



Projet ObsHyper Bilan de mi-parcours

Emilie SIMON

Jany LAHLOUH

Julien DUCHEMANN

Pierre-Olivier PEDRENO

Objet du rapport

Pour donner suite à la réunion du 16 février 2021 avec nos encadrants messieurs Comblet et Tanguy et madame Gautron, l'équipe ObsHyper se propose ici de rendre compte de l'avancement du projet, des tests réalisées et de ce qu'il reste à accomplir afin d'obtenir un radar fonctionnel.

En premier lieu, ce rapport s'accompagne d'un nouveau diagramme de Gantt prenant en compte les accomplissements et changements qui ont caractérisés les premières semaines de l'acte 2 du projet. Ce diagramme peut être comparé au premier diagramme de Gantt établi le 19 janvier 2021 pour constater de l'avancement du projet.

Durant la phase 2 du projet, l'équipe s'est concentrée sur trois points : l'amélioration des algorithmes développés pour la phase 1, la reprise des manipulations du matériel (permise avec la levée du confinement) et enfin la mise en relation de ces deux premiers aspects.

Table des matières

Objet du rapport	2
Manipulations du matériel	4
Mise en place du filtrage adapté	5
Détection et localisation	7
Tracking.....	8
Outils mis en place pour pouvoir effectuer les tests	10
Taches restantes	10
Figures.....	11
Référence.....	11

Manipulations du matériel

La levée des cours à distance a permis à l'équipe de (re-)prendre en main le matériel et la chaîne RADAR. Nous avons donc repris la chaîne énoncée dans le rapport en fin de semestre 3, en y apportant les modifications suivantes :

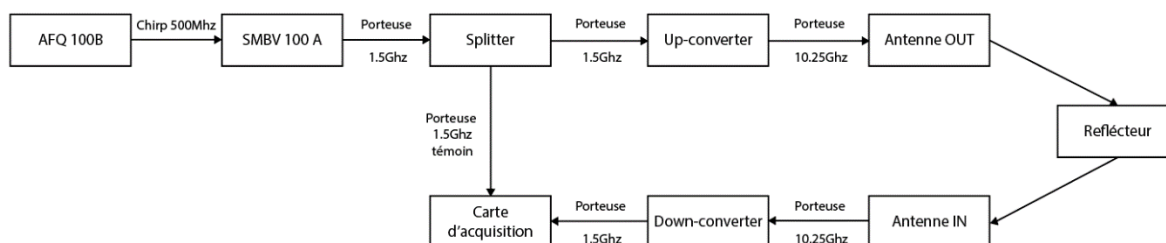


Figure 1 : Chaîne RADAR validée et exploitée

La présence du splitter permet de procéder à une acquisition sur deux canaux, d'une part le signal en sortie de l'antenne réceptrice et d'autre part le signal tel qu'il est émis. L'avantage est de pouvoir procéder à la corrélation mathématique de ces deux signaux et ainsi d'effectuer, en post-traitement, un filtrage dit "adapté". Le principal inconvénient d'un tel montage réside dans l'atténuation causée par une multiplication des canaux, par instants. Il semblerait même qu'on puisse perdre tout signal retour au sein d'une pièce fermée, comme nous avons pu l'expérimenter mardi 16/02.

Toutefois, après des tests semi-unitaires (puisque chaque test sur un appareil, nécessite le concours des précédents), nous avons obtenus des résultats relativement probants pour une telle chaîne qui aboutissent sans traitement aux analyses spectrales suivantes :

