



Projet ObsHyper

Emilie SIMON

Jany LAHLOUH

Julien DUCHEMANN

Pierre-Olivier PEDRENO

Résumé

ObsHyper est un projet ayant pour ambition de créer un système radar fonctionnel à partir des différents éléments constituant le standard de n'importe quel radar. Il s'agira donc d'appréhender la théorie relative à l'univers des systèmes d'observations, dont une certaine partie est intégrée à notre cursus, d'appréhender les différents organes des-dit systèmes, ainsi que de mettre en pratique cette théorie pour programmer différents algorithmes capables de mettre en œuvres différents assemblages radars. Une fois les prémices d'un système fonctionnel obtenues, il sera pertinent de suggérer des améliorations à ce système telles que le suivi de cible, le traitement "temps réel", ou encore la complexité de nos algorithmes.

Pour l'heure, nous avons compris le rôle et le fonctionnement de chacune des parties d'un radar, et avons pu les mettre en œuvre individuellement et collectivement, toutefois nous avons été interrompus par le confinement et manquons d'expérience pratique. Du point de vue théorique, nous avons mis au point deux algorithmes, l'un traitant le signal reçu par le radar et l'autre capable d'isoler le bruit du signal utile.

Mots clés : RADAR, acquisition, traitement, signal, suivi de cible.

OBShyper is a project with the ambition to create a functional radar system from the different elements that make up the standard of any radar. It will thus be a question of apprehending the theory relating to the universe of observation systems, a certain part of which is integrated into our curriculum, of apprehending the various organs of these systems, as well as putting this theory into practice in order to program various algorithms capable of implementing different radar assemblies. Once the beginnings of a functional system are obtained, it will be relevant to suggest improvements to this system such as target tracking, real-time processing, or the complexity of our algorithms.

So far, we have understood the role and operation of each part of a radar and have been able to implement them individually and collectively, however we have been interrupted by the lockdown and we suffer from a lack practical experience. From a theoretical point of view, we have developed two algorithms, one processing the signal received by the radar and the other able to isolate the noise from the wanted signal.

Key words : RADAR, acquirement, processing, signal, tracking.

Table des matières

Introduction	4
I – Etat de l’art	4
I.1 – Le radar : éléments de base	4
I.2 – Traitement du signal	7
I.3 – Détection	8
I.3.1 – Taux de fausses alarmes	8
I.3.2 – Taux de fausses alarmes constant (CFAR)	8
I.3.3 – Test de l’algorithme CFAR	9
I.4 – Principe du tracking	11
I.4.1 – Algorithme des plus proches voisins (K-NN pour K Nearest Neighbours).....	12
I.4.2 – Probabilistic data association filter PDAF	12
I.4.3 – Multiple Hypothesis Tracking (MHT).....	13
I.4.4 – Filtrage de Kalman	14
II – Les principes de l’ingénierie système	14
II.1 – Exigences et architecture fonctionnelle	15
II.2 – Architecture physique	15
III – L’application de la méthode agile au sein de l’équipe du projet ObsHyper	18
Conclusion	21
Bibliographie	22
Table des figures	22
Table des tableaux	23
Annexes	23

Introduction

Au cours de ce projet nous avons comme objectif principal de créer en autonomie la totalité de la chaîne composant un radar hyperfréquence afin de pouvoir, à terme, à l'aide de ce dernier, détecter un objet ainsi que sa distance. L'objectif est simple et limité à la détection d'objet à une faible distance. Pour ce faire, l'équipe encadrante et l'ENSTA Bretagne mettent à notre disposition tous les composants nécessaires : générateurs, oscilloscopes, analyseurs, convertisseurs.

Face au re-confinement et à l'éloignement de l'équipe, le projet a été réorganisé pour comporter deux aspects :

- Un aspect pratique avec utilisation du matériel pour répondre à la problématique originelle, qui correspond aux travaux effectués pré-confinement et qui restent pour lors inachevés. Parmi eux et en tête de liste, la prise en main des différents organes à notre disposition, notamment au travers de tests unitaires visant à comprendre le fonctionnement, le rôle, et l'intérêt de chacun d'entre eux, dans le but de pouvoir suggérer une chaîne d'assemblage cohérente et sensée. Cette phase avait lieu sur les créneaux de projet, en lieu et place du bâtiment E et se serait par la suite poursuivi par des essais en extérieur (tout le matériel étant installé sur une baie mobile).
- Un aspect théorique, basé principalement sur le traitement du signal que l'on a acquis. Cet aspect du projet aurait été abordé qu'importe le confinement, toutefois, ce dernier a accéléré les travaux de ce point de vue ci, suivant également les indications de nos encadrants qui nous ont aiguillé sur les différentes parties théoriques que nous abordons plus loin dans ce rapport.

I – Etat de l'art

I.1 – Le radar : éléments de base

Un radar hyperfréquence est en général constitué de 5 éléments : un émetteur, un duplexeur, une ou plusieurs antennes, un récepteur et enfin un moyen d'affichage. L'affichage le plus classique étant le Plan Position Indicator (PPI) qui restitue une vue en deux dimensions de l'espace à 360 degrés autour du radar.

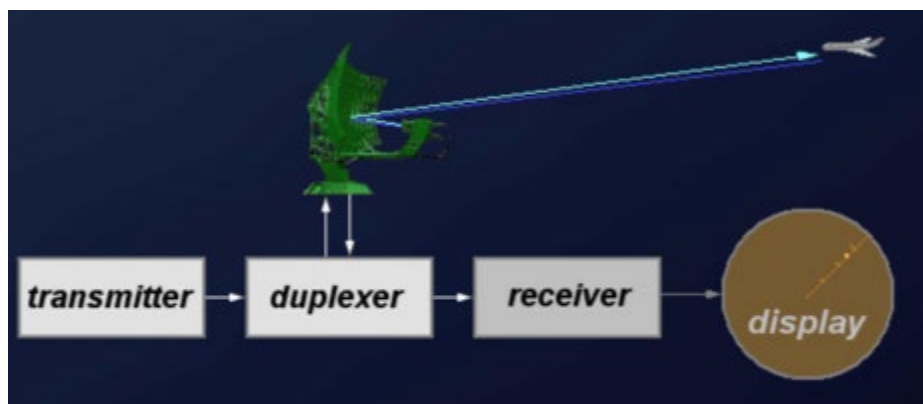


Figure 1 : Chaîne radar type

Dans notre cas, le duplexeur n'est pas nécessaire car le système possède 2 antennes, une pour l'émetteur et une pour le récepteur. Ainsi le duplexeur qui est un commutateur qui relie alternativement l'antenne à l'émetteur puis au récepteur radio n'est pas nécessaire.

Pour notre projet, l'émetteur (ou générateur de signaux) sera le générateur de signaux en bande de base RS AFQ 100B. Dans le jargon des télécommunications, le terme de bande de base (ou en anglais baseband) désigne une technique de transmission dans laquelle le signal est envoyé directement sur le canal après Codage en ligne sans passer par un codage canal (sans modulation). Ce générateur peut produire des signaux radar complexes avec des impulsions courtes et des temps de montée et de descente rapide. Il possède également un bon rapport signal sur bruit. Les impulsions émises sont de très courtes durées, ce qui permet à la bande passante d'atteindre de très grandes valeurs, on parle alors d'UltraLargeBand (ULB), utilisée notamment pour la communication en haut débit sur courte distance.



Figure 2 : RS AFQ 100B

Le générateur de signaux en bande de base RS AFQ 100B est ensuite couplé au générateur de signaux vectoriels RS SMBV 100 A qui transpose les signaux en bande de base en signal RF. Ce dernier pourrait servir de générateur signaux mais dispose de caractéristiques moins intéressantes que son homologue. Il sert uniquement dans notre cas de modulateur numérique et analogique : après avoir généré le signal il le transforme pour l'adapter au canal de transmission. Ici le canal est hertzien sachant qu'il aurait pu être filaire où optique. C'est la modulation. On va donner à l'onde dite porteuse de nouvelles caractéristiques d'amplitude de fréquences et de phases. On a vu que ce générateur avait la particularité de posséder une modulation de type IQ, La modulation d'amplitude en quadrature qui modifie la porteuse elle-même et une onde en quadrature autrement dit déphasé de 90 degrés. Ce qui permet de modifier simultanément la phase et l'amplitude.



