

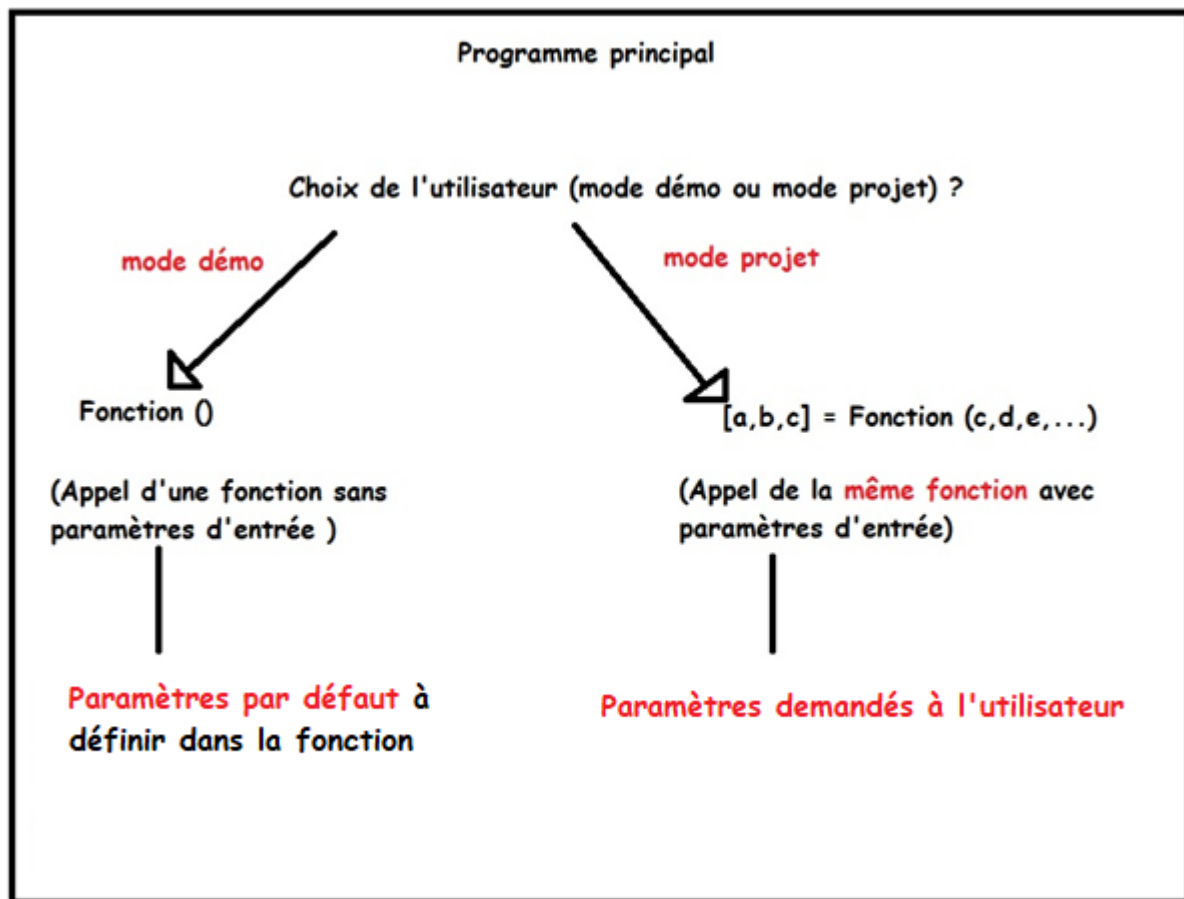
PROJET MATLAB

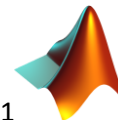
Ce projet comporte 2 sujets de projet indépendants. Il appartient à groupe de choisir un des sujets :

LIRE ATTENTIVEMENT CE QUI SUIT

Au plus tard 48h avant la soutenance, nous retourner les informations suivantes :

- Chaque équipe devra fournir ses livrables (scripts matlab) :





Organisation de la soutenance

Présentation du travail de groupe	5'
Questions-Réponses	10'
Délibération	5'
Total	20'

- Une note /20 sera attribuée à chaque équipe au vu de la qualité de son travail selon le barème suivant.

Compétences		Critères d'évaluation	Barème
C1	Objectifs - Résultats attendus	Les objectifs ont-ils été atteints ? Les résultats obtenus sont-ils conformes aux résultats attendus ?	A/B/C/D
C2	Syntaxe Matlab	Respect des règles de codage Matlab (syntaxe d'une fonction, non redondance de lignes de codes, bon usage des opérations arithmétiques, qualité des commentaires, clarté du code, conformité et cohérence du code avec l'algorithmie) ?	A/B/C/D (x2)
C3	Respect du cahier des charges	Le groupe a-t-il bien réalisé une fonction pouvant être utilisée en mode démo et en mode projet ?	A/B/C/D
C4	Valorisation des résultats - Affichage	Qualité de l'IHM / L'affichage des résultats est-il correct ? Le visuel de l'affichage est-il de bonne qualité ? (titre, labels, limites des axes, variable en abscisse, IHM, ...)	A/B/C/D
C5	Qualité des réponses aux questions	Le code et le projet sont-ils maîtrisés par l'équipe ? L'équipe sait-elle expliquer les lignes de codes et les résultats obtenus ? L'équipe maîtrise-t-elle le sujet posé ?	A/B/C/D (x2)

$A = 5 \text{ pts} / B = 4 \text{ pts} / C = 2 \text{ pts} / D = 1 \text{ pt}$

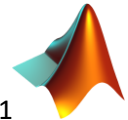
Note finale : $NF = (C1+2C2+C3+C4+2C5)/7$

$NF : A \geq 4.6 / B = [3.6 ; 4.5] / C = [1.6 ; 3.5] / D \leq 1.5$

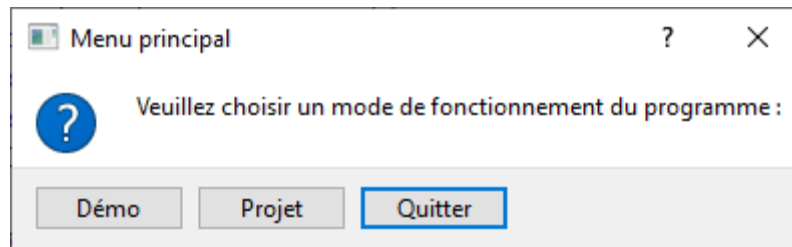
Exemple 1 : B A B B A = $(4 + 10 + 4 + 4 + 10) / 7 = 32/7 = 4.5 = B$

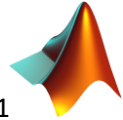
Exemple 2 : A A B B A = $(5+10+4+4+10)/7 = 33/7 = 4.7 = A$

Exemple 3 : B B B B B = $(4+8+4+4+8)/7 = 28/7 = 4 \Rightarrow B$



Exemple d'IHM pour le choix entre mode Démo et projet





SUJET 1 : SURECHANTILLONNAGE D'UN SIGNAL PAR TECHNIQUES D'INTERPOLATION

Introduction

Dans ce sujet, nous nous proposons de sur-échantillonner un signal d'un facteur entier M : ce qui signifie calculer $(M-1)$ valeurs intermédiaires entre 2 échantillons consécutifs.

