrairement à l'ancienne mesure où les points pouvaient se trouver à 20 positions différentes, ici le nombre maximum de positions est de 4 et forme tout simplement un rectangle autour de la position moyenne, ce qui montre déjà un gain de précision important sur les mesures les plus lointaines de la base. L'histogramme à la figure 44 confirme cette observation.

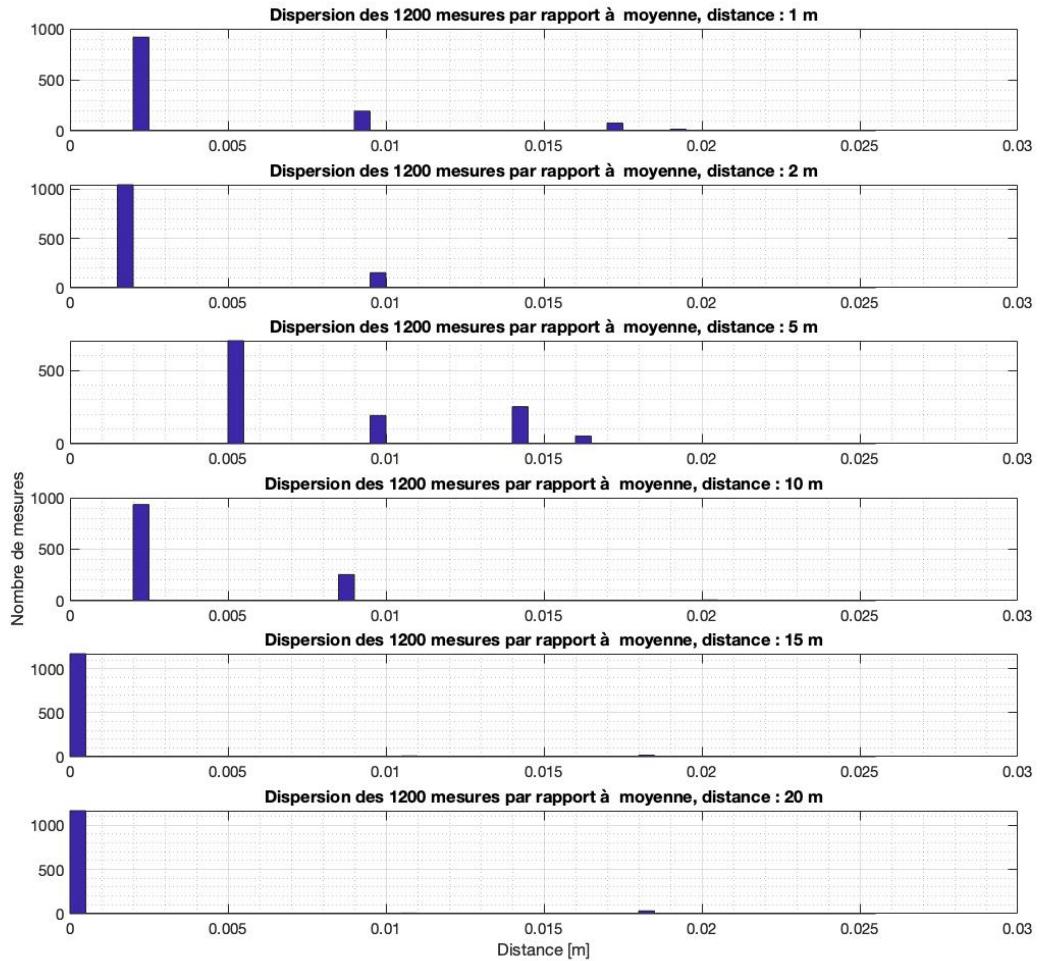


FIGURE 44 – Dispersion des mesures par rapport à la moyenne

L'histogramme montre bien que lorsque l'on fait des acquisitions dans des endroits bien dégagés, la grande majorité des mesures sont à moins d'un centimètre de la position moyenne alors que cette valeur se trouvait plutôt autour de 2 centimètres pour la mesure dans le jardin. Le tableau 45 indique l'écart type ainsi que la distances des mesures les plus lointaines.

| distance base/rover [m] | écart type [cm] | distance max [cm] | rayon de 2,5 cm |
|-------------------------|-----------------|-------------------|-----------------|
| 1                       | 0,45            | 1,9               | 100 %           |
| 2                       | 0,27            | 1,9               | 100 %           |
| 5                       | 0,4             | 1,6               | 100 %           |
| 10                      | 0,31            | 2                 | 100 %           |
| 15                      | 0,24            | 1,8               | 100 %           |
| 20                      | 0,3             | 1,8               | 100 %           |

FIGURE 45 – Tableau récapitulatif de la précision des distances par coordonnée GPS

Si l'on compare les tableaux 39 et 45, on constate effectivement que les écarts types sont beaucoup plus stables lorsque la distance augmente et que l'antenne du rover reste dans une zone sans obstacle. Cette constatation peut également être faite avec les distances maximum entre la moyenne et les points les plus éloignés. Alors que certains points se trouvaient à plus de 5,5 cm pour une distance de 20 mètre entre la base et le rover, la distance n'est maintenant plus que de 1,8 cm. On note que comme tous les points se trouvent à des distances de 2 cm ou moins de la moyenne, 100 % des points sont dans un cercle de 2,5 cm de rayon, alors que u-blox n'en annonçait que 50%.

Le dernier graphique de cette mesure c'est celui qui indique les distances entre les différents points de mesures et c'est celui qui possédait les plus grosses erreurs lors de la mesure faite dans mon jardin. Les mesures les plus impactées, probablement à cause des obstacles environnant, étaient celles qui se trouvaient éloignées de la base. Le graphique 46 représente l'histogramme des distances calculées entre chaque point de mesures lorsqu'il n'y a plus d'obstacle autour.

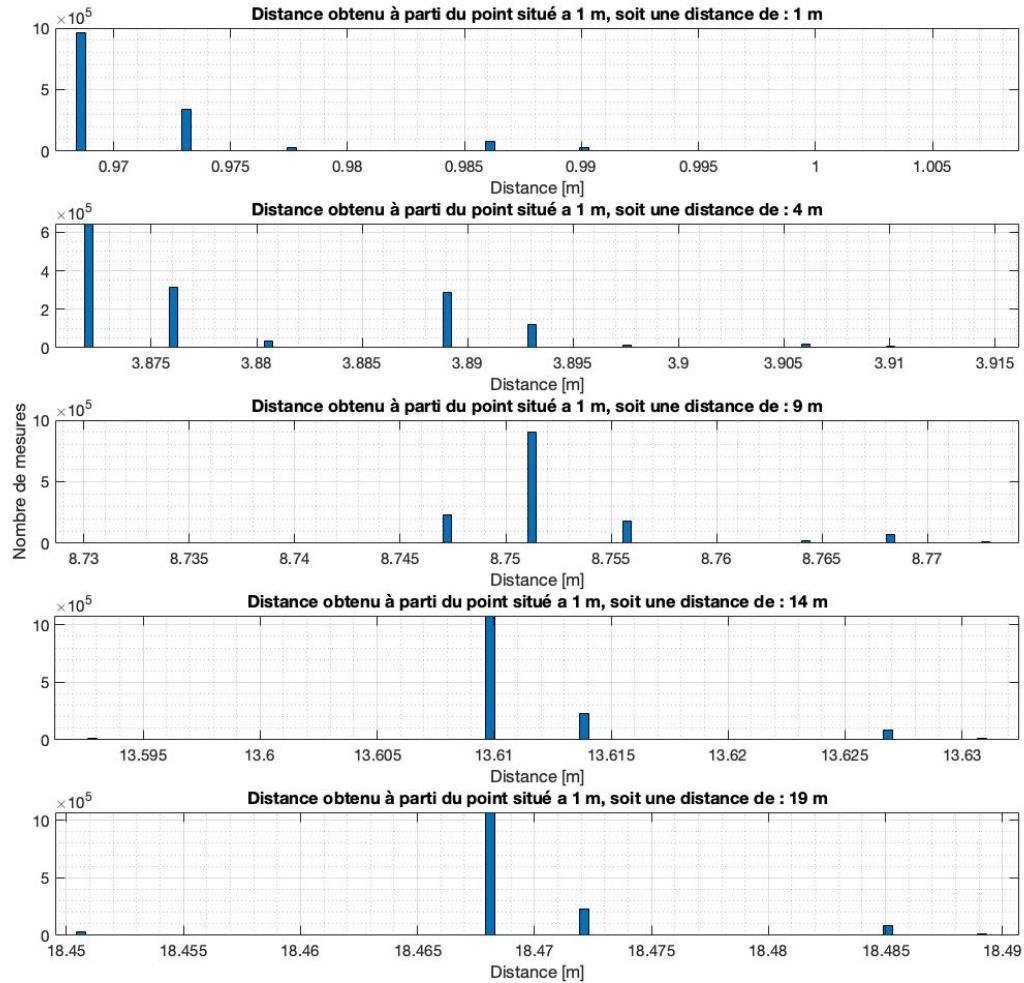


FIGURE 46 – Distances obtenues à partir des points se trouvant à 1 m vers les points se trouvant à 2,5,10,15 et 20 m

On peut constater que comme les mesures sont moins dispersées autour des moyennes que lors de la mesure dans le jardin, les distances entre les points de mesures sont plus concentrées autour de la moyenne. Mais il subsiste une erreur beaucoup trop importante. Cette erreur n'est apparemment pas dû aux obstacles des signaux comme j'en avais fait l'hypothèse lorsque j'avais fait l'analyse de cette même mesure mais faite dans le jardin. Le tableau 47 permet de chiffrer ces erreurs et de les comparer avec le tableau 41.

| distance rover/rover [m] | distance moyenne [m] | erreur [m] | erreur [%] |
|--------------------------|----------------------|------------|------------|
| 1                        | 0,97                 | 0,028      | 2,83 %     |
| 4                        | 3,88                 | 0,121      | 3,02 %     |
| 9                        | 8,75                 | 0,248      | 2,75 %     |
| 14                       | 13,61                | 0,39       | 2,77 %     |
| 19                       | 18,47                | 0,53       | 2,79 %     |

FIGURE 47 – Tableau récapitulatif de la précision des distances par coordonnée GPS

En comparant les deux tableaux, on peut constater que les pourcentages d'erreur relativ aux distances mesurées étaient comprises entre 4,25 et 5%, sauf pour la distance de 9 m qui n'avait aucune erreur. Elles se situent maintenant entre 2,75 et 3,02%. Ce sont certes des baisses significatives d'erreur mais l'erreur restante masque complètement la précision des modules GPS RTK car si mesurer 97 cm au lieu de 1 mètres par GPS RTK peut sembler vraiment bien, mesurer 18,47 mètres au lieu de 19 en champ dégagé n'est clairement pas la précision attendue. L'erreur maximum devrait se trouver à environ 3,7 cm, ce qui correspond au point le plus éloigné de la moyenne de 1 m (1,9 cm) et le point le plus éloignée de la moyenne de 20 m (1,8 cm). Il doit donc très probablement subsister une erreur dans la procédure, mais je ne suis pas en mesure de la déterminer à l'heure actuelle.

#### 4.6 Correction de la formule de calcul de distance

Après avoir constaté que des erreurs subsistaient dans les mesures, j'ai cherché de nouvelles explications afin de les justifier. J'ai fini par trouver une erreur qui se trouve directement dans la formule pour calculer les distances. Cette formule est la formule 2 que je vais rappeler ci dessous.

$$D = 60 \cdot \arccos(\sin(Lat_d) \cdot \sin(Lat_a) + \cos(Lat_d) \cdot \cos(Lat_a) \cdot \cos(Lon_a - Lon_d)) \cdot 1851.85 \quad (3)$$

Cette équation n'est pas fausse en soit, mais son utilisation dans matlab avec des coordonnées GPS nécessite d'utiliser les fonctions sind, cosd et acosd car tous les angles des coordonnées terrestres sont en degré et non en radian. Cette erreur n'a pas été facile à trouver car les résultats obtenus avec l'équation erronée étaient très proches des valeurs attendues et comme il y avait d'autres imprécisions notamment dues aux obstacles dans le jardin, cette erreur n'a pu être détectée qu'une fois que les mesures en champ dégagé ont été faites. La formule à utiliser dans matlab est donc celle de l'équation 4

$$D = 60 \cdot \text{acosd}(\text{sind}(Lat_d) \cdot \text{sind}(Lat_a) + \text{cosd}(Lat_d) \cdot \text{cosd}(Lat_a) \cdot \text{cosd}(Lon_a - Lon_d)) \cdot 1851.85 \quad (4)$$

Mais cette erreur cause un problème un peu inattendu. Alors que les angles comme 0,1 radian sont maintenant correctement interprétés à 0,1 degré, la taille des angles

à fortement été réduite.<sup>7</sup> De ce fait, le nombre de décimales nécessaires pour les calculs a augmenté.

Les plus petits angles utilisés dans les mesures étant des valeurs de  $10^{-8}[\text{deg}]$  comme sur la longitude du graphique 43, mais matlab n'arrive plus à calculer les distances entre les points à cause d'erreurs d'arrondis. Les grandes distances ne sont pas affectées par ce problème car les angles sont plus grands, mais cette formule rend impossible le calcul de distances entre les points d'une même mesure afin d'en connaître l'écart type ou de faire un histogramme de la dispersion des mesures par exemple.

Le graphique 48 permet d'illustrer le problème. Le premier graphique étant le même que le premier de la figure 43 et la seule grandeur qui varie entre les graphiques est l'indicateur de taille de l'échelle des longitude, la grille reste constante. Les 3 premiers graphiques utilisent la formule 3 qui est erronée et les 3 autres utilisent la formule 4 qui est la version corrigée.

---

7. 1 radian étant beaucoup plus grand que 1 degrés

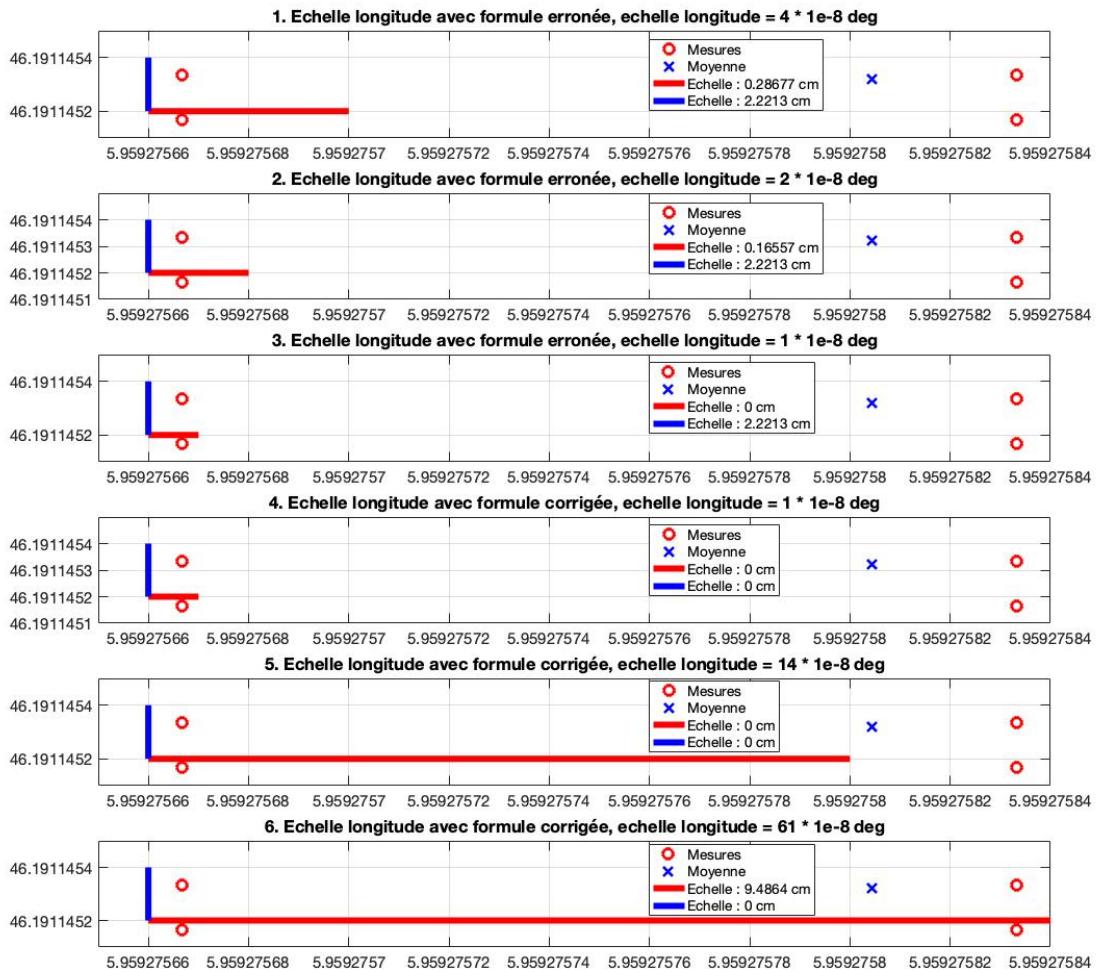


FIGURE 48 – Comparaison des distances calculables dans matlab avant et après la correction de formule

Le premier graphique montre que l'échelle des longitudes est représentée sur  $4 \cdot 10^{-8}[\text{deg}]$  et que cette distance correspond à 0,28677 cm. Le second graphique montre que si l'on divise par deux la distance à mesurer, matlab indique une distance qui n'est pas la moitié de la distance initiale. Comme les distances sont mesurées à la même latitude, la valeur obtenue devrait être exactement la moitié de la valeur initiale, l'erreur provient donc des arrondis de calculs dans matlab. Il est important de noter que bien que la distance du premier graphique est moins soumise aux erreurs d'arrondi, elle n'y échappe pas pour autant, et comme elle n'est pas possible à quantifier car aucune des formules présentées ne permet de calculer d'aussi petites distances, le calcul d'erreur et de pourcentage d'erreur entre ces deux valeurs n'a que peu de sens.

Mais lorsque l'on regarde le troisième graphique, les erreurs deviennent extrêmement

problématiques. L'échelle des longitudes du troisième graphique est censée indiquer la distance que représente  $1 \cdot 10^{-8}[\text{deg}]$ , mais les erreurs de calculs sont tellement importantes que matlab indique que la distance de l'échelle des longitudes est de 0.

Les trois premiers graphiques montrent que les erreurs avec la formule 3 empêchent de calculer des distances de l'ordre de  $10^{-8}[\text{deg}]$  qui correspondent pourtant aux ordres de grandeurs de la précision nécessaire pour calculer les distances entre des points d'une même mesure.

Les 3 graphiques suivants montrent les résultats des distances obtenues avec la formule corrigée. Le premier indique que, comme avec la formule erronée, il n'est pas possible d'obtenir la distance entre deux points distant de  $10^{-8}[\text{deg}]$ . Le second montre pourquoi il n'est pas possible de calculer d'écart type ou de faire l'histogramme des distances des mesures car les distances des points les plus lointains de la moyenne ont toujours une valeur de 0. Finalement, le dernier graphique montre à partir de quel angle il est possible de calculer une distance, et cet angle est de  $61 \cdot 10^{-8}[\text{deg}]$ , soit près de 3 fois la longueur complète du graphique.

Suite à cet exemple, il est clair qu'il n'est pas possible de calculer les petites distances entre les différentes mesures car les angles sont trop petits et matlab fait trop d'arrondis, mais la formule corrigée doit permettre de calculer les distances de 1 à 20 mètres qui séparent les moyennes de chaque point de mesure. Le tableau 49 compare les distances moyennes des deux mesures (avec obstacles autour et sans) avec la formule erronée et la formule corrigée.

|                     | Dist avec obstacles [m] |          | Dist sans obstacles [m] |          |
|---------------------|-------------------------|----------|-------------------------|----------|
| Formule<br>Dist [m] | erronée                 | corrigée | erronée                 | corrigée |
| 1                   | 0,95                    | 0,99     | 0,97                    | 1,00     |
| 4                   | 3,83                    | 4,00     | 3,88                    | 3,98     |
| 9                   | 9,00                    | 9,40     | 8,75                    | 8,99     |
| 14                  | 13,36                   | 13,97    | 13,61                   | 13,98    |
| 19                  | 18,12                   | 18,94    | 18,47                   | 18,97    |

FIGURE 49 – Tableau de comparaison des distances en utilisant la formule erronée et corrigée

Ce tableau montre que grâce à la formule corrigée, les erreurs de distances obtenues entre deux moyennes en terrain dégagé sont de 3 cm maximum. On peut également remarquer que mesurer des distances avec des obstacles autour peut donner de très bons résultats avec 6 cm d'erreur pour une distance de 20 mètres, sauf pour la mesure de 9 mètres qui elle, cette fois ci, est vraiment erronée. Ce résultat semble cohérent car la mesure de 9 mètres avait été faite alors qu'une branche se trouvait 1 mètre au dessus, ce qui l'a complètement faussée.