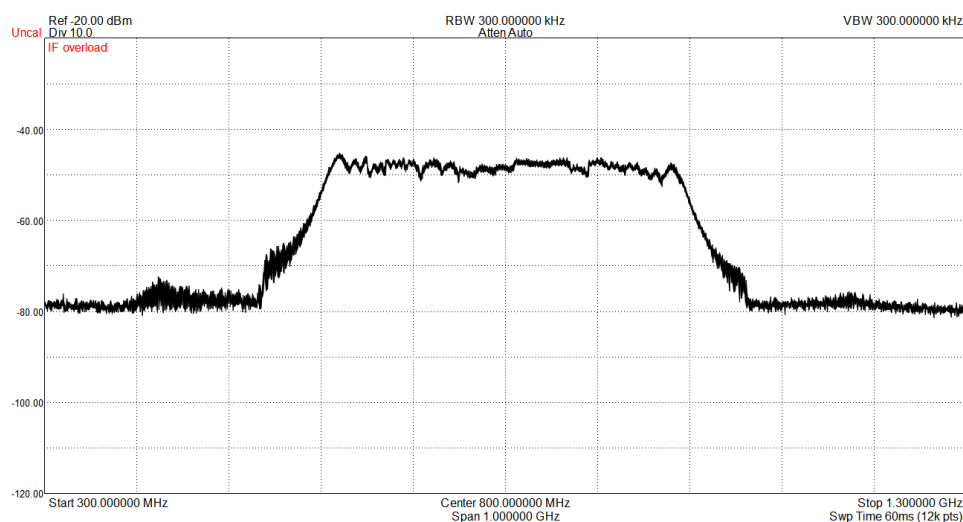


Figure 2 : Signal témoin en sortie du splitter

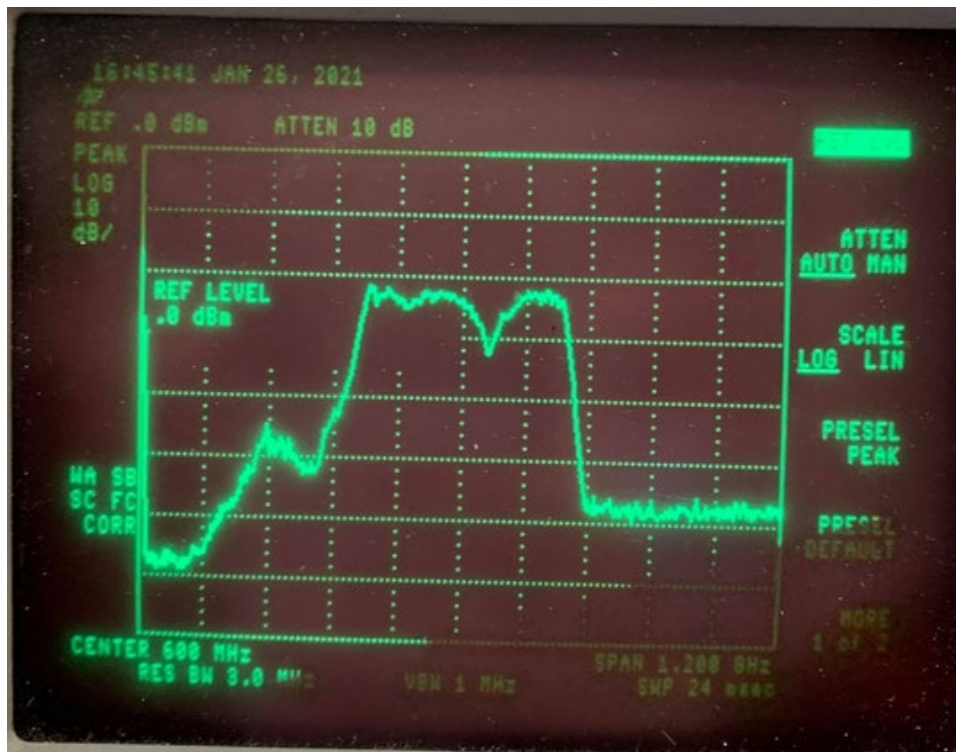


Figure 3 : Signal lu après réflexion sur un mur à 1 mètre

On y lit donc quelques pertes sur certaines fréquences et une perte en amplitude qui reste raisonnable. Nous avons donc pris le parti d'exploiter ce montage afin de réaliser les algorithmes qui suivent.

A l'heure de la rédaction de ce rapport, il nous reste à expérimenter l'usage des converters 15GHz et étudier la possibilité d'en tirer parti : peuvent-ils apporter une atténuation moins faible et donc un meilleur SNR ? Se pencher sur la question des diagrammes d'antennes et leurs positions physiques relatives : est-ce que ça a un réel impact ? Dans quelle mesure ? Et enfin, prendre en main le logiciel K6 Pulse Sequencer qui permet de générer des formes de signaux de notre conception, lisibles par le R&S AFQ 100B.