Figure 3 : RS SMBV 100A

Enfin, le générateur de signaux vectoriels RS SMBV 100A est branché au convertisseur GeoSync Microwave. Ce dernier sert à la fois de Up Converter à l'émission et de Down Converter à la réception. En d'autres termes, son rôle est d'effectuer une translation en fréquence entre la bande L (fournie et reçue par le générateur de signaux vectoriels) et des fréquences décryptables pour le transpondeur qui reçoit l'onde électromagnétique émise par l'antenne.



These block converter systems provide frequency translation between the transponder band and L-band frequencies.

The two independent converter assemblies are "hot swappable" through the rear of the chassis. Each converter tray has independent control of the front panel. The front panel can be replaced without loss of signal.

BLOCK DOWNCONVERTERS

Input (GHz)	Output (MHz)	LO (GHz)	Conv Tray Model #
9.75-10.25	250-750	9.5	D10
14.0-14.5	250-750	13.75	D14.25
14.75-15.25	250-750	14.5	D15

BLOCK UPCONVERTERS

Input (GHz)	Output (GHz)	LO (GHz)	Conv Tray Model #
250-750	9.75-10.25	9.5	U10
250-750	14.0-14.5	13.75	U14.25
250-750	14.75-15.25	14.5	U15

Figure 4 : Convertisseur GeoSync Microwave

À la fin, il est éventuellement possible d'analyser le signal avec un oscilloscope. Le schéma complet est donc le suivant :

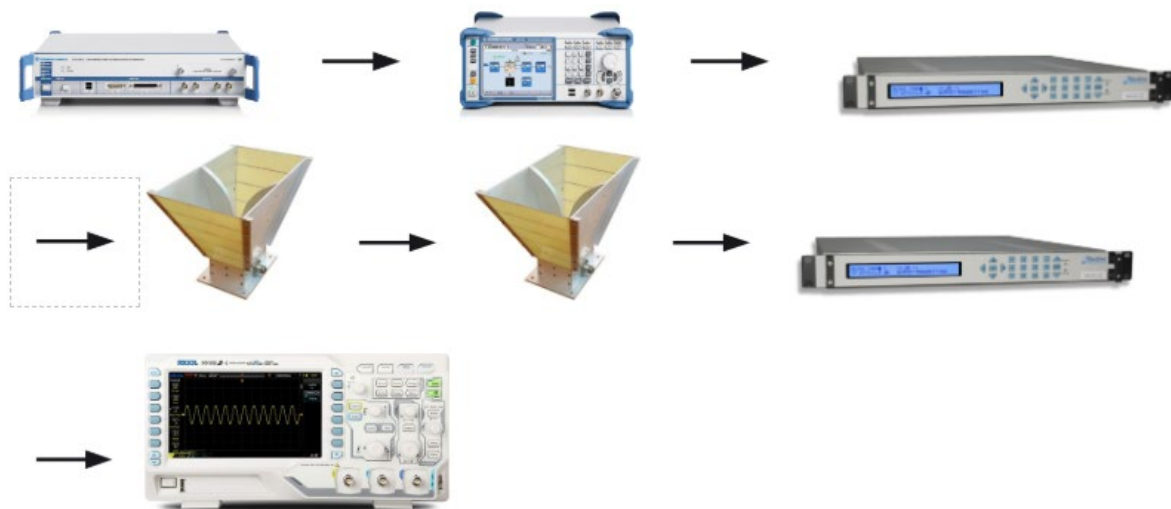


Figure 5 : Montage type de notre système

I.2 – Traitement du signal

Pour analyser un signal reçu, il est intéressant de l'afficher en tant que tel et d'afficher son spectre. Python permet cette analyse via les fonctions de l'algorithme 1, fourni en annexe :

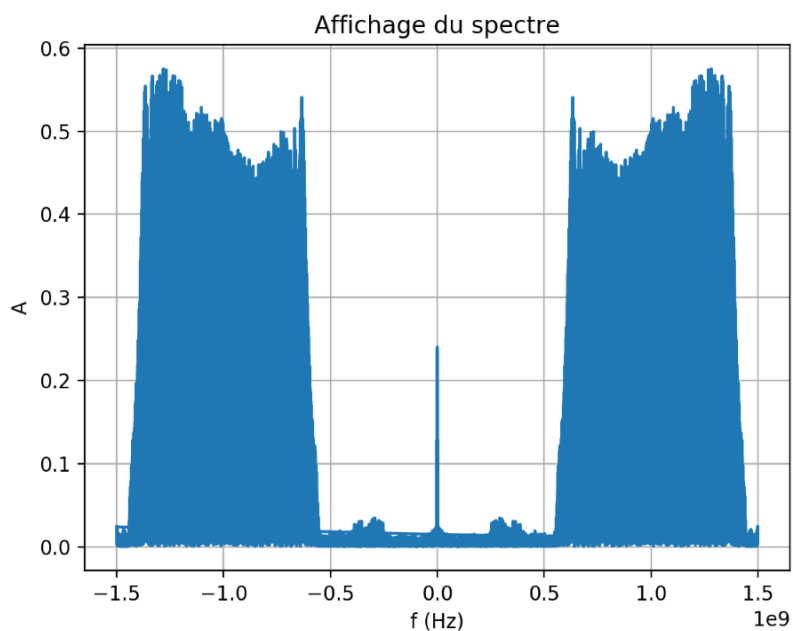


Figure 6 : Spectre obtenu en sortie d'algorithme

On remarque que la largeur de bande est très importante, cela est due à la durée très brève des impulsions du signal. Python n'est pas très adapté à l'analyse des signaux ultra large bande.

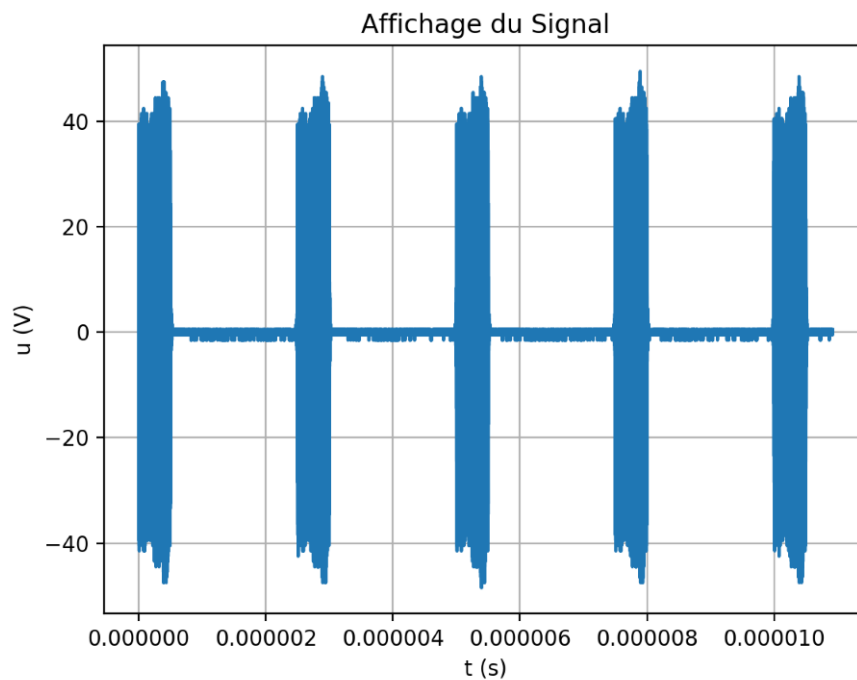


Figure 7 : Signal obtenu en sortie d'algorithme

I.3 – Détection

La détection consiste à mettre en évidence les cibles potentielles, à partir du signal reçu et préalablement traité. Pour ce faire il faut être en mesure d'isoler les pics correspondants à des cibles du bruit environnant. La complexité de la manipulation est de réussir à faire ceci en évitant de provoquer de fausses alarmes dues au bruit. Pour ce faire nous utilisons un seuil de détection, lorsque ce seuil est dépassé nous considérons qu'il y a présence d'une cible. Néanmoins puisque qu'un bruit ou d'autres signaux peuvent interférer et eux aussi dépasser ce seuil, nous sommes confrontés à de fausses alarmes. Il faut donc une façon robuste de choisir le seuil de détection afin de limiter le taux de fausses alarmes, c'est ce que nous allons développer ici.