La formule corrigée permet donc de calculer correctement des distances, mais attention car elle est également soumise aux erreurs d'arrondi. Le tableau 49 prouve que

la formule marche correctement pour des distances de 1 mètre, pour des distances plus petites, il faudra s'assurer que les arrondis ne viennent pas entacher la précision souhaitée.

## 5 Hardware

Cette section contient le détail de chaque partie du circuit. La partie schématique est composée à chaque fois de la partie du schéma qui réalise la fonction, des explications pour comprendre comment marche cette partie du système et quelques indications sur les choix techniques qui ont été fait.

Le but recherché lors de la réalisation du routage est d'obtenir un PCB compact, où les antennes sont correctement positionnées pour ne pas se déranger entre elles et où les alimentations sont correctement réparties afin d'assurer leurs stabilités lors des appels de courants.

### 5.1 Schématique

#### 5.1.1 Microcontrôleur et bluetooth

Pour le microcontrôleur, le choix a d'abord été fait sur le NRF52832 de Nordic Semiconductor. Ce microncontrôleur a l'avantage d'intégrer une communication bluetooth et d'être disponible dans un package QFN, ce qui permet de le souder à la main. Mais ce microcontrôleur ne possède pas suffisamment de GPIO, il a donc fallu en utiliser un, un peu plus gros, le NRF52840. Ce dernier possède les mêmes caractéristiques que le précédent mais n'est disponible que dans un package aQFN ce qui empêche de le souder à la main, il faudra donc faire monter le PCB. La figure 50 montre la schématique de cet IC.

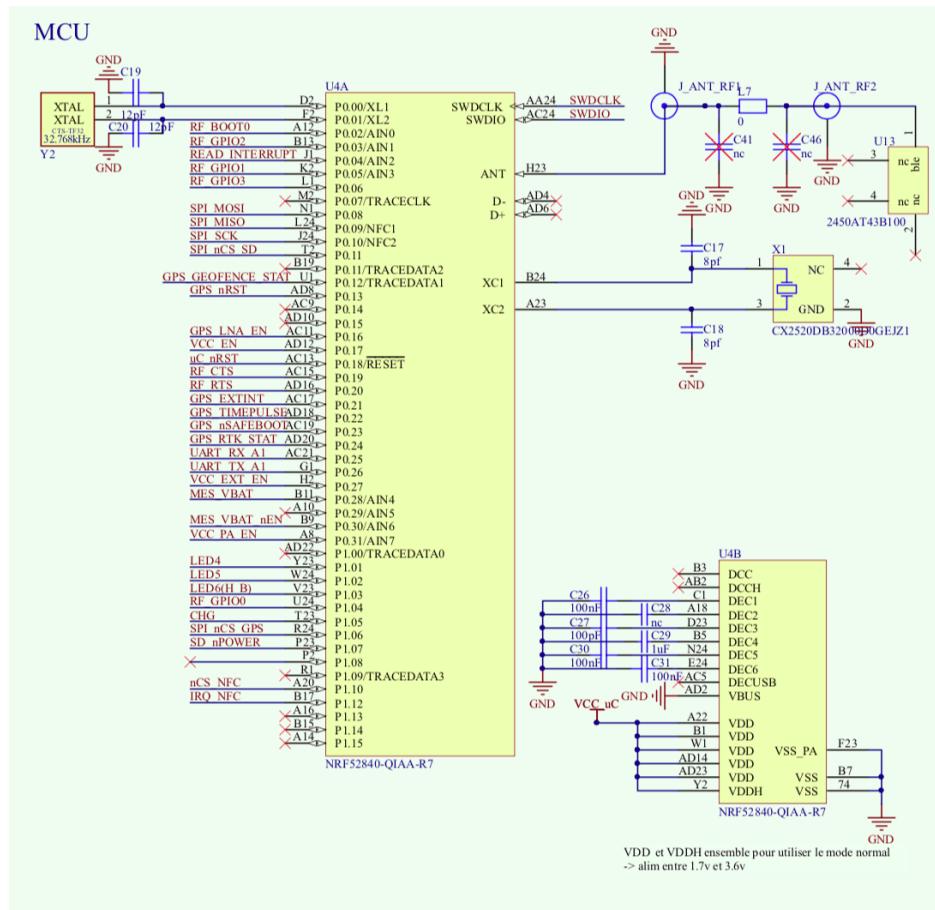


FIGURE 50 – Schématique du microcontrôleur et du bluetooth

Sur ce schéma, le microcontrôleur est découpé en deux parties (U4A et U4B) pour faciliter la lecture du schéma. La partie U4A est destinée aux GPIO et la partie U4A ne contient que l'alimentation et le découplage du NRF52840. La partie antenne bluetooth se trouve sur la pin ANT du microcontrôleur. On y retrouve deux connecteurs UFL, un montage en pi et l'antenne.

### 5.1.2 GPS

Le chip choisi pour gérer le GPS est le NEO-M8P de ublox. Cet IC a comme avantage de pourvoir servir soit de base, soit de rover et peut fournir une position avec une précision maximum de 2.5 cm grâce à l'algorithme RTK [26]. Mais la puce GPS seule ne suffit pas. Comme le montre la figure 51, le signal GPS doit passer par plusieurs autres chips avant d'être reçu et de pouvoir être interprété.

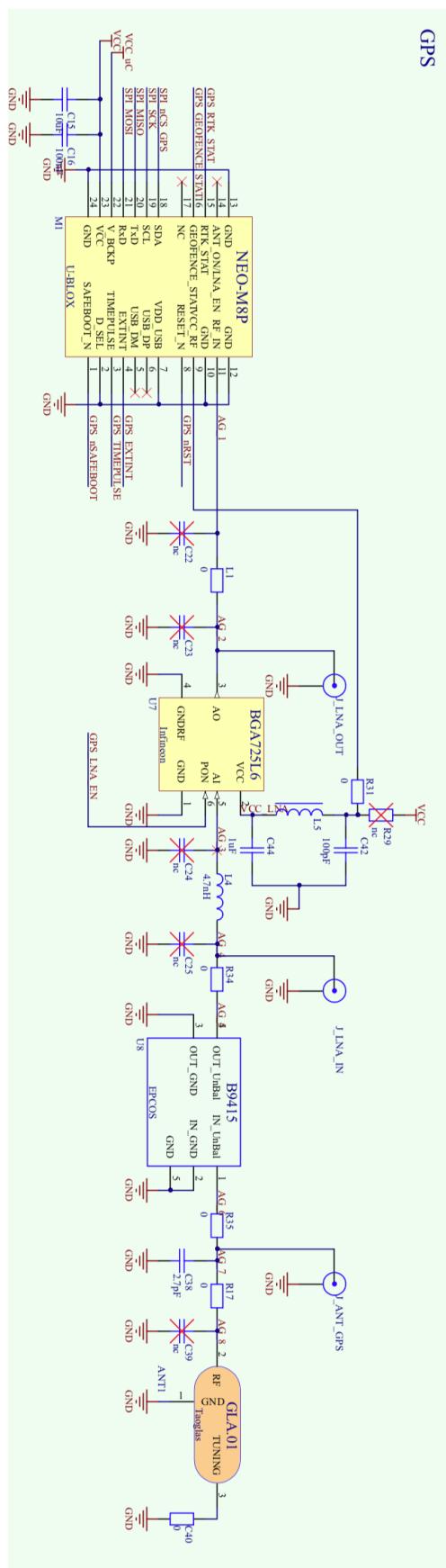


FIGURE 51 – Schématique du GPS

Le premier chip de la chaîne est le NEO-M8P, qui gère les informations GPS et communique en SPI avec le microcontrôleur. Le second circuit (BGA725L6) est un LNA. Le B9415 est un filtre adapté aux fréquences GPS et le GLA.01 est l'antenne. Un montage en pi a été prévu entre chaque chip s'il fallait ajuster l'impédance.

### 5.1.3 Communication radio

La communication radio provient d'un module développé par la HEIG-VD et permet au GPS de recevoir les facteurs de correction de la base pour utiliser le RTK. La figure 52 montre la schématique du module radio.

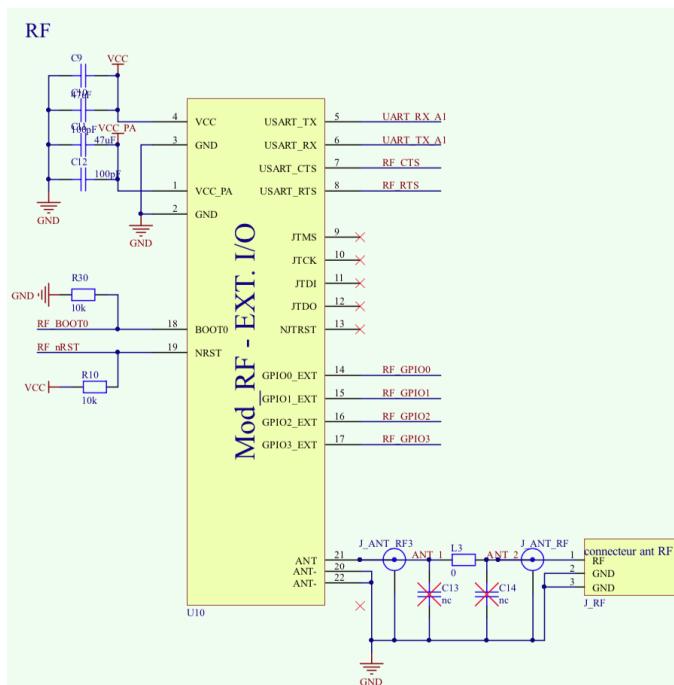


FIGURE 52 – Schématique du module radio

Les deux connecteurs UFL en bas et la cellule en pi servent à adapter l'antenne RF. L'antenne est une antenne PIFA réalisée sur un pcb, la conception est détaillée dans le chapitre 5.2.4

### 5.1.4 Alimentation

La partie alimentation est constituée de 3 régulateurs de tension et du circuit pour la gestion de la batterie. Le schéma de l'alimentation se trouve à la figure 53

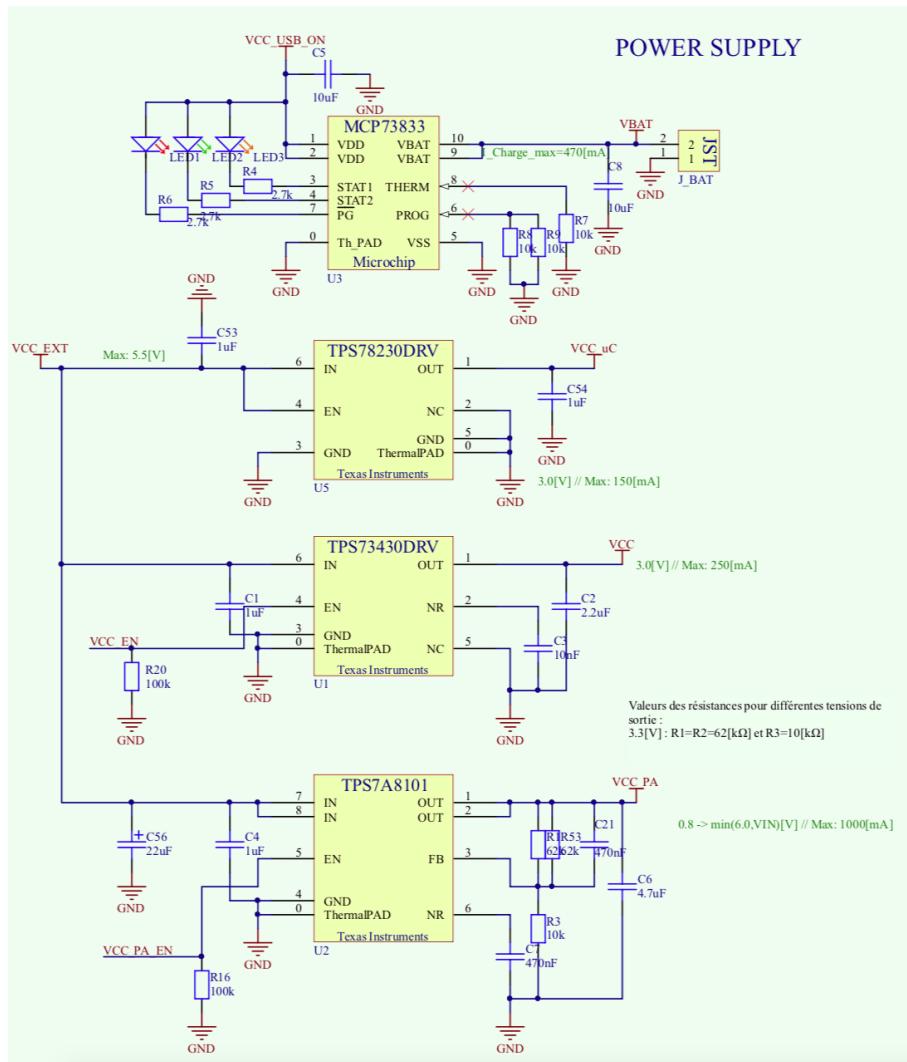


FIGURE 53 – Schématique de l'alimentation

Le chip qui gère la batterie possède 3 leds. La rouge s'allume lorsque le circuit est allumé, la verte s'allume lorsque la batterie est chargée et la orange s'allume lorsque la batterie charge. Ces leds ne fonctionnent que pour la charge filaire qui est uniquement présente pour le développement de la carte. La charge de la batterie se fait avec un courant de 200mA

En dessous du gestionnaire de batterie, il y a 3 régulateurs. Le premier génère une tension de 3V et peut fournir 150mA pour alimenter le microcontrôleur et le chip pour la communication NFC. Le second régulateur est également un régulateur 3V mais 250mA. Il alimente une partie du circuit RF, toute la partie GPS et la carte SD. Le dernier régulateur ajustable est fixé à 3.3V pour alimenter la partie puissance du circuit RF.

La puce GPS possède 2 entrées d'alimentation : une principale et une de secours au cas où la première ne suffit plus. Cet IC est donc alimenté par le VCC mais peut

utiliser le VCC du microcontrôleur si jamais le VCC ne suffisait pas.

### 5.1.5 Charge par induction

La solution utilisée pour recharger le système est une charge par induction. Cela facilite l'étanchéité du boîtier car il n'y a aucun port de connexion où l'eau pourrait entrer dans le boîtier. La figure 54 montre le schéma de la charge par induction.

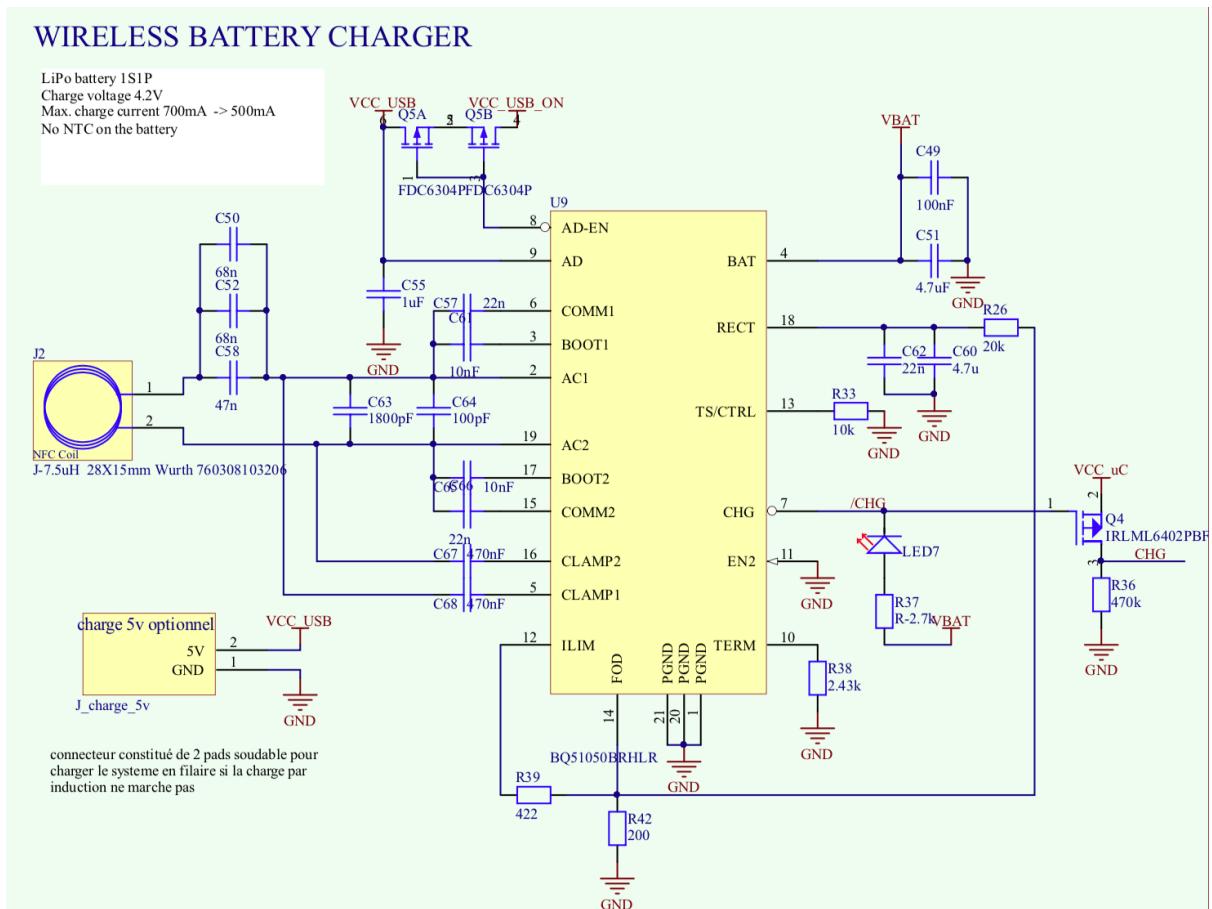


FIGURE 54 – Schématique de la charge par induction

Sur cette figure, J2 est la bobine de charge qui se trouve dans le boîtier pour charger le système. Le chip BQ5150B est le chip qui gère la charge sans fil, il ajuste et stabilise la tension pour la charge de la batterie. Il protège également la batterie des surtensions. Le circuit dispose d'une led qui indique lorsque le système se charge par induction. Le transistor Q4 permet d'indiquer au microcontrôleur que la batterie est en charge.

Ce chip permet également de charger la batterie par un câble. Cette solution est prévue pour faciliter le debug de la carte. Pour faire ceci, un connecteur supplémentaire a été ajouté, il s'agit de J\_charge\_5v. Ce connecteur est constitué de deux pads où il est possible de souder des fils dessus avec une tension de 5v pour charger

la batterie. Cette solution n'est plus accessible lorsque le circuit est dans un boîtier étanche mais permet lors des tests, de recharger la batterie sans être obligé de laisser le système sur un socle de recharge, ce qui facilite le débogage de la carte. L'IC priorise la charge filaire, si une tension est présente sur J\_charge\_5v , le chip va bloquer la charge par induction, puis mettre en conduction les deux transistors Q5A et Q5B afin de renvoyer la tension de charge sur le chip U3 qui se trouve dans la partie alimentation, c'est donc à lui de gérer la charge de la batterie.

### 5.1.6 ON/OFF et reset

Dans le but de maintenir l'étanchéité des circuits lorsqu'ils se trouvent dans un boîtier, les boutons choisis pour ce système sont en fait des capteurs magnétiques. Les figures 55 et 56 montrent comment sont réalisés les câblages de ces capteurs magnétiques.

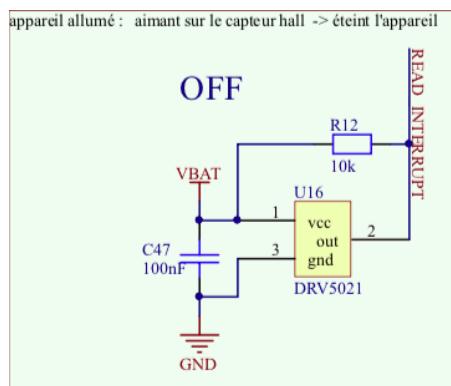


FIGURE 55 – Schématique pour éteindre le système

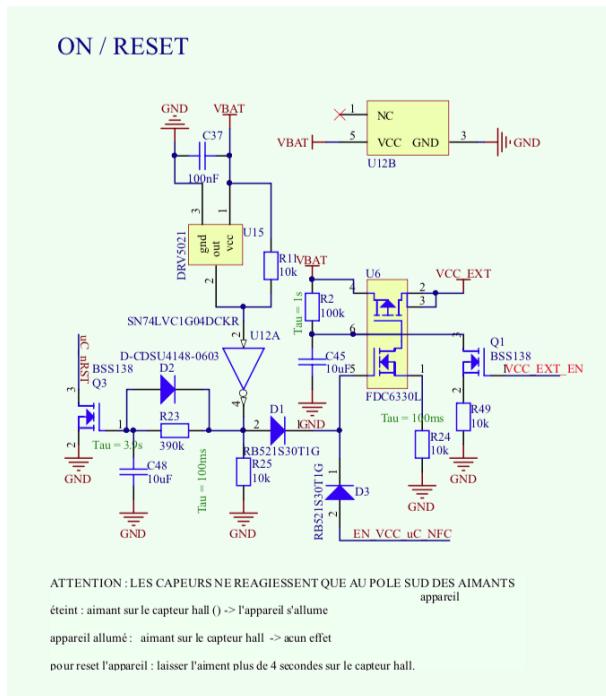


FIGURE 56 – Schématique pour allumer ou reset le système

Le capteur magnétique choisi est le DRV5021, c'est un capteur à effet hall qui est sensible aux pôles sud des aimants. Si l'on approche un aimant avec un champ suffisamment important, alors la sortie bascule à l'état bas et le reste tant que le champ est présent.