Mise en place du filtrage adapté

Le filtre adapté est un filtre linéaire qui permet d'optimiser le rapport signal sur bruit (SNR). Ce filtre est utilisé pour la détection d'un signal inconnu en corrélation avec un signal connu. D'un point de vue technique, cela revient à effectuer une intercorrélation entre le signal émis, connu, et le signal reçu, inconnu et bruité. Si on considère une seule impulsion, le résultat est un sinus cardinal centré sur "t_delay", le temps de retard du signal reçu par rapport au signal émis.

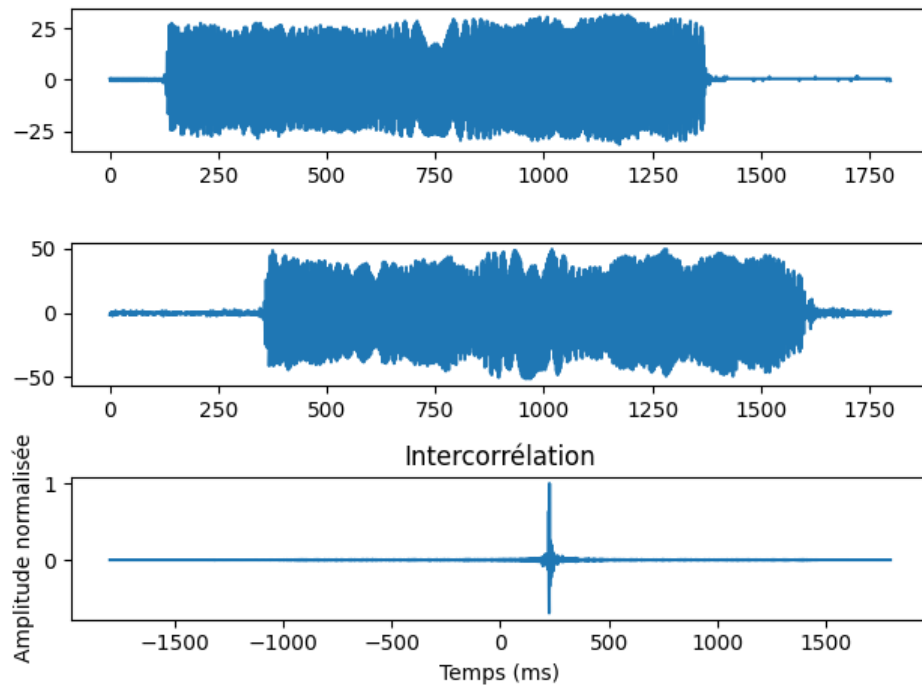


Figure 4 : Filtrage adapté sur une impulsion unique

Lorsqu'on effectue l'intercorrélation sur l'intégralité des signaux reçus et émis (l'entièreté du train d'impulsion), le résultat est un ensemble de $2n-1$ sinus cardinaux, avec n le nombre d'impulsions.