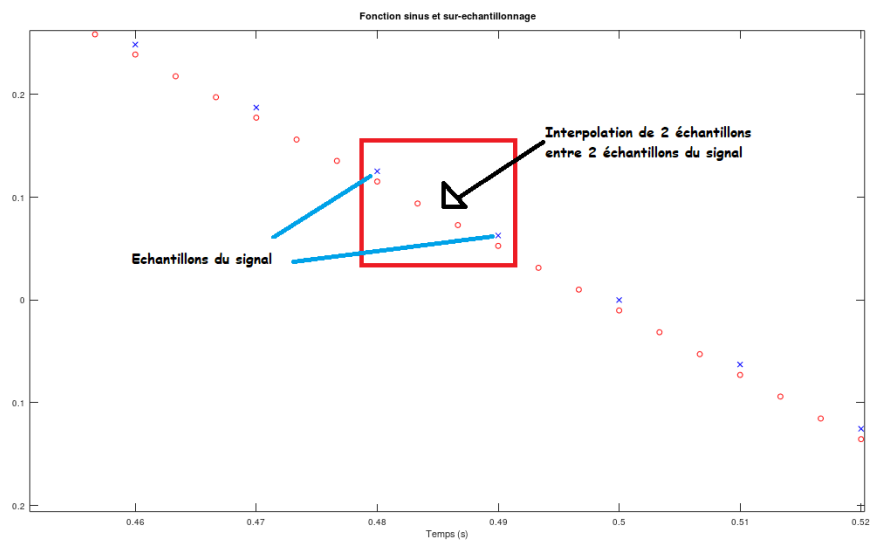
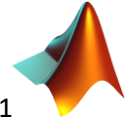


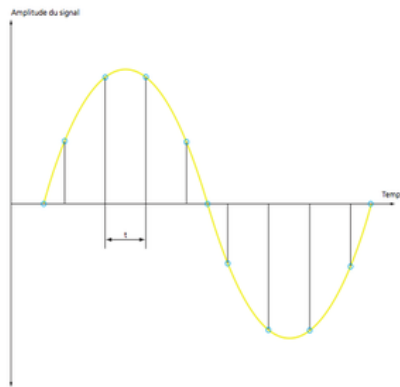
Figure 1 : interpolation d'un facteur 3 : 2 échantillons sont créés entre 2 échantillons du signal d'origine



Contexte applicatif du projet

Pourquoi sur-échantillonner un signal ?

Basse fréquence d'échantillonnage



Haute fréquence d'échantillonnage

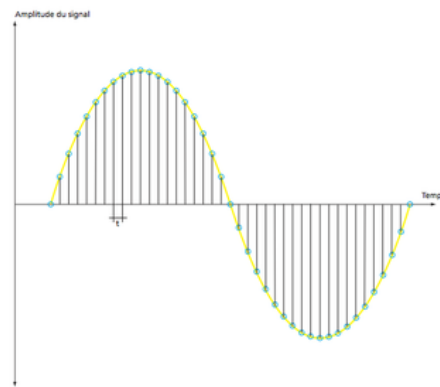
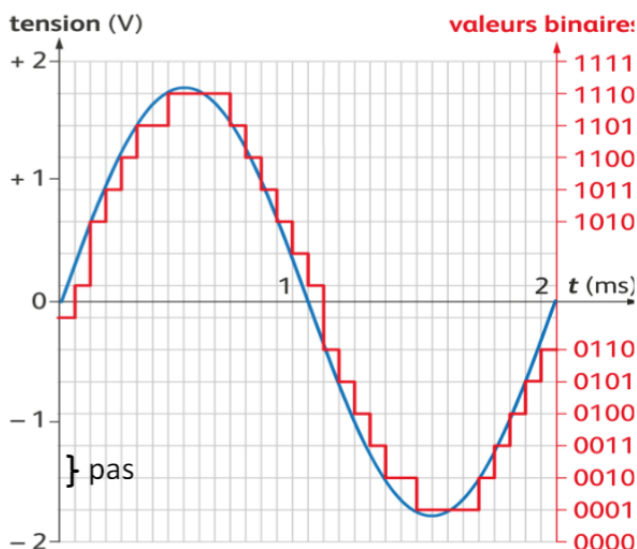


Figure 2 : Un même signal échantillonné à 2 fréquences d'échantillonnage différentes

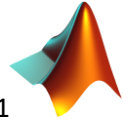
Des applications très spécifiques peuvent faire appel à ces opérations. Signalons par exemple le **changement de fréquence d'échantillonnage** : on veut passer de la fréquence d'échantillonnage 44,1 kHz à 49 kHz. Une solution est de sur-échantillonner d'un facteur 10 puis de sous-échantillonner d'un facteur 9.

Autre exemple, on veut passer d'une représentation sur 16 bits à une représentation sur 14 bits. Ignorer 2 bits conduit à perdre 12 dB sur le **rapport signal/bruit** de quantification. **On peut regagner environ 6 dB** si on sur-échantillonne d'un facteur 4.



$$\text{SNR de quantification} \\ \text{SNR} = 6,02 N + 1,76\text{dB} \\ (N = \text{nombre de bits})$$

Figure 3 : Exemple d'un codage sur 4 bits d'un signal



Donnons un dernier exemple : lors de la restitution des signaux audio-fréquences (parole, musique), le sur-échantillonnage fournit une solution simple **pour ne pas avoir à réaliser de filtre passe-bas analogique** en sortie de la conversion numérique/analogique.

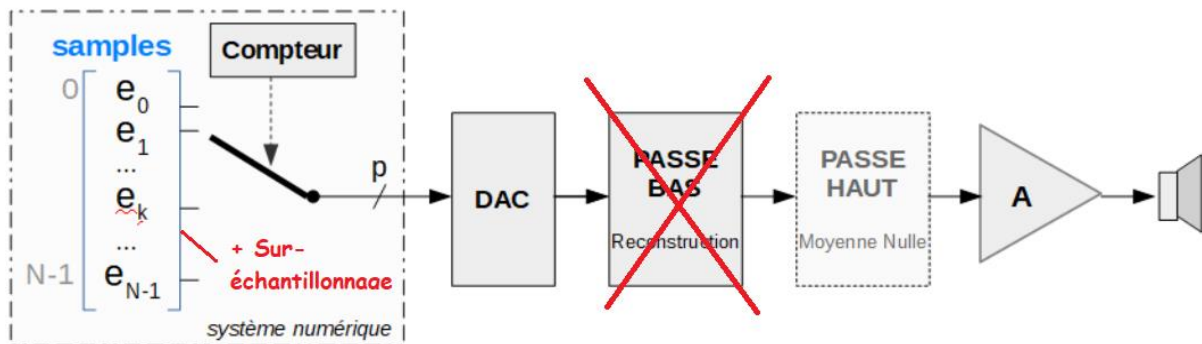
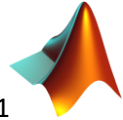


Figure 4 : Chaîne de restitution d'un signal analogique



Objectifs du projet

Il s'agit de réaliser 3 techniques d'interpolations :

- Celle proposée par Matlab (interp) : vu en TP MATLAB n°5
- Interpolation par technique zero-padding : fonction à concevoir par l'apprenti(e)
- Interpolation linéaire : fonction à concevoir par l'apprenti(e)

Puis pour l'aspect applicatif :