Dans le cas de la figure 55, cela permet de déclencher une interruption sur le microcontrôleur. Il sera alors capable de couper l'alimentation de tous les circuits grâce au transistor Q1 que l'on retrouve sur la partie droite de la figure 56.

Pour la figure 56, le capteur à effet hall est suivi d'un inverseur qui est alimenté directement par la batterie, ce qui a pour effet qu'un état haut apparaît en sortie de U12A lorsqu'un champ magnétique est présent. Cet état haut va directement sur la gate d'un des transistors de U6, ce qui a pour conséquence de mettre en conduction les deux transistors, le premier pilotant la gate du deuxième. La tension de la batterie se trouve alors connectée sur VCC\_EXT qui est la tension d'entrée des 3 régulateurs de la partie alimentation. Toute la carte est alimentée tant que le champ magnétique est présent sur le capteur magnétique. Le microcontrôleur doit ensuite prendre le relais et fixer son signal VCC\_EXT\_EN à l'état haut pour maintenir le transistor du haut du chip U6 conducteur, ce qui permet de maintenir toute la carte alimentée lorsque l'aimant est enlevé et que le champ magnétique s'arrête.

Pour obtenir un reset, il faut laisser un aimant sur le capteur qui gère l'allumage pendant plus de 3.9 secondes. Un reset se fait grâce au condensateur C48 qui va se charger et faire conduire le transistor afin de déclencher le reset du microcontrôleur.

### 5.1.7 NFC

La communication NFC est utilisée pour vérifier les informations d'un rover (vérification du numéro de dossard associer au coureur avant le début d'une course par exemple). Le NFC est également utilisé pour reprogrammer le micro-contrôleur ou le module radio une fois que le système a été rendu étanche car il n'y a plus de connecteur accessible. Le NFC permet en plus d'allumer le système. Pour rendre le NFC aussi polyvalent, il a fallu passer par un chip dédié au NFC car certaines fonctions ne sont pas disponibles sur la partie NFC incorporée dans le NRF52840. La gestion du NFC est faite grâce au chip AS3953A. La figure 57 montre le schéma de la partie NFC.

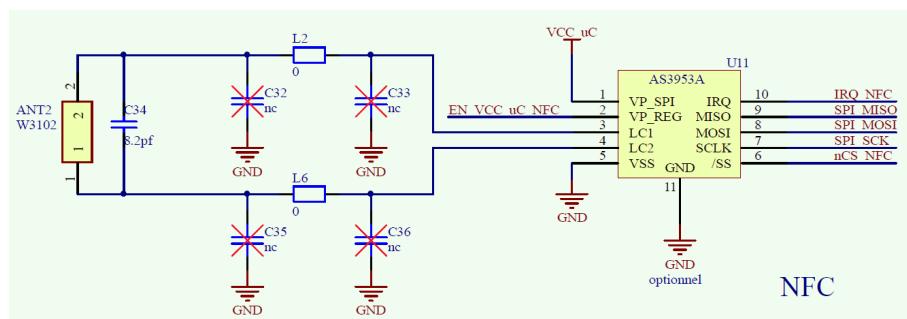


FIGURE 57 – Schématique du NFC

Sur le schéma, on retrouve le AS3953A pour gérer le NFC ainsi que l'antenne avec un montage en pi en cas de nécessité pour adapter l'antenne. Le signal EN\_VCC\_uC\_NFC est un signal fait pour réveiller le système à partir d'une connexion NFC. Il permet d'allumer le circuit en activant provisoirement les transistors du chip U6 qui se trouvent dans la partie on et reset.

### 5.1.8 SD ou MMC

Un emplacement pour une carte micro SD est disponible pour y stocker l'historique des points GPS. La figure 58 montre le schéma du connecteur pour carte micro SD.

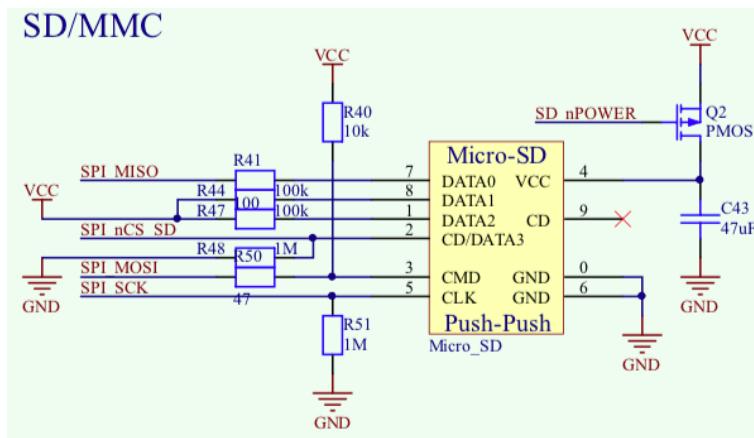


FIGURE 58 – Schématique du connecteur micro SD

La communication entre la carte micro SD et le microcontrôleur ne nécessite pas de chip intermédiaire mais utilise le protocole SPI. Un signal permet au microcontrôleur de couper l'alimentation de la carte mémoire si besoin.

### 5.1.9 Connecteurs de programmation et leds

Deux connecteurs supplémentaires sont présents sur la carte. Le jtag qui est présent pour programmer le microcontrôleur et le FTDI afin de pouvoir lire le bus UART et vérifier les données entre le microcontrôleur et le module RF.

Comme les leds sont connectées au microcontrôleur, elle peuvent indiquer des informations de batterie, de qualité des signaux GPS ou le fonctionnement du RTK en fonction du code qui sera utilisé.

Les figures 59 et 60 montrent les schémas utilisés pour les connecteurs et pour les leds.

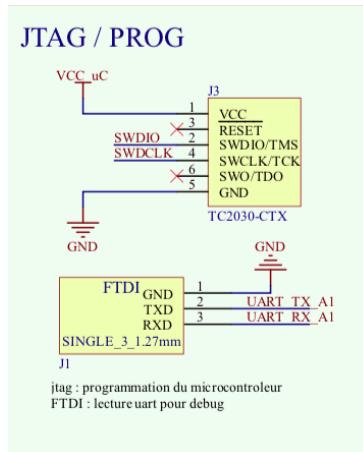


FIGURE 59 – Schématique des connecteurs

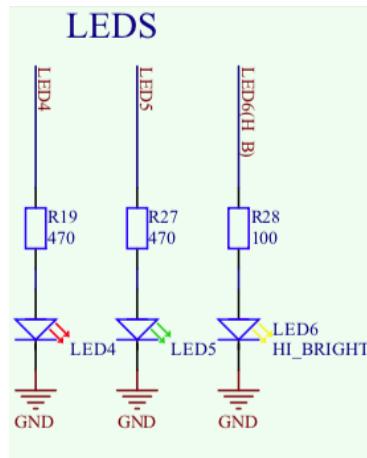


FIGURE 60 – Schématique des leds

## 5.2 Layout

Cette section contient des informations sur la réalisation du PCB. Le PCB a des dimensions de 60 mm x 100 mm et est constitué de 4 couches. La figure 61 montre le layer stack du PCB.

| # | Name           | Material      | Type        | Weight | Thickness | Dk  | Df |
|---|----------------|---------------|-------------|--------|-----------|-----|----|
| ✓ | Top Overlay    |               | Overlay     |        |           |     |    |
| ✓ | Top Solder     | Solder Resist | Solder Mask |        | 0,01mm    | 3,5 |    |
| ✓ | 1 Top Layer    |               | Signal      | 1/2oz  | 0,018mm   |     |    |
| ✓ | Dielectric 2   | PP-006        | Prepreg     |        | 0,36mm    | 3,9 |    |
| ✓ | 2 gnd          |               | Signal      | 1oz    | 0,035mm   |     |    |
| ✓ | Dielectric 1   | FR-4          | Dielectric  |        | 0,71mm    | 3,9 |    |
| ✓ | 3 vcc          |               | Signal      | 1oz    | 0,035mm   |     |    |
| ✓ | Dielectric 3   | PP-006        | Prepreg     |        | 0,36mm    | 3,9 |    |
| ✓ | 4 Bottom Layer |               | Signal      | 1/2oz  | 0,018mm   |     |    |
| ✓ | Bottom Solder  | Solder Resist | Solder Mask |        | 0,01mm    | 3,5 |    |
| ✓ | Bottom Overlay |               | Overlay     |        |           |     |    |

FIGURE 61 – Layer stack du PCB

Les couches 1 et 4 sont les couches où sont routés les signaux. La couche 2 est une couche de GND et la couche 3 est dédiée aux alimentations.

### 5.2.1 Disposition des composants

Presque tous les composants de ce PCB se trouvent sur la face top du circuit afin de faciliter le montage et d'en diminuer le coût. Seul le module radio se trouve en bottom avec ses condensateurs de découplage. Les figures 62 et 63 représentent le routage visible sur les couches top et bottom. La figure 64 montre le routage sans les plans d'alimentation ni le plan de masse.

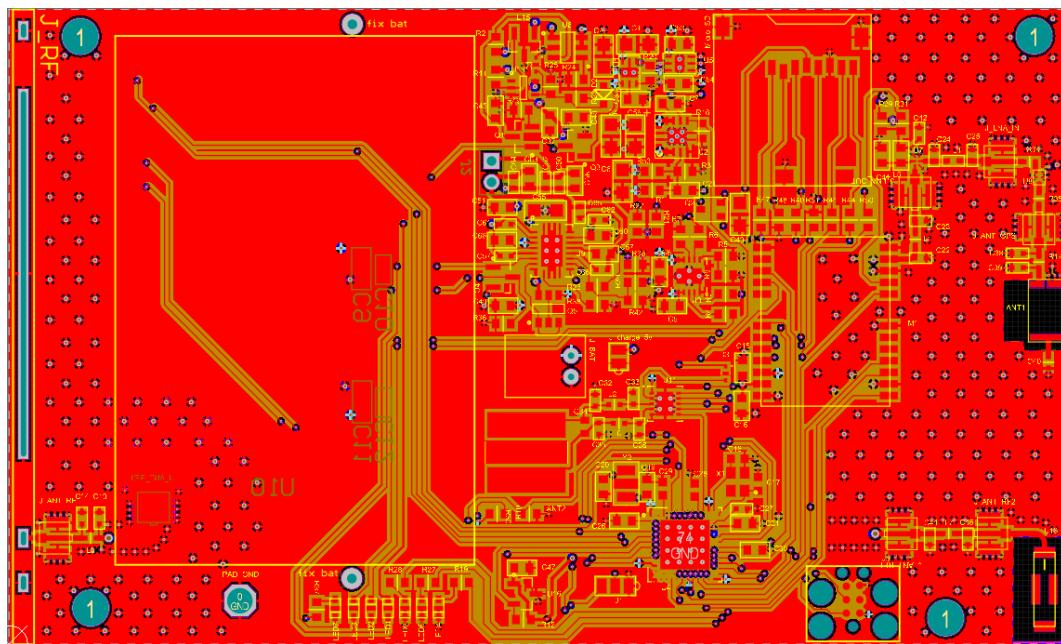


FIGURE 62 – Routage de la couche top

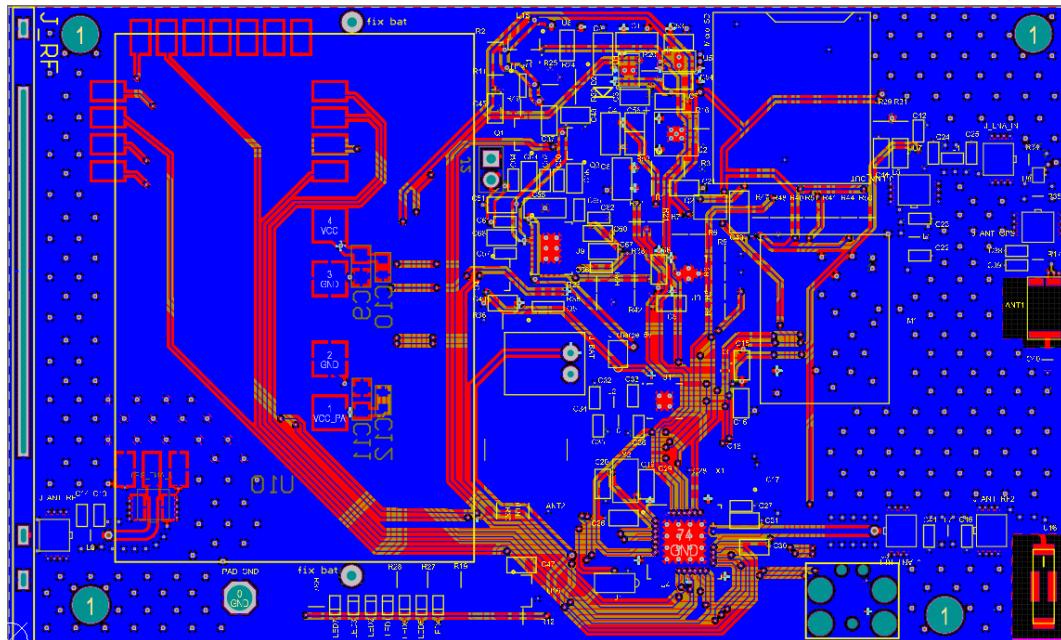


FIGURE 63 – Routage de la couche bottom

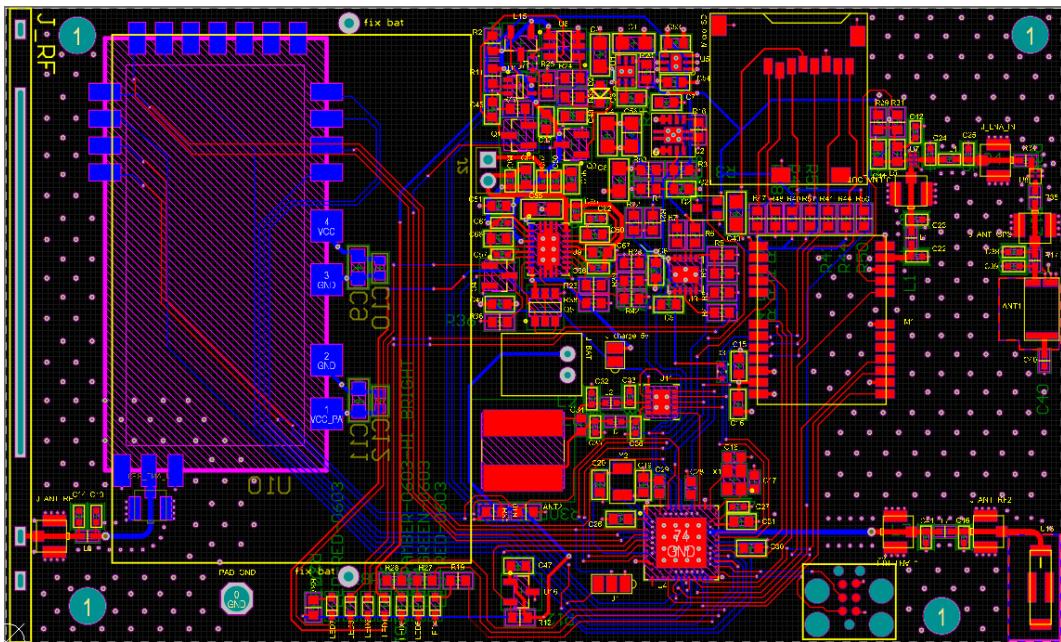


FIGURE 64 – Routage des couches top et bottom sans le plan de masse ni les plans d'alimentation

Sur toute la hauteur de la partie gauche se trouve un connecteur pour accueillir l'antenne RF qui est utilisée par le module pour recevoir les facteurs de correction provenant de la base pour permettre l'utilisation du RTK. L'énorme partie vide juste à coté sert pour fixer la batterie. Le fait que la batterie soit juste à coté de l'antenne permet d'augmenter la capacité de l'antenne afin de contrer l'effet inductif et obtenir une impédance plus proche de  $50 \Omega$ .

Sur la partie droite, il y a le NEO-M8P pour le GPS, avec toute la chaîne de composants pour traiter le signal reçu des satellites. L'antenne GPS se trouve à mi-hauteur sur le bord droit du PCB et l'antenne bluetooth se trouve dans le coin inférieur droit.

Dans la partie centrale du circuit il y tout le reste de l'électronique. C'est à dire la partie alimentation, la charge par induction, le microcontrôleur et le NFC. L'antenne NFC et la charge par induction ont plutôt intérêt à être centré sur la carte, contrairement aux autres antennes. Comme la fonction de charge par induction implique de poser l'appareil sur un pad, le plus logique est que la bobine soit centrée afin de simplement poser l'appareil au centre du pad et ne pas avoir à excentrer l'appareil pour qu'il charge. Pour les mêmes raisons, il est plus judicieux de mettre le NFC au centre car cette fonction nécessite de poser deux appareils l'un sur l'autre.

Les deux capteurs magnétiques pour allumer, éteindre et reset la carte se trouvent à droite de la batterie. Le ON / reset se trouve en haut, et le OFF se trouve en bas, à droite des leds.

### 5.2.2 Routage des antennes

Parmi toutes les antennes qui figurent sur la carte, les antennes GPS, RF et bluetooth sont les plus sensibles aux perturbations. Afin d'avoir les meilleures performances possibles, chaque signal provenant de ces antennes est isolé le plus possible des autres et est routé de la manière la plus directe possible. Chaque signal est entouré par une protection de via (via shielding) qui permet de protéger la piste des perturbations EM. Afin de connaître à quelle distance peuvent figurer ces via, il faut déjà connaître la distance d'isolation qui doit se trouver autour de la piste. En entrant les paramètres d'épaisseur, de conductivité et de permittivité du PCB, les dimensions de la piste et la fréquence du signal dans Linecalc, le logiciel est capable de calculer une isolation constante à laisser autour de la piste pour un fonctionnement optimal. La figure 65 montre le résultat obtenu.

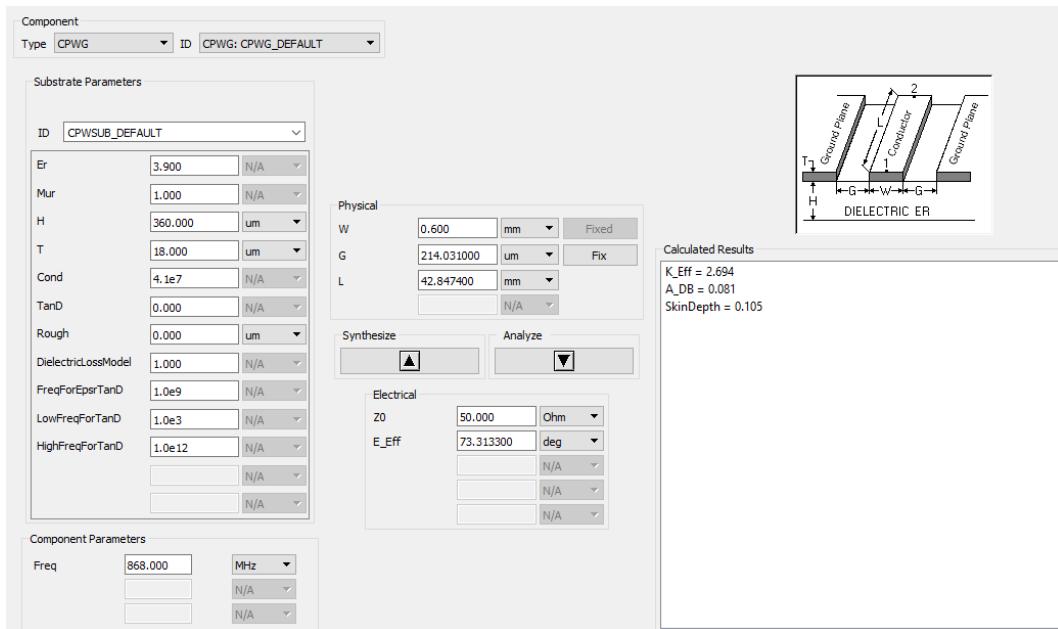


FIGURE 65 – Configuration de Linecalc pour l'antenne 868 MHz

Linecalc indique que pour une fréquence de 868 MHz et une impédance de  $50 \Omega$ , l'isolation doit être de 0.214 mm autour de la piste. Une fois que cette isolation est fixée sur altium, il est possible d'ajouter les via autour de la piste à une distance suffisante pour conserver l'isolation constante.