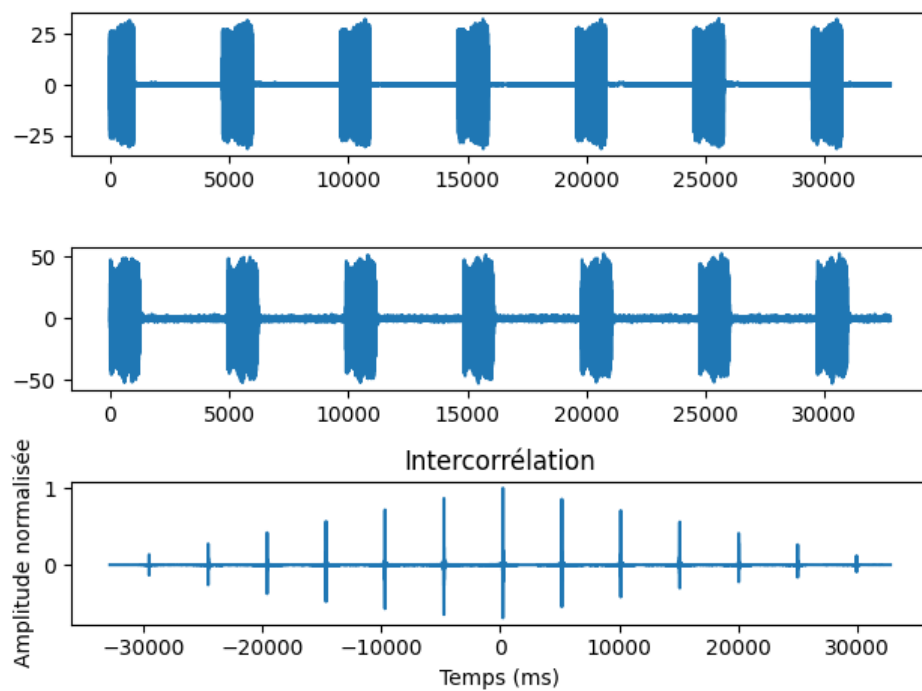


Figure 5 : Filtrage adapté sur plusieurs impulsions

Cela s'explique par le fait que l'intercorrélation est une mesure de similitude entre deux signaux par translation temporelle. Ainsi, le sinus cardinal central est le sinus cardinal avec l'amplitude la plus forte car il correspond au recouvrement exact des deux trains d'impulsions. Le sinus cardinal à sa droite (respectivement à sa gauche) correspond à un recouvrement exact moins la première impulsion reçue et la dernière impulsion émise (respectivement la première impulsion émise et la dernière impulsion reçue). Ainsi l'amplitude de ses sinus cardinaux est moindre. Pour les sinus cardinaux suivants, on continue la translation et on répète le même raisonnement, jusqu'à arriver aux sinus cardinaux extrêmes. Ces sinus cardinaux témoignent du recouvrement des impulsions extrêmes des signaux émis et reçus avec perte de toutes les autres impulsions du train. Perte due à la translation temporelle de l'intercorrélation.

Afin de vérifier expérimentalement la cohérence de ce modèle, bien qu'elle ne soit plus à prouver, nous avons réalisé les tests unitaires aboutissant aux deux figures ci-dessus. Il s'agit ici, après avoir validé notre montage de la chaîne physique, d'exploiter une impulsion unique, en espérant retrouver un sinus cardinal, puis plusieurs impulsions (avec antenne et cibles fixes) en espérant cette fois retrouver une forme de sinus cardinaux d'amplitudes croissantes puis décroissantes, régulièrement espacés. Comme vous pouvez le constater, nos attentes théoriques ont été honorées, ce qui confirme sans surprises la pertinence du filtrage adapté.

Détection et localisation

L'utilisation des premiers signaux acquis a mis en lumière la nécessité d'améliorer la mise en forme des acquisitions afin de les rendre parfaitement utilisables. En effet, dans un premier temps afin de pouvoir obtenir un résultat correct avec le filtrage adapté il nous faut effectuer l'intercorrélation entre une émission et les retours contenus dans le temps d'écoute qui lui est associé. Pour ce faire nous allons découper chaque acquisition en plusieurs pistes, chacune étant un et un seul temps d'écoute. En pratique cela sera réalisé sous la forme d'une matrice dont chaque colonne sera un temps d'écoute. Ainsi toutes les colonnes seront traitées séparément par le filtrage adapté. Le résultat du filtrage sera lui aussi une matrice.

Ce résultat sera ainsi sous une forme parfaitement appropriée pour l'entrée des algorithmes de détection et de localisation. L'algorithme CFAR traitera chacune des colonnes séparément, puis les résultats seront utilisés par l'algorithme de localisation qui utilisera pleinement la mise en forme réalisée précédemment. Les distances cibles-radar des cibles marquées par l'algorithme de détection pourront être calculées piste par piste.

Cette solution nous permet de résoudre le problème de détermination d'appartenance d'un signal retour avec l'impulsion qui est à son origine. En effet, puisque le signal retour aura été prédécoupé en temps d'écoute la détermination est aisée. Ce constat nous a aussi mener à faire un choix quant aux spécificités des émissions que nous réaliserons dans les semaines à venir, dans un premier temps nous allons choisir une période de récurrence, entre deux impulsions, assez grande pour lever toute ambiguïté d'appartenance des signaux retours.