- Créer une fonction permettant de supprimer N bits sur un signal numérique

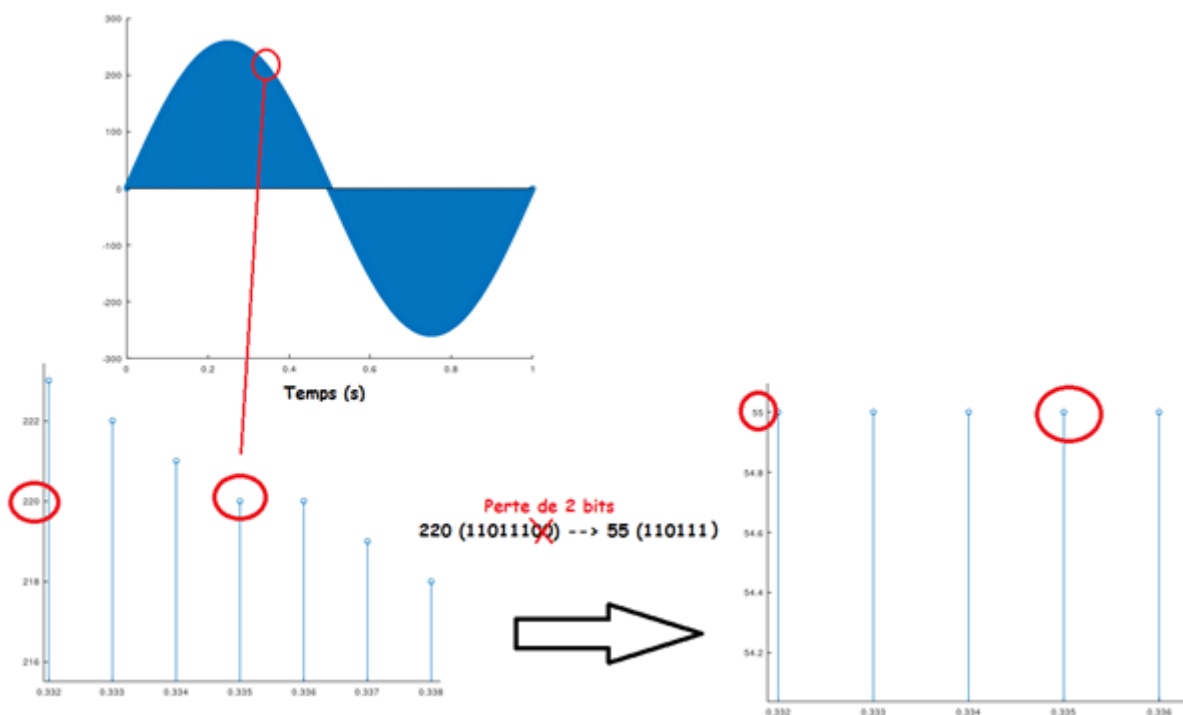
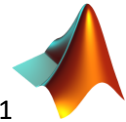


Figure 5 : Exemple de la suppression de 2 bits sur un signal numérique



Le programme doit prévoir de demander à l'utilisateur en mode projet :

- Quelle méthode d'interpolation à utiliser :
 - Option 1 : 'interp'
 - Option 2 : 'zero-padding'
 - Option 3 : 'lineaire' (dans tout le projet, seul un taux de sur-échantillonnage = 2 sera appliqué pour la méthode 'lineaire')
- 1. Quel taux de sur-échantillonnage (uniquement pour la fonction interp et zero-padding) à appliquer :
 - un taux de sur-échantillonnage de 2 signifie qu'on crée 1 échantillon entre 2 échantillons de signal : $F_e \rightarrow 2 * F_e \rightarrow$ l'intervalle entre 2 échantillons temporels est divisé par 2 → **mettre à jour la variable temps !**
 - un taux de sur-échantillonnage de 3 signifie qu'on crée un 2 échantillons entre 2 échantillons de signal : $F_e \rightarrow 3 * F_e \rightarrow$ l'intervalle entre 2 échantillons temporels est divisé par 3 → **mettre à jour la variable temps !**
 - etc ...
- 2. Le nombre de bits à supprimer sur le **signal d'étude** (fichier signal numérique fourni par l'enseignant)

Le signal d'étude sur 8 bits (bits à supprimer et à sur-échantillonner) ainsi que la **fréquence d'échantillonnage** vous seront fournis dans le **fichier signal.dat** : ils pourront être chargés via la commande :

>> load signal (commande à mettre dans votre script matlab).

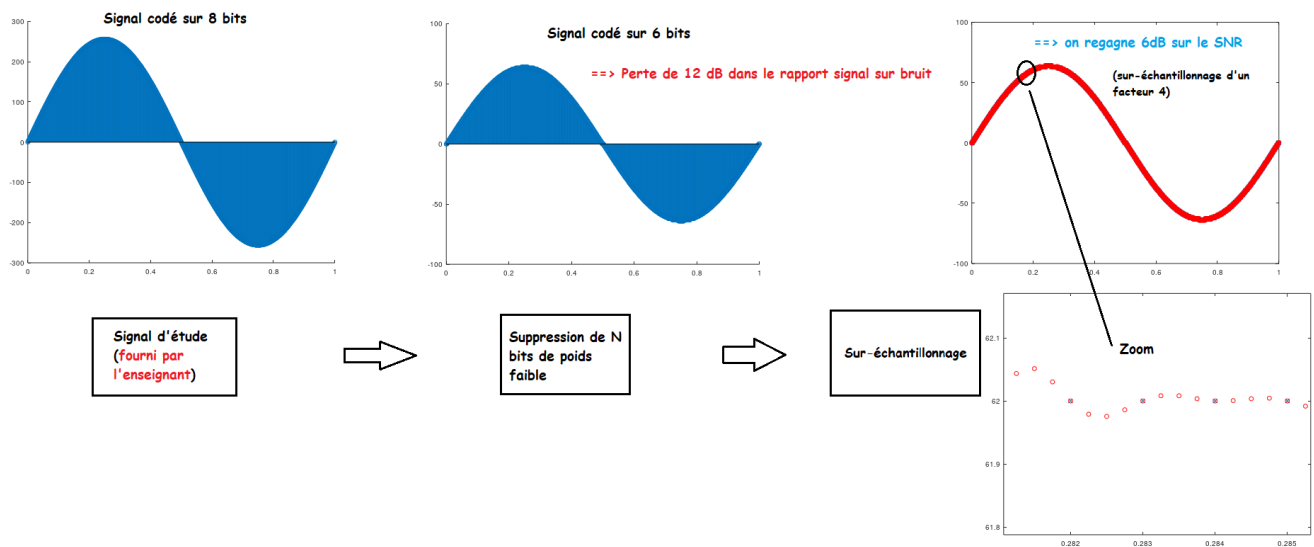


Figure 6 : Scénario en mode projet

Si le mode démo est choisi par l'utilisateur :

3. On appellera la **méthode d'interpolation linéaire** (taux de sur-échantillonnage égal à 2) sur un **signal sinusoïdal à générer par l'apprenti** :

- Caractéristiques du signal :
 - Durée : 10 secondes
 - Fréquence d'échantillonnage : 10 Hz

