

Implantation d'une chaîne

Compte-rendu

-

Lucas COQUARD
Antoine NOËL

Table des matières

1.	Introduction	2
2.	Fichiers de configuration	2
3.	Calcul des passages des satellites	3
3.2	Définition d'une orbite.....	3
3.3	Fichiers TLE	4
3.4	Calcul de la position géocentrique de la station	5
3.5	Calculs position topocentrique du satellite	6
3.5.1	Temps Julien	7
3.6	Implantation des différents calculs dans le logiciel	8
4.	Génération d'un planning d'observation sur 7 jours.....	9
4.2	Initialisation.....	9
4.3	Méthode Planning_Maker	9
4.4	Méthode chooseWindows	10
4.5	Méthode determineAddableWindows	10
5.	Moteur.....	12
5.2	Cas n°1.....	13
5.3	Cas n°2.....	13
5.3.1	Résolution du cas n°2.....	14
5.4	Cas n°3.....	14
5.4.1	Résolution du cas n°3.....	15
5.5	Implantation des solutions	15
5.6	Rotctld	16
5.6.1	Utilisation de Rotctld.....	16
6.	SDR	17
6.1	Paramètres du SDR	17
6.1.1	Plage de fréquences	17
6.1.2	Plage de gain.....	17
6.1.3	Plage de taux d'échantillonnage (sample rate).....	17
6.1.4	Format de données supportées	17
6.2	Format d'enregistrement	18
6.3	Test sur une fréquence radio.....	18
6.4	Enregistrement des satellites	19
6.5	Améliorations	20
7.	Conclusion	20

1. Introduction

La station s'intéresse principalement aux satellites météorologiques NOAA et METEOR, qui évoluent en orbite basse à des altitudes comprises entre 800 et 850 km. Ces satellites transmettent leurs données sur des fréquences situées entre 137 et 140 MHz. À notre arrivée sur le projet, la station de réception de satellites était en place, mais nécessitait une automatisation complète pour fonctionner de manière autonome.

Cette station comprend deux parties principales : une partie mécanique (antenne et moteur) et une partie logicielle. Initialement, le logiciel GPredict était utilisé pour contrôler le moteur et régler les fréquences de réception. Bien que performant, GPredict ne permettait de suivre qu'un seul passage de satellite à la fois, nécessitant une reconfiguration manuelle pour chaque nouveau satellite, ce qui limitait son autonomie.

Compte tenu de la situation initiale, l'objectif de ce projet est de complètement remplacer l'interface GPredict par un logiciel conçu par nos soins. Ce logiciel, écrit en python, doit être capable d'assurer les mêmes fonctions : calcul des fenêtres de visibilité des satellites, positionnement de l'antenne et récupération des données.

2. Fichiers de configuration

L'un des objectifs de ce projet est de pouvoir rendre ce logiciel portable. En effet, qu'importe le lieu de la station ou les satellites voulant être observés, le logiciel doit fonctionner. Pour cela, des fichiers de configuration sont nécessaires. Ces fichiers seront aux formats TOML (Tom's Obvious Minimal Language). Ce format de fichier de configuration est conçu pour être simple, lisible et facilement modifiable, il est donc adapté pour ce projet.

Dans ces fichiers de configurations, plusieurs éléments vont être inscrits :

- Les paramètres de la station : incluant le nom, la latitude, la longitude et l'altitude de la station
- Les paramètres du moteur : incluant l'IP, le port de connexion et l'IP du logiciel de contrôle
- Les différents dossiers d'enregistrements
- Les paramètres de téléchargement des fichiers orbitaux
- La liste des satellites avec leurs paramètres : incluant son nom, son élévation minimale recherchée, sa culmination minimale recherchée, sa fréquence, le format d'enregistrement et sa bande passante.

Maintenant que le format des fichiers et leur contenu sont définis, l'étape suivante consiste à développer un programme orienté objet capable de les interpréter. Ce programme devra extraire les informations contenues dans les fichiers et les organiser dans des objets correspondant aux éléments décrits. Pour réaliser cela, plusieurs classes de configuration sont créées afin de stocker les informations dans des objets. Parmi elles, la classe *ConfigurationReader* est développée pour initialiser les objets de configuration.

Le diagramme UML de ce programme est présenté ci-dessous :

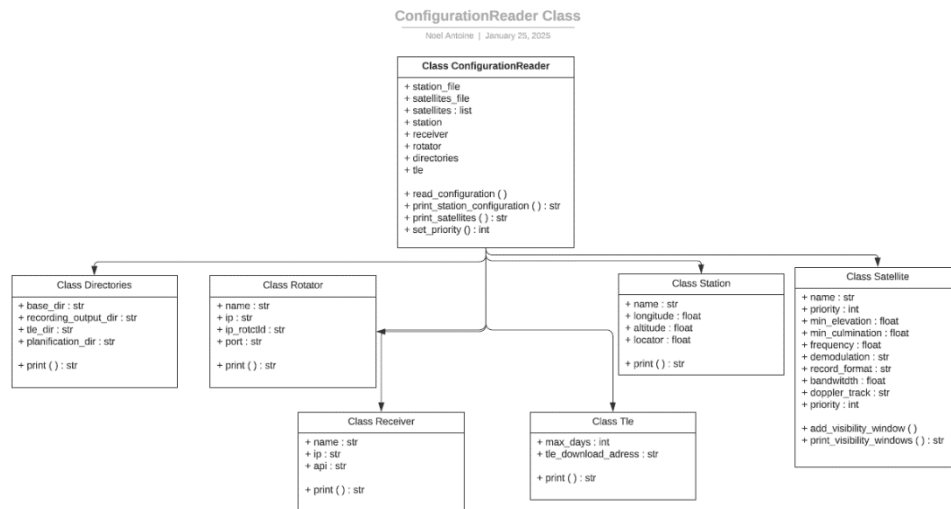


Figure 1 – Diagramme de classes UML

Comme le spécifie ce diagramme, différentes méthodes ont été implantées afin d'obtenir des informations sur les éléments qui composent chaque classe.

3. Calcul des passages des satellites

Les satellites voulant être reçus ne sont pas des satellites géostationnaires, ceux-ci possèdent une orbite.

3.2 Définition d'une orbite

L'orbite d'un satellite est le chemin qu'il suit autour d'un corps céleste, dans le cas présent : la Terre, sous l'effet de la gravité. Ce mouvement orbital est le résultat d'un équilibre entre la force gravitationnelle, qui attire le satellite vers le centre de la Terre, et la vitesse tangentielle du satellite qui le pousse à se déplacer en ligne droite. Si la vitesse est suffisante, le satellite ne s'écrase pas sur Terre, mais continue à "tomber" autour d'elle. Les satellites suivent généralement une orbite elliptique, conformément à la première loi de Kepler.