En plus de la protection autour de la piste avec des via, d'autres via ont été placées régulièrement sur la zone autour de l'antenne (via stitching) afin d'avoir le meilleur plan de masse possible autour de l'antenne. Comme cette dernière utilise le plan de masse autour comme capacité pour rayonner, il faut avoir un plan de masse le plus grand et le plus stable possible.

### 5.2.3 Routage des alimentations

Les trois régulateurs de tension pour alimenter les différentes parties du circuit se trouvent centrés sur la partie haute du PCB.

Cet emplacement est le plus éloigné des antennes qui se trouvent à gauche et à droite du PCB, ainsi que le microcontrôleur qui se trouve en bas de la carte. Cela permet de les préserver le mieux possible du bruit pouvant venir des alimentations. Celles-ci sont routées grâce à des plans d'alimentation sur la couche 3 afin d'avoir le meilleur découplage possible avec le plan de masse qui se trouve en couche 2. Le routage des plans d'alimentation se trouve à la figure 66 et le plan de masse à la figure 67.

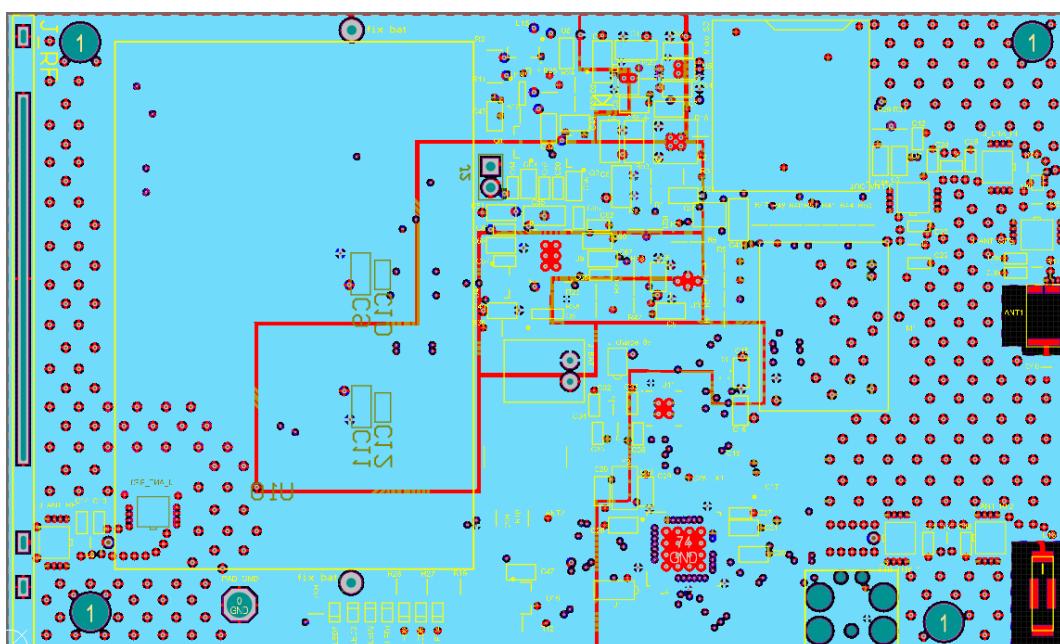


FIGURE 66 – Routage des plans d'alimentations sur la couche 3

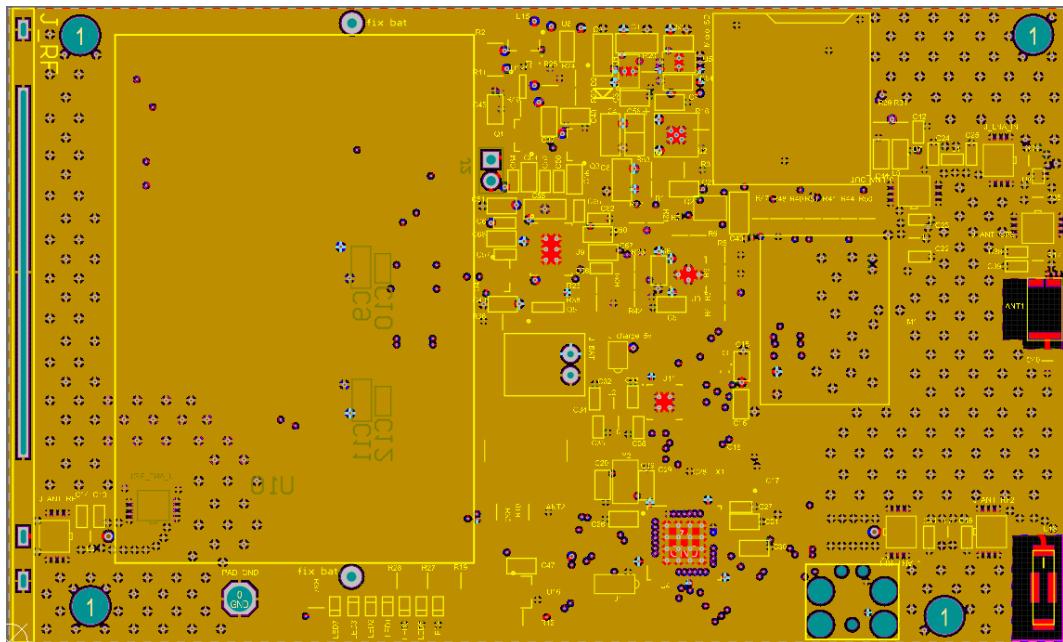


FIGURE 67 – Plan de GND sur la couche 2

#### 5.2.4 Réalisation de l'antenne pour la communication radio

Cette section va détailler la réalisation de l'antenne qui sera utilisée pour la communication radio entre la base et le rover. Contrairement aux autres antennes utilisées comme les antennes GPS, Bluetooth et NFC, l'antenne radio sera fabriquée à l'école afin de réduire les coûts du produit final. L'antenne est une antenne PIFA qui viens se fixer sur le coté gauche, perpendiculairement au circuit. Elle va être modélisée sur ADS puis redessinée sur Altium pour enfin pouvoir être réalisée sur un PCB.

Une antenne PIFA est une antenne avec une forme semblable à celle présente à la figure 68

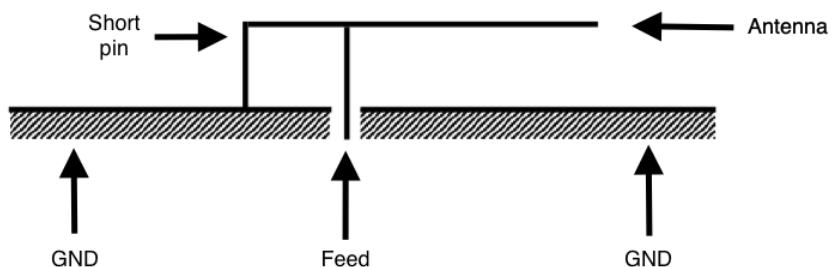


FIGURE 68 – Exemple d'antenne PIFA

On remarque que l'antenne a un point d'entrée du signal (Feed), un point à la masse (Short pin) et un bras en l'air qui rayonne (Antenna). Différents facteurs entrent en compte pour créer une antenne PIFA. La longueur influence la fréquence de résonance de l'antenne, la distance entre le bras de l'antenne et la masse influence la capacitance et la distance entre les points feed et short pin influencent l'inductance de l'antenne. Bien entendu tous ces éléments sont reliés, la modification de la longueur affectera aussi la surface de l'antenne et donc sa capacitance par exemple.

Avant de commencer à designer l'antenne, il faut configurer le substrat du PCB utilisé dans ADS. Ces informations se trouvent sur la figure 69

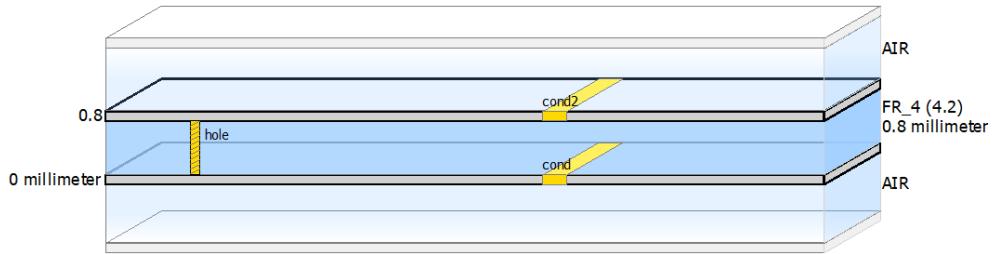


FIGURE 69 – Substrat utilisé pour l'antenne RF

L'épaisseur de FR-4 utilisée est de 0.8 mm car avec des épaisseurs plus fines, les antennes ne peuvent pas être gravées correctement

Une fois que ces informations sont complétées, il faut dessiner une antenne sur ADS. Pour simplifier la simulation, l'antenne est dessinée sur le même plan que le PCB du circuit, même si à la fin elle sera fixée perpendiculairement.

Il faut commencer par dessiner un plan de masse qui a les mêmes dimensions que notre circuit, soit 60 mm x 100 mm. Il faut ensuite dessiner une première antenne, la simuler, puis l'ajuster jusqu'à obtenir les caractéristiques souhaitées. Dans notre cas, l'antenne doit fonctionner à 868 MHz et avoir une impédance de  $50 \Omega$ . Elle doit avoir une longueur maximum de 60 mm et une hauteur d'environ 8mm afin de ne pas déborder du PCB en longueur et de ne pas trop dépasser la batterie en hauteur. Ça permet de garder l'appareil dans une taille raisonnable. L'antenne pourra être un peu plus haute que les 8 mm de la batterie car un système de fixation vient sur la batterie. L'antenne obtenue est représentée à la figure 70.



FIGURE 70 – Antenne PIFA 868 MHz réalisée sur ADS

Ses dimensions sont de 50.2 mm le long et 8.2 mm de haut. Les dimensions finales seront légèrement plus grandes car l'antenne sera gravée au centre d'un PCB et pas au bord.

La figure 71 montre les caractéristiques techniques de l'antenne.

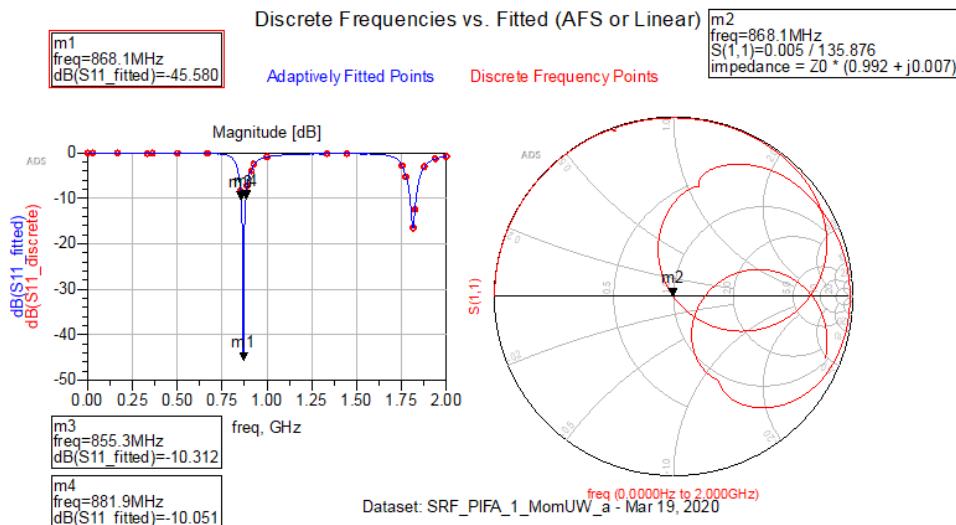


FIGURE 71 – Résultats de simulation de l'antenne PIFA sur ADS

Cette antenne a un facteur S11 de -45.5 dB à 868.1 MHz et une bande passante de

26.6 MHz à -10 dB. L'équation 5 permet d'obtenir l'impédance de l'antenne.

$$Z_0 \cdot (0.992 + 0.007j) = 50 \cdot (0.992 + 0.007j) = 49.6 + 0.35j[\Omega] \quad (5)$$

L'antenne possède une impédance de  $49.6 + 0.35j[\Omega]$ .

Maintenant que l'antenne PIFA simulée convient, il faut la redessiner sur Altium en reportant les mesures afin de pouvoir la réaliser sur un PCB. La figure 72 présente l'antenne dessinée sur Altium. Afin de pouvoir ajuster le plus précisément possible la fréquence de résonance de l'antenne, sa longueur a été légèrement augmentée sur le design Altium. Il suffira de la raccourcir petit à petit avec un cutter pour ajuster sa fréquence.

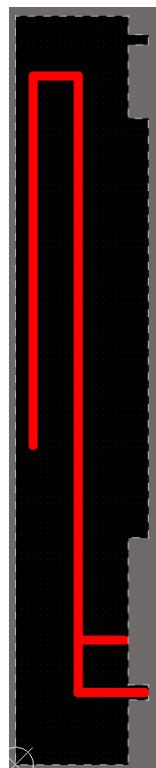


FIGURE 72 – Antenne PIFA redessinée sur Altium

Les dimensions finales du PCB pour l'antenne sont de 60 mm x 9 mm. Comme cette antenne a été dessinée sur altium, cela permet de contrôler précisément les positions des points de connexion et de fixation de l'antenne en important provisoirement le PCB de l'antenne dans celui du circuit complet. La figure 73 montre l'antenne PIFA en face de son connecteur qui se trouve sur le PCB du circuit.

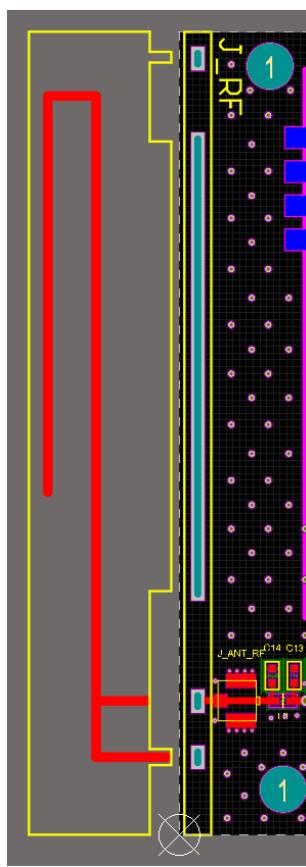


FIGURE 73 – Antenne positionnée à coté du PCB du module

La grande encoche au centre de l'antenne permet non seulement de maintenir mécaniquement l'antenne en place, mais également de pouvoir ajouter de la capacité à l'antenne si l'impédance n'était pas correcte. Pour le faire, il suffit d'ajouter un morceau de scotch cuivré au dos de l'antenne, et de le souder sur le grand pad central du PCB qui est connecté à la masse. Il ne reste plus qu'à ajuster la taille du scotch pour obtenir la correction d'impédance.

## 6 Continuité et améliorations du projet

Cette section détaille la suite du travail qui devrait être effectué afin d'obtenir un module GPS opérationnel. Certains points ci-dessous proposent des solutions à des problématiques qui n'ont pas pu être résolues par manque de temps ou d'idée sur le moment.

Comme le PCB n'a jamais été réalisé, la première chose à faire est de faire fabriquer le PCB, le monter et le tester. Il y aura sûrement des modifications à apporter, rien que pour adapter les impédances de chaque antenne ainsi que leurs fréquences de résonance si besoin.

Une fois que le PCB fonctionnel, il faudra mettre en place la communication entre le microcontrôleur et le GPS puis configurer GPS notamment grâce aux commandes de configuration UBX détaillée dans le chapitre 3.3.1, ce qui devrait permettre de récupérer les informations de positions dans le microcontrôleur. Ces informations devraient ensuite pouvoir être renvoyées par bluetooth sur un appareil à partir duquel il serait possible de récupérer la position du GPS. Une étape intermédiaire pourrait être de stocker les points GPS sur la carte mémoire micro SD puis de vérifier les données de cette carte sur un ordinateur, ce qui permettrait de s'affranchir des contraintes du bluetooth dans un premier temps.

Une fois que les données sont correctement reçues par le téléphone, l'idéal serait de pouvoir les renvoyer par internet 3G/4G afin de récupérer ces données régulièrement sur une page internet. Cette fonctionnalité permettrait de centraliser chaque position de plusieurs modules GPS et d'avoir une vue globale de chaque participant dans le cadre d'un événement sportif comme une course par exemple.

En ce qui concerne les mesures, toute la partie sur la dispersion des mesures serait à confirmer après avoir trouvé un moyen d'empêcher les arrondis dans matlab. Une des solutions que j'ai essayée est de changer le nombre de digits avec lesquels matlab calcule (32 par défauts) grâce à la commande VPA, mais le temps m'a manqué et je n'ai pas réussi à mettre en place un code permettant de calculer de petites distances sans subir d'erreur d'arrondi.

J'ai également trouvé un moyen pour m'assurer que les coordonnées indiquées par le GPS soient exactement celles où il se trouve et qu'il n'y ait pas une forme d'offset dans les coordonnées obtenues. Cette vérification peut être effectuée grâce aux cadastres situés un peu partout en Suisse. Les coordonnées de ceux-ci sont disponibles depuis le site "cadastre.ch" [30] qui regroupe les liens des plans de chaque canton sur lesquels les cadastres sont indiqués. La Suisse ayant son propre système de coordonnées, la conversion vers le système standard peut être fait sur le site "admin.ch" [31].

Les tests de portée du module radio n'ont pas été faits car celui qui équipe les modules de u-blox n'est pas le même que celui qui se trouve sur le PCB. Ils devront probablement être faits une fois que le PCB sera monté.

## 7 Conclusion

Ce projet m'a permis d'en apprendre beaucoup sur le fonctionnement des différentes technologies liées au GPS ainsi qu'à son évolution. Il m'a également permis de mettre en pratique le design d'antennes radio notamment en combinant les logiciels Linecalc et ADS afin d'obtenir une antenne compacte et avec des caractéristiques techniques simulées satisfaisantes. J'ai également beaucoup appris sur le placement des composants sur les PCB multicouches afin de séparer chaque partie du circuit tout en conservant un design compact. Bien que je n'ai pas eu l'opportunité de réaliser le montage du PCB et par conséquent les mesures avec le produit terminé, j'ai pu effectuer différentes mesures dans diverses conditions grâce aux modules u-blox. Ces mesures m'ont notamment confirmé la précision centimétrique lors des mesures de distances entre différentes positions de l'antenne GPS.