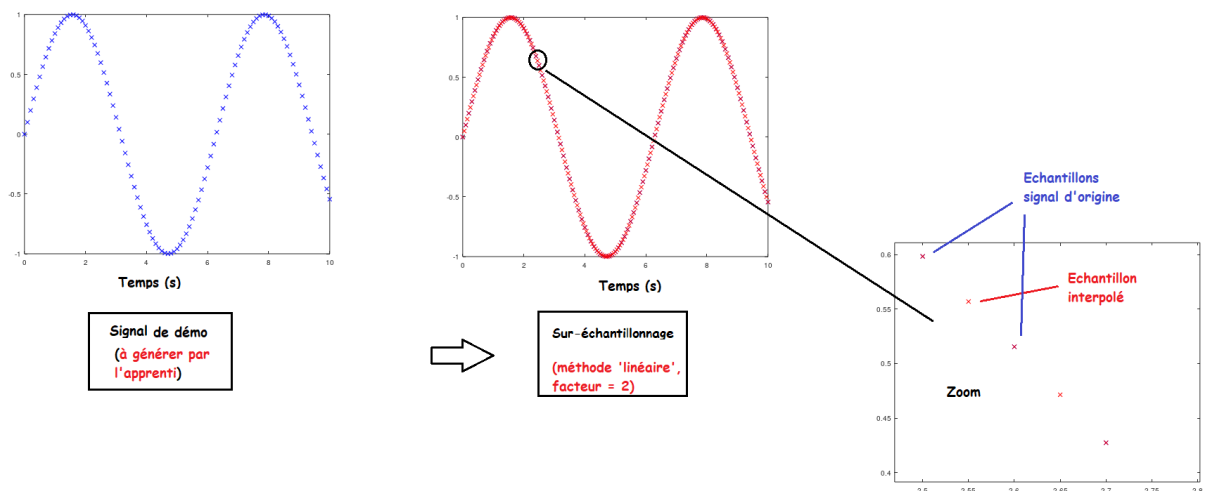
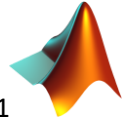


Figure 7 : Scénario en mode démo



Principe de la méthode 'zero-padding'

La méthode 'zero-padding' consiste à rajouter des zéros dans le spectre complexe du signal : ceci aura pour effet d'augmenter « artificiellement » votre fréquence d'échantillonnage et donc diminuer l'intervalle de temps entre 2 échantillons temporels. Ci-dessous l'algorithme à programmer en Matlab dans votre projet (Figure 8)

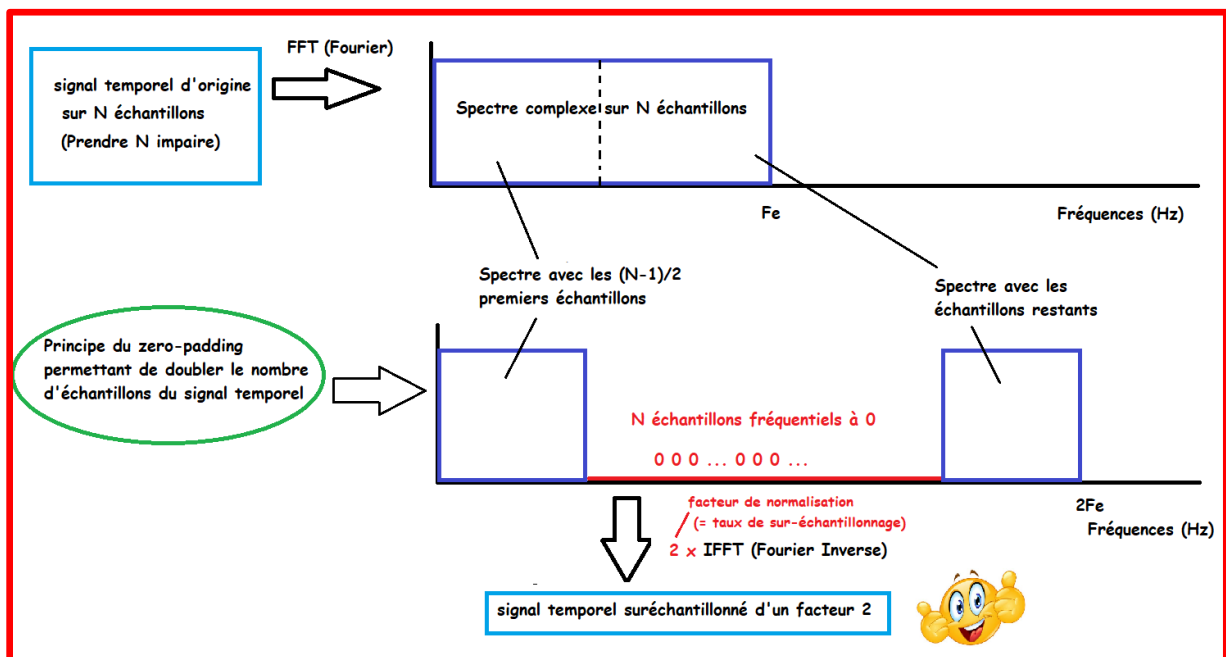
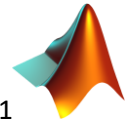


Figure 8 : Algorithme à développer sous Matlab (à adapter en fonction du taux de sur-échantillonnage fourni par l'utilisateur : Ici taux de sur-échantillonnage = 2)



En fait, comme le montre la figure suivante, ceci revient à rajouter des échantillons à 0 sur les bords du spectre centré (obtenu par FFTSHIFT) :

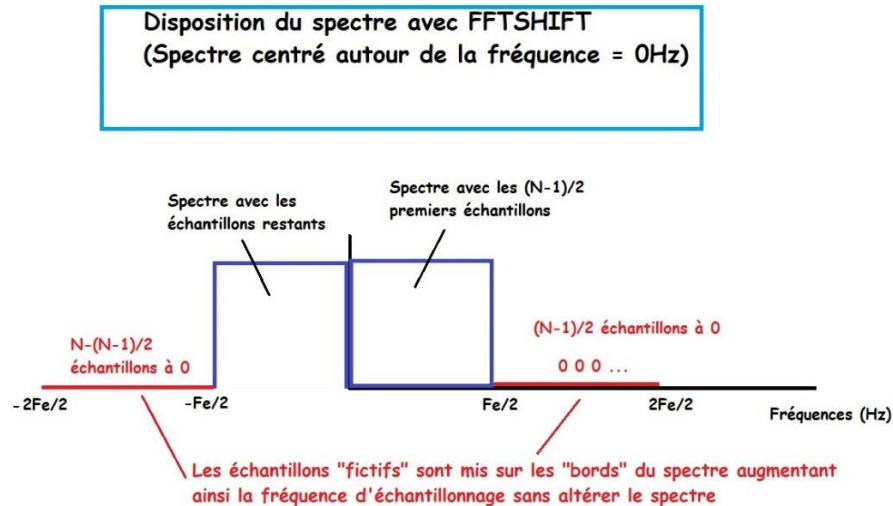


Figure 9 : Ajout de zéros sur les bords du spectre centré

Principe de la méthode 'lineaire'

Nous allons aborder dans ce projet, le cas le plus simple de l'interpolation linéaire. Le schéma ci-dessous présente le principe de la méthode à implémenter en Matlab :