I.3.1 Taux de fausses alarmes

La première notion importante du problème de détection, que nous souhaitons maîtriser au mieux est donc le taux de fausses alarmes, c'est un facteur primordial pour assurer la fiabilité de notre système radar.

Une fausse alarme est « une décision de détection de cible radar erronée causée par du bruit ou d'autres signaux interférents dépassant le seuil de détection »[1]. Le taux de fausses alarmes (FAR) est calculé à l'aide de la formule suivante :

$$\text{FAR} = \text{Fausses cibles par PRT} / \text{Nombre de cellules}$$

Ainsi de fausses alarmes sont générées lorsque le bruit dépasse un niveau seuil prédéfini, par la présence de signaux parasite.

- Si le seuil est réglé trop haut : il y aura très peu de fausses alarmes mais le rapport signal sur bruit requis empêchera la détection de cibles valides.
- Si le seuil est réglé trop bas : le grand nombre de fausses alarmes masquera la détection de cibles valides.
- Si le seuil est réglé de façon optimale : probabilité de détection = 83%, mais une fausse alarme se produit, FAR = $1,5 \cdot 10^{-3}$

- Le seuil est défini de façon variable : taux de fausses alarmes constant.

Le taux de fausses alarmes dépend du niveau de toutes les interférences (bruit, encombrement, brouillage). A courte distance l'influence du clutter fixe (ensemble d'échos retournés par l'environnement) est plus importante que le bruit. A longue distance c'est le plus bruit qui a le plus gros impact. Donc le taux de fausses alarmes dépend de la plage. Il faut noter que le taux de fausses alarmes augmente à courte distance.

I.3.2 Taux de fausses alarmes constant (CFAR)

Afin de maîtriser ce point clé nous allons utiliser un schéma conservant un taux de fausses alarmes constant et faisant varier le seuil de détection en fonction de l'environnement. Ainsi le seuil de détection restera au-dessus du niveau du bruit ce qui nous permettra d'isoler les cibles.

Il existe un grand nombre de type de circuit CFAR, qui sont pour la plupart basés sur le CFAR à moyenne de cellules que nous allons mettre en œuvre ici.

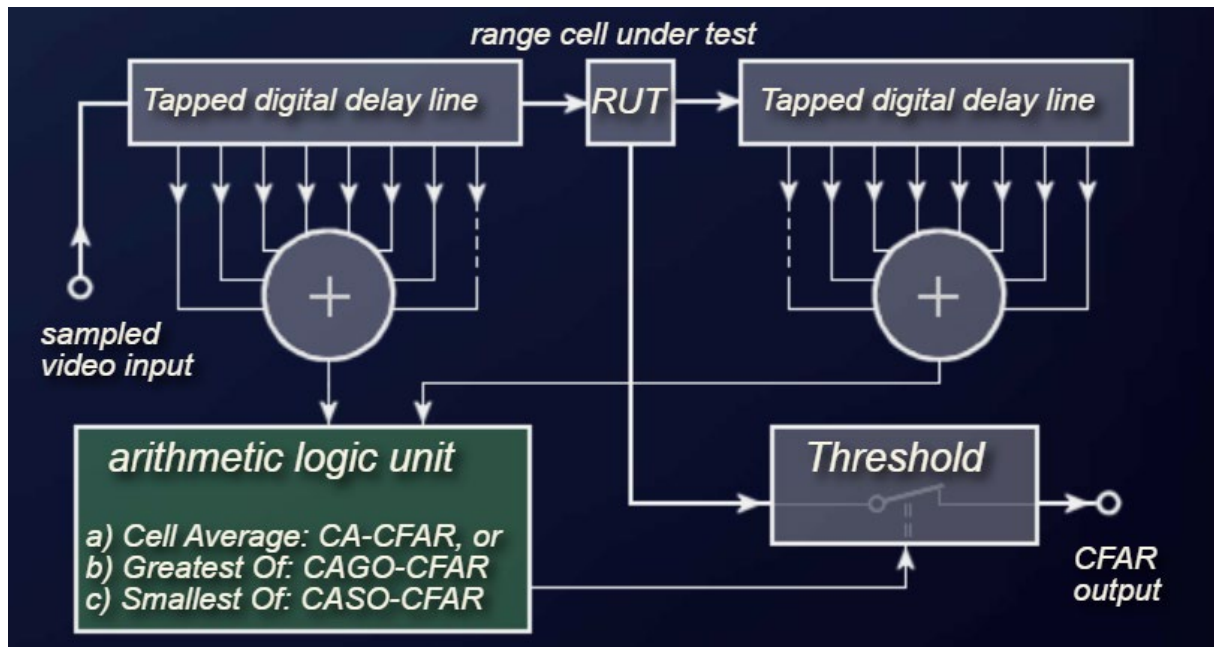


Figure 8 : Principe du "Cell-averaging CFAR"

Le signal reçu est découpé en "cellules" permettant de comparer sa valeur à un instant donné à ses valeurs précédentes et suivantes. Ainsi le circuit estime le niveau d'interférence (clutter et bruit) grâce aux cellules en amont et en aval de la cellule sous test, et utilise cette estimation pour décider s'il y a ou non une cible dans la cellule sous test. Le bruit est considéré comme homogène sur la plage testée donc l'estimation faite autour de la cellule de test est valable pour cette dernière. Ce processus est répété sur toutes les cellules. Nous aurons théoriquement un taux de fausses alarmes constant qui est indépendant du niveau du clutter et de bruit. De plus des cellules de gardes sont ajoutées afin de ne pas prendre en compte le signal écho présent juste avant et juste après la cible, qui présente une forte amplitude.^[2]

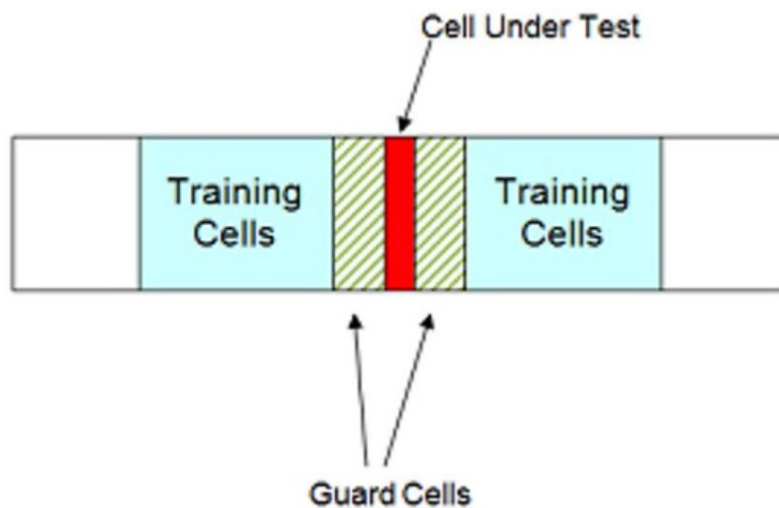


Figure 9 : Schéma de principe des cellules testées

I.3.3 Test de l'algorithme CFAR

Une première implémentation de l'algorithme CFAR a été réalisé en python, présenté en annexe. Il est immédiatement apparu que cet algorithme est très sensible au nombre de cellules d'entraînement et au nombre de cellules de gardes que nous choisissons. Tout d'abord l'algorithme a été testé sur des sinusoïdes bruitées générées de façon aléatoire. Les résultats sont présentés ci-dessous, où les signaux considérés comme cible par l'algorithme sont marqués par un point rouge :