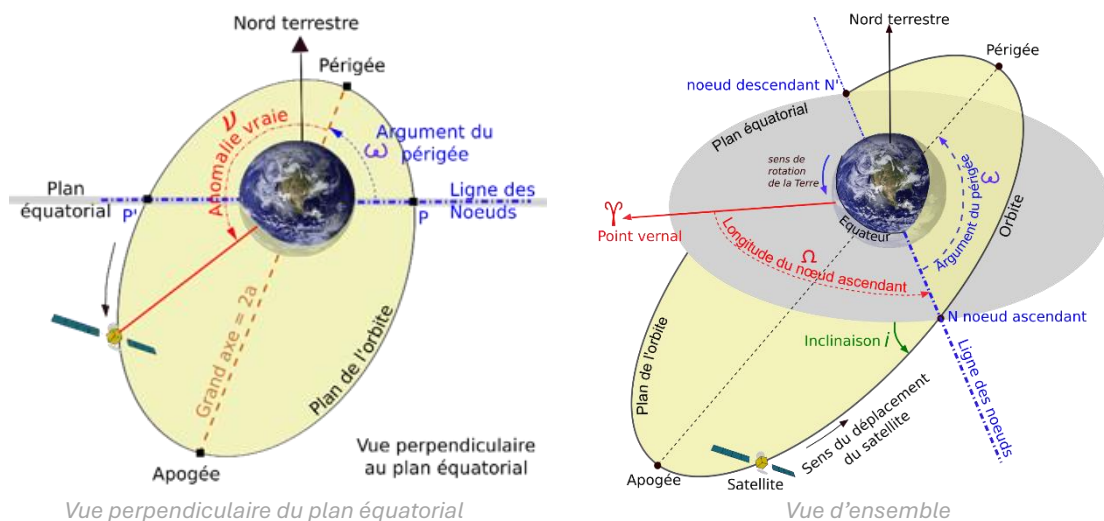


Figure 2 – Orbite d'un satellite autour de la Terre

Pour décrire avec précision cette trajectoire elliptique ou circulaire, il est nécessaire de recourir à des paramètres orbitaux. Ces paramètres permettent de définir la forme, la taille, l'orientation et la position de l'orbite dans l'espace. La liste des paramètres est présentée ci-dessous :

- Demi-grand axe
- Excentricité
- Périogée
- Apogée
- Inclinaison orbitale
- Argument du périogée
- Longitude du nœud ascendant

Le demi-grand axe représente la moitié du plus grand diamètre de l'ellipse orbitale. Il détermine la taille de l'orbite et est directement lié à la période orbitale, c'est-à-dire le temps que met le satellite pour accomplir un tour complet autour de la Terre. Plus le demi-grand axe est grand, plus la période orbitale est longue.

L'excentricité mesure à quel point l'orbite est allongée. Une excentricité de 0 correspond à une orbite parfaitement circulaire, tandis qu'une valeur proche de 1 indique une orbite très elliptique. Dans une orbite elliptique, la distance entre le satellite et la Terre varie, avec un point le plus proche appelé périogée et un point le plus éloigné appelé apogée.

L'inclinaison orbitale est un angle qui définit l'orientation du plan orbital par rapport à un plan de référence, généralement l'équateur terrestre. Une inclinaison de 0° signifie que le satellite orbite au-dessus de l'équateur, tandis qu'une inclinaison de 90° correspond à une orbite polaire, où le satellite passe au-dessus des pôles. L'inclinaison influence la couverture géographique du satellite et son interaction avec les forces gravitationnelles.

L'argument du périogée est un angle qui décrit l'orientation de l'orbite dans son plan. Il mesure la position du périogée par rapport au nœud ascendant qui est le point où le satellite traverse le plan de référence en montant. Ce paramètre est crucial pour déterminer comment l'orbite est orientée dans l'espace.

En plus de ces six paramètres orbitaux, il y a l'anomalie moyenne. L'anomalie moyenne est un angle qui indique la position du satellite sur son orbite à un instant donné. Il est calculé en supposant une orbite circulaire et permet de prédire où se trouvera le satellite à un moment précis. Ces paramètres, combinés, fournissent une description complète de l'orbite et permettent aux ingénieurs et aux scientifiques de contrôler et de prévoir le mouvement des satellites avec une grande précision.

3.3 Fichiers TLE

Afin de pouvoir suivre les satellites, il faut connaître leurs paramètres orbitaux. Ces paramètres sont définis dans des fichiers TLE (Two-Line Elements).

```
NOAA 15  
1 25338U 98030A 25016.08210717 .00000281 00000+0 13327-3 0 9990  
2 25338 98.5505 44.5894 0011213 65.1724 295.0621 14.26857031387798
```

Figure 3 – Exemple de fichier TLE pour le satellite NOAA 15

Ce format de données est utilisé pour décrire les orbites des objets en orbite autour de la Terre, comme les satellites. Il contient des informations essentielles sous forme de deux lignes de texte,

comprenant l'inclinaison, l'excentricité, la longitude du nœud ascendant, et d'autres éléments nécessaires pour calculer la position de l'objet à un moment donné. Ces paramètres sont récupérés et déterminés grâce à un processus qui allie observation et calcul. Ce processus est réalisé par des organisations spécialisées comme le NORAD (North American Aerospace Defense Command).

Les satellites en orbite sont surveillés en permanence à l'aide de radars et de télescope optiques répartis sur la terre. Ces instruments permettent de mesurer la position, la vitesse et la trajectoire d'un satellite à différents moments.

À partir de ces observations, les calculs peuvent être effectués. Ceux-ci prennent en compte les lois de la mécanique céleste, comme la première loi de Kepler et les lois du mouvement de Newton, ainsi que les perturbations orbitales causées par l'influence de la lune et du soleil ou le freinage atmosphérique.

Dû à ces perturbations, les paramètres orbitaux ne sont pas fixes dans le temps, ceux-ci peuvent varier chaque semaine. Afin d'avoir les positions les plus exactes lors des passages au-dessus de la station, il est nécessaire de télécharger les fichiers TLE tous les 7 jours.

Un programme a donc été conçu afin de pouvoir télécharger les TLE. Ce programme vérifie si le fichier correspondant au satellite existe ou si le fichier a une ancienneté supérieure à 7 jours.

Dans ce cas, le programme lance une requête de téléchargement au site celestack.com.

3.4 Calcul de la position géocentrique de la station

Une fois les paramètres orbitaux téléchargés, il faut les utiliser. Avant de commencer le calcul des passages des satellites au-dessus de la station, il faut placer la station sur des coordonnées GPS standards WGS84. Le Système géodésique mondial de 1984 est un système de référence géodésique utilisé par de nombreux appareils GPS comme base uniforme pour les informations de position sur Terre. Le WGS84 permet de relier une position tridimensionnelle (x, y, h) à une coordonnée cohérente, au format latitude (Φ), longitude (λ), hauteur ellipsoïdale (H). La latitude (Φ) représente l'angle entre l'équateur et la position, la longitude (λ) correspond à l'angle entre le méridien de Greenwich et la position, et la hauteur ellipsoïdale (H) désigne la distance verticale entre la position et la surface de l'ellipsoïde de référence ici celle du WGS84.

Les calculs permettant de convertir les coordonnées géographiques en coordonnées cartésiennes sont présentés ci-dessous :

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2(\phi)}}$$

- **a** (le demi-grand axe de l'ellipsoïde) = 6 378 137.0 m
- **e** (première excentricité de l'ellipsoïde) $\simeq$ 0.081 819
- **N** étant le rayon de la courbure dans la direction verticale

$$x = (N + h) \cos(\phi) \cos(\lambda)$$

$$y = (N + h) \cos(\phi) \sin(\lambda)$$

$$z = (N(1 - e^2) + h) \sin(\phi)$$

Avec :

- ϕ : latitude,
- λ : longitude,
- h : altitude (notre station ayant une altitude très faible, celle-ci sera égale à 0)

Ces calculs permettent de mettre la station de l'IUT de Toulon dans un référentiel géocentrique.

3.5 Calculs position topocentrique du satellite

Il faut maintenant calculer la position topocentrique du satellite. Celle-ci correspond à la position du satellite par rapport à la station au sol. Elle permet d'obtenir les informations essentielles au suivi des satellites :

- Altitude
- Azimut
- Distance

L'image ci-dessous illustre la position du satellite dans le ciel de la station de réception, tout en mettant en évidence les éléments essentiels pour le suivi des satellites:

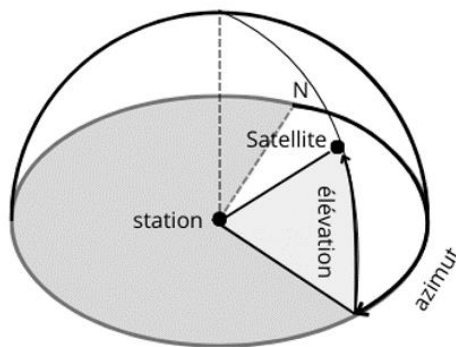


Figure 4 – Visualisation de la position topocentrique du satellite

Avant de commencer les calculs de position topocentrique, il faut dans un premier temps calculer la position géostationnaire ($x_{sat}, y_{sat}, z_{sat}$) à un temps t . Cela peut se faire grâce au TLE récupéré et des algorithmes de calcul d'orbite.