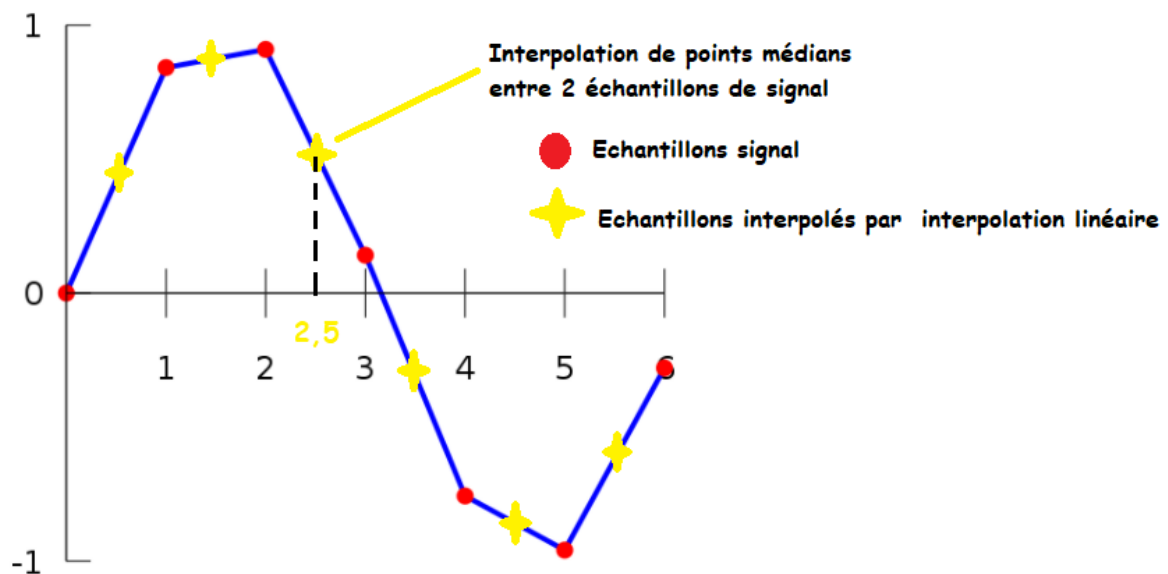
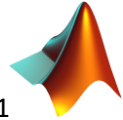


Figure 10 : Calcul de points interpolés (moyenne arithmétique de 2 points adjacents de signal)

Exemple :



Par exemple, si nous souhaitons déterminer $f(2,5)$ alors que l'on connaît les valeurs de $f(2) = 0,9093$ et $f(3) = 0,1411$, cette méthode consiste à prendre la moyenne des deux valeurs sachant que 2,5 est le milieu des deux points. On obtient par conséquent $f(2,5) \approx \frac{0,9093 + 0,1411}{2} = 0,5252$.

Principe de la suppression de N bits sur un signal



Figure 11 : Décalage de 2 bits vers la droite pour supprimer 2 bits

Récapitulatif « Projet Suréchantillonnage »

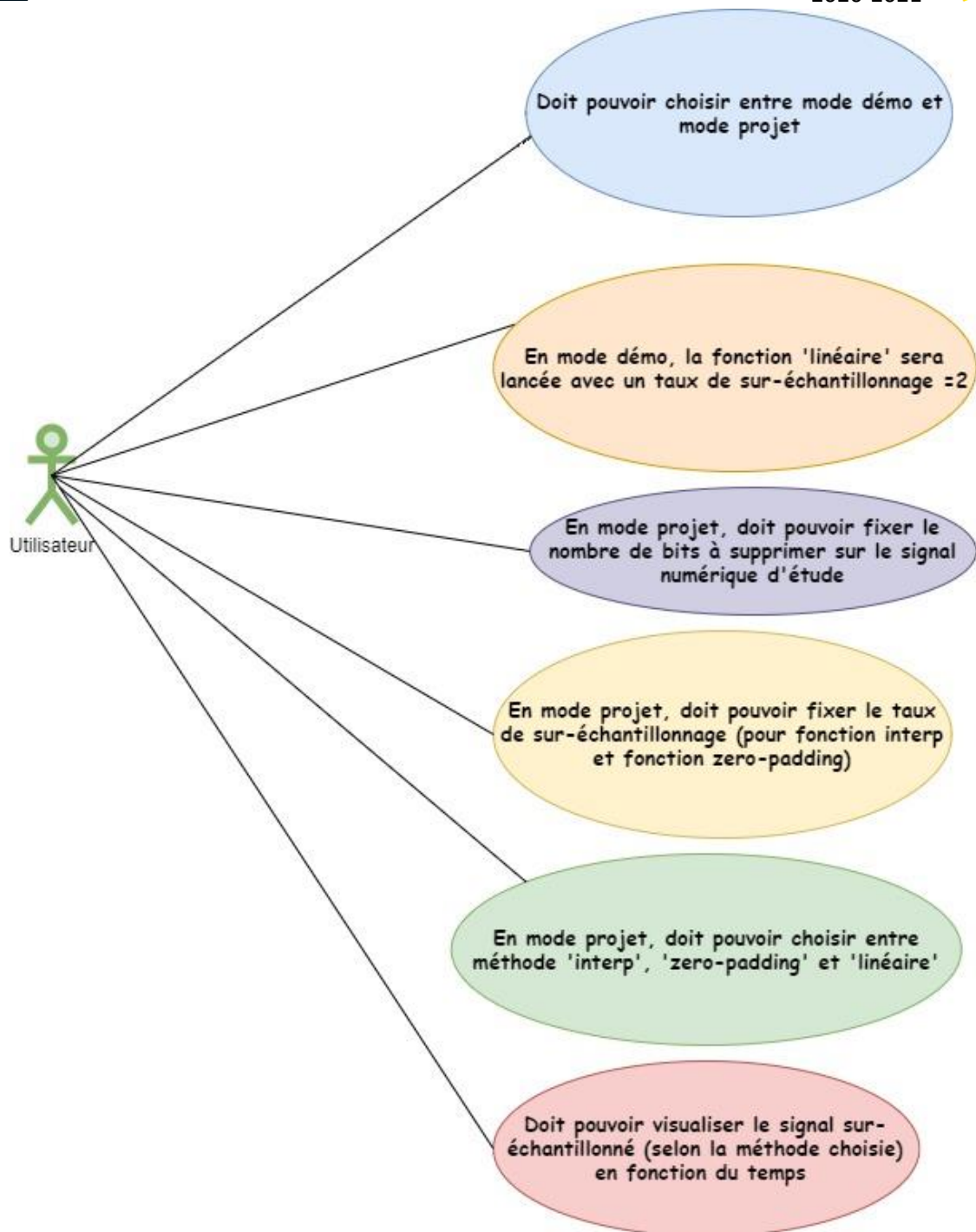
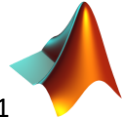


Figure 12 : Récapitulatif du cahier des charges vis-à-vis du « client » (utilisateur)