## 8 Signatures

Pougny, le 7 juin 2020

Aurélien Berthelot

## 9 Liste des abréviations

|       |  |
|-------|--|
| ARP   | Antenna Reference Point                            |
| ASCII | American Standard Code for Information Interchange |
| BPSK  | Binary Phase Shift Keying                          |
| C/A   | Coarse Acquisition                                 |
| CEP   | Circular Error Probability                         |
| DD    | Degrés Décimaux                                    |
| DGNSS | Differential Global Navigation Satellite System    |
| DGPS  | Differential Global Positioning System             |
| DM    | Degrés Minutes                                     |
| EGNOS | European Geostationary Navigation Overlay Service  |
| FAA   | Federal Aviation Administration                    |
| GLL   | Positionnement Géographique Longitude - Latitude   |
| GNSS  | Global Navigation Satellite System                 |
| GPIO  | General Purpose InputOutput                        |
| GPS   | Global Positioning System                          |
| IC    | Integrated Circuit                                 |
| LBAS  | Local Based Augmentation System                    |
| LNA   | Low Noise Amplifier                                |
| MSM   | Multiple Signal Messages                           |
| NFC   | Near Field Communication                           |
| NMEA  | National Marine & Electronics Association          |
| PCB   | Printed Circuit Board                              |
| PIFA  | Planar Inverted-F antenna                          |
| PPM   | Partie Par Million                                 |
| QNF   | Quad Flat No-leads                                 |
| RF    | Radio-Fréquence                                    |
| RTCM  | Radio Technical Commission for Maritime Services   |
| RTK   | Real Time Kinematic                                |
| SA    | Selective Availability                             |
| SBAS  | Satellite Based Augmentation System                |
| SDMC  | System for Differential Corrections and Monitoring |
| SPI   | Serial Peripheral Interface                        |
| TTFF  | Time To First Fix                                  |
| UART  | Universal Asynchronous Receiver Transmitter        |
| UHF   | Ultra High Frequency                               |
| USB   | Universal Serial Bus                               |
| USCG  | United States Coast Guard                          |
| UTC   | Universel Temps Coordonné                          |
| WAAS  | Wide Area Augmentation System                      |

## Références

- [1] Wikipédia. Système de positionnement par satellites, consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/Systeme\\_de\\_positionnement\\_par\\_satellites](https://fr.wikipedia.org/wiki/Systeme_de_positionnement_par_satellites).
- [2] Wikipédia. Transit (satellite), consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/TRANSIT\\_\(satellite\)](https://fr.wikipedia.org/wiki/TRANSIT_(satellite)).
- [3] Philippe Varnoteaux. La localisation par satellites : 40 ans de révolutions, consulté en : Mars 2020.  
<https://www.air-cosmos.com/article/la-localisation-par-satellite-s-40-ans-de-rvolutions-2858>.
- [4] Wikipédia. Global positioning system, consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/Global\\_Positioning\\_System](https://fr.wikipedia.org/wiki/Global_Positioning_System).
- [5] THOMAS. Systèmes d'augmentations gnss : Abas, sbas et gbas, consulté en : Mars 2020.  
<http://thomas-pilote.over-blog.com/2018/12/systemes-augmentation-gps-abas-sbas-et-gbas.html>.
- [6] Triangulation, consulté en : Mars 2020.  
<http://gps-tpe.e-monsite.com/pages/le-fonctionnement-du-gps/triangulation.html>.
- [7] Navstar gps, consulté en : Mars 2020.  
<https://eduscol.education.fr/orbito/system/navstar/gps00.htm>.
- [8] Denis DELBECQ. Le gps, atout de la coalition, consulté en : Avril 2020.  
[https://www.liberation.fr/evenement/2003/03/27/le-gps-atout-de-la-coalition\\_459770](https://www.liberation.fr/evenement/2003/03/27/le-gps-atout-de-la-coalition_459770).
- [9] Wikipédia. Gps différentiel, consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/GPS\\_d%C3%A9f%C3%A9rentiel](https://fr.wikipedia.org/wiki/GPS_d%C3%A9f%C3%A9rentiel).
- [10] Sylvie Ladet Renaud Lahaye. Les principes du positionnement par satellite : Gnss, consulté en : Avril 2020.  
[https://www6.inrae.fr/cahier\\_des\\_techniques/content/download/3806/34077/version/2/file/05\\_CH1\\_LAHAYE\\_gnss.pdf](https://www6.inrae.fr/cahier_des_techniques/content/download/3806/34077/version/2/file/05_CH1_LAHAYE_gnss.pdf).
- [11] Gps – fonctionnement, consulté en : Mars 2020.  
<http://3grt.fr/comprendre-le-gps/>.
- [12] L'équipe TERIA. Les systèmes d'augmentation de la précision lbas et sbas, consulté en : Mars 2020.  
<https://www.reseau-teria.com/2019/12/30/les-systemes-daugmentation-sbas-et-lbas/>.
- [13] News Desk. Australian government announces the development of augmentation system, consulté en : Avril 2020.  
<https://www.geospatialworld.net/news/australian-government-announces-the-development-of-sbas/>.
- [14] Sbas in the world, consulté en : Mars 2020.  
[https://gssc.esa.int/navipedia/index.php/File:SBAS\\_in\\_the\\_world.png](https://gssc.esa.int/navipedia/index.php/File:SBAS_in_the_world.png).

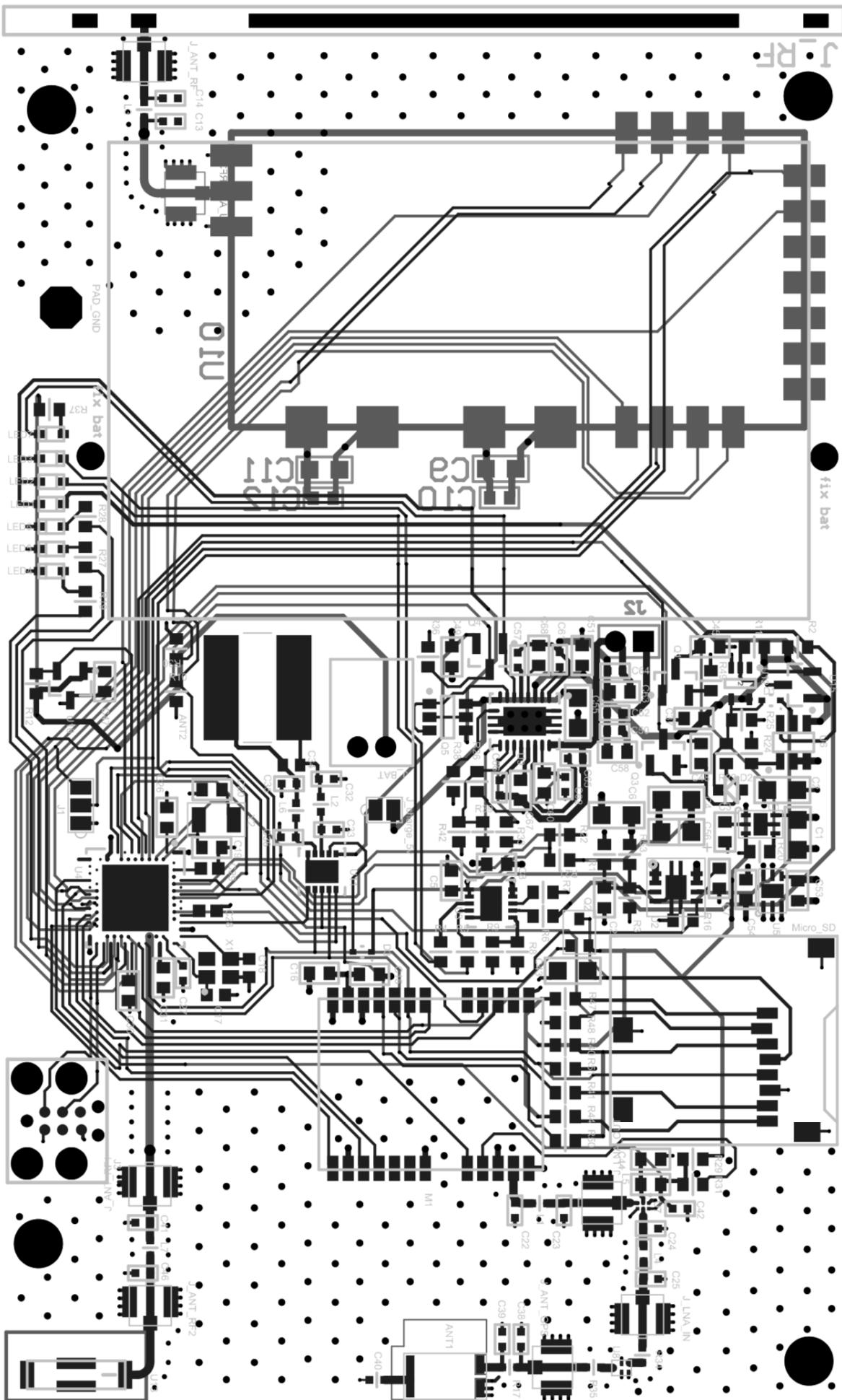
- [15] Wikipédia. Wide area augmentation system, consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/Wide\\_Area\\_Augmentation\\_System](https://fr.wikipedia.org/wiki/Wide_Area_Augmentation_System).
- [16] Wikipédia. European geostationary navigation overlay service, consulté en : Avril 2020.  
[https://fr.wikipedia.org/wiki/European\\_Geostationary\\_Navigation\\_Overlay\\_Service](https://fr.wikipedia.org/wiki/European_Geostationary_Navigation_Overlay_Service).
- [17] Wikipédia. Récepteur gps, consulté en : Mars 2020.  
[https://fr.wikipedia.org/wiki/Récepteur\\_GPS](https://fr.wikipedia.org/wiki/Récepteur_GPS).
- [18] Le signal gps, consulté en : Avril 2020.  
[http://www.geologie.ens.fr/~ecalais/teaching/geopositionnement-gnss/m2\\_geopositionnement\\_2.pdf](http://www.geologie.ens.fr/~ecalais/teaching/geopositionnement-gnss/m2_geopositionnement_2.pdf).
- [19] René REYT. Le g.p.s suite - les signaux g.p.s, consulté en : Avril 2020.  
<https://www.rene-reyt.fr/documents/electronique/le-g-p-s-suite-les-signaux-g-p-s/>.
- [20] Sreejith Hrishikesan. Bpsk system with block diagram, consulté en : Avril 2020.  
<https://www.electronicsandcommunications.com/2019/02/bpsk-system-with-block-diagram.html>.
- [21] About rtk, consulté en : Mars 2020.  
<https://docs.datagnss.com/d303-docs/common/about-rtk/#about-rtk>.
- [22] Gnss measurements, consulté en : Avril 2020.  
<https://www.novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/gnss-measurements/>.
- [23] Wikipédia. Cinématique temps réel, consulté en : Mars 2020.  
[https://fr.wikipedia.org/wiki/Cinématique\\_temps\\_réel](https://fr.wikipedia.org/wiki/Cinématique_temps_réel).
- [24] Lillian Havill. Gps basic theory, consulté en : Mars 2020.  
<https://slideplayer.com/slide/1508088/>.
- [25] RP. Types d'erreurs gps, consulté en : Avril 2020.  
<http://gerssat.chez-alice.fr/gps/GPS2.htm>.
- [26] u blox. Neo-m8p, consulté en : Mars 2020.  
[https://www.u-blox.com/sites/default/files/NEO-M8P\\_DataSheet\\_%28UBX-15016656%29.pdf](https://www.u-blox.com/sites/default/files/NEO-M8P_DataSheet_%28UBX-15016656%29.pdf).
- [27] u blox. Receiver description, consulté en : Avril 2020.  
[https://www.u-blox.com/sites/default/files/products/documents/u-blox8-M8\\_ReceiverDescrProtSpec\\_%28UBX-13003221%29.pdf](https://www.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_%28UBX-13003221%29.pdf).
- [28] LP du Giennois. Les trames nmea, consulté en : Mai 2020.  
<http://www.cedricaoun.net/eie/trames%20NMEA183.pdf>.
- [29] Wikipédia. Orthodromie, consulté en : Mai 2020.  
<https://fr.wikipedia.org/wiki/Orthodromie>.
- [30] Donnée de la mensuration officielle - obtention, consulté en : Mai 2020.  
<https://www.cadastre.ch/fr/services/service/data.html>.
- [31] Navref, consulté en : Mai 2020.  
<https://www.swisstopo.admin.ch/fr/cartes-donnees-en-ligne/calculation-services/navref.html>.

## 10 Annexes

Cette section contient tous les codes matlab dont les graphiques ont été présenté dans ce rapport

| Nom du fichier                             | Commentaire  |
|--|--|
| Schema_complet                             | Schématique complète du module, chaque partie figure indépendamment dans la section 5.1.   |
| PCB_NB                                     | Rootage du PCB top et bottom en transparence.  |
| rover_stab_8h                              | Code matlab utilisé pour traiter les données de la mesure de stabilité du rover.   |
| rover_distance_50_cm                       | Code matlab utilisé pour traiter les données de la mesure d'une distance de 50 cm entre deux points.                                       |
| rover_dist_1_2_5_10_15_20_m_jardin         | Code matlab utilisé pour traiter les données des mesures faites à 1, 2, 5, 10, 15 et 20 m de la base dans le jardin.                       |
| rover_dist_1_2_5_10_15_20_m_sans_obstacles | Code matlab utilisé pour traiter les données des mesures faites à 1, 2, 5, 10, 15 et 20 m de la base dans un champ loin de tout obstacles. |
| comp_dist_min_avec_sans_erreurs            | Code matlab utilisé pour montrer l'incidence les erreurs d'arrondis de matlab.   |
| comp_dist_1_a_20_m_avec_sans_erreurs       | Code matlab utilisé pour comparer les distances obtenues entre la formule erroné et la formule corrigée.                                   |





```
%%%%%%%%%%%%%
%
% rover_stab_8h.m
% Aurélien Berthelot
% Code pour traiter la mesure de stabilité de 8h du rover
%
% ATTENTION, CE CODE UTILISE UNE FORMULE DE CALCUL DE DISTANCE ERROREE
%
%%%%%%%%%%%%%
clc; clear; close all;
load workspace;

% conversion degré minutes (dm) en degré décimale (dd)
data_8h_dd.Latitude = fix(data_8h_dm.Latitude/100) + (data_8h_dm.Latitude - fix(
(data_8h_dm.Latitude/100)*100)/60;
data_8h_dd.Longitude = fix(data_8h_dm.Longitude/100) + (data_8h_dm.Longitude - fix(
(data_8h_dm.Longitude/100)*100)/60;
data_8h_dd.Time = data_8h_dm.Time;

moy_latitude_dd = mean(data_8h_dd.Latitude);
moy_longitude_dd = mean(data_8h_dd.Longitude);

% définition de la grille d'affichage du graph
origine_x = 5.9642426;
step_x = 1e-7;
fin_x = 5.9642432;
origine_y = 46.1469236;
step_y = 1e-7;
fin_y = 46.1469244;

% calcul des distances de la grille
lat_1 = origine_y;
lat_2 = origine_y + step_y;

lon_1 = origine_x;
lon_2 = origine_x + step_x;

%distance d'un carré du plot en longitude
%depart : lat_1 lon_1
%arrivée : lat_1 lon_2

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(
(lon_2-lon_1))) * 1851.85

%distance d'un carré du plot en latitude
%depart : lat_1 lon_1
%arrivée : lat_2 lon_1

distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_1))) * 1851.85

markersize = 10;
linewidth = 3;

% affichage de la dispersion des points
figure();
hold on
```

```

p = plot(data_8h_dd.Longitude, data_8h_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p =plot(moy_longitude_dd,moy_latitude_dd,'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p =plot([origine_x origine_x+step_x],[origine_y origine_y], 'r'); % longueur d'un carré
p.LineWidth=linewidth*2;
p =plot([origine_x origine_x],[origine_y origine_y+step_y], 'b'); % largeur d'un carré
p.LineWidth=linewidth*2;
axis([origine_x fin_x origine_y fin_y])
legend('Mesures','Position moyenne',[ 'Echelle : ~ ',num2str(
(distance_carre_horizontale*100), ' cm'), ' Echelle : ~ ',num2str(
(distance_carre_verticale*100), ' cm') ])
xlabel('Longitude [°], format DD')
ylabel('Latitude [°], format DD')
title("Mesure de stabilité de la position du rover pendant 8h")
hold off
grid on;

%calcul de la distance entre chaque point et la moyenne des mesures
distance_point_moyenne = zeros(size(data_8h_dd.Latitude,1),1);
for j=1:size(data_8h_dd.Longitude)
    distance_point_moyenne(j) = (60*acos(sin(moy_latitude_dd)*sin(data_8h_dd.Latitude(j))*
+cos(moy_latitude_dd)*cos(data_8h_dd.Latitude(j))*cos(data_8h_dd.Longitude(j)-*
moy_longitude_dd))) *1851.85 ;
end

ecart_type = std(distance_point_moyenne)

distance_max = max(distance_point_moyenne)

step_dist = 0.0005;
% histogramme des mesures
figure();
echelle_m = 0:step_dist:distance_max+0.002;
[ret_histc] = histc(distance_point_moyenne,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
xlabel("Distance [m]")
ylabel("Nombre de mesures")
title("Dispersion des 28800 mesures par rapport à la position moyenne")
set(gca,'FontSize',15)
grid on;
grid minor

div_dist_pt_moy = reshape(distance_point_moyenne,3600,8);

%histogramme des mesures heures par heures
figure();
echelle_m = 0:0.0005:distance_max+0.002;
[ret_histc_2] = histc(div_dist_pt_moy,echelle_m);
b = bar(echelle_m,ret_histc_2,'stacked','FaceColor','flat');
b(8).CData = [0 0 0];
xlabel("Distance [m]")
ylabel("Nombre de mesures")

```

```
title("Dispersion des mesures par rapport à la position moyenne, divisée par heures")
set(gca,'FontSize',15)
legend("1re heure","2e heure","3e heure","4e heure","5e heure","6e heure","7e heure","8e heure")
grid on;
grid minor

dist_pre_neo_m8p = 0.025+step_dist;
dist_2_5_cm = dist_pre_neo_m8p/step_dist;
nb_pt_2_5_cm = (sum(ret_histc_2(1:dist_2_5_cm,:)))'

dist_pre_neo_m8p = 0.004+step_dist;
dist_0_4_cm = dist_pre_neo_m8p/step_dist;
nb_pt_0_4_cm = (sum(ret_histc_2(1:dist_0_4_cm,:)))'
```

```
%%%%%
%
% rover_distance_50_cm.m
% Aurélien Berthelot
% Code pour evaluer deux points distant de 50 cm
%
% ATTENTION, CE CODE UTILISE UNE FORMULE DE CALCUL DE DISTANCE ERROREE
%
%%%%%

clc; close all; clear;
load workspace;
% conversion degré minutes (dm) en degré décimale (dd)
d_50cm_pt1_dd.Latitude = fix(d_50cm_pt1_dm.Latitude/100) + (d_50cm_pt1_dm.Latitude - fix(d_50cm_pt1_dm.Latitude/100)*100)/60;
d_50cm_pt1_dd.Longitude = fix(d_50cm_pt1_dm.Longitude/100) + (d_50cm_pt1_dm.Longitude - fix(d_50cm_pt1_dm.Longitude/100)*100)/60;
d_50cm_pt1_dd.Time = d_50cm_pt1_dm.Time;

% conversion degré minutes (dm) en degré décimale (dd)
d_50cm_pt2_dd.Latitude = fix(d_50cm_pt2_dm.Latitude/100) + (d_50cm_pt2_dm.Latitude - fix(d_50cm_pt2_dm.Latitude/100)*100)/60;
d_50cm_pt2_dd.Longitude = fix(d_50cm_pt2_dm.Longitude/100) + (d_50cm_pt2_dm.Longitude - fix(d_50cm_pt2_dm.Longitude/100)*100)/60;
d_50cm_pt2_dd.Time = d_50cm_pt2_dm.Time;

% calcul des moyennes
moy_latitude_pt1_dd = mean(d_50cm_pt1_dd.Latitude);
moy_longitude_pt1_dd = mean(d_50cm_pt1_dd.Longitude);

moy_latitude_pt2_dd = mean(d_50cm_pt2_dd.Latitude);
moy_longitude_pt2_dd = mean(d_50cm_pt2_dd.Longitude);

% axes du graphique
origine_x = 5.9642562;
step_x = 2e-7;
fin_x = 5.9642578;
origine_y = 46.146922;
step_y = 1e-6;
fin_y = 46.146928;

% calcul des distances de la grille
lat_1 = origine_y;
lat_2 = origine_y + step_y;

lon_1 = origine_x;
lon_2 = origine_x + step_x;

% distance d'un carré du plot en longitude
% départ : lat_1 lon_1
% arrivée : lat_1 lon_2

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(lon_2-lon_1))) * 1851.85;

% distance d'un carré du plot en latitude
% départ : lat_1 lon_1
% arrivée : lat_2 lon_1
```

```

distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_1))) * 1851.85;

markersize = 10;
linewidth = 3;

%affichage dispersion des points
figure();
hold on
p = plot(d_50cm_pt1_dd.Longitude, d_50cm_pt1_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p =plot(d_50cm_pt2_dd.Longitude, d_50cm_pt2_dd.Latitude, 'bo');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p =plot(moy_longitude_pt1_dd,moy_latitude_pt1_dd,'xm');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p =plot(moy_longitude_pt2_dd,moy_latitude_pt2_dd,'gx');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p =plot([origine_x origine_x+step_x],[origine_y origine_y], 'r'); % longueur d'un carré
p.LineWidth=linewidth*2;
p =plot([origine_x origine_x],[origine_y origine_y+step_y], 'b'); % largeur d'un carré
p.LineWidth=linewidth*2;
axis([origine_x fin_x origine_y fin_y])
legend('Mesures de position du point 1','Mesures de position du point 2','Position
moyenne du point 1','Position moyenne du point 2',['Echelle : ',num2str(
distance_carre_horizontale*100), ' cm'], ['Echelle : ',num2str(
distance_carre_verticale*100), ' cm'])
xlabel('Longitude [°], format DD')
ylabel('Latitude [°], format DD')
title("Mesure d'une distance de 50 cm")
hold off
grid on;

% distance entre la moyenne des points 1 et 2
distance_moyenne = (60*acos(sin(moy_latitude_pt1_dd)*sin(moy_latitude_pt2_dd)+cos(
(moy_latitude_pt1_dd)*cos(moy_latitude_pt2_dd)*cos(moy_longitude_pt2_dd)-
moy_longitude_pt1_dd))) * 1851.85

%calcul de la distance entre chaque point et la moyenne des mesures
distance_pt_moy_et_pt_1 = zeros(size(d_50cm_pt1_dd.Latitude));
for i=1:size(d_50cm_pt1_dd.Latitude,1)
    distance_pt_moy_et_pt_1(i) = (60*acos(sin(d_50cm_pt1_dd.Latitude(i))*sin(
(moy_latitude_pt1_dd)+cos(d_50cm_pt1_dd.Latitude(i))*cos(moy_latitude_pt1_dd)*cos(
(d_50cm_pt1_dd.Longitude(i)-moy_longitude_pt1_dd))) *1851.85;
end
dis_max_moy_point_1 = max(distance_pt_moy_et_pt_1)
ecart_type_moy_pt1 = std(distance_pt_moy_et_pt_1)

distance_pt_moy_et_pt_2 = zeros(size(d_50cm_pt2_dd.Latitude));
for i=1:size(d_50cm_pt2_dd.Latitude,1)
    distance_pt_moy_et_pt_2(i) = (60*acos(sin(d_50cm_pt2_dd.Latitude(i))*sin(
(moy_latitude_pt2_dd)+cos(d_50cm_pt2_dd.Latitude(i))*cos(moy_latitude_pt2_dd)*cos(
(d_50cm_pt2_dd.Longitude(i)-moy_longitude_pt2_dd))) *1851.85;
end