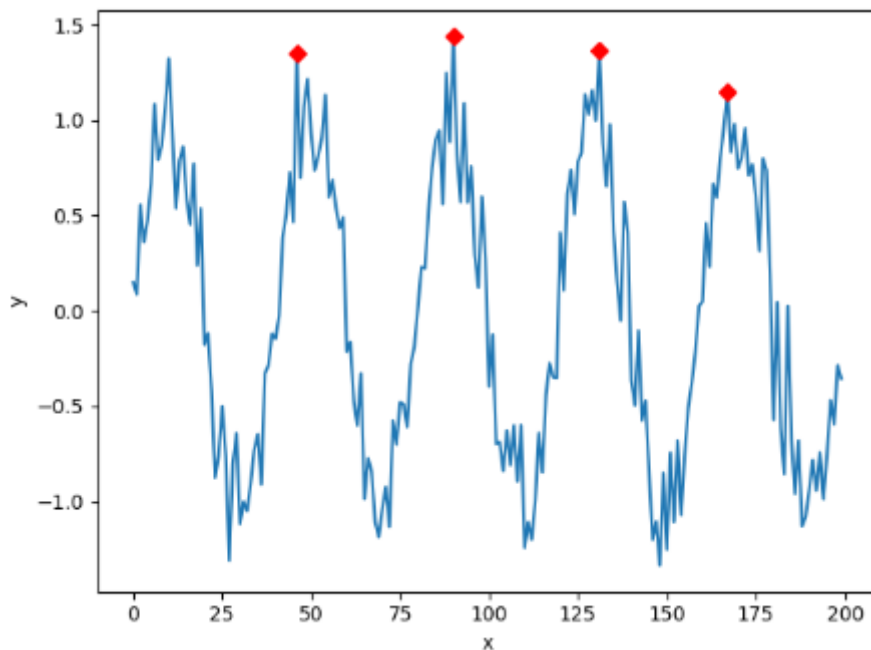


Figure 10 : Résultat de l'algorithme CFAR

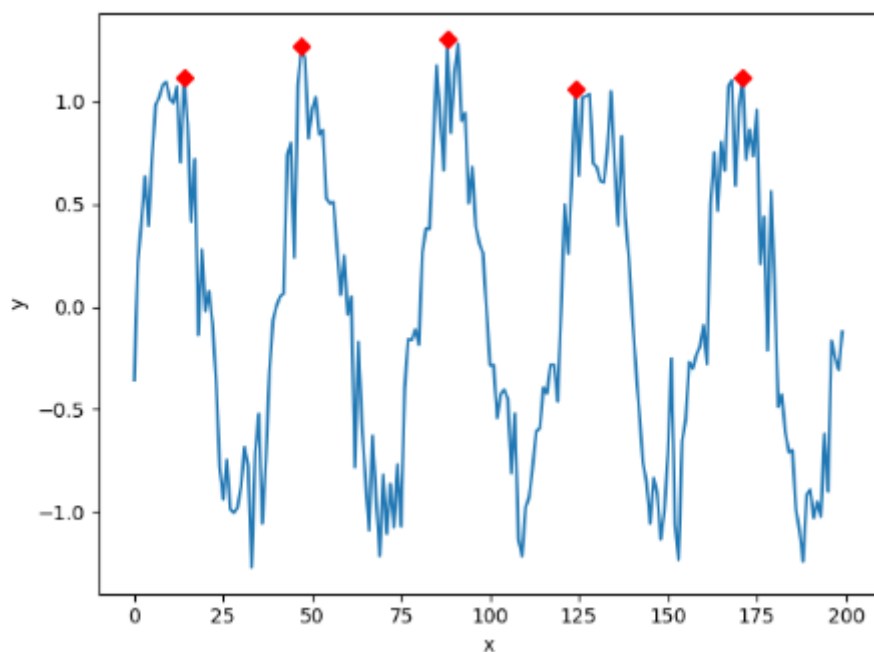


Figure 11 : Résultat de l'algorithme CFAR

Par la suite, l'algorithme a été testé sur le signal acquis avant le re-confinement et présenté à la partie précédente, nous avons obtenus les résultats ci-dessous :

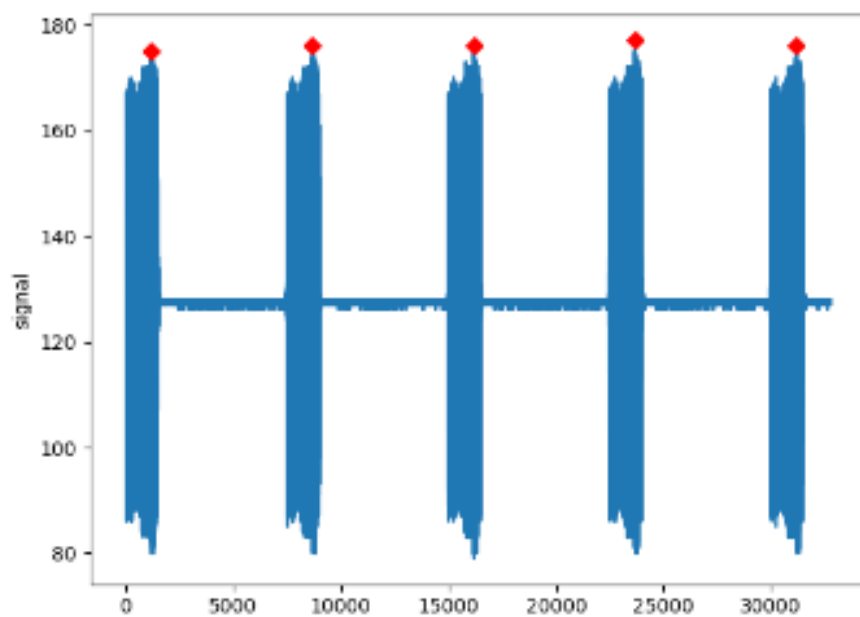


Figure 12 : Résultat de l'algorithme CFAR sur un signal type de nos appareils

Une prochaine étape de notre travail sera de mieux maîtriser le choix du nombre de cellules d'entraînement et de garde mais également d'appréhender une approche 2D avec une prise en compte de l'effet Doppler.

I.4 – Principe du tracking :

Le pistage ou tracking consiste en la dernière étape du traitement de l'information réceptionnée par l'équipement en ce qui concerne notre projet. Elle intervient après la détection. L'étape de détection met à disposition de l'algorithme de tracking des *plots* : ces derniers doivent être liés à une cible et donc à la piste (*track*) liée à cette dernière. Les pistes sont donc une représentation du trajet de la cible. Le but de l'algorithme de tracking est de mettre à jour ces pistes pour permettre de suivre une cible à l'aide des différentes acquisitions du radar.

On va ici décrire un procédé simplifié et très général de la méthode de pistage^[1]. On décrira ensuite les différentes méthodes et différents algorithmes à disposition pour implémenter le pistage en insistant plus particulièrement sur les deux algorithmes que nous avons décidé d'implémenter pour notre projet : l'algorithme des k plus proches voisins (K-NN) et le filtrage de Kalman.

Tout d'abord, chaque piste (*track*) a son historique de données c'est à dire tous les plots qui lui ont été attribué. Ces derniers sont stockés dans une liste.

On commence le tracking ou pistage quand les capteurs détectent correctement un point d'intérêt, c'est à dire qu'on prend soin d'éliminer des données parasites pour éliminer les faux départs. On peut se reporter à la section *Détection* I.3 pour plus de précisions. Une fois ce point communiqué à l'algorithme de pistage, nous sommes face à 3 situations différentes :

- Le point ne correspond à rien : on le *drop* i.e on décide qu'il n'est pas digne d'intérêt après comparaison avec les données disponibles.
- *Track initiation* : On associe le point à une nouvelle piste et donc au pistage d'un autre objet que l'on ne suivait pas avant.
- *Plot to track association* : l'algorithme reconnaît la donnée comme faisant partie d'une piste existante. On met donc à jour la piste en ajoutant ce nouveau plot à la liste de ses plots.