Pour obtenir la position topocentrique, il faut calculer le vecteur relatif entre le satellite et la station au sol :

$$\vec{r}_{topo} = \vec{r}_{sat} - \vec{r}_{station}$$

Avec :

- $\vec{r}_{sat}$: la position du satellite dans le référentiel géocentrique
- $\vec{r}_{station}$: la position de la station dans le référentiel géocentrique

Grâce à ce vecteur relatif 3 composantes peuvent être calculées.

La composante Nord :

Cette composante est la projection du vecteur relatif sur l'axe local pointant vers le nord géographique. Elle permet d'indiquer à quelle distance le satellite se trouve dans la direction nord ou dans la direction sud par rapport à la station.

$$N = -\sin(\phi) \cos(\lambda) x_{topo} - \sin(\phi) \sin(\lambda) y_{topo} + \cos(\phi) z_{topo}$$

Avec :

- ϕ : latitude de la station,
- λ : longitude de la station.

La composante Est

Cette composante est la projection du vecteur relatif sur l'axe local pointant vers l'Est géographique. Elle permet d'indiquer à quelle distance le satellite se trouve dans la direction Est ou ouest par rapport à la station.

$$E = -\sin(\lambda) x_{topo} + \cos(\lambda) y_{topo}$$

La composante Haut :

Cette composante est la projection du vecteur relatif sur l'axe local pointant vers le haut (direction radiale, perpendiculaire à la surface de l'ellipsoïde de référence). Elle permet d'indiquer la distance du satellite par rapport à la station et si le satellite se trouve au-dessus ou en dessous de la station.

$$U = \cos(\phi) \cos(\lambda) x_{topo} + \cos(\phi) \sin(\lambda) y_{topo} + \sin(\phi) z_{topo}$$

Grâce à ces composantes les coordonnées topocentriques peuvent être calculées :

$$distance = \|\vec{r}_{topo}\|$$

$$azimut = \arctan2(E, N)$$

$$altitude = \arcsin\left(\frac{U}{distance}\right)$$

Tous ces calculs permettent donc de calculer la position précise du satellite à un instant t dans le ciel local de la station. Cependant, l'instant t utilisé dans les calculs ne correspond pas directement au temps universel coordonné (UTC), qui est la valeur de temps utilisée au quotidien. Pour effectuer des calculs astronomiques précis, il est nécessaire de convertir cet instant en temps julien.

3.5.1 Temps Julien

Le temps Julien est un système de mesure de temps principalement utilisé en astronomie afin de s'affranchir de problèmes de chronologie à longue période comme le changement de calendrier ou de fuseaux horaires. Ce système de temps compte le nombre de jours écoulés depuis une date de référence : le 1^{er} janvier 4713 avant J-C à midi (UTC). Ce système compte le nombre de jours et de fractions de jours écoulés depuis la date de référence. Cette méthode offre une précision accrue pour les calculs de précisions tels que le positionnement d'un satellite dans le ciel.

3.6 Implantation des différents calculs dans le logiciel

Heureusement, une librairie Python contenant la totalité des calculs existe et permet de simplifier le développement du logiciel.

Cette librairie se nomme Skyfield, c'est une librairie open source conçue pour effectuer des calculs astronomiques et spatiaux. Cette librairie offre des fonctionnalités pour convertir entre différents systèmes de coordonnées, prédire des événements astronomiques (comme les levers, culmination et coucher des objets), et suivre les satellites en temps réel.

Elle est donc parfaitement adaptée pour le projet. Grâce à ces différentes fonctions, il est possible d'effectuer tous les calculs cités précédemment rapidement, et permet de commencer à calculer les passages des différents satellites au-dessus de la station de réception.

En suivant la logique de calcul présenté ci-haut, il faut dans un premier temps calculer les coordonnées géocentriques de la station, puis grâce au TLE du satellite voulu, ainsi qu'à la méthode *find_events* qui est incluse dans la librairie Skyfield, les différentes fenêtres de visibilité peuvent être trouvées. Ces fenêtres comprennent 3 heures :

- L'heure de lever (rise)
- L'heure de culmination (culminate)
- L'heure de coucher (set)

Ces passages sont calculés en utilisant le système de temps Julien tels que présenté précédemment afin de garantir la précision et la fiabilité des résultats.

L'heure de lever et de coucher correspond au moment où le satellite atteint une élévation supérieure ou égale à l'élévation minimale définie dans le fichier de configuration. L'heure de culmination, quant à elle, correspond au moment où le satellite atteint son point le plus haut dans le ciel. Cet instant est utilisé pour calculer la position topocentrique du satellite, ce qui permet de déterminer son altitude maximale lors du passage. Cette altitude maximale est ensuite comparée à la culmination maximale spécifiée dans le fichier de configuration afin d'effectuer un pré-tri. Ce tri est crucial, car il permet d'éliminer les passages où le satellite est trop bas dans le ciel, ce qui pourrait rendre la récupération du signal difficile, voire impossible.

Dans le cas où la culmination serait assez haute, le passage est ajouté dans la liste *visibility_windows* présente dans chaque objet qui correspond à un satellite. Une fois tous les passages calculés, chaque passage est associé à son satellite correspondant via la classe *Observation*. Ces objets *Observation*, qui contiennent les informations sur les passages, sont ensuite regroupés dans une liste appelée *observations*. Cette liste contient ainsi tous les passages de tous les satellites souhaités. Pour finir, la liste *observations* est triée dans l'ordre chronologique des débuts de passage des satellites, afin de faciliter la conception d'un planning.

4. Génération d'un planning d'observation sur 7 jours

Une fois, toutes les fenêtres de visibilité de tous les satellites choisis calculées, il est nécessaire d'en faire le tri. En effet, en augmentant le nombre de satellites observés, il est inévitable que plusieurs d'entre eux soit observables au même moment.

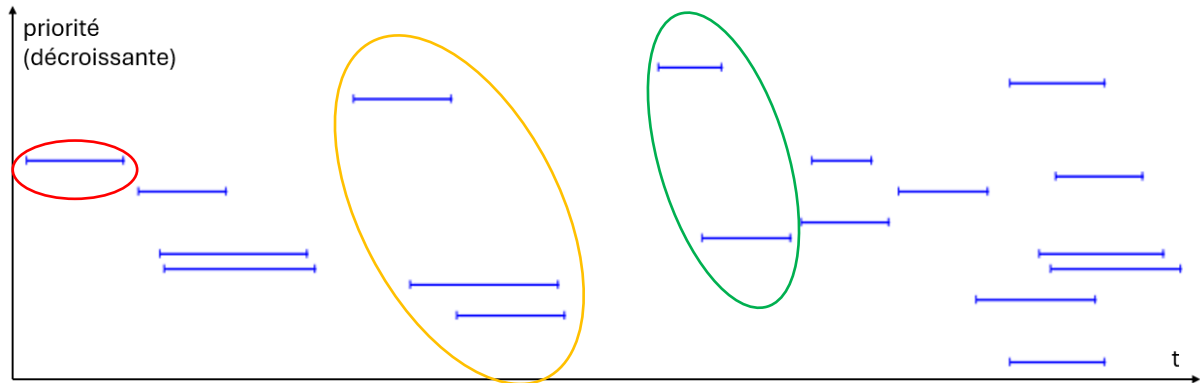


Figure 5 – Planning contenant toutes les fenêtres de visibilité et mise en évidence des différents cas

Dans la figure ci-dessus, trois cas distincts peuvent être mis en évidence. Le premier cas, en rouge, est un cas où la fenêtre de visibilité n'en chevauche aucune autre, la sélection de cette fenêtre est donc évidente. Dans le second cas, en vert, deux fenêtres se chevauchent, ce qui rend le choix plus difficile. Pour effectuer cette sélection, la priorité accordée par l'utilisateur est essentielle. Lors de la création du fichier TOML, l'ordre d'apparition des satellites va venir accorder une certaine importance aux différents satellites, le premier dans la liste étant considéré comme le plus important. Le dernier cas, en orange, est encore plus particulier. Comme dit plus haut, il est possible que plus de deux satellites soient observables en même temps. Dans ce cas, il faut pouvoir choisir, la première fenêtre ainsi que la dernière tout en excluant celle au milieu.