Concernant la sortie de la partie de détection et de localisation, elle est mise sous forme d'un tableau comportant les distances au radar des cibles détectés (chaque colonne étant toujours un temps d'écoute). Cette sortie type "1D" a été choisi après la réunion du 16 février car la détermination

de coordonnées spatiales “2D” plus complètes semble difficile avec notre matériel. La sortie permettra néanmoins une mise en œuvre des méthodes de tracking étudiées.

Tracking

Les premiers algorithmes de tracking développés lors de la phase 1 du projet s'exécutaient avec en entrée des coordonnées cartésiennes (x, y) 2D. Comme évoqué précédemment, nous sommes restreints par le matériel en conséquent, l'algorithme CFAR ne peut fournir qu'une entrée (x) 1D aux algorithmes de tracking. La totalité du code concernant l'algorithme des k plus proches voisins a donc été revue pour pouvoir s'exécuter simplement avec la distance comme entrée. L'algorithme possède toujours 3 fonctions : initialisation de nouvelles pistes, suppression de pistes non mise à jour depuis trop longtemps et mise à jour de pistes existantes.

La mise à jour de piste existante est fonctionnelle et a été testée avec en entrée 6 coordonnées imposées représentant une piste et une coordonnée (là aussi imposée) représentant le point à tester. Cette partie de l'algorithme sera à nouveau testée quand le système physique aura fourni ses premières données.

L'initialisation de piste a été mise en place. L'algorithme est en mesure de déterminer si un point correspond effectivement à la piste existante ou s'il correspond à une nouvelle piste. Pour cela, l'algorithme se base sur un seuil que l'on a déterminé arbitrairement et qui a vocation à être changé selon l'objet tracké.

Comme l'initialisation fonctionne correctement, nous avons ensuite développé un algorithme capable de suivre deux cibles en même temps et ainsi prenant le relais du premier algorithme quand une nouvelle piste est initialisée. Là encore, ce nouvel algorithme a été testé avec deux pistes créées artificiellement.

Actuellement l'algorithme fonctionne en mode “pas à pas” c'est à dire avec un plot ou deux à tester. Pour suivre son exécution, une représentation graphique a aussi été mise en place pour pouvoir représenter les pistes existantes, le nouveau point mais aussi le résultat de l'exécution de l'algorithme.

Prenons comme exemple d'exécution la situation initiale suivante : nous avons une piste déjà existante ainsi qu'un point à tester non rattaché à cette même piste. Les deux figures suivantes illustrent les actions de l'algorithme.