```

```

dis_max_moy_point_2 = max(distance_pt_moy_et_pt_2)
ecart_type_moy_pt2 = std(distance_pt_moy_et_pt_2)

% distance de chaque points de départ vers chaque point d'arrivé
distance_pt_1_et_pt_2 = zeros(size(d_50cm_pt2_dd.Latitude,1), size(d_50cm_pt2_dd.‹
Latitude,1));
for i=1:size(d_50cm_pt1_dd.Latitude,1)
    for j=1:size(d_50cm_pt2_dd.Latitude,1)
        distance_pt_1_et_pt_2(i,j) = (60*acos(sin(d_50cm_pt1_dd.Latitude(i))*sin‹
(d_50cm_pt2_dd.Latitude(j))+cos(d_50cm_pt1_dd.Latitude(i))*cos(d_50cm_pt2_dd.Latitude(j))‹
*cos(d_50cm_pt2_dd.Longitude(j)-d_50cm_pt1_dd.Longitude(i)))) *1851.85;
    end
end

dist_pts = reshape(distance_pt_1_et_pt_2,[],1);

ecart_type_dist = std(dist_pts)
dist_moyenne = mean(dist_pts)
distance_max = max(dist_pts)
distance_min = min(dist_pts)

% echelle histogramme
step_dist = 0.0005;
dist = 0.5;
start_ech = distance_min - step_dist;
stop_ech = distance_max + step_dist;
echelle_m = start_ech-step_dist:step_dist:stop_ech+step_dist;

% histogramme
[ret_histc] = histc(dist_pts,echelle_m);

% vérification que tous les points sont dans l'histogramme 3600*3600
tot_points_on_graph = sum(ret_histc)

% affichage de l'histogramme des distances
figure();
bar(echelle_m,ret_histc,'stacked');
xlabel("Distance [m]")
ylabel("Nombre de mesures")
title("Distance obtenu à parti de chaque point de départ vers chaque point d'arrivée")
set(gca,'FontSize',15)
grid on;
grid minor

% calcul du pourcentage de points dans des cercles de rayon de 0.4 et 2.5 cm
nb_step_tot = round(( stop_ech + step_dist - start_ech + step_dist)/step_dist)
nb_step_50_cm = round(nb_step_tot /(stop_ech - step_dist - start_ech - step_dist) * (dist‐
- start_ech ))

% calcul de la précision en fonction d'un rayon
nb_step_r_3_mm = 0.003/step_dist;
precision_3_mm = sum(ret_histc(nb_step_50_cm-nb_step_r_3_mm :‹
nb_step_50_cm+nb_step_r_3_mm ))/tot_points_on_graph *100

nb_step_r_4_mm = 0.004/step_dist;
precision_4_mm = sum(ret_histc(nb_step_50_cm-nb_step_r_4_mm :‹
nb_step_50_cm+nb_step_r_4_mm))/tot_points_on_graph *100

```

06/06/20 14:37 /Users/aurelienbe.../rover\_distance\_50\_cm.m 4 of 4

---

```
nb_step_r_1_cm = 0.01/step_dist;
precision_1_cm = sum(ret_histc(nb_step_50_cm-nb_step_r_1_cm :&
nb_step_50_cm+nb_step_r_1_cm ))/tot_points_on_graph *100

nb_step_r_2_5_cm = 0.025/step_dist;
precision_2_5cm = sum(ret_histc(nb_step_50_cm-nb_step_r_2_5_cm :&
nb_step_50_cm+nb_step_r_2_5_cm ))/tot_points_on_graph *100
```

```
%%%%%%%%%%%%%
%
% rover_dist_1_2_5_10_15_20_m_jardin.m
% Aurélien Berthelot
% Code pour traiter les mesures de distances et de dispersion des points
% dans mon jardin (environnement avec des obstacles autour)
%
% ATTENTION, CE CODE UTILISE UNE FORMULE DE CALCUL DE DISTANCE ERROREE
%
%%%%%%%%%%%%%
clc; close all; clear;
load workspace;

% conversion degré minutes (dm) en degré décimale (dd)
data_1m_dd.Latitude = fix(data_1m_dm.Latitude/100) + (data_1m_dm.Latitude - fix(
(data_1m_dm.Latitude/100)*100)/60;
data_1m_dd.Longitude = fix(data_1m_dm.Longitude/100) + (data_1m_dm.Longitude - fix(
(data_1m_dm.Longitude/100)*100)/60;
data_1m_dd.Time = data_1m_dm.Time;

data_2m_dd.Latitude = fix(data_2m_dm.Latitude/100) + (data_2m_dm.Latitude - fix(
(data_2m_dm.Latitude/100)*100)/60;
data_2m_dd.Longitude = fix(data_2m_dm.Longitude/100) + (data_2m_dm.Longitude - fix(
(data_2m_dm.Longitude/100)*100)/60;
data_2m_dd.Time = data_2m_dm.Time;

data_5m_dd.Latitude = fix(data_5m_dm.Latitude/100) + (data_5m_dm.Latitude - fix(
(data_5m_dm.Latitude/100)*100)/60;
data_5m_dd.Longitude = fix(data_5m_dm.Longitude/100) + (data_5m_dm.Longitude - fix(
(data_5m_dm.Longitude/100)*100)/60;
data_5m_dd.Time = data_5m_dm.Time;

data_10m_dd.Latitude = fix(data_10m_dm.Latitude/100) + (data_10m_dm.Latitude - fix(
(data_10m_dm.Latitude/100)*100)/60;
data_10m_dd.Longitude = fix(data_10m_dm.Longitude/100) + (data_10m_dm.Longitude - fix(
(data_10m_dm.Longitude/100)*100)/60;
data_10m_dd.Time = data_10m_dm.Time;

data_15m_dd.Latitude = fix(data_15m_dm.Latitude/100) + (data_15m_dm.Latitude - fix(
(data_15m_dm.Latitude/100)*100)/60;
data_15m_dd.Longitude = fix(data_15m_dm.Longitude/100) + (data_15m_dm.Longitude - fix(
(data_15m_dm.Longitude/100)*100)/60;
data_15m_dd.Time = data_15m_dm.Time;

data_20m_dd.Latitude = fix(data_20m_dm.Latitude/100) + (data_20m_dm.Latitude - fix(
(data_20m_dm.Latitude/100)*100)/60;
data_20m_dd.Longitude = fix(data_20m_dm.Longitude/100) + (data_20m_dm.Longitude - fix(
(data_20m_dm.Longitude/100)*100)/60;
data_20m_dd.Time = data_20m_dm.Time;

%calcul des moyennes
moy_1m_lat_dd = mean(data_1m_dd.Latitude);
moy_1m_lon_dd = mean(data_1m_dd.Longitude);

moy_2m_lat_dd = mean(data_2m_dd.Latitude);
moy_2m_lon_dd = mean(data_2m_dd.Longitude);
```

```
moy_5m_lat_dd = mean(data_5m_dd.Latitude);
moy_5m_lon_dd = mean(data_5m_dd.Longitude);

moy_10m_lat_dd = mean(data_10m_dd.Latitude);
moy_10m_lon_dd = mean(data_10m_dd.Longitude);

moy_15m_lat_dd = mean(data_15m_dd.Latitude);
moy_15m_lon_dd = mean(data_15m_dd.Longitude);

moy_20m_lat_dd = mean(data_20m_dd.Latitude);
moy_20m_lon_dd = mean(data_20m_dd.Longitude);

% axes du graphique
origine_x = 5.9642562;
step_x = 2e-7;
fin_x = 5.9642578;
origine_y = 46.146922;
step_y = 1e-6;
fin_y = 46.146928;

% calcul des distances de la grille
lat_1 = origine_y;
lat_2 = origine_y + step_y;

lon_1 = origine_x;
lon_2 = origine_x + step_x;

%distance d'un carré du plot en longitude
%depart : lat_1 lon_1
%arrivee : lat_1 lon_2

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(lon_2-lon_1))) * 1851.85;

%distance d'un carré du plot en latitude
%depart : lat_1 lon_1
%arrivee : lat_2 lon_1

distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(lon_1-lon_1))) * 1851.85;

markersize = 10;
linewidth = 3;

%affichage vue générale
figure(1);
hold on
p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_1m_lon_dd, moy_1m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_2m_dd.Longitude, data_2m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_2m_lon_dd, moy_2m_lat_dd, 'xb');
```

```
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_5m_dd.Longitude, data_5m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_5m_lon_dd, moy_5m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_10m_dd.Longitude, data_10m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_10m_lon_dd, moy_10m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_15m_dd.Longitude, data_15m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_15m_lon_dd, moy_15m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_20m_dd.Longitude, data_20m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_20m_lon_dd, moy_20m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

xlabel('Longitude [°], format DD')
ylabel('Latitude [°], format DD')
title("Mesure de points distant de 1 à 20 m de la base")
legend('Mesures ','Moyenne')
hold off
grid on;

% axes du graphique
step_x = 0.0000001;
step_y = 00.0000001;
nb_step_x = 10;
nb_step_y = 10;

%affichage dispersions de chaque mesure
figure(2);

subplot(6,1,1);
x_start = 5.96424085;
y_start = 46.1469125;

lat_1 = y_start ;
lat_2 = lat_1 + 5*step_y;

lon_1 = x_start +0.5*step_y;
lon_2 = lon_1 + step_x;

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(
```

```

(lon_2-lon_1)) * 1851.85;
distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_1))) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro', moy_1m_lon_dd, moy_1m_lat_dd,
'xb',[lon_1 lon_2], [lat_1 lat_1], 'r' , [lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 1 m")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100), ' cm'],
['Echelle : ',num2str(distance_carre_verticale*100), ' cm'])

subplot(6,1,2);
p = plot(data_2m_dd.Longitude, data_2m_dd.Latitude, 'ro', moy_2m_lon_dd, moy_2m_lat_dd,
'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.96424715;
y_start = 46.146905;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 2 m")

subplot(6,1,3);
p = plot(data_5m_dd.Longitude, data_5m_dd.Latitude, 'ro', moy_5m_lon_dd, moy_5m_lat_dd,
'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.96426665;
y_start = 46.1468812;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 5 m")

subplot(6,1,4);
p = plot(data_10m_dd.Longitude, data_10m_dd.Latitude, 'ro', moy_10m_lon_dd,
moy_10m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.96430095;
y_start = 46.1468390;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])

```

```

grid on
set(gca,'FontSize',15)
ylabel('Latitude [°], format DD')
title("Dispersion des points à 10 m")

subplot(6,1,5);
p = plot(data_15m_dd.Longitude, data_15m_dd.Latitude, 'ro',moy_15m_lon_dd,✓
moy_15m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.96433045;
y_start = 46.1468033;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 15 m")

subplot(6,1,6);
p = plot(data_20m_dd.Longitude, data_20m_dd.Latitude, 'ro',moy_20m_lon_dd,✓
moy_20m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.96436205;
y_start = 46.1467643;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 20 m")
xlabel('Longitude [°], format DD')

% distance entre la moyenne à 1m et à 2m
distance_moyenne_1_2 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_2m_lat_dd)+cos(moy_1m_lat_dd)✓
*cos(moy_2m_lat_dd)*cos(moy_2m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 5m
distance_moyenne_1_5 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_5m_lat_dd)+cos(moy_1m_lat_dd)✓
*cos(moy_5m_lat_dd)*cos(moy_5m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 10m
distance_moyenne_1_10 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_10m_lat_dd)+cos✓
(moy_1m_lat_dd)*cos(moy_10m_lat_dd)*cos(moy_10m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 15m
distance_moyenne_1_15 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_15m_lat_dd)+cos✓
(moy_1m_lat_dd)*cos(moy_15m_lat_dd)*cos(moy_15m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 20m
distance_moyenne_1_20 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_20m_lat_dd)+cos✓
(moy_1m_lat_dd)*cos(moy_20m_lat_dd)*cos(moy_20m_lon_dd-moy_1m_lon_dd))) * 1851.85;

distance_moyenne = [distance_moyenne_1_2 distance_moyenne_1_5 distance_moyenne_1_10✓
distance_moyenne_1_15 distance_moyenne_1_20]

%calcul de la distance entre chaque point et la moyenne des mesures à

```

```
%chaque distances
distance_pt_moy_1m = zeros(size(data_1m_dd.Latitude));
for i=1:size(data_1m_dd.Latitude,1)
    distance_pt_moy_1m(i) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(moy_1m_lat_dd)-
+cos(data_1m_dd.Latitude(i))*cos(moy_1m_lat_dd)*cos(data_1m_dd.Longitude(i)-
moy_1m_lon_dd)) *1851.85;
end
dis_max_moy_1m = max(distance_pt_moy_1m);
ecart_type_moy_1m = std(distance_pt_moy_1m);

distance_pt_moy_2m = zeros(size(data_2m_dd.Latitude));
for i=1:size(data_2m_dd.Latitude,1)
    distance_pt_moy_2m(i) = (60*acos(sin(data_2m_dd.Latitude(i))*sin(moy_2m_lat_dd)-
+cos(data_2m_dd.Latitude(i))*cos(moy_2m_lat_dd)*cos(data_2m_dd.Longitude(i)-
moy_2m_lon_dd)) *1851.85;
end
dis_max_moy_2m = max(distance_pt_moy_2m);
ecart_type_moy_2m = std(distance_pt_moy_2m);

distance_pt_moy_5m = zeros(size(data_5m_dd.Latitude));
for i=1:size(data_5m_dd.Latitude,1)
    distance_pt_moy_5m(i) = (60*acos(sin(data_5m_dd.Latitude(i))*sin(moy_5m_lat_dd)-
+cos(data_5m_dd.Latitude(i))*cos(moy_5m_lat_dd)*cos(data_5m_dd.Longitude(i)-
moy_5m_lon_dd)) *1851.85;
end
dis_max_moy_5m = max(distance_pt_moy_5m);
ecart_type_moy_5m = std(distance_pt_moy_5m);

distance_pt_moy_10m = zeros(size(data_10m_dd.Latitude));
for i=1:size(data_10m_dd.Latitude,1)
    distance_pt_moy_10m(i) = (60*acos(sin(data_10m_dd.Latitude(i))*sin(
(moy_10m_lat_dd)+cos(data_10m_dd.Latitude(i))*cos(moy_10m_lat_dd)*cos(data_10m_dd.-
Longitude(i)-moy_10m_lon_dd))) *1851.85;
end
dis_max_moy_10m = max(distance_pt_moy_10m);
ecart_type_moy_10m = std(distance_pt_moy_10m);

distance_pt_moy_15m = zeros(size(data_15m_dd.Latitude));
for i=1:size(data_15m_dd.Latitude,1)
    distance_pt_moy_15m(i) = (60*acos(sin(data_15m_dd.Latitude(i))*sin(
(moy_15m_lat_dd)+cos(data_15m_dd.Latitude(i))*cos(moy_15m_lat_dd)*cos(data_15m_dd.-
Longitude(i)-moy_15m_lon_dd))) *1851.85;
end
dis_max_moy_15m = max(distance_pt_moy_15m);
ecart_type_moy_15m = std(distance_pt_moy_15m);

distance_pt_moy_20m = zeros(size(data_20m_dd.Latitude));
for i=1:size(data_20m_dd.Latitude,1)
    distance_pt_moy_20m(i) = (60*acos(sin(data_20m_dd.Latitude(i))*sin(
(moy_20m_lat_dd)+cos(data_20m_dd.Latitude(i))*cos(moy_20m_lat_dd)*cos(data_20m_dd.-
Longitude(i)-moy_20m_lon_dd))) *1851.85;
end
dis_max_moy_20m = max(distance_pt_moy_20m);
ecart_type_moy_20m = std(distance_pt_moy_20m);

dis_max_moy = [dis_max_moy_1m dis_max_moy_2m dis_max_moy_5m dis_max_moy_10m-
dis_max_moy_15m dis_max_moy_20m]
ecart_type_moy = [ecart_type_moy_1m ecart_type_moy_2m ecart_type_moy_5m]
```

```
ecart_type_moy_10m ecart_type_moy_15m ecart_type_moy_20m]

step_dist = 0.0005;
nb_step_0_4_cm = 9;
nb_step_2_5_cm = 51;

% histogramme de la dispersion des mesures
figure(3);
subplot(6,1,1);
echelle_m = 0:step_dist:0.06;
[ret_histc] = histc(distance_pt_moy_1m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 1 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_1m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_1m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,2);
[ret_histc] = histc(distance_pt_moy_2m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 2 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_2m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_2m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,3);
[ret_histc] = histc(distance_pt_moy_5m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 5 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_5m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_5m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,4);
[ret_histc] = histc(distance_pt_moy_10m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
ylabel("Nombre de mesures")
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 10 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_10m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_10m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,5);
[ret_histc] = histc(distance_pt_moy_15m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
```

```

title("Dispersion des 1200 mesures par rapport à moyenne, distance : 15 m")
set(gca,'FontSize',15)
grid on;
grid minor
cercle_0_4_cm_15m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_15m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,6);
[ret_histc] = histc(distance_pt_moy_20m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
xlabel("Distance [m]")
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 20 m")
set(gca,'FontSize',15)
grid on;
grid minor
cercle_0_4_cm_20m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_20m = sum(ret_histc(1:nb_step_2_5_cm));

pourcent_cercle_0_4_cm = [cercle_0_4_cm_1m cercle_0_4_cm_2m cercle_0_4_cm_5m
cercle_0_4_cm_10m cercle_0_4_cm_15m cercle_0_4_cm_20m]/1200*100
pourcent_cercle_2_5_cm = [cercle_2_5_cm_1m cercle_2_5_cm_2m cercle_2_5_cm_5m
cercle_2_5_cm_10m cercle_2_5_cm_15m cercle_2_5_cm_20m]/1200*100

distance_1m_2m = zeros(size(data_1m_dd.Latitude,1), size(data_2m_dd.Latitude,1));
distance_1m_5m = zeros(size(data_1m_dd.Latitude,1), size(data_5m_dd.Latitude,1));
distance_1m_10m = zeros(size(data_1m_dd.Latitude,1), size(data_10m_dd.Latitude,1));
distance_1m_15m = zeros(size(data_1m_dd.Latitude,1), size(data_15m_dd.Latitude,1));
distance_1m_20m = zeros(size(data_1m_dd.Latitude,1), size(data_20m_dd.Latitude,1));

% distance de chaque points de départ (1m) vers chaque point d'arrivé (2,5,10,15,20 m)
for i=1:size(data_1m_dd.Latitude,1)
    for j=1:size(data_2m_dd.Latitude,1)
        distance_1m_2m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_2m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_2m_dd.Latitude(j))*cos(data_2m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_5m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_5m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_5m_dd.Latitude(j))*cos(data_5m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_10m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_10m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_10m_dd.Latitude(j))*cos(data_10m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_15m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_15m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_15m_dd.Latitude(j))*cos(data_15m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_20m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_20m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_20m_dd.Latitude(j))*cos(data_20m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
    end
end

dist_1m_2m = reshape(distance_1m_2m,[],1);
dist_1m_5m = reshape(distance_1m_5m,[],1);
dist_1m_10m = reshape(distance_1m_10m,[],1);
dist_1m_15m = reshape(distance_1m_15m,[],1);
dist_1m_20m = reshape(distance_1m_20m,[],1);
dist_pts = [dist_1m_2m dist_1m_5m dist_1m_10m dist_1m_15m dist_1m_20m];