Le code est structuré en plusieurs parties principales. Il commence par l'importation des modules nécessaires, tels que *datetime*, *os*, et *matplotlib.pyplot*, ainsi que des modules personnalisés comme *TLE_Loader*, *VisibilyWindowComputer*, et *ConfigurationReader*. Ensuite, des constantes sont définies pour gérer l'état des observations (*IDLE*, *SELECTED*, *EXCLUDED*, *OVERLAPS_PREVIOUS*) et pour accéder aux indices des fenêtres de visibilité (*START_TIME*, *END_TIME*).

La classe principale, *Plannifier*, est le cœur du programme. Elle contient les méthodes pour sélectionner les fenêtres de visibilité, déterminer les fenêtres ajoutables, et générer un planning. Enfin, la fonction principale du script charge les configurations, calcule les fenêtres de visibilité, et génère un planning qui est ensuite visualisé sous forme de graphique.

4.2 Initialisation

La classe *Plannifier* est initialisée avec deux paramètres principaux : *observations*, qui est une liste d'observations de satellites, et *satellites*, qui est une liste de satellites. Une liste vide, *planning*, est également initialisée pour stocker les observations sélectionnées.

4.3 Méthode *Planning_Maker*

La méthode *Planning_Maker* est responsable de la création du planning. Elle commence par initialiser l'état de toutes les observations à *IDLE*. Ensuite, elle entre dans une boucle infinie où

elle appelle deux méthodes principales : *chooseWindows* pour sélectionner les fenêtres de visibilité, et *determineAddableWindows* pour déterminer si des fenêtres supplémentaires peuvent être ajoutées. La boucle se termine lorsqu'il n'y a plus de fenêtres ajoutables. Enfin, la méthode *Planning_append* est appelée pour ajouter les observations sélectionnées au planning.

4.4 Méthode chooseWindows

La méthode *chooseWindows* parcourt les observations et décide quelles fenêtres de visibilité doivent être sélectionnées en fonction de leur priorité et de leur chevauchement avec d'autres fenêtres. Si une fenêtre ne chevauche pas la suivante, elle est marquée comme *SELECTED*. En cas de chevauchement, la fenêtre avec la priorité la plus élevée est sélectionnée, et l'autre est marquée comme *EXCLUDED* ou *OVERLAPS_PREVIOUS*. La méthode gère également les cas où une observation précédente peut être réactivée si elle ne chevauche pas la nouvelle observation sélectionnée.

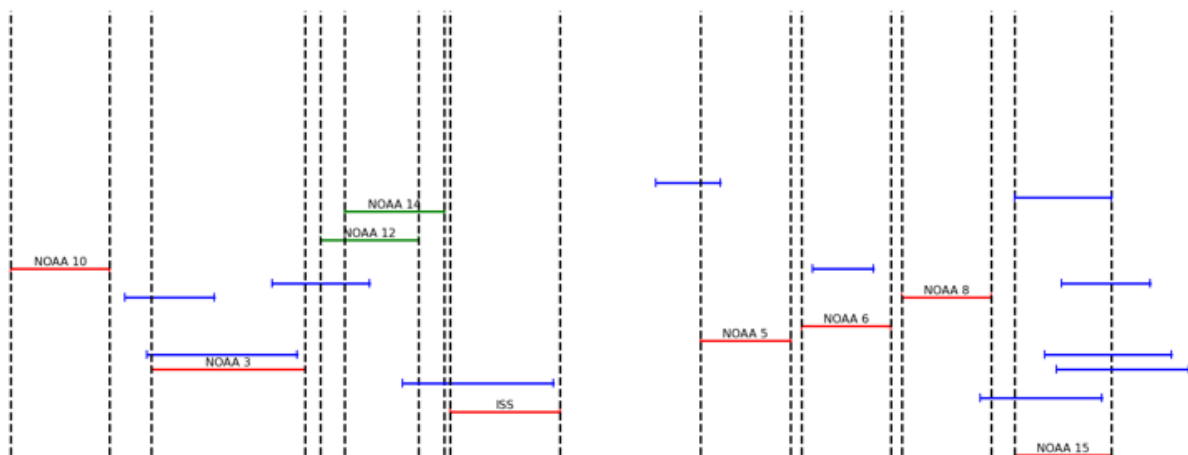


Figure 6 – Planning généré par la méthode *chooseWindows*

Cependant, l'algorithme qui avait été réalisé manquait de fiabilité. En effet, dans les cas où un nombre important de fenêtres se superposant s'enchaînaient, certaines fenêtres qui auraient dû être sélectionnées ne l'étaient pas (voir fenêtres en vert Figure 6). C'est pourquoi il a fallu créer la méthode *determineAddableWindows*.

4.5 Méthode determineAddableWindows

La méthode *determineAddableWindows* détermine s'il existe des fenêtres de visibilité qui peuvent être ajoutées entre les fenêtres déjà sélectionnées. Elle parcourt les observations et vérifie si des fenêtres *EXCLUDED* ou *OVERLAPS_PREVIOUS* peuvent être insérées entre les fenêtres *SELECTED*. Si c'est le cas, ces fenêtres sont marquées comme *IDLE* et peuvent être réévaluées dans les itérations suivantes. Les observations qui ne sont ni *SELECTED* ni *IDLE* sont supprimées de la liste des observations.

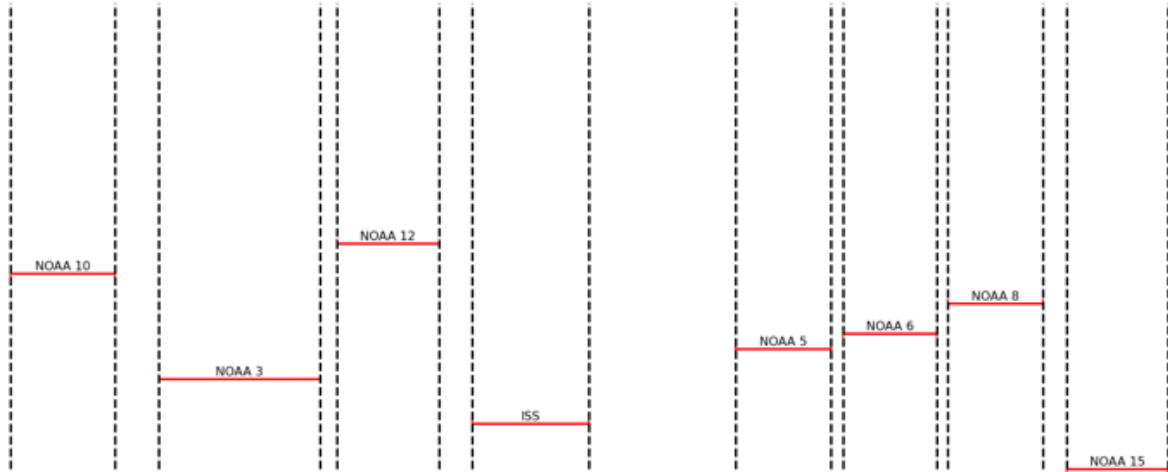


Figure 7 – Planning final obtenu en combinant les deux méthodes

En combinant ces deux algorithmes, la liste qui contenait toutes les observations possibles sur les sept prochains jours ne contient désormais que les fenêtres de visibilité qui ont été sélectionnées comme utilisables. De cette manière, les deux scripts suivants ont seulement besoin de la parcourir, élément par élément, pour procéder au suivi et à l'enregistrement des satellites.