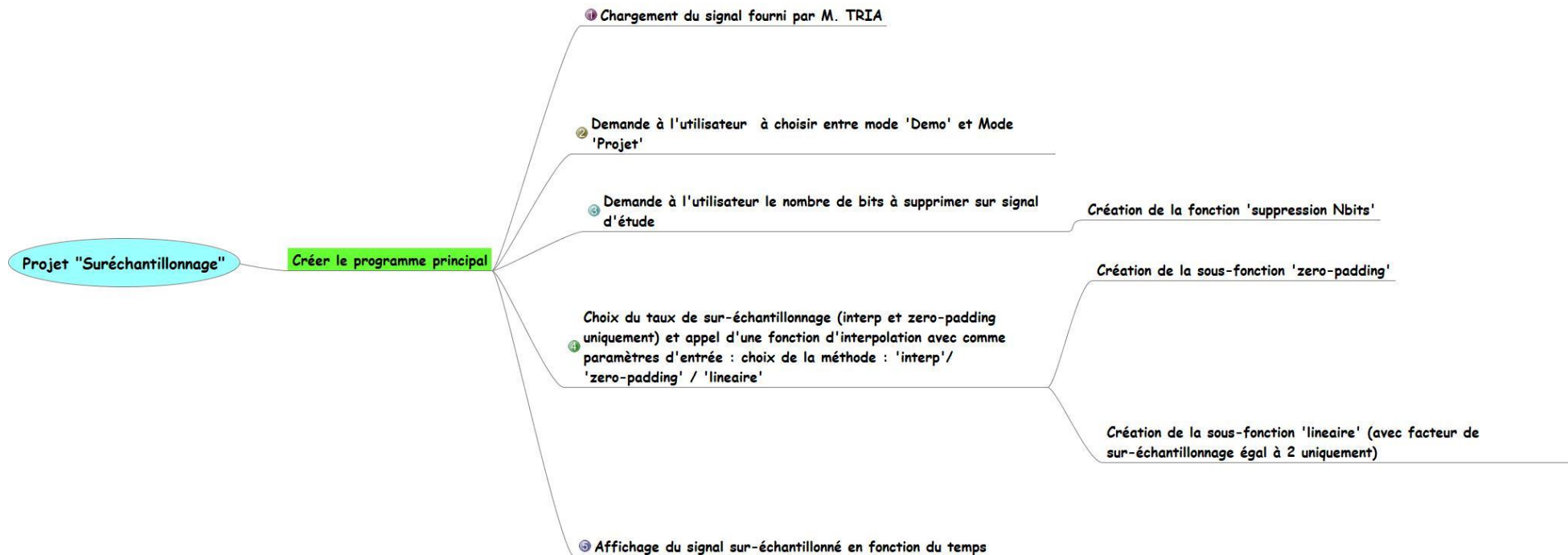
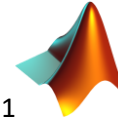
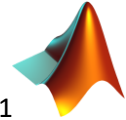


Figure 13 : Travail à réaliser par l'équipe



SUJET 2 : DETECTION RADAR PAR ALGORITHME CFAR

Introduction

Le principe de l'algorithme CFAR consiste à calculer un seuil à partir d'un taux de fausse alarme fixé par l'utilisateur. Au dessus de ce seuil, on décide qu'une cible est détectée (voir Figure 14).

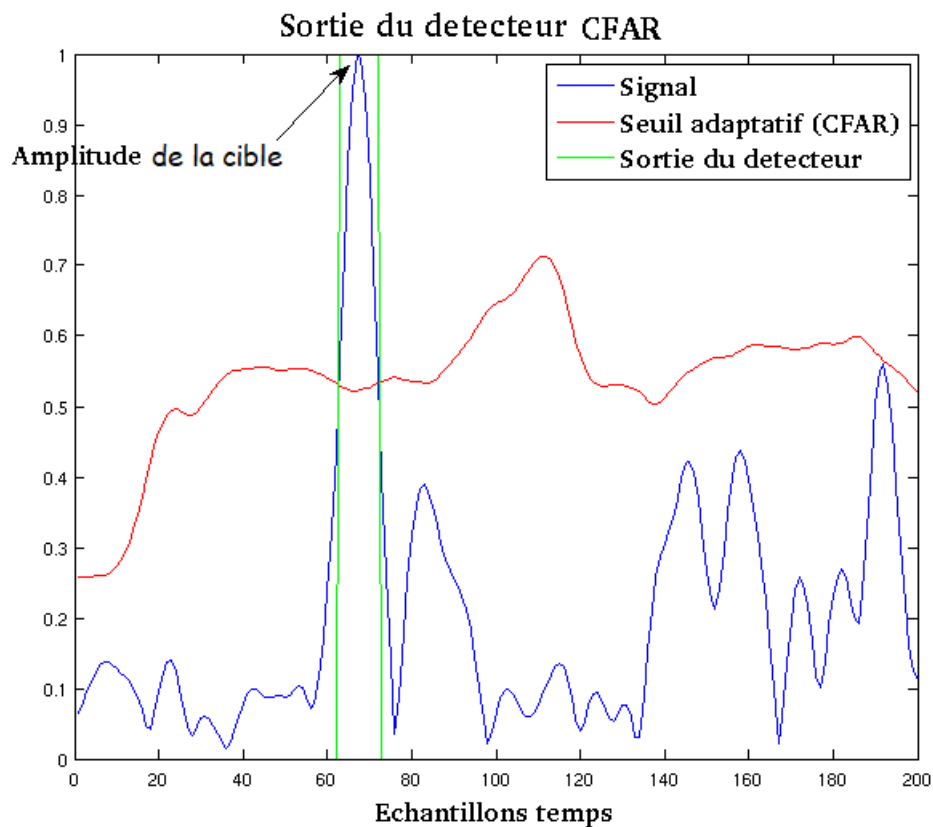
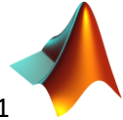


Figure 14 : Création d'un seuil qui s'adapte aux données : l'algorithme CFAR



Contexte expérimental du projet

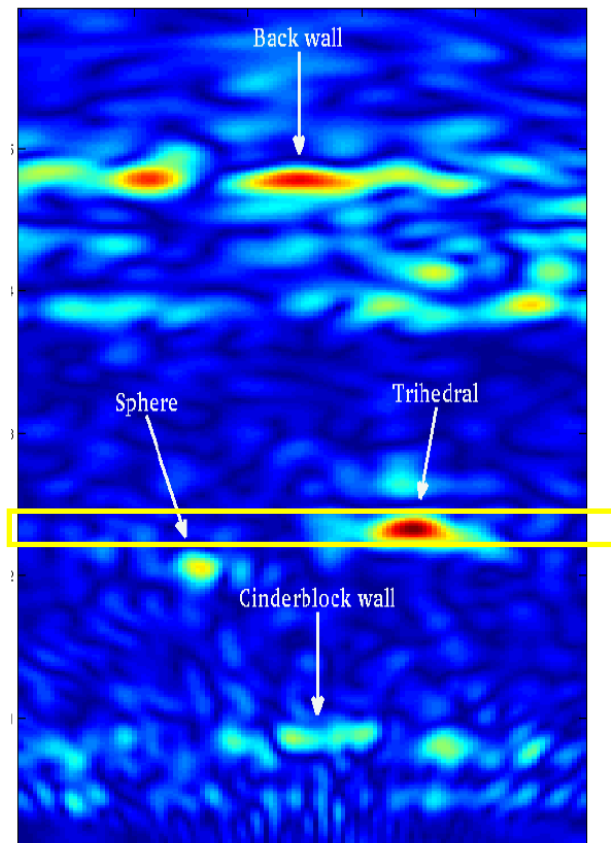
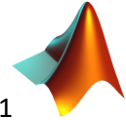
Tout d'abord, je tiens à remercier Robert J. Burkholder, Professeur émérite du Département Génie Electrique de l'Université de l'Ohio qui a eu la gentillesse de nous autoriser à utiliser les données présentées ci-dessous.



Le signal étudié dans ce projet est récolté par un radar (à gauche) se déplaçant à travers le mur. Dans un premier temps, le radar irradie une scène de l'autre côté du mur. Cette scène contient 2 cibles (trièdre et sphère métalliques). Ces cibles renvoient le signal vers le radar.