Pour pouvoir réaliser ces étapes, on effectue une prédiction de la nouvelle position basée sur les dernières données recueillies (pour chaque pistes). Pour pouvoir prédire la nouvelle position, on se base sur le dernier état connu de la cible (position, direction, vitesse, accélération). Puis on compare cette prédiction à notre plot détecté. Pour cela on utilise principalement 4 algorithmes :

I.4.1 – Algorithme des plus proches voisins (K-NN pour K Nearest Neighbours) :

Cette méthode se base sur un principe plutôt simple : on se sert des k plots (dont on sait à quelle piste ils sont associés) les plus proches du dernier plot relevé par l'algorithme de détection pour déterminer à quelle piste il appartient.^[2]

L'algorithme calcule tout d'abord la distance du dernier plot à chaque plot détecté au préalable pour établir la liste des k plus proches voisins de notre point d'intérêt. Dans notre cas, on utilise la distance euclidienne. D'autres distances sont parfois utilisées comme la distance de Mahalanobis.

Puis chacun de ces k plots est classé : dans notre cas, une classe représente une piste. Il nous reste à déterminer à quelle classe le plot d'intérêt appartient. On choisit de lui attribuer la classe majoritaire parmi ses k plus proches voisins. Parce qu'on fait le choix d'un "vote majoritaire" on fixe la valeur de k à un nombre impair (ici 3) pour éviter un cas d'égalité et donc éviter de bloquer l'algorithme qui ne dispose pas d'élément pour privilégier une classe plutôt que l'autre.^[3] Pour une première version de l'algorithme K-NN, se reporter à l'annexe.

I.4.2 – Probabilistic data association filter PDAF

Ce type de filtre sert uniquement à associer un plot à une piste. On ne peut pas s'en servir pour initier une piste ou pour la terminer. Comme son nom l'indique, on se base ici sur une approche statistique. On recherche les plots qui ont le plus de chances (statistiquement parlant) de correspondre à la position prévue. Pour cela, on met en place une zone de validation autour du point estimé, c'est à dire une zone dans laquelle chaque plot qui s'y trouve peut correspondre à la cible (cela permet de réduire l'espace dans lequel on fait la recherche). Parmi tous les plots se trouvant dans cette zone, il faut alors trouver celui qui vient bien de la cible et le rattacher la piste correspondante.^[4]

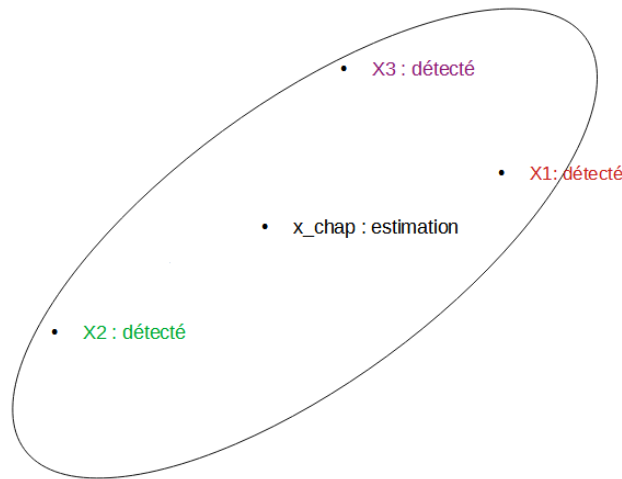


Figure 13 : zone de validation autour de la position estimée

Le filtre, dans sa version la plus simple, va alors calculer l'espérance de chaque plot détecté et chercher à minimiser l'erreur quadratique moyenne. Chaque plot possède un poids différent. Le plot avec l'erreur quadratique moyenne la plus faible sera retenu pour être intégré à la piste de la cible. Ce type d'algorithme se montre efficace quand on cherche à suivre une cible seulement au milieu d'un nuage de données.^[5]

I.4.3 – Multiple Hypothesis Tracking (MHT)

L'algorithme de pistage avec hypothèse multiple est une version plus avancée du PDAF. Avec ce type d'algorithme, une piste peut être mise à jour par plusieurs données. Il en résulte un schéma (appelé « arbre ») qui bifurque dans plusieurs directions possibles créant plusieurs pistes possibles, plusieurs alternatives. Chaque arbre correspond à une cible. L'algorithme calcule ensuite la probabilité de chaque branche et sélectionne la combinaison la plus probable, offrant ainsi la trajectoire de la cible.^[6]

Cet algorithme prend en compte toutes les mesures transmises par l'étape de détection. L'ajout d'une mesure ne fait qu'étendre les différentes branches de l'arbre comme on peut le voir sur cette représentation graphique tirée des travaux de A. Amditis et al.^[7]

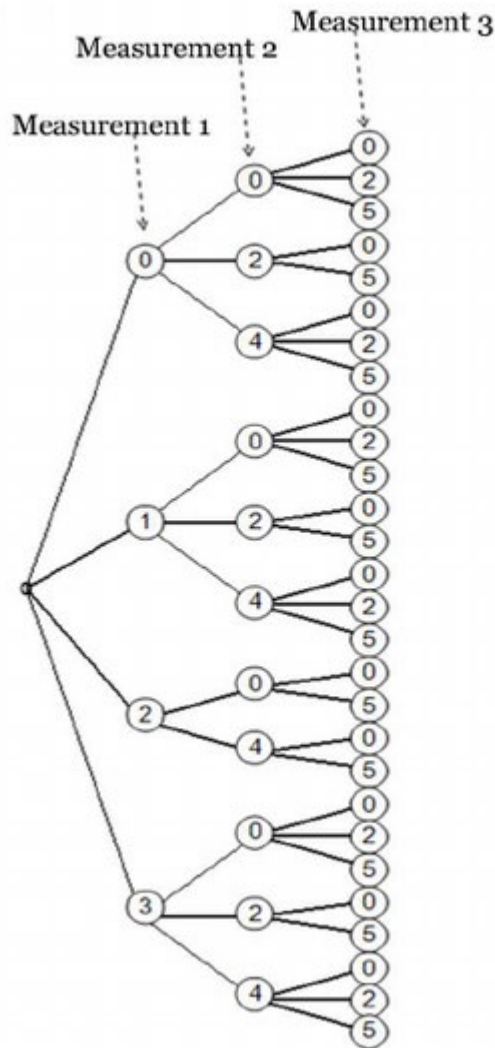


Figure 14 : Arbre formé par les différentes hypothèses ^[7]

Le fait que l'algorithme conserve toutes les pistes possibles fait qu'il est particulièrement intéressant de l'utiliser quand le mouvement de la cible traquée est trop erratique.

I.4.4 – Filtrage de Kalman

Le filtrage de Kalman est le second algorithme que nous tenterons d'implémenter pour analyser nos acquisitions et tenter de pister une cible. C'est une des méthodes de filtrage les plus utilisées aujourd'hui. Un filtre de Kalman possède deux aspects : un concernant la *prédiction* et l'autre concernant la *correction* des mesures.

Contrairement au filtre précédent, celui-ci est récursif, on ne prend en compte que la valeur mesurée précédente.^[7] Cette valeur qui peut être bruitée est corrigée à l'aide des mesures corrigées précédentes, pour obtenir une valeur plus proche de la valeur réelle, puis à l'aide de cette dernière, on prédit la nouvelle position au temps présent. Le filtre a l'avantage notoire de corriger son erreur au cours de son utilisation.