De surcroît, il a été identifié que cet algorithme aurait pu être simplifié de manière significative. En tenant compte du fait que chaque satellite se voit attribuer une priorité unique, il est impossible que deux satellites de même priorité soient observables simultanément. Ainsi, une approche plus efficace aurait consisté à commencer par sélectionner toutes les fenêtres de visibilité associée au satellite dont la priorité est la plus élevée. Ensuite, il aurait suffi d'appliquer la méthode *chooseWindows* pour identifier les fenêtres de visibilité des satellites de priorité inférieure ($p+1$) qui auraient pu être insérées.

Cette méthode aurait permis de simplifier le processus en évitant de trier toutes les fenêtres de visibilité de manière simultanée. Au lieu de cela, l'algorithme aurait pu se concentrer d'abord sur les satellites de priorité la plus élevée, puis insérer progressivement les fenêtres de visibilité des satellites de priorité plus faible, en fonction des disponibilités restantes.

5. Moteur

Pour suivre les différents satellites défilants, la station satellite est équipée d'un moteur permettant de bouger à la fois en azimut (rotation horizontale) et en élévation (rotation verticale). Cette capacité de mouvement permet d'assurer un suivi précis des trajectoires des satellites dans le ciel. Le moteur permet une rotation horizontale allant de 0° à 450° en azimut et une rotation verticale de 0° à 180° en élévation. La plage de mouvement est représentée ci-dessous :

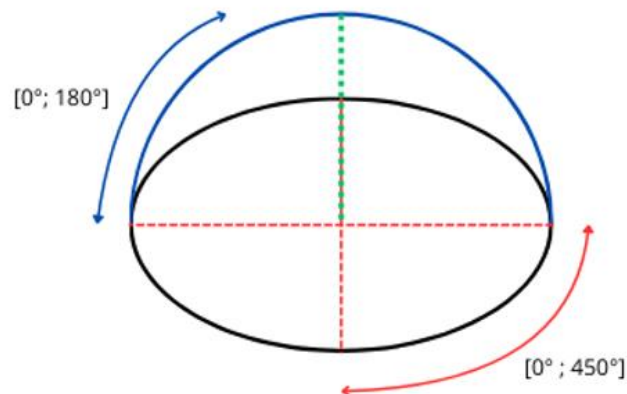
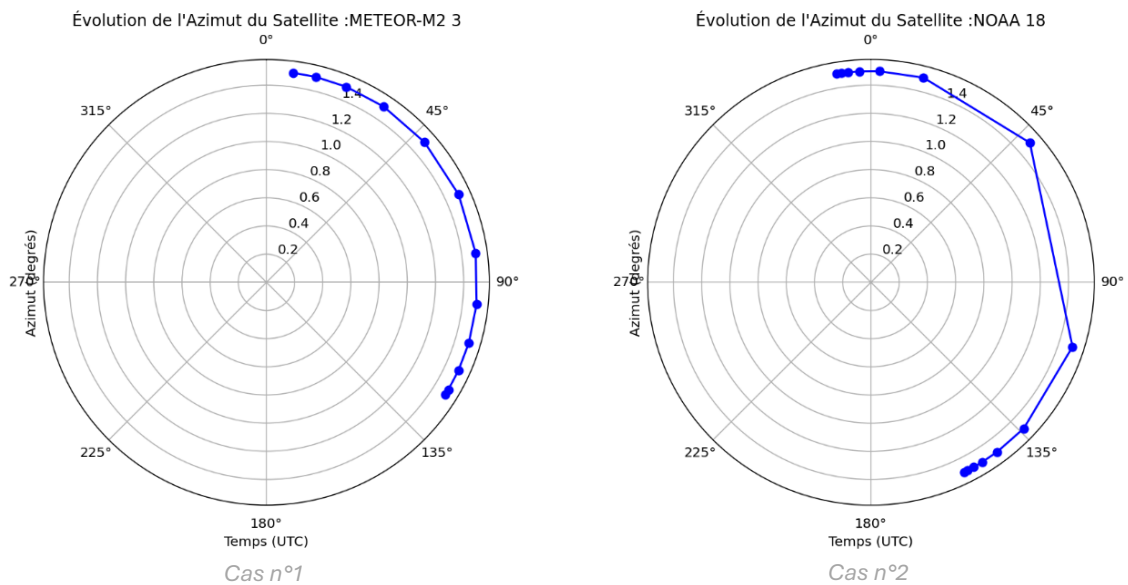


Figure 8 – Plage de mouvement du moteur

Les orbites des satellites sont déterminées par les différents paramètres mentionnés précédemment, qui définissent une trajectoire unique pour chaque satellite. En utilisant la bibliothèque Skyfield, les trajectoires de plusieurs satellites ont été calculées, permettant d'analyser et d'observer les différents cas de passage possibles. Les calculs fournissent un angle d'azimut compris dans l'intervalle $[0^\circ ; 360^\circ]$. Trois cas distincts ont été observés :



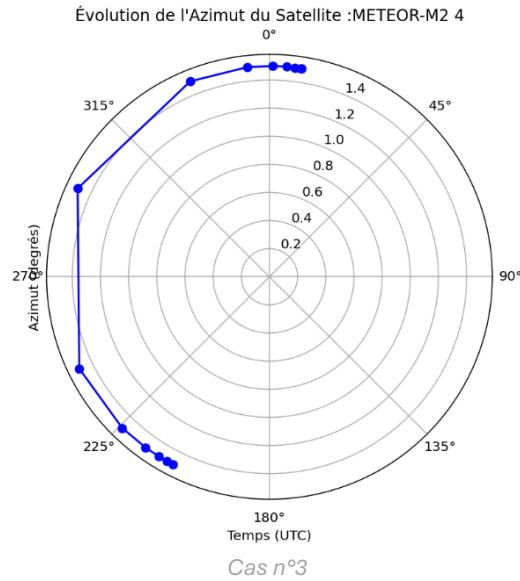


Figure 9 – Mise en évidence des différents cas possibles

Dans les cas présentés ci-haut, différents problèmes ont été identifiés.

5.2 Cas n°1

Le cas n°1 est un cas idéal pour le suivi du satellite. Le passage ne présente aucun risque de blocage, car il se trouve dans la plage de mouvement $[0^\circ ; 360^\circ]$. Le mouvement du moteur est ainsi représenté :

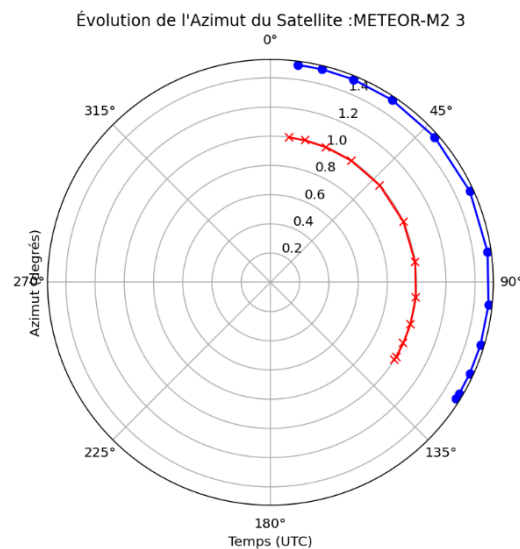


Figure 10 – Mouvement du moteur (rouge) dans le cas n°1

Le déplacement réel du moteur (en rouge) suit exactement l'azimut du satellite sans bloquer.