Figure 15 : Détection de 2 cibles métalliques : sphère et trièdre



Cadre de notre projet :
coupe 1D pour détecter le
trièdre

Figure 16 : Image radar mettant en évidence les 2 cibles métalliques

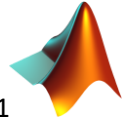
Principe de l'algorithme radar CFAR

Le seuil de décision est calculé en utilisant un voisinage autour d'une cellule test (échantillon dont la valeur sera comparée au seuil calculé). Dans les travaux de détection de cibles en radar [Messali, 2005 ; Ghandhi et al., 1988 ; Saniie, 1992], 3 variantes de l'algorithme CFAR existent :

1. CA (Cell Averaging) ;
2. CAGO (Cell Averaging - Greatest Of);
3. CASO (Cell Averaging - Smallest Of).

Voici les différentes étapes de l'algorithme CFAR :

1. Sélection d'un échantillon de signal
2. Création d'un groupe d'échantillons situé à gauche et création d'un groupe d'échantillons à droite :
 - a. On prendra 25 échantillons à gauche et 25 échantillons à droite
3. Moyennage du voisinage à gauche et moyennage du voisinage à droite
4. Choix du variant de l'algorithme :
 - a. $Z = Z1+Z2$ [CA : Cell Averaging]



- b. $Z = \text{MAX}(Z1, Z2)$ [GO : Greatest Of]
- c. $Z = \text{min}(Z1, Z2)$ [SO : Smallest Of]

- 5. Calcul du seuil final $S : Z \times T$ (*calcul du paramètre T à partir du PFA (taux de fausse alarme) : donné par l'enseignant*)
- 6. Comparaison de l'échantillon de signal avec S et Décision D :
 - a. Si l'Echantillon $\geq S \Rightarrow$ échantillon déclaré comme appartenant à une cible $\Rightarrow D = 1$

L'algorithme doit être effectué en balayant tous les échantillons du signal \Rightarrow pour chaque échantillon du signal correspond une valeur de Seuil S et une valeur de décision D

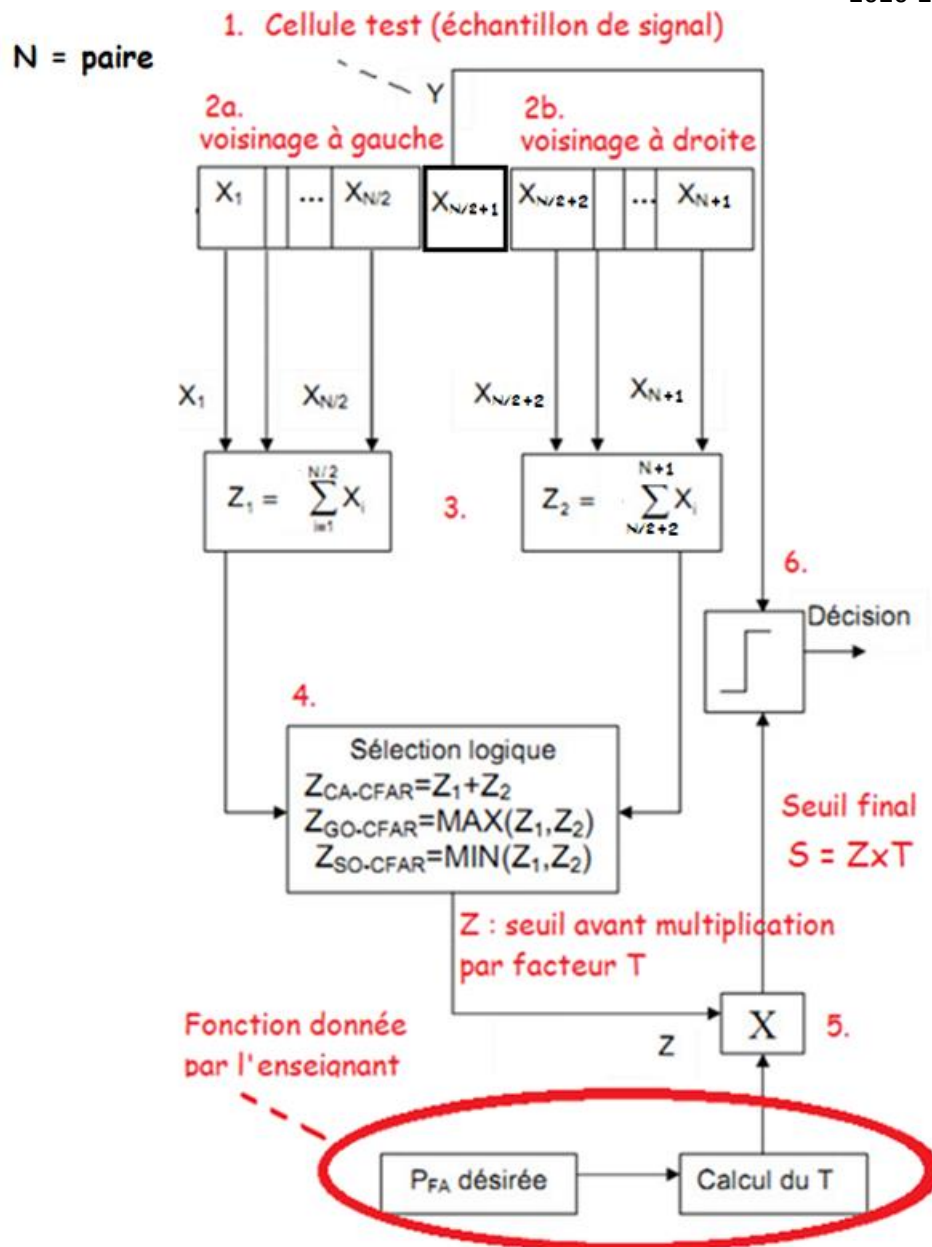
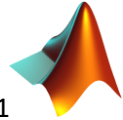


Figure 17 : Les différentes étapes de l'algorithme CFAR

Objectif du projet

Il s'agit de développer l'algorithme CFAR présenté précédemment sous ses 3 variantes :

- CFAR-CA
- CFAR-GO

- CFAR-SO

Le programme doit prévoir de demander à l'utilisateur :

- Quel taux de fausse alarme souhaite t-il ?
 - Prendre un taux de fausse alarme pfa autour de 0.1
- Quelle méthode ci-dessus souhaite t-il exécuter ?

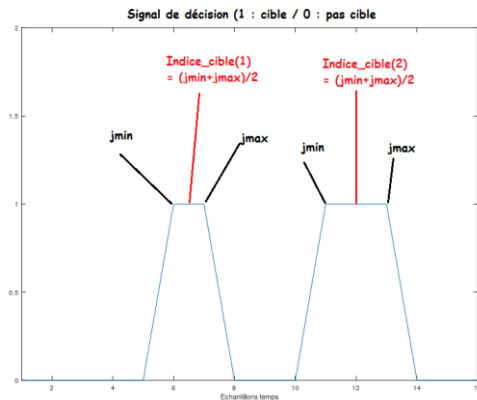
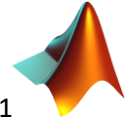
Si le mode démo est choisi par l'utilisateur :

4. On appellera la méthode CFAR-CA avec un taux de fausse alarme $pfa = 0.1$

Le résultat à afficher :

5. Il s'agira d'afficher le signal en superposant la fonction seuil S et le résultat de la détection D (si échantillon signal $> S \Rightarrow$ Décision = 1 sinon Décision = 0)
6. Indiquer le nombre de cibles et leur localisation en mètres

Algorithme « Nombres de cibles et localisation »



Nb_cibles = 0 ; i = 1 % Nb_cibles : nombre de cibles

Tant que (i <= longueur du signal)

Si signal(i) == 1 % Si on détecte une cible

Nb_cibles = Nb_cibles+1; % on incrémente Nb_cibles

jmin = i ; j = jmin ; % on identifie la limite inf de la cible

Tant que signal(j) == 1 % on balaye le rectangle de la cible

j = j+1 ;