Pour une première version de l'algorithme soutenant le filtrage de Kalman, se reporter à l'annexe. Comme pour l'algorithme K-NN, cette méthode a été implémentée "seule" c'est à dire sans avoir un jeu de données d'entrée. Une prochaine étape de notre travail consistera à améliorer ces deux algorithmes pour qu'ils puissent s'exécuter efficacement avec en entrée, les données fournies par l'algorithme de détection.

II – Les principes de l'ingénierie système :

Comme nous sommes dans la première partie du projet système, nous sommes ici dans la phase de *conception* du cycle de vie du système. Autrement dit, dans la première partie du cycle en v. Nous allons chercher à exposer dans cette partie les exigences, l'architecture fonctionnelle et l'architecture physique qui ont soutenus la conception de notre projet.

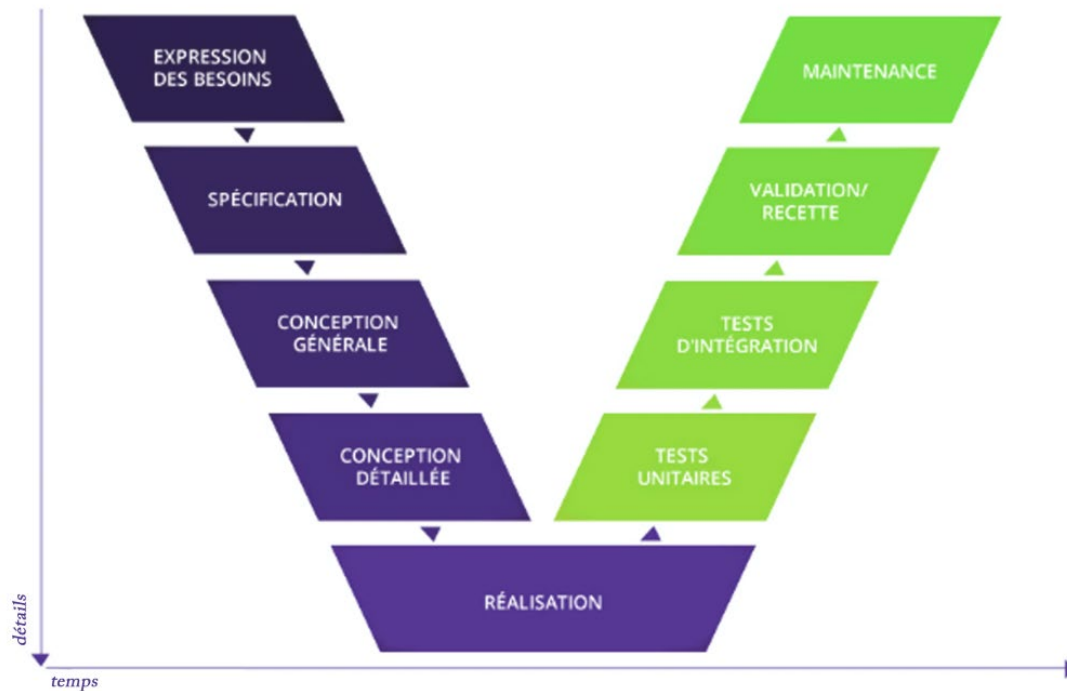


Figure 15 : cycle en V

II.1 – Exigences et architecture fonctionnelle

Fonction	Description
FP : Détection	Détecter un objet immobile ou en mouvement et rendre compte de sa vitesse radiale. Utiliser l'algorithme CFAR
FS : Matière	Rapporter la matière de l'objet détecté par le radar
FC : Filtrage	Réduire le bruit environnant pour analyser correctement l'écho d'ondes électromagnétiques/ Augmenter le rapport signal sur bruit
FS : Simulation des signaux	Produire des signaux proches de ceux obtenus avec la carte d'acquisition
FP : Suivi de cible	Utiliser la technique du beamforming pour orienter électroniquement le lobe d'antenne en fonction du mouvement de la cible
FS : Distinguer	Trouver la distance minimale nécessaire pour détecter deux objets proches
FS : Performance	Trouver la taille minimale requise pour que l'objet soit détecté par l'antenne. Trouver les distances minimales et maximales de l'objet à l'antenne

Tableau 1 : tableau des fonctions du système

II.2 – Architecture physique

La conception de l'architecture physique du système se base sur les exigences exprimées précédemment et sur des principes physiques. Chaque composant est sélectionné pour répondre à un besoin précis. L'équipe encadrante a imposé une liste de matériel à utiliser. Il nous faut donc comprendre le rôle de chaque élément puis effectuer la connectique entre ces derniers.

Comme exposé dans la partie I.1, la base du fonctionnement d'un radar est l'émission et la réception d'une onde. Lors du premier sprint nous avons donc cherché à mettre en place une chaîne de composants pouvant générer une onde hyperfréquence, nous avons aussi réfléchi au moyen d'acquiescer cette dernière.

Le radar est un radar hyperfréquence. On doit donc générer une onde ayant une fréquence de l'ordre de la dizaine de GHz (giga hertz). On choisit l'onde et sa fréquence en fonction du milieu de propagation de l'onde et de l'utilisation du radar. On ne veut pas un radar très longue portée qui soit capable de traverser plusieurs milieux avant de trouver sa cible (qui nécessiterait une longueur d'onde de l'ordre du mètre) mais un radar courte portée pouvant détecter une cible en extérieur et visible : le choix d'une fréquence de 10 GHz est donc cohérent avec notre utilisation. Le signal que nous cherchons à émettre à terme est composé de *pulses*. Avant d'en arriver là, nous cherchons d'abord à générer et modifier correctement un signal simple de type sinusoïde. Cette dernière est générée à une fréquence relativement basse (un peu moins d'un GHz). Il nous faut donc augmenter considérablement la fréquence, ce qui est fait à l'aide d'un Up Converter. Il n'y a ici aucune perte d'informations car l'augmentation de la fréquence ne modifie pas la largeur de bande (500 MHz) qui contient toutes les informations portées par le signal. Bien sûr pour analyser le signal renvoyé par l'objet traqué par le radar, on diminuera la fréquence pour revenir à l'ordre de grandeur des valeurs de départ.

L'ensemble des composants, leur rôle ainsi que les connections entre eux sont exposés dans le tableau et la figure ci-dessous :

Composant	Fonction dans la chaîne	Explications
Générateur 100B	Génération de l'onde	Génération de la sinusoïde
Générateur 100A	Modulation en fréquence du signal émis	Va synthétiser un signal complexe, à bande plus large (500 MHz) et augmenter la fréquence du signal (1GHz). On travaille avec des fréquences hautes mais le générateur 100A ne suffit pas à atteindre nos fréquences de travail
Splitter	Intermédiaire	Le signal d'arrivée est dupliqué. Cela permet de lui appliquer 2 traitements différents

Up converter	Augmentation de la fréquence du signal avant de le transmettre à l'antenne radar	On atteint les très hautes fréquences : 10/15 GHz
Antenne	Emission et réception de l'onde	L'antenne a pour but d'envoyer l'onde sur l'objet à détecter et de réceptionner l'onde émise par l'objet
Down converter	Diminution de la fréquence du signal revenant du radar	Le down converter fait partie de la chaîne chargée de réceptionner l'onde en provenance du radar. On revient donc à nos fréquences de base (1GHz) pour analyser le retour du radar
Oscilloscope	Analyse du signal	Analyse temporelle du signal envoyé vers l'antenne
Analyseur de spectre	Analyse du signal	Analyse spectrale du signal envoyé vers l'antenne. Cependant l'analyseur ne peut pas être utilisé avec de très hautes fréquences

Tableau 2 : tableau des composants

La chaîne que nous avons mise en place pour l'émission de l'onde est la suivante :