5.3 Cas n°2

Dans le cas n°2, le satellite commence son passage au Nord-Ouest (azimut = 350°) et termine au Sud-Est (azimut = 153°), en passant par l'Est (azimut = 90°). Ce trajet pose un problème de suivi, car le moteur, ayant une plage de mouvement limitée de 0° à 45° , ne peut pas couvrir l'intégralité du passage sans rencontrer une limitation mécanique. En effet, les angles entre 0° et 90° peuvent également être représentés entre 360° et 450° . De ce fait, en partant du principe que le passage

commence à 350°, il devrait théoriquement se terminer à 513°, ce qui dépasse la capacité du moteur. Inversement, si l'on considère que le mouvement se termine à 153°, cela implique qu'il commence à -7°, ce qui est tout aussi impossible. Ainsi, sans mesures correctives, les données du début ou de la fin du passage seront inévitablement perdues, car le moteur bloquera le suivi à un certain point.

5.3.1 Résolution du cas n°2

Pour résoudre ce problème, il faut s'appuyer sur la capacité de l'antenne à bouger à la verticale. Sachant que l'élévation peut aller jusqu'à 180°, il est donc possible d'inverser l'antenne (la retourner de 180°). Cette inversion permet de pointer l'antenne à l'angle voulu tout en étant dans une plage de mouvement acceptable afin d'éviter toute contrainte mécanique empêchant le suivi du satellite. Cette solution est représentée ci-dessous :

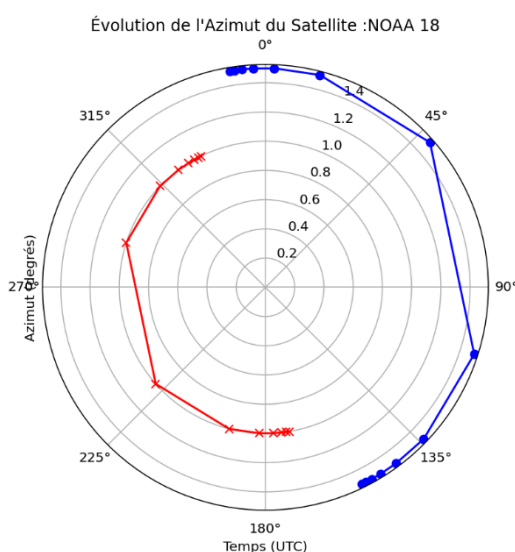


Figure 11 – Mouvement du moteur (rouge) dans le cas n°2

Le mouvement horizontal réel du moteur (représenté en rouge) est inversé de 180°. Concrètement, cela signifie que le point de départ, initialement à 350°, est maintenant à 170°, et que la fin du passage, qui était à 153°, est désormais à 333°. Parallèlement, l'élévation du moteur (non représentée sur le schéma) est également inversée pour s'adapter à cette nouvelle orientation, garantissant que l'antenne pointe correctement vers l'azimut de départ. Ce changement permet d'éviter tout blocage de mouvement dû à des contraintes mécaniques et assure un suivi constant.

5.4 Cas n°3

Dans ce troisième cas, le problème est similaire au cas précédent. Le passage débute à un azimut de 7° et se termine à un azimut de 213°, en passant par l'Ouest (azimut = 270°). Là encore, une limitation mécanique apparaît. Cette limitation est due aux résultats fournis par les calculs, qui amènent la station à un angle d'azimut de 7° avant de la faire se déplacer vers la gauche en passant par le Nord (azimut = 0°). Ce mouvement déclenche une contrainte liée à la plage de mouvement du moteur, qui ne peut pas suivre une trajectoire passant en dessous de 0° ou au-delà de 450° sans rencontrer de blocage. Ainsi, sans correction, une partie du passage ne peut pas être suivie correctement.

5.4.1 Résolution du cas n°3

La solution à ce problème repose sur l'utilisation de la plage de mouvement horizontal étendue du moteur, qui va de 0° à 450°. Comme la valeur de départ est inférieure à 90°, elle peut simplement être convertie dans l'intervalle [360° ; 450°]. Ce qui donnerait pour l'exemple un azimuth de fin égale à 367°. Cette conversion permet d'éviter la limitation mécanique et de garantir un suivi continu du satellite sans interruption. Le mouvement modifié est illustré ci-contre :

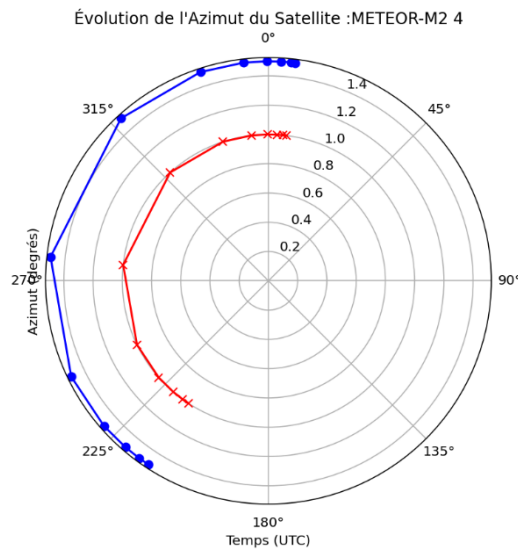


Figure 12 - Mouvement du moteur (rouge) dans le cas n°3

Les valeurs problématiques envoyées au moteur sont converties dans l'intervalle [360° ; 450°]. Dans le cas présent, ce sont les valeurs contenues entre la valeur de départ (7°) et la valeur de blocage (0°) qui sont converties sur cet intervalle.

5.5 Implantation des solutions

Pour implanter les deux solutions présentées ci-dessus, il est nécessaire de connaître à l'avance la trajectoire du satellite. Le calcul des trajectoires se fait dans la fonction *calcul_trajectory*. En utilisant l'heure de lever, l'heure de coucher du passage du satellite, ainsi que ses TLE, il est possible de déterminer la position du satellite à un instant *t*. Ce moment *t* est incrémenté dans la fonction afin d'obtenir la position prévisionnelle du satellite toutes les minutes pendant toute la durée de son passage. Ces positions sont ensuite intégrées dans un tableau qui servira à choisir la meilleure solution pour suivre le satellite sans interruption.

Le choix de la solution s'effectue dans la fonction *select_trajectory_normalization*. Cette fonction parcourt l'intégralité de la liste créée précédemment, lorsqu'un écart entre l'azimut actuel et l'azimut suivant est supérieur à 180°, cela indique un risque de blocage. Pour résoudre ce problème, la fonction effectue une vérification : si l'azimut de départ ou l'azimut de fin est inférieur à 90°, alors la modification de l'azimut peut être appliquée en utilisant l'intervalle [360° ; 450°]. Dans le cas contraire, la rotation est inversée de 180° pour éviter tout problème de suivi. Ce choix de normalisation est enregistré dans une variable, qui sert à indiquer comment les valeurs d'azimut doivent être modifiées.

5.6 Rotctld

Maintenant que les problèmes liés aux contraintes mécaniques sont réglés, il faut réussir à envoyer les valeurs modifiées au moteur. Pour cela, le logiciel Rotctld est utilisé.

Rotctld est un serveur logiciel qui permet de contrôler à distance des systèmes de positionnement d'antenne. Ce logiciel fait partie du projet Hamlib, qui est une bibliothèque open source dédiée au contrôle d'équipement radioamateurs. Ce logiciel prend en charge différents protocoles de communication, ce qui est un point crucial, car cela lui permet de s'adapter à divers types de moteurs et de systèmes de contrôle d'antenne. Cette flexibilité est en parfait accord avec l'objectif de rendre le logiciel portable.