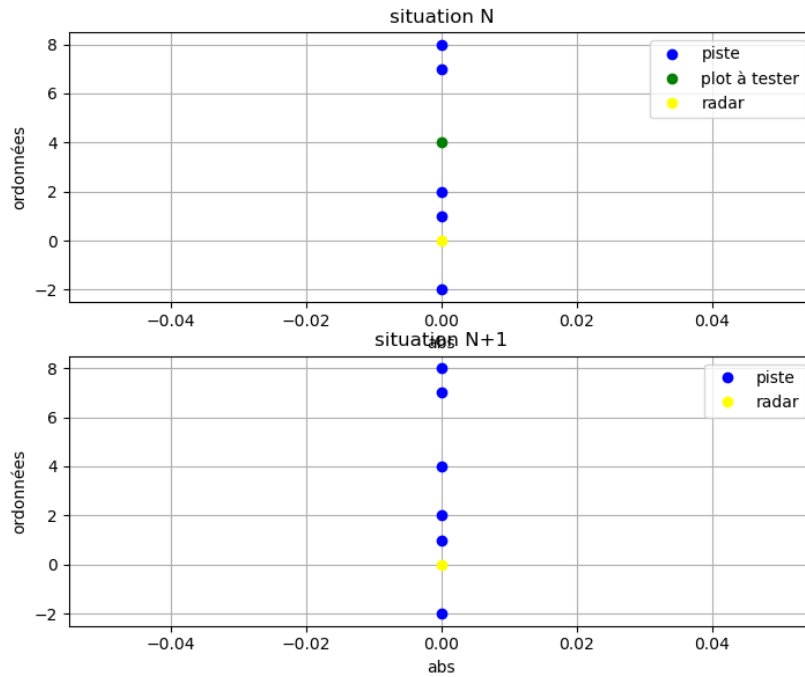


Figure 6 : (haut) situation initiale – (bas) ajout du point à la piste existante

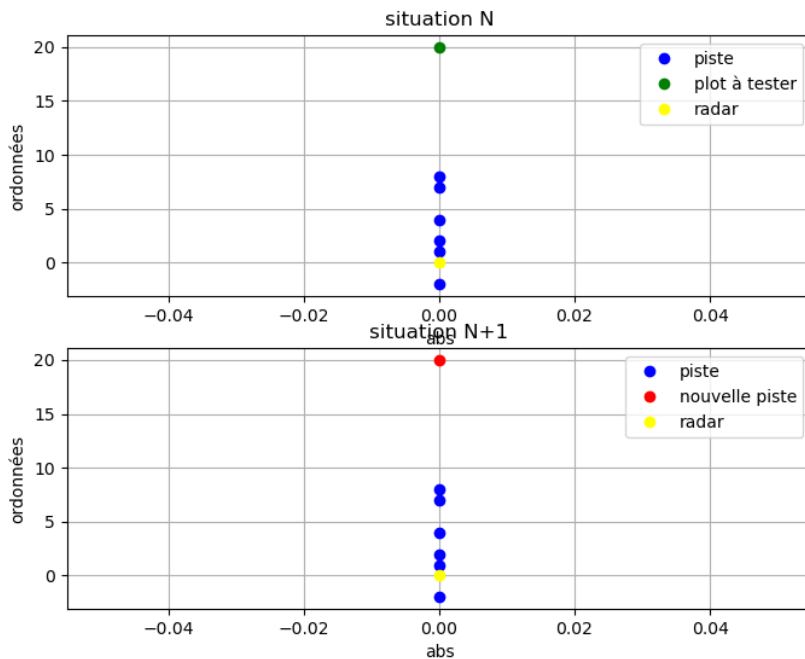


Figure 7 : (haut) ajout d'un second point – (bas) création d'une nouvelle piste pour ce point

Il nous reste encore à améliorer cet algorithme. Tout d'abord, le seuil déterminant si oui ou non une nouvelle piste est créée doit être déterminé autrement que par un nombre imposé et doit pouvoir changer selon l'objet tracké et sa vitesse. Puis il faut encore mettre en place l'algorithme permettant de mettre fin au suivi d'une piste quand cette dernière n'est plus mise à jour. Ensuite il faut connecter l'algorithme KNN à la sortie du CFAR et effectuer des tests avec des coordonnées réelles issues des acquisitions.