```

```

ecart_type_dist_1m_2m = std(dist_1m_2m);
ecart_type_dist_1m_5m = std(dist_1m_5m);
ecart_type_dist_1m_10m = std(dist_1m_10m);
ecart_type_dist_1m_15m = std(dist_1m_15m);
ecart_type_dist_1m_20m = std(dist_1m_20m);

distance_max_1m_2m = max(dist_1m_2m);
distance_max_1m_5m = max(dist_1m_5m);
distance_max_1m_10m = max(dist_1m_10m);
distance_max_1m_15m = max(dist_1m_15m);
distance_max_1m_20m = max(dist_1m_20m);
distance_max = [distance_max_1m_2m distance_max_1m_5m distance_max_1m_10m
                distance_max_1m_15m distance_max_1m_20m];

distance_min_1m_2m = min(dist_1m_2m);
distance_min_1m_5m = min(dist_1m_5m);
distance_min_1m_10m = min(dist_1m_10m);
distance_min_1m_15m = min(dist_1m_15m);
distance_min_1m_20m = min(dist_1m_20m);
distance_min = [distance_min_1m_2m distance_min_1m_5m distance_min_1m_10m
                distance_min_1m_15m distance_min_1m_20m];

step_dist = 0.0005;
dist = 0.5;

% histogramme des distances entre chaque points
figure(4);
tab_dist = [ 2-1 5-1 10-1 15-1 20-1];

for i=1:5

    start_ech = distance_min(:,i) - step_dist;
    stop_ech = distance_max(:,i) + step_dist;
    echelle_m = start_ech-step_dist:step_dist:stop_ech+step_dist;
    [ret_histc] = histc(dist_pts(:,i),echelle_m);

    % verification que tous les points sont dans l'histogramme 3600*3600
    tot_points_on_graph = sum(ret_histc) ;

    % affichage
    subplot(5,1,i);
    bar(echelle_m,ret_histc,'stacked');
    xlabel("Distance [m]")
    if i == 3
        ylabel("Nombre de mesures")
    end
    title(['Distance obtenu à parti du point situé à 1 m, soit une distance de : ', num2str(tab_dist(i)), ' m'])
    set(gca,'FontSize',15)
    grid on;
    grid minor

end

```

```
%%%%%%%%%%%%%
%
% rover_dist_1_2_5_10_15_20_m_sans_obstacles.m
% Aurélien Berthelot
% Code pour traiter les mesures de distances et de dispersion des points
% dans un environnement dégagé
%
% ATTENTION, CE CODE UTILISE UNE FORMULE DE CALCUL DE DISTANCE ERREURÉE
%
%%%%%%%%%%%%%
clc; close all; clear;
load workspace;

% conversion degré minutes (dm) en degré décimale (dd)
data_1m_dd.Latitude = fix(data_1m_dm.Latitude/100) + (data_1m_dm.Latitude - fix(
(data_1m_dm.Latitude/100)*100)/60;
data_1m_dd.Longitude = fix(data_1m_dm.Longitude/100) + (data_1m_dm.Longitude - fix(
(data_1m_dm.Longitude/100)*100)/60;
data_1m_dd.Time = data_1m_dm.Time;

data_2m_dd.Latitude = fix(data_2m_dm.Latitude/100) + (data_2m_dm.Latitude - fix(
(data_2m_dm.Latitude/100)*100)/60;
data_2m_dd.Longitude = fix(data_2m_dm.Longitude/100) + (data_2m_dm.Longitude - fix(
(data_2m_dm.Longitude/100)*100)/60;
data_2m_dd.Time = data_2m_dm.Time;

data_5m_dd.Latitude = fix(data_5m_dm.Latitude/100) + (data_5m_dm.Latitude - fix(
(data_5m_dm.Latitude/100)*100)/60;
data_5m_dd.Longitude = fix(data_5m_dm.Longitude/100) + (data_5m_dm.Longitude - fix(
(data_5m_dm.Longitude/100)*100)/60;
data_5m_dd.Time = data_5m_dm.Time;

data_10m_dd.Latitude = fix(data_10m_dm.Latitude/100) + (data_10m_dm.Latitude - fix(
(data_10m_dm.Latitude/100)*100)/60;
data_10m_dd.Longitude = fix(data_10m_dm.Longitude/100) + (data_10m_dm.Longitude - fix(
(data_10m_dm.Longitude/100)*100)/60;
data_10m_dd.Time = data_10m_dm.Time;

data_15m_dd.Latitude = fix(data_15m_dm.Latitude/100) + (data_15m_dm.Latitude - fix(
(data_15m_dm.Latitude/100)*100)/60;
data_15m_dd.Longitude = fix(data_15m_dm.Longitude/100) + (data_15m_dm.Longitude - fix(
(data_15m_dm.Longitude/100)*100)/60;
data_15m_dd.Time = data_15m_dm.Time;

data_20m_dd.Latitude = fix(data_20m_dm.Latitude/100) + (data_20m_dm.Latitude - fix(
(data_20m_dm.Latitude/100)*100)/60;
data_20m_dd.Longitude = fix(data_20m_dm.Longitude/100) + (data_20m_dm.Longitude - fix(
(data_20m_dm.Longitude/100)*100)/60;
data_20m_dd.Time = data_20m_dm.Time;

%calcul des moyennes
moy_1m_lat_dd = mean(data_1m_dd.Latitude);
moy_1m_lon_dd = mean(data_1m_dd.Longitude);

moy_2m_lat_dd = mean(data_2m_dd.Latitude);
moy_2m_lon_dd = mean(data_2m_dd.Longitude);
```

```
moy_5m_lat_dd = mean(data_5m_dd.Latitude);
moy_5m_lon_dd = mean(data_5m_dd.Longitude);

moy_10m_lat_dd = mean(data_10m_dd.Latitude);
moy_10m_lon_dd = mean(data_10m_dd.Longitude);

moy_15m_lat_dd = mean(data_15m_dd.Latitude);
moy_15m_lon_dd = mean(data_15m_dd.Longitude);

moy_20m_lat_dd = mean(data_20m_dd.Latitude);
moy_20m_lon_dd = mean(data_20m_dd.Longitude);

% axes du graphique
origine_x = 5.9642562;
step_x = 2e-7;
fin_x = 5.9642578;
origine_y = 46.146922;
step_y = 1e-6;
fin_y = 46.146928;

% calcul des distances de la grille
lat_1 = origine_y;
lat_2 = origine_y + step_y;

lon_1 = origine_x;
lon_2 = origine_x + step_x;

%distance d'un carré du plot en longitude
%depart : lat_1 lon_1
%arrivee : lat_1 lon_2

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(lon_2-lon_1))) * 1851.85;

%distance d'un carré du plot en latitude
%depart : lat_1 lon_1
%arrivee : lat_2 lon_1

distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(lon_1-lon_1))) * 1851.85;

markersize = 10;
linewidth = 3;

%affichage vue générale
figure(1);
hold on
p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_1m_lon_dd, moy_1m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_2m_dd.Longitude, data_2m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_2m_lon_dd, moy_2m_lat_dd, 'xb');
```

```

p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_5m_dd.Longitude, data_5m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_5m_lon_dd, moy_5m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_10m_dd.Longitude, data_10m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_10m_lon_dd, moy_10m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_15m_dd.Longitude, data_15m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_15m_lon_dd, moy_15m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

p = plot(data_20m_dd.Longitude, data_20m_dd.Latitude, 'ro');
p.MarkerSize = markersize;
p.LineWidth=linewidth;
p = plot(moy_20m_lon_dd, moy_20m_lat_dd, 'xb');
p.MarkerSize = markersize;
p.LineWidth=linewidth;

xlabel('Longitude [°], format DD')
ylabel('Latitude [°], format DD')
title("Mesure de points distant de 1 à 20 m de la base")
legend('Mesures ','Moyenne')
hold off
grid on;

% axes du graphique
step_x = 0.0000001;
step_y = 00.000001;
nb_step_x = 19;
nb_step_y = 4;

%affichage dispersions de chaque mesure
figure(2);

subplot(6,1,1);
x_start = 5.95927565;
y_start = 46.1911451;

lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 2*step_x;

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(

```

```
(lon_2-lon_1)) * 1851.85;
distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_1)) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd,
'xb',[lon_1 lon_2], [lat_1 lat_1], 'r', [lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 1 m")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100), ' cm'],
['Echelle : ',num2str(distance_carre_verticale*100), ' cm'])

subplot(6,1,2);
p = plot(data_2m_dd.Longitude, data_2m_dd.Latitude, 'ro',moy_2m_lon_dd, moy_2m_lat_dd,
'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.95928165;
y_start = 46.1911532;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 2 m")

subplot(6,1,3);
p = plot(data_5m_dd.Longitude, data_5m_dd.Latitude, 'ro',moy_5m_lon_dd, moy_5m_lat_dd,
'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.95929899;
y_start = 46.1911772;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 5 m")

subplot(6,1,4);
p = plot(data_10m_dd.Longitude, data_10m_dd.Latitude, 'ro',moy_10m_lon_dd,
moy_10m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.95932799;
y_start = 46.1912174;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
```

```

grid on
set(gca,'FontSize',15)
ylabel('Latitude [°], format DD')
title("Dispersion des points à 10 m")

subplot(6,1,5);
p = plot(data_15m_dd.Longitude, data_15m_dd.Latitude, 'ro',moy_15m_lon_dd,'
moy_15m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.95935699;
y_start = 46.1912576;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 15 m")

subplot(6,1,6);
p = plot(data_20m_dd.Longitude, data_20m_dd.Latitude, 'ro',moy_20m_lon_dd,'
moy_20m_lat_dd, 'xb');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
x_start = 5.95938599;
y_start = 46.1912977;
axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("Dispersion des points à 20 m")
xlabel('Longitude [°], format DD')

% distance entre la moyenne à 1m et à 2m
distance_moyenne_1_2 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_2m_lat_dd)+cos(moy_1m_lat_dd)*
*cos(moy_2m_lat_dd)*cos(moy_2m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 5m
distance_moyenne_1_5 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_5m_lat_dd)+cos(moy_1m_lat_dd)*
*cos(moy_5m_lat_dd)*cos(moy_5m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 10m
distance_moyenne_1_10 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_10m_lat_dd)+cos(moy_1m_lat_dd)*
*cos(moy_10m_lat_dd)*cos(moy_10m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 15m
distance_moyenne_1_15 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_15m_lat_dd)+cos(moy_1m_lat_dd)*
*cos(moy_15m_lat_dd)*cos(moy_15m_lon_dd-moy_1m_lon_dd))) * 1851.85;

% distance entre la moyenne à 1m et à 20m
distance_moyenne_1_20 = (60*acos(sin(moy_1m_lat_dd)*sin(moy_20m_lat_dd)+cos(moy_1m_lat_dd)*
*cos(moy_20m_lat_dd)*cos(moy_20m_lon_dd-moy_1m_lon_dd))) * 1851.85;

distance_moyenne = [distance_moyenne_1_2 distance_moyenne_1_5 distance_moyenne_1_10
distance_moyenne_1_15 distance_moyenne_1_20]

%calcul de la distance entre chaque point et la moyenne des mesures à

```

```
%chaque distances
distance_pt_moy_1m = zeros(size(data_1m_dd.Latitude));
for i=1:size(data_1m_dd.Latitude,1)
    distance_pt_moy_1m(i) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(moy_1m_lat_dd)*
+cos(data_1m_dd.Latitude(i))*cos(moy_1m_lat_dd)*cos(data_1m_dd.Longitude(i)-
moy_1m_lon_dd)) *1851.85;
end
dis_max_moy_1m = max(distance_pt_moy_1m);
ecart_type_moy_1m = std(distance_pt_moy_1m);

distance_pt_moy_2m = zeros(size(data_2m_dd.Latitude));
for i=1:size(data_2m_dd.Latitude,1)
    distance_pt_moy_2m(i) = (60*acos(sin(data_2m_dd.Latitude(i))*sin(moy_2m_lat_dd)*
+cos(data_2m_dd.Latitude(i))*cos(moy_2m_lat_dd)*cos(data_2m_dd.Longitude(i)-
moy_2m_lon_dd)) *1851.85;
end
dis_max_moy_2m = max(distance_pt_moy_2m);
ecart_type_moy_2m = std(distance_pt_moy_2m);

distance_pt_moy_5m = zeros(size(data_5m_dd.Latitude));
for i=1:size(data_5m_dd.Latitude,1)
    distance_pt_moy_5m(i) = (60*acos(sin(data_5m_dd.Latitude(i))*sin(moy_5m_lat_dd)*
+cos(data_5m_dd.Latitude(i))*cos(moy_5m_lat_dd)*cos(data_5m_dd.Longitude(i)-
moy_5m_lon_dd)) *1851.85;
end
dis_max_moy_5m = max(distance_pt_moy_5m);
ecart_type_moy_5m = std(distance_pt_moy_5m);

distance_pt_moy_10m = zeros(size(data_10m_dd.Latitude));
for i=1:size(data_10m_dd.Latitude,1)
    distance_pt_moy_10m(i) = (60*acos(sin(data_10m_dd.Latitude(i))*sin(
(moy_10m_lat_dd)+cos(data_10m_dd.Latitude(i))*cos(moy_10m_lat_dd)*cos(data_10m_dd.-
Longitude(i)-moy_10m_lon_dd))) *1851.85;
end
dis_max_moy_10m = max(distance_pt_moy_10m);
ecart_type_moy_10m = std(distance_pt_moy_10m);

distance_pt_moy_15m = zeros(size(data_15m_dd.Latitude));
for i=1:size(data_15m_dd.Latitude,1)
    distance_pt_moy_15m(i) = (60*acos(sin(data_15m_dd.Latitude(i))*sin(
(moy_15m_lat_dd)+cos(data_15m_dd.Latitude(i))*cos(moy_15m_lat_dd)*cos(data_15m_dd.-
Longitude(i)-moy_15m_lon_dd))) *1851.85;
end
dis_max_moy_15m = max(distance_pt_moy_15m);
ecart_type_moy_15m = std(distance_pt_moy_15m);

distance_pt_moy_20m = zeros(size(data_20m_dd.Latitude));
for i=1:size(data_20m_dd.Latitude,1)
    distance_pt_moy_20m(i) = (60*acos(sin(data_20m_dd.Latitude(i))*sin(
(moy_20m_lat_dd)+cos(data_20m_dd.Latitude(i))*cos(moy_20m_lat_dd)*cos(data_20m_dd.-
Longitude(i)-moy_20m_lon_dd))) *1851.85;
end
dis_max_moy_20m = max(distance_pt_moy_20m);
ecart_type_moy_20m = std(distance_pt_moy_20m);

dis_max_moy = [dis_max_moy_1m dis_max_moy_2m dis_max_moy_5m dis_max_moy_10m
dis_max_moy_15m dis_max_moy_20m]
ecart_type_moy = [ecart_type_moy_1m ecart_type_moy_2m ecart_type_moy_5m]
```

```
ecart_type_moy_10m ecart_type_moy_15m ecart_type_moy_20m]

step_dist = 0.0005;
nb_step_0_4_cm = 9;
nb_step_2_5_cm = 51;

% histogramme de la dispersion des mesures
figure(3);
subplot(6,1,1);
echelle_m = 0:step_dist:0.025;
[ret_histc] = histc(distance_pt_moy_1m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 1 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_1m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_1m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,2);
[ret_histc] = histc(distance_pt_moy_2m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 2 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_2m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_2m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,3);
[ret_histc] = histc(distance_pt_moy_5m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 5 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_5m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_5m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,4);
[ret_histc] = histc(distance_pt_moy_10m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
ylabel("Nombre de mesures")
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 10 m")
set(gca,'FontSize',15)
grid on;
grid minor;
cercle_0_4_cm_10m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_10m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,5);
[ret_histc] = histc(distance_pt_moy_15m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
```

```

title("Dispersion des 1200 mesures par rapport à moyenne, distance : 15 m")
set(gca,'FontSize',15)
grid on;
grid minor
cercle_0_4_cm_15m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_15m = sum(ret_histc(1:nb_step_2_5_cm));

subplot(6,1,6);
[ret_histc] = histc(distance_pt_moy_20m,echelle_m);
tot_points_on_graph = sum(ret_histc);
bar(echelle_m,ret_histc,'histc');
xlabel("Distance [m]")
title("Dispersion des 1200 mesures par rapport à moyenne, distance : 20 m")
set(gca,'FontSize',15)
grid on;
grid minor
cercle_0_4_cm_20m = sum(ret_histc(1:nb_step_0_4_cm));
cercle_2_5_cm_20m = sum(ret_histc(1:nb_step_2_5_cm));

pourcent_cercle_0_4_cm = [cercle_0_4_cm_1m cercle_0_4_cm_2m cercle_0_4_cm_5m
cercle_0_4_cm_10m cercle_0_4_cm_15m cercle_0_4_cm_20m]/1200*100
pourcent_cercle_2_5_cm = [cercle_2_5_cm_1m cercle_2_5_cm_2m cercle_2_5_cm_5m
cercle_2_5_cm_10m cercle_2_5_cm_15m cercle_2_5_cm_20m]/1200*100

distance_1m_2m = zeros(size(data_1m_dd.Latitude,1), size(data_2m_dd.Latitude,1));
distance_1m_5m = zeros(size(data_1m_dd.Latitude,1), size(data_5m_dd.Latitude,1));
distance_1m_10m = zeros(size(data_1m_dd.Latitude,1), size(data_10m_dd.Latitude,1));
distance_1m_15m = zeros(size(data_1m_dd.Latitude,1), size(data_15m_dd.Latitude,1));
distance_1m_20m = zeros(size(data_1m_dd.Latitude,1), size(data_20m_dd.Latitude,1));

% distance de chaque points de départ (1m) vers chaque point d'arrivé (2,5,10,15,20 m)
for i=1:size(data_1m_dd.Latitude,1)
    for j=1:size(data_2m_dd.Latitude,1)
        distance_1m_2m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_2m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_2m_dd.Latitude(j))*cos(data_2m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_5m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_5m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_5m_dd.Latitude(j))*cos(data_5m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_10m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_10m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_10m_dd.Latitude(j))*cos(data_10m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_15m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_15m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_15m_dd.Latitude(j))*cos(data_15m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
        distance_1m_20m(i,j) = (60*acos(sin(data_1m_dd.Latitude(i))*sin(data_20m_dd.
Latitude(j))+cos(data_1m_dd.Latitude(i))*cos(data_20m_dd.Latitude(j))*cos(data_20m_dd.
Longitude(j)-data_1m_dd.Longitude(i)))) *1851.85;
    end
end

dist_1m_2m = reshape(distance_1m_2m,[],1);
dist_1m_5m = reshape(distance_1m_5m,[],1);
dist_1m_10m = reshape(distance_1m_10m,[],1);
dist_1m_15m = reshape(distance_1m_15m,[],1);
dist_1m_20m = reshape(distance_1m_20m,[],1);
dist_pts = [dist_1m_2m dist_1m_5m dist_1m_10m dist_1m_15m dist_1m_20m];

```

```

ecart_type_dist_1m_2m = std(dist_1m_2m);
ecart_type_dist_1m_5m = std(dist_1m_5m);
ecart_type_dist_1m_10m = std(dist_1m_10m);
ecart_type_dist_1m_15m = std(dist_1m_15m);
ecart_type_dist_1m_20m = std(dist_1m_20m);

distance_max_1m_2m = max(dist_1m_2m);
distance_max_1m_5m = max(dist_1m_5m);
distance_max_1m_10m = max(dist_1m_10m);
distance_max_1m_15m = max(dist_1m_15m);
distance_max_1m_20m = max(dist_1m_20m);
distance_max = [distance_max_1m_2m distance_max_1m_5m distance_max_1m_10m<
distance_max_1m_15m distance_max_1m_20m];

distance_min_1m_2m = min(dist_1m_2m);
distance_min_1m_5m = min(dist_1m_5m);
distance_min_1m_10m = min(dist_1m_10m);
distance_min_1m_15m = min(dist_1m_15m);
distance_min_1m_20m = min(dist_1m_20m);
distance_min = [distance_min_1m_2m distance_min_1m_5m distance_min_1m_10m<
distance_min_1m_15m distance_min_1m_20m];

step_dist = 0.0005;
dist = 0.5;

% histogramme des distances entre chaque points
figure(4);
tab_dist = [ 2-1 5-1 10-1 15-1 20-1];

for i=1:5

    start_ech = distance_min(:,i) - step_dist;
    stop_ech = distance_max(:,i) + step_dist;
    echelle_m = start_ech-step_dist:step_dist:stop_ech+step_dist;
    [ret_histc] = histc(dist_pts(:,i),echelle_m);

    % verification que tous les points sont dans l'histogramme 3600*3600
    tot_points_on_graph = sum(ret_histc) ;

    % affichage
    subplot(5,1,i);
    bar(echelle_m,ret_histc,'stacked');
    xlabel("Distance [m]")
    if i == 3
        ylabel("Nombre de mesures")
    end
    title(['Distance obtenu à parti du point situé à 1 m, soit une distance de : ',<
num2str(tab_dist(i)), ' m'])
    set(gca,'FontSize',15)
    grid on;
    grid minor

end