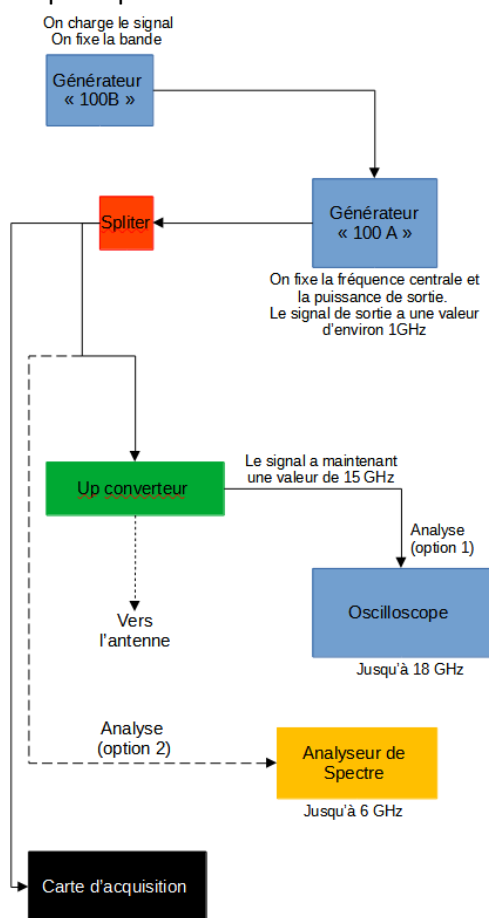


Figure 16 : Chaîne mise en place pour générer le signal

La mise en place de la chaîne doit s'accompagner de différents tests sur les performances de chaque élément. Ces tests unitaires sont nécessaires pour établir les caractéristiques et performances de chaque composant de la ligne et sont donc cruciaux pour pouvoir déterminer les caractéristiques du radar mais aussi sa performance globale.

Les premiers tests unitaires ont été entamés lors du sprint numéro 1 (06/10/2020 au 20/10/2020). Ces derniers devaient être plus poussés mais en raison du confinement national, nous n'avons pas pu avoir accès au matériel pour les poursuivre.

III – L'application de la méthode Agile au sein de l'équipe du projet ObsHyper :

Le projet s'est articulé en trois temps lors de ce semestre. La première phase a été pour nous un temps d'appropriation du sujet et des méthodes agiles qui allaient nous permettre de gérer l'organisation et le fonctionnement de l'équipe. Cette phase du projet a été ponctuée de nombreux rendez-vous avec les encadrants afin d'orienter l'équipe et de la familiariser avec le matériel mis à disposition par l'ENSTA Bretagne. En effet face à l'angle d'approche très large du sujet et la quantité d'informations, les encadrants ont pris soin de nous accompagner sur le plan théorique afin d'être sûrs que nous ayons les bases nécessaires à l'implémentation de ce projet. Cette phase a duré jusqu'au 6 octobre 2020. De plus, nous avons pendant cette période, nous avons commencé à mettre en place la méthode Agile dont nous nous servons pour gérer notre équipe. Le rôle de Product Owner a été attribué à Pierre-Olivier et celui de Scrum Master à Emilie. Nous avons aussi commencé à utiliser régulièrement l'outil Trello pour suivre l'avancement des tâches que nous devons remplir, ceci afin d'être prêts pour commencer les sprints.

Au 6 octobre 2020, nous avons lancé le premier sprint. Dès le commencement du projet, les tâches respectives de chacun des membres de l'équipe ont été choisies par affinités et cela s'est poursuivi lors de la répartition des tâches du sprint 1. Chacun d'entre nous représentant un maillon de la chaîne censée nous mener au produit fini. Nous sommes dépendants du travail de celui qui nous précède dans ladite chaîne. A la fin de sprint, nous avons mis en œuvre la chaîne permettant de générer le signal à destination de l'antenne et avons commencé à exploiter les résultats d'acquisitions simples. Il a été mis en avant, au cours de ces premières semaines, que notre force ici, est notre bonne entente et une aisance de communication non dissimulable, d'autant que l'intérêt que chacun porte personnellement à ce projet est manifeste. Ce qui a été rapporté lors de la première réunion avec notre encadrante *Méthode Agile* Pascale Gautron.

Younouss, jusqu'en octobre notre partenaire, a dû nous quitter, faute de pouvoir rejoindre l'école, nous avons sans peine redistribué sa charge de travail équitablement entre nous et avons poursuivi nos activités à quatre. Or, le 20 octobre 2020, nous avons été pour la deuxième fois confronté à un confinement et à la mise en place des cours à distance. Après une semaine de réadaptation et de consultation avec nos encadrants, nous avons lancé le sprint 2 avec un mode de fonctionnement plus rigide. L'équipe s'est alors articulée comme suit : Julien et Pierre-Olivier ont travaillé, à partir du confinement, sur l'acquisition des données depuis un fichier type récupéré sur les outils de l'école avant le confinement, et le traitement du signal afin de les rendre exploitables par Jany qui travaille sur un algorithme de détection de cible, suivi de près par Emilie qui travaille sur le suivi de ces mêmes cibles. Les recherches pour établir un état de l'art dans ces différents domaines et les premiers développements des algorithmes nécessaires se sont poursuivis jusqu'à la fin du sprint 3 (01/12/2020). Puis pour le dernier sprint du semestre (le sprint 4), nous avons décidé de nous concentrer sur l'écriture du rapport de projet et la mise en forme des différents attendus.

Pour autant, et tant que c'est possible, nous avons tenu à ce que chacun d'entre nous soit capable de comprendre les travaux des autres, c'est pourquoi, au-delà des tâches assignées à proprement parler, nous avons pris soin de nous intéresser en profondeur aux travaux de nos pairs, afin de profiter, à terme, d'un maximum d'agilité.

Le suivi des sprints s'est fait à l'aide des outils Mahara et Trello dont voici les accès :

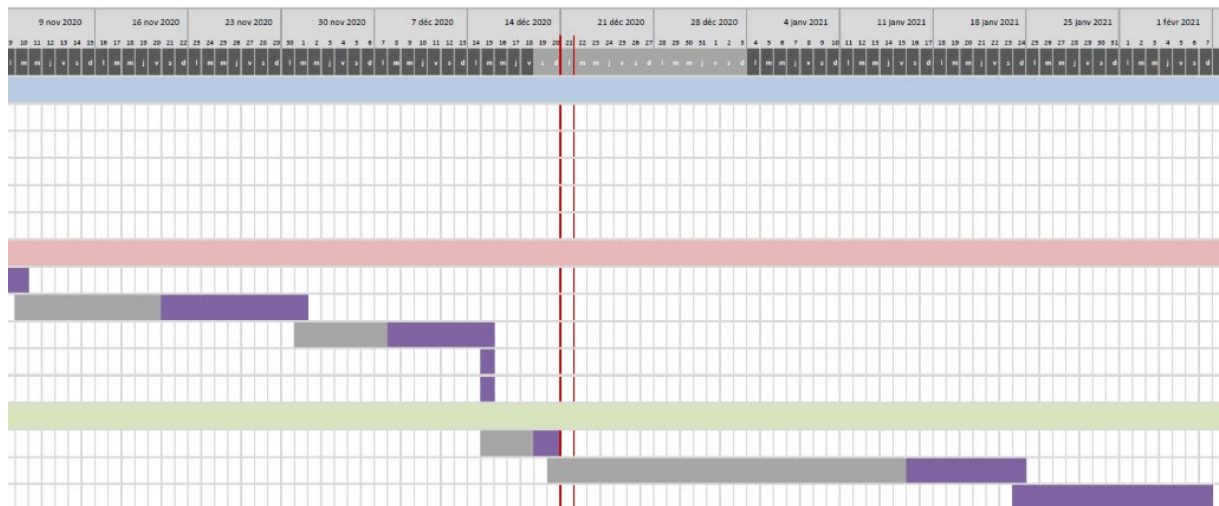
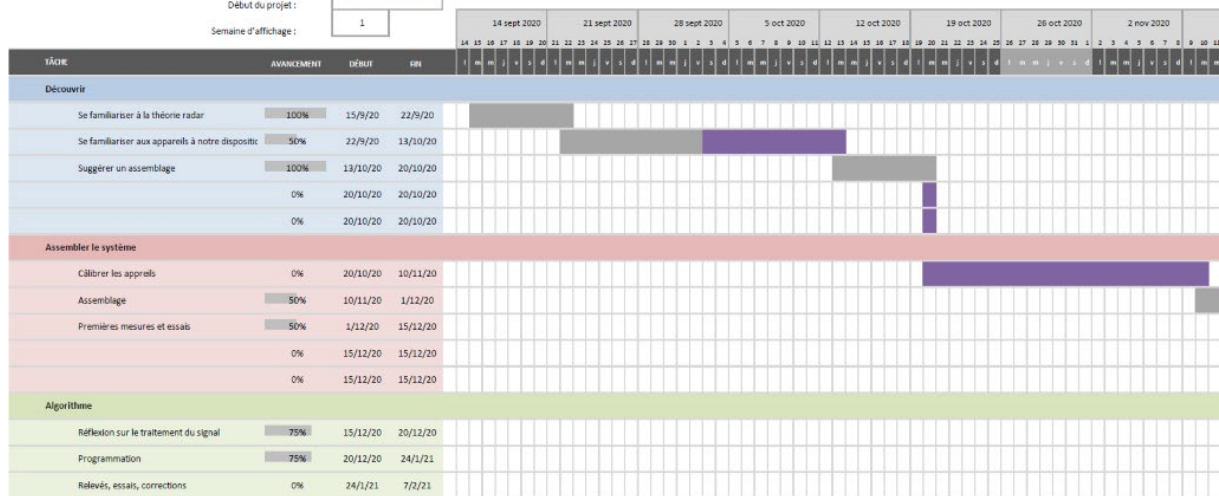
- Mahara : <https://mahara.ensta-bretagne.fr/view/groupviews.php?group=346>
- Trello : <https://trello.com/b/YCVPYzwc/obshyper>