La connexion avec ce logiciel s'effectue via une commande réseau ou via des sockets TCP/IP.

Actuellement la station utilise le protocole GS232b, cependant les commandes qui contrôlent le moteur ne sont pas universelles, chaque protocole a ses propres commandes. C'est sur ce point que le logiciel Rotctld est utile. Celui-ci intègre ses propres commandes puis agit comme un traducteur pour tous les protocoles. La liste des commandes utiles pour utiliser Rotctld est présentée ci-dessous :

- `p : get_pos ()`. Cette commande permet de recevoir la position actuelle de l'antenne.
- `P : set_pos (Azimut, Elevation)`. Cette commande permet de définir la position de l'antenne.

Seules ces deux commandes, sont nécessaires à une gestion efficace de la position du moteur.

5.6.1 Utilisation de Rotctld

Le logiciel Python se connecte à Rotctld via un socket TCP/IP. Cette méthode de communication permet à deux applications de communiquer en utilisant le protocole TCP/IP. Il permet l'envoi et la réception de données de manière fiable. Un socket est défini par une adresse IP (celle de la machine où Rotctld est lancé) et un port (ici, le port est défini sur 4545).

Le socket permet l'envoi des deux commandes présentées ci-dessus à Rotctld et assure la bonne réception des données.

La communication entre le logiciel Python et le moteur est désormais fonctionnelle.

Initialement, le logiciel Rotctld était lancé manuellement via l'invite de commande linux. Cette méthode offrait une liberté importante, car le logiciel pouvait être lancé sur n'importe quel ordinateur connecté au réseau. Cependant, cette méthode n'est pas viable. Lors de différents tests, des messages d'erreur provenant du logiciel provoquaient une incapacité de communiquer avec le moteur. Le seul moyen de rétablir la connexion était de relancer le logiciel Rotctld. Ce problème compromettait l'objectif d'une utilisation automatique de la station, car il nécessitait la présence d'un opérateur sur site pour relancer manuellement le logiciel.

Pour résoudre ce problème, une solution a été trouvée, celle de lancer le logiciel via Python. Cette solution permet de garder en mémoire le numéro de process du logiciel afin de le relancer en cas de problème.

Avec le problème de fiabilité du logiciel Rotctld désormais résolu, l'objectif de suivre des satellites de manière automatique est enfin atteignable.

6. SDR

Désormais, la station est entièrement gérée par un programme qui assure la génération d'un planning fiable des fenêtres d'observation et le suivi précis des satellites grâce au moteur. Cependant, il reste une étape cruciale à automatiser : l'enregistrement des données envoyées par les satellites. Pour cela, la station utilise un SDR (Software-Defined Radio), un dispositif piloté par le programme, qui permet de recevoir et de traiter des signaux radio de manière logicielle. Ce programme gère l'ensemble du processus, de la réception des signaux à leur enregistrement, en passant par le contrôle des paramètres du SDR, garantissant ainsi une autonomie complète de la station.

Le programme récupère les paramètres nécessaires, tels que la fréquence, directement depuis l'objet *satellite* correspondant à chaque fenêtre d'observation. Ainsi, pour chaque passage de satellite, la fréquence et les autres paramètres (comme le taux d'échantillonnage) sont automatiquement ajustés, garantissant une configuration adaptée à chaque satellite.

6.1 Paramètres du SDR

Le SDR utilisé dans le projet est un RTL-SDR dont les paramètres sont les suivants :

6.1.1 Plage de fréquences

Le SDR peut être utilisé pour des fréquences allant de 24 MHz à 1764 MHz. Pour rappel, les satellites qui nous intéressent sont les NOAA et les METEOR dont les fréquences se situent entre 139 MHz et 140 MHz. Ce SDR est donc parfaitement adapté.

6.1.2 Plage de gain

Le gain est réglable entre 0 dB et 49.6 dB, ce qui est idéal puisque sans amplification, les signaux reçus des satellites sont très faibles. De plus, il est possible d'activer ou non l'AGC (Automatic Gain Control).

6.1.3 Plage de taux d'échantillonnage (sample rate)

Le SDR supporte deux plages de taux d'échantillonnages distinctes :

- 0.225001 MS/s à 0.3 MS/s pour les signaux à faible bande passante
- 0.900001 MS/s à 3.2 MS/s pour les signaux à forte bande passante

Ces plages permettent d'ajuster la résolution du signal en fonction de la bande passante occupée par le signal de chaque satellite (indiquée dans le fichier TOML). Par ailleurs, dans le cas où la bande passante d'un satellite ne serait pas comprise dans les plages acceptées par le SDR, il est nécessaire de prendre un taux d'échantillonnage supérieur pour éviter toute perte de données. Pour l'instant, le taux d'échantillonnage est fixé à 0.900001 MS/s

6.1.4 Format de données supportées

Le SDR supporte plusieurs formats de données pour le streaming, notamment :

- CS8 (Complex Signed 8 bits) : format natif du SDR.
- CS16 (Complex Signed 16 bits) : format couramment utilisé pour une meilleure précision.
- CF32 (Complex Float 32 bits) : format offrant une haute précision pour les calculs complexes.

Ces formats permettent de choisir entre une résolution plus élevée (CF32) et une taille de fichier réduite (CS8 ou CS16). Ainsi, pour avoir le meilleur compromis entre résolution et taille de fichier, il faudrait enregistrer dans le format CS16. Cependant, dans le programme, les données sont d'abord reçues et traitées en CF32 car, en pratique, l'enregistrement direct du CS16 ne fonctionne pas. Par la suite, les données en CF32 sont converties en CS16 pour les traiter ultérieurement. Cette méthode permet de contourner les limitations techniques. De plus, le format CS16 est particulièrement adapté pour le décodage des signaux satellites avec des logiciels comme SatDump, car il offre une précision suffisante tout en conservant une taille de fichier raisonnable.

6.2 Format d'enregistrement

Après conversion, ces données seront enregistrées dans un fichier WAV, avec un nom de fichier composé :

- Du nom du satellite
- Du jour
- De l'heure de fin du passage

En sachant que les fichiers WAV sont normalement utilisés pour stocker de l'audio, il est possible de questionner le choix de ce type de fichier. Cependant, ce format présente plusieurs avantages pour l'enregistrement des signaux des satellites. Contrairement à d'autres types de fichiers, comme le MP3 par exemple, le WAV stocke les données de manière non-compressée. Dans notre cas, cela garantit qu'il n'y a pas de perte de qualité ou de détails. Dans le contexte de l'enregistrement de satellites météo, l'enregistrement des données dans un fichier compressé pourrait entraîner une baisse de qualité de l'image, voire une perte d'informations cruciales pour la reconstruction des images. D'autre part, les fichiers WAV peuvent stocker les données brutes, y compris les signaux complexes (I/Q). En utilisant des canaux stéréos, il est possible de stocker dans deux canaux séparés la partie réelle et la partie imaginaire du signal.

Ainsi, les caractéristiques du format WAV en font un choix adapté pour l'enregistrement des signaux des satellites.

6.3 Test sur une fréquence radio

Avant de passer à la réception des données des satellites, des tests ont été effectués avec une fréquence radio standard (100 MHz) pour s'assurer du bon fonctionnement du SDR ainsi que du logiciel associé. L'avantage de tester la station sur une fréquence radio réside dans le fait que les signaux radio FM sont omnidirectionnels. De ce fait, il n'est pas nécessaire de contrôler le moteur simultanément, ce qui a permis d'entamer la phase de test sans attendre que la partie moteur fonctionne.