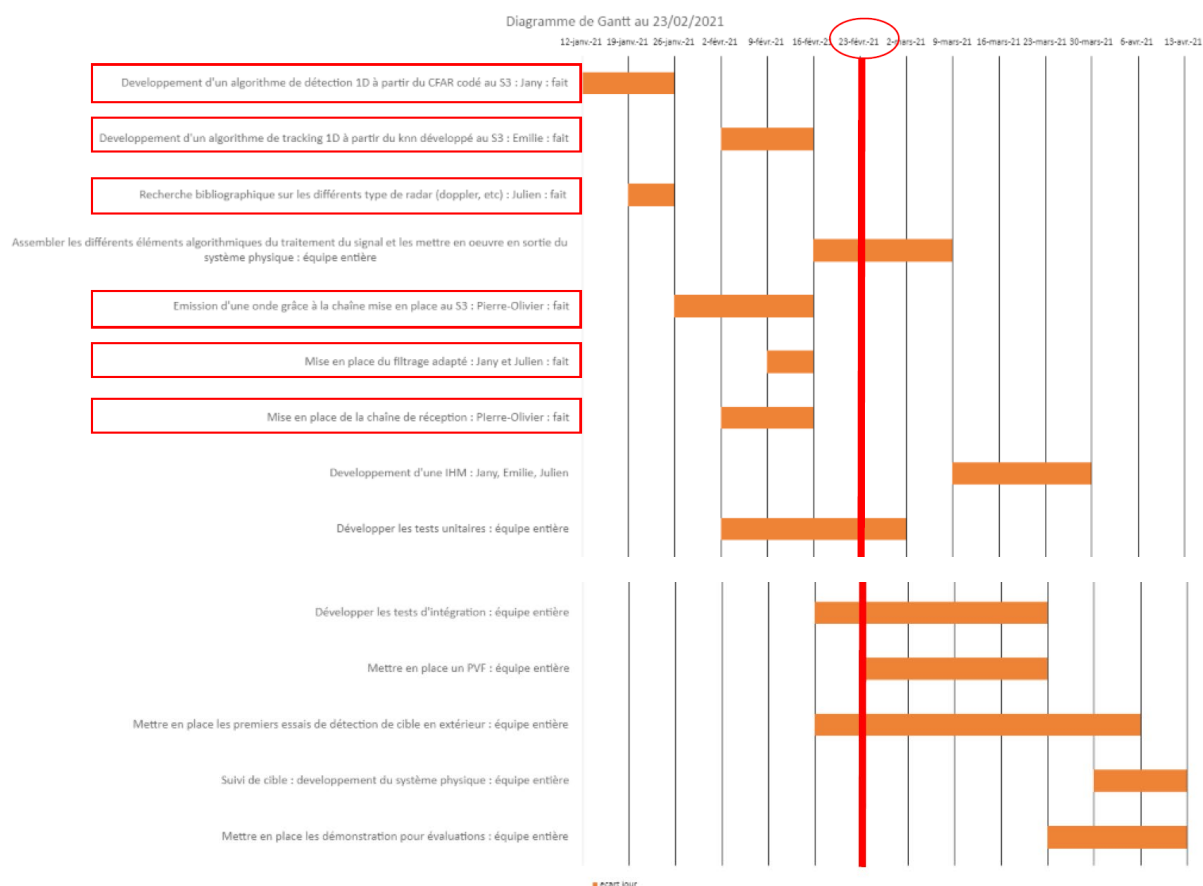
Outils mis en place pour pouvoir effectuer les tests

En parallèle du travail sur la chaîne de traitement, la mise en place d'une bibliothèque a été commencé. Elle comporte d'ores et déjà des fonctions permettant d'améliorer certains calculs de chaînes principales, et des fonctions qui permettront de calculer et de vérifier certaines des capacités de notre système ainsi que la qualité des signaux acquis et traités.

Les premiers algorithmes de traitement sont testés selon différentes configurations : en premier avec un signal "propre" et simple type sinusoïdal, puis avec ce même signal auquel on a artificiellement rajouté un bruit gaussien et enfin avec des "vrais" signaux issus des acquisitions avec le matériel.

Taches restantes

Les tâches restantes, au même titre que les tâches abouties ou en cours de réalisation, peuvent être résumées et sont lisibles dans le diagramme de Gantt, dont la dernière édition, la plus à jour possible pour ce rapport, est daté du 23/02/21.



Figures 8 & 9 : Diagramme de Gantt en date du 23/02/2021

Figures

Figure 1 : Chaîne RADAR validée et exploitée – Pierre-Olivier Pedreno

Figure 2 : Signal témoin en sortie du splitter – Pierre-Olivier Pedreno

Figure 3 : Signal lu après réflexion sur un mur à 1 mètre – Pierre-Olivier Pedreno

Figure 4 : Filtrage adapté sur une impulsion unique – Julien Duchemann

Figure 5 : Filtrage adapté sur plusieurs impulsions – Julien Duchemann

Figure 6 : (haut) situation initiale – (bas) ajout du point à la piste existante – Emilie Simon

Figure 7 : (haut) ajout d'un second point – (bas) création d'une nouvelle piste pour ce point – Emilie Simon

Figures 8 & 9 : Diagramme de Gantt en date du 23/02/2021 – Emilie Simon

Référence

Les différents codes et algorithmes mentionnés sont disponible sur Github à l'adresse infra. Il reste cependant important de noter qu'ils sont disposés sur cette plateforme afin de faciliter leur mise en commun pour le groupe, cependant, ils sont faits pour être exploités conjointement à la chaîne RADAR physique. Pour ce faire, vous pouvez utiliser le fichier *acquisition.txt* du dossier *data* qui résulte d'une acquisition respectant la figure 1.

<https://github.com/AspirantMoutarde/radar-signal-post-processing>