Fin Tant que

jmax = j-1 ; % on identifie la limite sup de la cible

% calcul de l'indice milieu

Indice_cibles(Nb_cibles) = round((jmin+jmax)/2);

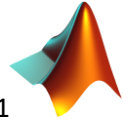
i = jmax+1 ; % i est mis à jour après la cible détectée

Sinon

i = i+1; % si pas de cibles détectées => on incrémente i

Fin Si

Fin Tant que



Récapitulatif « Détection Radar »

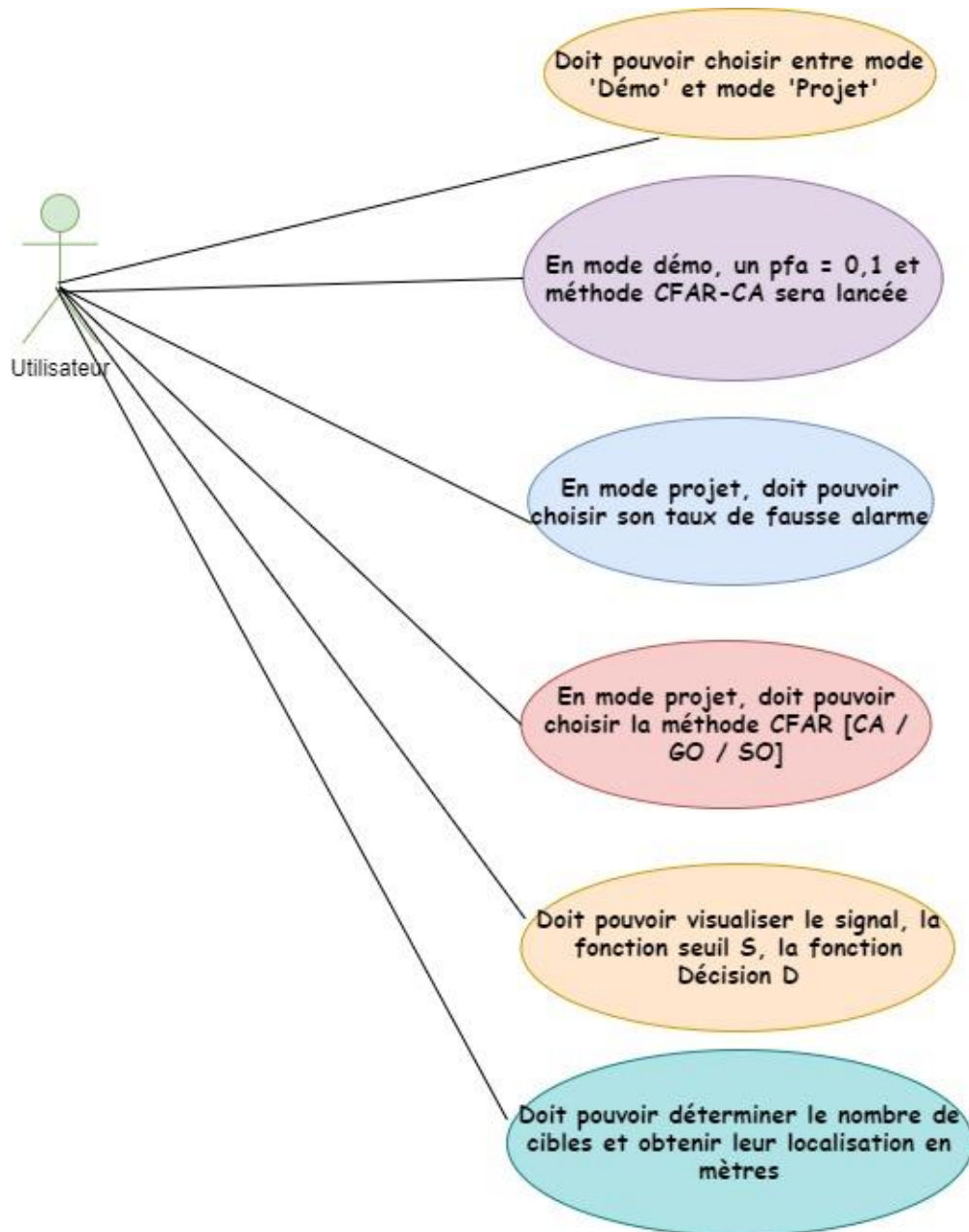


Figure 18 : Récapitulatif du cahier des charges vis-à-vis du « client » (utilisateur)

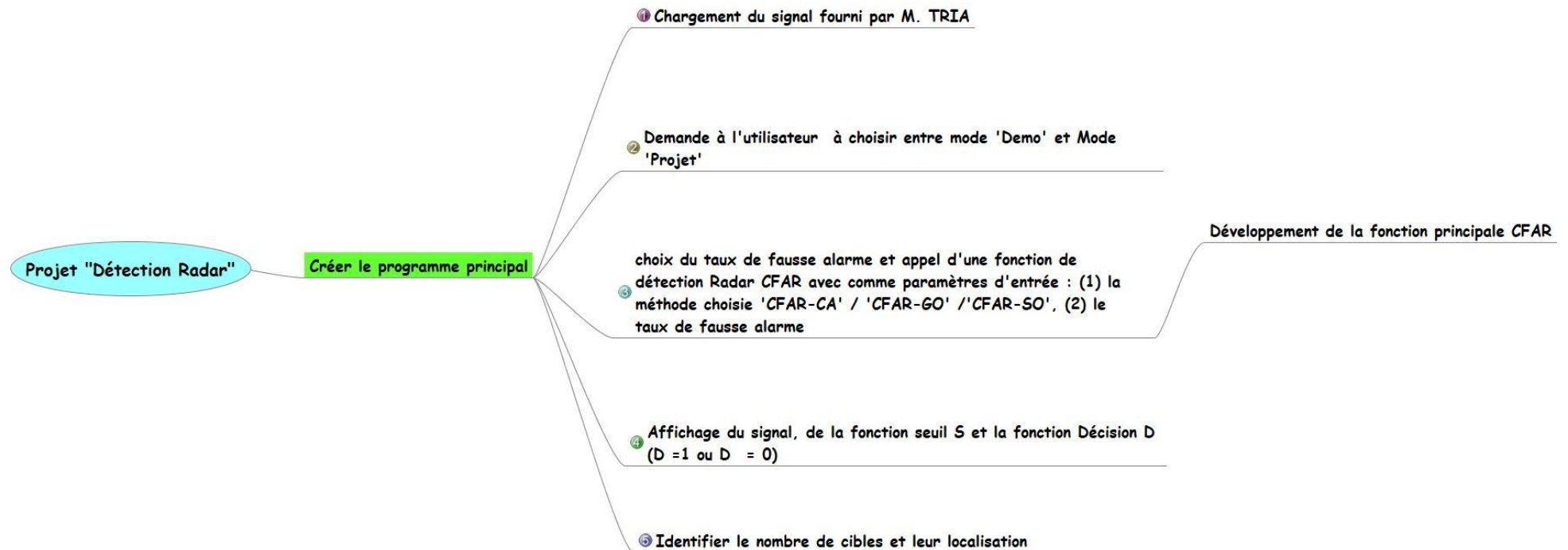
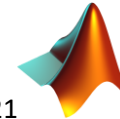


Figure 19 : Travail à réaliser par l'équipe