L'éloignement géographique a mis l'équipe au défi, notamment en ce qui concerne la communication et la compréhension des attendus. L'équipe s'est donc tournée vers les réseaux sociaux et les plateformes de communication de type Messenger pour garder le contact. Nous avons cependant conservé Teams pour communiquer avec nos encadrants et pour faire un point sur l'avancement de chacun toutes les deux semaines.



Figures 17 et 18 : Diagramme de Gantt en début de projet

Début du projet :	mar, 15/09/2020
semaine d'affichage :	1



{ 20 }

Conclusion

Au cours de ce premier semestre et lors de la phase de mise en place et de développement de notre projet nous avons pu travailler simultanément sur différents fronts. Tout d'abord nous nous sommes concentrés sur les principes physiques et la théorie soutenant le fonctionnement d'un radar hyperfréquences, cela nous a conduit naturellement à la réalisation d'une partie de ce dernier à l'aide de composants sur étagère. Nous avons pu ainsi mettre en place une chaîne produisant une onde hyperfréquence prête à être communiquée à une antenne de radar. Nous avons aussi travaillé sur la programmation et le traitement du signal, partie essentielle pour exploiter les signaux communiqués par un radar. Nous avons donc implémenté différents codes avec des buts précis : récupérer un signal bruité pour le rendre exploitable, détecter dans ce signal des points trahissant la présence d'une cible et enfin exploiter ces points pour effectuer le suivi de ces cibles. Un à un, ces codes sont fonctionnels, nous cherchons donc maintenant à les faire travailler ensemble et, à terme, avec en entrées nos acquisitions faites sur le terrain. Nous avons aussi travaillé sur nos capacités relationnelles afin de relever les défis imposés par le confinement. Les leçons que nous en avons tiré notamment sur la communication et la gestion de la charge de travail nous seront utiles car les défis auxquels nous avons fait face seront renouvelés car la reprise des cours en présentiel est prévu en février. Nous avons pour la seconde partie du projet de nouveaux objectifs et perspectives. Nous voulons travailler avec le matériel et pouvoir à terme générer des ondes plus complexes dans le but d'avoir un radar fonctionnel capable de détecter un objet sur le terrain de sport de l'ENSTA Bretagne par exemple. Concernant la programmation, nous souhaitons améliorer nos algorithmes comme avancé précédemment mais aussi travailler en parallèle sur d'autres aspects comme le traitement de l'information en temps réel, la furtivité mais aussi, si cela est possible, l'implémentation d'une interface homme-machine.

Bibliographie :

- [1] : Christian Wolff. (1998). *Radartutorial*. <https://www.radartutorial.eu/index.en.html>
- [2] : David Roche. (date non communiquée). *Pixees*.
https://pixees.fr/informatiquelycee/n_site/nsi_prem_knn.html
- [3] : Onel Harrison. (2018). *Towards Data Science*. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
- [4] : bar-shalom, Yaakov & Daum, Fred & Huang, Jim. (2010). The probabilistic data association filter. *Control Systems, IEEE*. 29. 82-100.10.1109/MCS.2009.934469.
https://www.researchgate.net/publication/224083228_The_probabilistic_data_association_filter
- [5] : Kirubarajan, Thia & bar-shalom, Yaakov. (2004). Probabilistic Data Association Techniques for Target Tracking in Clutter. *Proceedings of the IEEE*. 92. 536 - 557. 10.1109/JPROC.2003.823149.
https://www.researchgate.net/publication/2986277_Probabilistic_Data_Association_Techniques_for_Target_Tracking_in_Clutter
- [6] : Kim, C., Li, F., Ciptadi, A., & Rehg, J. M. (2015). Multiple hypothesis tracking revisited. In *Proceedings of the IEEE international conference on computer vision* (pp. 4696-4704)
https://www.cc.gatech.edu/~ckim314/papers/MHTR_ICCV2015.pdf
- [7] : Amditis, Angelos & Thomaidis, George & Maroudis, Pantelis & Lytrivis, Panagiotis & Karaseitanidis, Ioannis. (2012). Multiple Hypothesis Tracking Implementation. 10.5772/33583. https://www.researchgate.net/publication/224829356_Multiple_Hypothesis_Tracking_Implementation

Table des figures :

- Figure 1 : Chaîne radar type - www.radartutorial.eu
- Figure 2 : RS AFQ 100B - www.rohde-schwarz.com
- Figure 3 : RS SMBV 100A - www.rohde-schwarz.com
- Figure 4 : Convertisseur GeoSync Microwave - www.geosyncmicrowave.com
- Figure 5 : Montage type de notre système – J. Duchemann
- Figure 6 : Spectre obtenu en sortie d’algorithme – J. Duchemann
- Figure 7 : Signal obtenu en sortie d’algorithme – J. Duchemann
- Figure 8 : Principe du “Cell-averaging CFAR”
- Figure 9 : Schéma de principe des cellules testées
- Figure 10 : Résultat de l’algorithme CFAR – J. Lahlouh
- Figure 11 : Résultat de l’algorithme CFAR – J. Lahlouh
- Figure 12 : Résultat de l’algorithme CFAR sur un signal type de nos appareils – J. Lahlouh
- Figure 13 : “zone de validation autour de la position estimée” – E. Simon
- Figure 14 : “arbre formé par les différentes hypothèses” nom original “Tree representation of the formed hypotheses” © 2012 DOI 10.5772/33583 [7]
- Figure 15 : “cycle en V” Max Godenn dans *Cycle en V* sur *Planilog*
<https://www.planilog.com/fr/glossaire/cycle-en-v>
- Figure 16 : “Chaîne mise en place pour générer le signal” – E. Simon
- Figures 17 et 18 : Diagramme de Gantt en début de projet – P-O. Pedreno
- Figure 19 et 20 : Diagramme de Gantt aux vacances de Noël – P-O. Pedreno

Table des tableaux :

Tableau 1 : “Tableau des composants” E. Simon

Annexes :

Algorithme de traitement du signal : <https://pastebin.com/GMzSXJtf>

Algorithme CFAR : <https://pastebin.com/FcR4fuQV>

Algorithme K-NN : <https://pastebin.com/J3UcCGg3>

Filtre de Kalman : <https://pastebin.com/LSWZcFM6>