Lors des premiers essais, des erreurs d'échantillonnage ont été rencontrées. Pour résoudre ce problème, la taille des blocs de données a été augmentée de 2048 à 4096 échantillons. Une fois ce paramètre modifié, les tests ont pu reprendre et se sont avérés concluants.

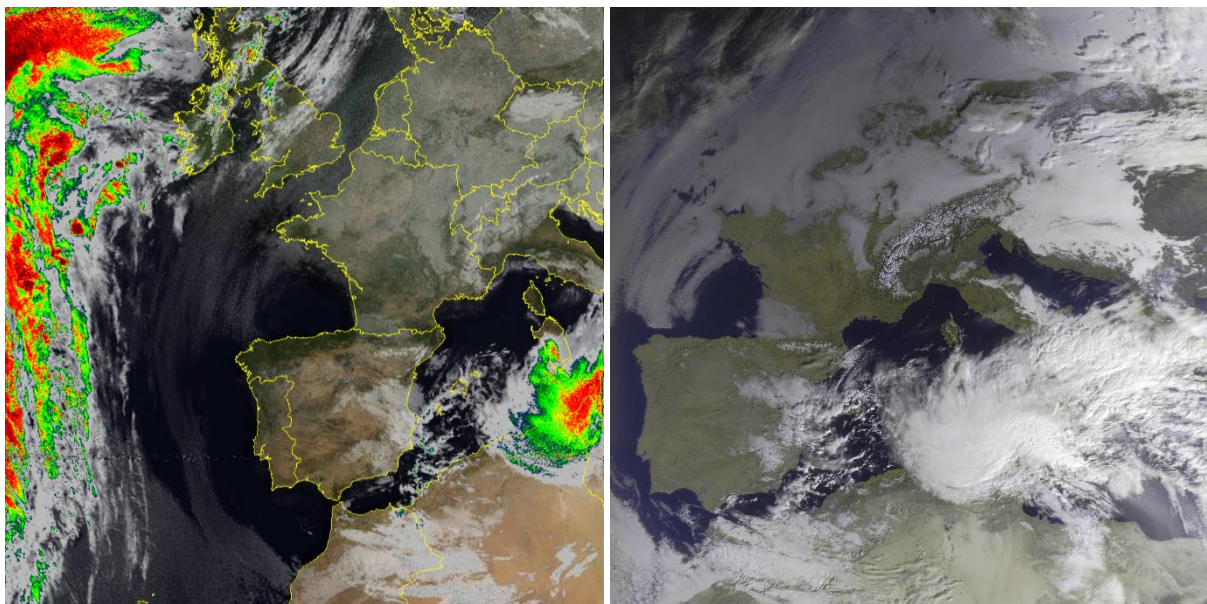
Ces tests ont permis de vérifier que le programme avait bien la capacité de contrôler les différentes caractéristiques du SDR (comme la fréquence et le taux d'échantillonnage) et d'enregistrer correctement des signaux avec une qualité suffisante. Pour s'en assurer, les fichiers WAV créés ont tous été lus sur SDR++, un logiciel de visualisation et d'analyse de signaux radio, permettant de vérifier que l'enregistrement audio ne présentait pas d'artefacts ou de pertes de données.

6.4 Enregistrement des satellites

Une fois que le système fonctionnait correctement sur une fréquence radio, l'étape suivante a consisté à passer à l'enregistrement des signaux des satellites, où le pointage précis de l'antenne devient essentiel.

Lors des premiers essais, les résultats n'étaient absolument pas concluants. En lisant les fichiers WAV sur SDR++ comme précédemment, les fichiers s'avéraient être seulement remplis de bruit, sans aucune donnée utile. Étant donné que l'enregistrement fonctionnait correctement sur une fréquence radio, la conclusion logique a été de penser que le moteur ne pointait pas correctement les satellites. Pour exclure cette possibilité, plutôt que d'utiliser le script d'enregistrement, un logiciel open source pour le décodage des signaux satellites, SatDump, a été utilisé. En passant par ce logiciel, l'enregistrement se faisait correctement. Il a été constaté que les paramètres du SDR définis dans le programme n'étaient pas optimaux. En effet, SatDump utilisait un gain de 48 dB avec AGC désactivé, des paramètres qui n'avaient pas été définis dans le script.

Après avoir ajusté ces paramètres, les signaux des satellites ont été correctement captés et enregistrés, confirmant ainsi que le problème venait bien du paramétrage du SDR. Avec ces paramètres optimisés, des données provenant de satellites NOAA et METEOR ont été reçues avec succès. Les fichiers WAV enregistrés ont été décodés à l'aide de SatDump, transformant ces données en image comme le montre la figure ci-dessous.



NOAA 19

METEOR M2-3

Figure 13 – Images reçues à l'aide de la station de l'IUT de Toulon

6.5 Améliorations

L'enregistrement des satellites fonctionne désormais correctement. Cependant, la taille des fichiers WAV créés est beaucoup trop importante (environ 2,2 Go par fichier) pour laisser la station tourner seule sur plusieurs jours, voire semaines. Pour réduire la taille des fichiers sans compromettre la qualité du signal, le taux d'échantillonnage qui était fixé à 900,001 kS/s, a été ajusté en fonction des signaux observés dans les fichiers WAV :

- 150 kHz pour le signal des satellites METEOR-M
- 50 kHz pour le signal des satellites NOAA

En tenant compte de ces informations, le taux d'échantillonnage a été réduit à la valeur minimale supportée par le SDR (225.001 kS/s), permettant ainsi de réduire drastiquement la taille des fichiers tout en conservant les données essentielles.

Après ces ajustements, le système fonctionne de manière optimale. Les signaux des satellites sont correctement captés, enregistrés et peuvent être traités pour extraire des données utiles.

7. Conclusion

Pour conclure sur l'état actuel du projet, le travail d'automatisation de la station de suivi automatique de satellites défilants est presque terminé. Tous les programmes semblent pouvoir fonctionner ensemble sans intervention humaine sur une période de sept jours et ainsi récupérer les données de tous les satellites qui passeraient durant cette période. Cependant, des problèmes sur la partie électronique du moteur nous empêchent de valider ce fonctionnement, laissant planer un certain doute sur le bon fonctionnement de la station.

Sans compter ce problème, certaines améliorations restent à apporter pour optimiser encore le système. En effet, actuellement, la station est limitée à un fonctionnement sur sept jours, ce qui correspond à la durée de validité des TLE des satellites, et nécessite deux machines différentes pour fonctionner : une pour la partie suivi et une pour la partie réception. La création d'un programme capable de gérer simultanément les tâches de suivi et de réception, en utilisant des threads par exemple, ainsi que de venir télécharger à nouveau les TLE tous les sept jours rendrait la station complètement automatique. De plus, actuellement, il est impossible de savoir si un problème est survenu sans se rendre sur les lieux. L'implantation d'un serveur MQTT pourrait donc être intéressante, car cela permettrait d'obtenir en continu des informations issues de la station. De plus migrer la station sur un réseau local permettrait de la fiabiliser. Actuellement, le réseau provoque des problèmes de connexion entre la suite logicielle qui a été développée et Rotctld, ce qui perturbe le suivi des satellites. Enfin, pour supprimer toutes les interventions manuelles, l'automatisation de la démodulation des signaux enregistrés via SatDump pourrait s'avérer pertinente. En interrogeant SatDump avec des lignes de commandes directement envoyées par le programme, il serait possible de démoduler tous les fichiers. Ainsi, quand l'utilisateur se rendra à la station, il aura juste à récupérer toutes les images. L'automatisation de la station pourrait par ailleurs rendre la station plus viable dans le temps. En effet, pour chaque passage, les données enregistrées sont d'environ 2 gigas, en supprimant ces fichiers après démodulation, la place gagnée sur le disque dur serait non négligeable.