```

```
%%%%%%%%%%%%%
%
% comp_dist_min_avec_sans_erreurs.m
% Aurélien Berthelot
% Code pour montrer l'impact des arrondis dans matlab
%
%%%%%%%%%%%%%
clc; close all; clear;
load workspace

% conversion degré minutes (dm) en degré décimale (dd)
data_1m_dd.Latitude = fix(data_1m_dm.Latitude/100) + (data_1m_dm.Latitude - fix(
(data_1m_dm.Latitude/100)*100)/60;
data_1m_dd.Longitude = fix(data_1m_dm.Longitude/100) + (data_1m_dm.Longitude - fix(
(data_1m_dm.Longitude/100)*100)/60;
data_1m_dd.Time = data_1m_dm.Time;

% calcul des moyennes
moy_1m_lat_dd = mean(data_1m_dd.Latitude);
moy_1m_lon_dd = mean(data_1m_dd.Longitude);

% axes du graphique
origine_x = 5.9642562;
step_x = 2e-7;
fin_x = 5.9642578;
origine_y = 46.146922;
step_y = 1e-6;
fin_y = 46.146928;

% calcul des distances de la grille
lat_1 = origine_y;
lat_2 = origine_y + step_y;

lon_1 = origine_x;
lon_2 = origine_x + step_x;

% distance d'un carré du plot en longitude
% départ : lat_1 lon_1
% arrivée : lat_1 lon_2

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(
(lon_2-lon_1))) * 1851.85;

% distance d'un carré du plot en latitude
% départ : lat_1 lon_1
% arrivée : lat_2 lon_1

distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_1))) * 1851.85;

markersize = 10;
linewidth = 3;

step_x = 0.00000001;
step_y = 00.000001;
nb_step_x = 19;
nb_step_y = 4;
```

```
% affichage et changement d'echelle a chaque graph (formule eronée)
figure();
subplot(6,1,1);
x_start = 5.95927565;
y_start = 46.1911451;

lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 4*step_x;

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(lon_2-lon_1))) * 1851.85;
distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(lon_1-lon_1))) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd, 'xb',[lon_1 lon_2], [lat_1 lat_1], 'r', [lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("1. Echelle longitude avec formule erronée, echelle longitude = 4 * 1e-8 deg")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100), ' cm'], ['Echelle : ',num2str(distance_carre_verticale*100), ' cm'])

subplot(6,1,2);
lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 2*step_x;

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(lon_2-lon_1))) * 1851.85;
distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(lon_1-lon_1))) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd, 'xb',[lon_1 lon_2], [lat_1 lat_1], 'r', [lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
```

```

title("2. Echelle longitude avec formule erronée, echelle longitude = 2 * 1e-8 deg")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100),' cm'],  

['Echelle : ',num2str(distance_carre_verticale*100),' cm'])

subplot(6,1,3);
lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 1*step_x;

distance_carre_horizontale = (60*acos(sin(lat_1)*sin(lat_1)+cos(lat_1)*cos(lat_1)*cos(
(lon_2-lon_1))) * 1851.85;
distance_carre_verticale = (60*acos(sin(lat_1)*sin(lat_2)+cos(lat_1)*cos(lat_2)*cos(
(lon_1-lon_2))) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd,  

'xb',[lon_1 lon_2], [lat_1 lat_1], 'r' ,[lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("3. Echelle longitude avec formule erronée, echelle longitude = 1 * 1e-8 deg")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100),' cm'],  

['Echelle : ',num2str(distance_carre_verticale*100),' cm'])

% changement de formule (formule corrigée)
subplot(6,1,4);
lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 1*step_x;

distance_carre_horizontale = (60*acosd(sind(lat_1)*sind(lat_1)+cosd(lat_1)*cosd(lat_1)*
*cosd(lon_2-lon_1))) * 1851.85;
distance_carre_verticale = (60*acosd(sind(lat_1)*sind(lat_2)+cosd(lat_1)*cosd(lat_2)*cosd(
(lon_1-lon_2))) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd,  

'xb',[lon_1 lon_2], [lat_1 lat_1], 'r' ,[lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("4. Echelle longitude avec formule corrigée, echelle longitude = 1 * 1e-8 deg")

```

```

legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100),' cm'],  

['Echelle : ',num2str(distance_carre_verticale*100),' cm'])

subplot(6,1,5);
lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 14*step_x;

distance_carre_horizontale = (60*cosd(sind(lat_1)*sind(lat_1)+cosd(lat_1)*cosd(lat_1))  

*cosd(lon_2-lon_1)) * 1851.85;
distance_carre_verticale = (60*cosd(sind(lat_1)*sind(lat_2)+cosd(lat_1)*cosd(lat_2)*cosd  

(lon_1-lon_1)) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd,  

'xb',[lon_1 lon_2], [lat_1 lat_1], 'r' ,[lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("5. Echelle longitude avec formule corrigée, echelle longitude = 14 * 1e-8 deg")
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100),' cm'],  

['Echelle : ',num2str(distance_carre_verticale*100),' cm'])

subplot(6,1,6);
lat_1 = y_start +1*step_y ;
lat_2 = lat_1 + 2*step_y;

lon_1 = x_start + step_x;
lon_2 = lon_1 + 61*step_x;

distance_carre_horizontale = (60*cosd(sind(lat_1)*sind(lat_1)+cosd(lat_1)*cosd(lat_1))  

*cosd(lon_2-lon_1)) * 1851.85;
distance_carre_verticale = (60*cosd(sind(lat_1)*sind(lat_2)+cosd(lat_1)*cosd(lat_2)*cosd  

(lon_1-lon_1)) * 1851.85;

p = plot(data_1m_dd.Longitude, data_1m_dd.Latitude, 'ro',moy_1m_lon_dd, moy_1m_lat_dd,  

'xb',[lon_1 lon_2], [lat_1 lat_1], 'r' ,[lon_1 lon_1], [lat_1 lat_2], 'b');
p(1).MarkerSize = markersize;
p(2).MarkerSize = markersize;
p(1).LineWidth = linewidth;
p(2).LineWidth = linewidth;
p(3).LineWidth=linewidth*2;
p(4).LineWidth=linewidth*2;

axis([x_start x_start+nb_step_x*step_x y_start y_start+nb_step_y*step_y])
grid on
set(gca,'FontSize',15)
title("6. Echelle longitude avec formule corrigée, echelle longitude = 61 * 1e-8 deg")

```

06/06/20 18:13 /Users.../comp\_dist\_min\_avec\_sans\_erreurs.m 5 of 5

```
legend('Mesures ','Moyenne',['Echelle : ',num2str(distance_carre_horizontale*100),' cm'],  
['Echelle : ',num2str(distance_carre_verticale*100),' cm'])
```

```
%%%%%%%%%%%%%
%
% comp_dist_1_a_20_m_avec_sans_erreurs.m
% Aurélien Berthelot
% Code pour comparer les précision avec les formules de calcul de distances
% éronnée et corrigée
%
%%%%%%%%%%%%%
clc;clear; close all;

load workspace_jardin.mat
load workspace_sans_obstacles.mat

% conversion degré minutes (dm) en degré décimale (dd)
data_1m_dd_j.Latitude = fix(data_1m_dm_j.Latitude/100) + (data_1m_dm_j.Latitude - fix(
    (data_1m_dm_j.Latitude/100)*100)/60;
data_1m_dd_j.Longitude = fix(data_1m_dm_j.Longitude/100) + (data_1m_dm_j.Longitude - fix(
    (data_1m_dm_j.Longitude/100)*100)/60;
data_1m_dd_j.Time = data_1m_dm_j.Time;

data_2m_dd_j.Latitude = fix(data_2m_dm_j.Latitude/100) + (data_2m_dm_j.Latitude - fix(
    (data_2m_dm_j.Latitude/100)*100)/60;
data_2m_dd_j.Longitude = fix(data_2m_dm_j.Longitude/100) + (data_2m_dm_j.Longitude - fix(
    (data_2m_dm_j.Longitude/100)*100)/60;
data_2m_dd_j.Time = data_2m_dm_j.Time;

data_5m_dd_j.Latitude = fix(data_5m_dm_j.Latitude/100) + (data_5m_dm_j.Latitude - fix(
    (data_5m_dm_j.Latitude/100)*100)/60;
data_5m_dd_j.Longitude = fix(data_5m_dm_j.Longitude/100) + (data_5m_dm_j.Longitude - fix(
    (data_5m_dm_j.Longitude/100)*100)/60;
data_5m_dd_j.Time = data_5m_dm_j.Time;

data_10m_dd_j.Latitude = fix(data_10m_dm_j.Latitude/100) + (data_10m_dm_j.Latitude - fix(
    (data_10m_dm_j.Latitude/100)*100)/60;
data_10m_dd_j.Longitude = fix(data_10m_dm_j.Longitude/100) + (data_10m_dm_j.Longitude - fix(
    (data_10m_dm_j.Longitude/100)*100)/60;
data_10m_dd_j.Time = data_10m_dm_j.Time;

data_15m_dd_j.Latitude = fix(data_15m_dm_j.Latitude/100) + (data_15m_dm_j.Latitude - fix(
    (data_15m_dm_j.Latitude/100)*100)/60;
data_15m_dd_j.Longitude = fix(data_15m_dm_j.Longitude/100) + (data_15m_dm_j.Longitude - fix(
    (data_15m_dm_j.Longitude/100)*100)/60;
data_15m_dd_j.Time = data_15m_dm_j.Time;

data_20m_dd_j.Latitude = fix(data_20m_dm_j.Latitude/100) + (data_20m_dm_j.Latitude - fix(
    (data_20m_dm_j.Latitude/100)*100)/60;
data_20m_dd_j.Longitude = fix(data_20m_dm_j.Longitude/100) + (data_20m_dm_j.Longitude - fix(
    (data_20m_dm_j.Longitude/100)*100)/60;
data_20m_dd_j.Time = data_20m_dm_j.Time;

%calcul des moyennes
moy_1m_lat_dd_j = mean(data_1m_dd_j.Latitude);
moy_1m_lon_dd_j = mean(data_1m_dd_j.Longitude);

moy_2m_lat_dd_j = mean(data_2m_dd_j.Latitude);
moy_2m_lon_dd_j = mean(data_2m_dd_j.Longitude);
```

```

moy_5m_lat_dd_j = mean(data_5m_dd_j.Latitude);
moy_5m_lon_dd_j = mean(data_5m_dd_j.Longitude);

moy_10m_lat_dd_j = mean(data_10m_dd_j.Latitude);
moy_10m_lon_dd_j = mean(data_10m_dd_j.Longitude);

moy_15m_lat_dd_j = mean(data_15m_dd_j.Latitude);
moy_15m_lon_dd_j = mean(data_15m_dd_j.Longitude);

moy_20m_lat_dd_j = mean(data_20m_dd_j.Latitude);
moy_20m_lon_dd_j = mean(data_20m_dd_j.Longitude);

% conversion degré minutes (dm) en degré décimale (dd)
data_1m_dd_s_o.Latitude = fix(data_1m_dm_s_o.Latitude/100) + (data_1m_dm_s_o.Latitude - fix(data_1m_dm_s_o.Latitude/100)*100)/60;
data_1m_dd_s_o.Longitude = fix(data_1m_dm_s_o.Longitude/100) + (data_1m_dm_s_o.Longitude - fix(data_1m_dm_s_o.Longitude/100)*100)/60;
data_1m_dd_s_o.Time = data_1m_dm_s_o.Time;

data_2m_dd_s_o.Latitude = fix(data_2m_dm_s_o.Latitude/100) + (data_2m_dm_s_o.Latitude - fix(data_2m_dm_s_o.Latitude/100)*100)/60;
data_2m_dd_s_o.Longitude = fix(data_2m_dm_s_o.Longitude/100) + (data_2m_dm_s_o.Longitude - fix(data_2m_dm_s_o.Longitude/100)*100)/60;
data_2m_dd_s_o.Time = data_2m_dm_s_o.Time;

data_5m_dd_s_o.Latitude = fix(data_5m_dm_s_o.Latitude/100) + (data_5m_dm_s_o.Latitude - fix(data_5m_dm_s_o.Latitude/100)*100)/60;
data_5m_dd_s_o.Longitude = fix(data_5m_dm_s_o.Longitude/100) + (data_5m_dm_s_o.Longitude - fix(data_5m_dm_s_o.Longitude/100)*100)/60;
data_5m_dd_s_o.Time = data_5m_dm_s_o.Time;

data_10m_dd_s_o.Latitude = fix(data_10m_dm_s_o.Latitude/100) + (data_10m_dm_s_o.Latitude - fix(data_10m_dm_s_o.Latitude/100)*100)/60;
data_10m_dd_s_o.Longitude = fix(data_10m_dm_s_o.Longitude/100) + (data_10m_dm_s_o.Longitude - fix(data_10m_dm_s_o.Longitude/100)*100)/60;
data_10m_dd_s_o.Time = data_10m_dm_s_o.Time;

data_15m_dd_s_o.Latitude = fix(data_15m_dm_s_o.Latitude/100) + (data_15m_dm_s_o.Latitude - fix(data_15m_dm_s_o.Latitude/100)*100)/60;
data_15m_dd_s_o.Longitude = fix(data_15m_dm_s_o.Longitude/100) + (data_15m_dm_s_o.Longitude - fix(data_15m_dm_s_o.Longitude/100)*100)/60;
data_15m_dd_s_o.Time = data_15m_dm_s_o.Time;

data_20m_dd_s_o.Latitude = fix(data_20m_dm_s_o.Latitude/100) + (data_20m_dm_s_o.Latitude - fix(data_20m_dm_s_o.Latitude/100)*100)/60;
data_20m_dd_s_o.Longitude = fix(data_20m_dm_s_o.Longitude/100) + (data_20m_dm_s_o.Longitude - fix(data_20m_dm_s_o.Longitude/100)*100)/60;
data_20m_dd_s_o.Time = data_20m_dm_s_o.Time;

%calcul des moyennes
moy_1m_lat_dd_s_o = mean(data_1m_dd_s_o.Latitude);
moy_1m_lon_dd_s_o = mean(data_1m_dd_s_o.Longitude);

moy_2m_lat_dd_s_o = mean(data_2m_dd_s_o.Latitude);
moy_2m_lon_dd_s_o = mean(data_2m_dd_s_o.Longitude);

moy_5m_lat_dd_s_o = mean(data_5m_dd_s_o.Latitude);
moy_5m_lon_dd_s_o = mean(data_5m_dd_s_o.Longitude);

```

```

moy_10m_lat_dd_s_o = mean(data_10m_dd_s_o.Latitude);
moy_10m_lon_dd_s_o = mean(data_10m_dd_s_o.Longitude);

moy_15m_lat_dd_s_o = mean(data_15m_dd_s_o.Latitude);
moy_15m_lon_dd_s_o = mean(data_15m_dd_s_o.Longitude);

moy_20m_lat_dd_s_o = mean(data_20m_dd_s_o.Latitude);
moy_20m_lon_dd_s_o = mean(data_20m_dd_s_o.Longitude);

%% calcul distance jardin formule erronée

distance_moyenne_1_2_j = (60*acos(sin(moy_1m_lat_dd_j)*sin(moy_2m_lat_dd_j)+cos(
(moy_1m_lat_dd_j)*cos(moy_2m_lat_dd_j)*cos(moy_2m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_5_j = (60*acos(sin(moy_1m_lat_dd_j)*sin(moy_5m_lat_dd_j)+cos(
(moy_1m_lat_dd_j)*cos(moy_5m_lat_dd_j)*cos(moy_5m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_10_j = (60*acos(sin(moy_1m_lat_dd_j)*sin(moy_10m_lat_dd_j)+cos(
(moy_1m_lat_dd_j)*cos(moy_10m_lat_dd_j)*cos(moy_10m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_15_j = (60*acos(sin(moy_1m_lat_dd_j)*sin(moy_15m_lat_dd_j)+cos(
(moy_1m_lat_dd_j)*cos(moy_15m_lat_dd_j)*cos(moy_15m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_20_j = (60*acos(sin(moy_1m_lat_dd_j)*sin(moy_20m_lat_dd_j)+cos(
(moy_1m_lat_dd_j)*cos(moy_20m_lat_dd_j)*cos(moy_20m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;

distance_moyenne_j = [distance_moyenne_1_2_j distance_moyenne_1_5_j
distance_moyenne_1_10_j distance_moyenne_1_15_j distance_moyenne_1_20_j]'

%% calcul distance jardin formule corrigée

distance_moyenne_1_2_j_corr = (60*acosd(sind(moy_1m_lat_dd_j)*sind(moy_2m_lat_dd_j)+cosd(
(moy_1m_lat_dd_j)*cosd(moy_2m_lat_dd_j)*cosd(moy_2m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_5_j_corr = (60*acosd(sind(moy_1m_lat_dd_j)*sind(moy_5m_lat_dd_j)+cosd(
(moy_1m_lat_dd_j)*cosd(moy_5m_lat_dd_j)*cosd(moy_5m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_10_j_corr = (60*acosd(sind(moy_1m_lat_dd_j)*sind(moy_10m_lat_dd_j)+cosd(
(moy_1m_lat_dd_j)*cosd(moy_10m_lat_dd_j)*cosd(moy_10m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_15_j_corr = (60*acosd(sind(moy_1m_lat_dd_j)*sind(moy_15m_lat_dd_j)+cosd(
(moy_1m_lat_dd_j)*cosd(moy_15m_lat_dd_j)*cosd(moy_15m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;
distance_moyenne_1_20_j_corr = (60*acosd(sind(moy_1m_lat_dd_j)*sind(moy_20m_lat_dd_j)+cosd(
(moy_1m_lat_dd_j)*cosd(moy_20m_lat_dd_j)*cosd(moy_20m_lon_dd_j-moy_1m_lon_dd_j))) * 1851.85;

distance_moyenne_j_corr = [distance_moyenne_1_2_j_corr distance_moyenne_1_5_j_corr
distance_moyenne_1_10_j_corr distance_moyenne_1_15_j_corr distance_moyenne_1_20_j_corr]

%% calcul distance sans obstacles formule erronée

distance_moyenne_1_2_s_o = (60*acos(sin(moy_1m_lat_dd_s_o)*sin(moy_2m_lat_dd_s_o)+cos(
(moy_1m_lat_dd_s_o)*cos(moy_2m_lat_dd_s_o)*cos(moy_2m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_5_s_o = (60*acos(sin(moy_1m_lat_dd_s_o)*sin(moy_5m_lat_dd_s_o)+cos(
(moy_1m_lat_dd_s_o)*cos(moy_5m_lat_dd_s_o)*cos(moy_5m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_10_s_o = (60*acos(sin(moy_1m_lat_dd_s_o)*sin(moy_10m_lat_dd_s_o)+cos(

```

```

(moy_1m_lat_dd_s_o)*cos(moy_10m_lat_dd_s_o)*cos(moy_10m_lon_dd_s_o-moy_1m_lon_dd_s_o))) *↵
1851.85;
distance_moyenne_1_15_s_o = (60*acos(sin(moy_1m_lat_dd_s_o)*sin(moy_15m_lat_dd_s_o)+cos↵
(moy_1m_lat_dd_s_o)*cos(moy_15m_lat_dd_s_o)*cos(moy_15m_lon_dd_s_o-moy_1m_lon_dd_s_o))) *↵
1851.85;
distance_moyenne_1_20_s_o = (60*acos(sin(moy_1m_lat_dd_s_o)*sin(moy_20m_lat_dd_s_o)+cos↵
(moy_1m_lat_dd_s_o)*cos(moy_20m_lat_dd_s_o)*cos(moy_20m_lon_dd_s_o-moy_1m_lon_dd_s_o))) *↵
1851.85;

distance_moyenne_s_o = [distance_moyenne_1_2_s_o distance_moyenne_1_5_s_o ↵
distance_moyenne_1_10_s_o distance_moyenne_1_15_s_o distance_moyenne_1_20_s_o]'

%% calcul distance sans obstacles formule corrigée

distance_moyenne_1_2_s_o_corr = (60*acosd(sind(moy_1m_lat_dd_s_o)*sind(moy_2m_lat_dd_s_o) ↵
+cosd(moy_1m_lat_dd_s_o)*cosd(moy_2m_lat_dd_s_o)*cosd(moy_2m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_5_s_o_corr = (60*acosd(sind(moy_1m_lat_dd_s_o)*sind(moy_5m_lat_dd_s_o) ↵
+cosd(moy_1m_lat_dd_s_o)*cosd(moy_5m_lat_dd_s_o)*cosd(moy_5m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_10_s_o_corr = (60*acosd(sind(moy_1m_lat_dd_s_o)*sind(moy_10m_lat_dd_s_o) ↵
+cosd(moy_1m_lat_dd_s_o)*cosd(moy_10m_lat_dd_s_o)*cosd(moy_10m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_15_s_o_corr = (60*acosd(sind(moy_1m_lat_dd_s_o)*sind(moy_15m_lat_dd_s_o) ↵
+cosd(moy_1m_lat_dd_s_o)*cosd(moy_15m_lat_dd_s_o)*cosd(moy_15m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;
distance_moyenne_1_20_s_o_corr = (60*acosd(sind(moy_1m_lat_dd_s_o)*sind(moy_20m_lat_dd_s_o) ↵
+cosd(moy_1m_lat_dd_s_o)*cosd(moy_20m_lat_dd_s_o)*cosd(moy_20m_lon_dd_s_o-moy_1m_lon_dd_s_o))) * 1851.85;

distance_moyenne_s_o_corr = [distance_moyenne_1_2_s_o_corr distance_moyenne_1_5_s_o_corr ↵
distance_moyenne_1_10_s_o_corr distance_moyenne_1_15_s_o_corr ↵
distance_moyenne_1_20_s_o_corr]'

```