

# Simulation d'un émetteur/récepteur ADS-B et traitement des données pour le suivi en temps réel des avions.

Rapport de projet TS229 des communications numériques.

Khabouze Mehdi

Octobre 2024

# Table des matières

<b>I</b>	<b>Introduction</b>	<b>4</b>
<b>II</b>	<b>Tache 1 : Couche physique ADS-B Prise en main de la chaine de communication ADSB</b>	<b>5</b>
1	Sous-tache 1.1	5
2	Sous-tache 1.2	7
3	Sous-tache 1.3	8
4	Sous-tache 1.4	9
5	Sous-tache 1.5	9
6	Sous-tache 1.6	10
7	Sous-tache 1.7	10
<b>III</b>	<b>Tache 2 : Couche physique ADS-B   Densité spectrale de puissance</b>	<b>12</b>
1	Sous-tache 2.1	12
2	Sous-tache 2.2	12
3	sous-tâche 2.3	13
4	sous-tâche 2.4	13
5	sous-tâche 2.5	14
6	sous-tâche 2.6	15
<b>IV</b>	<b>Tache 3 : Couche physique ADS-B Algorithmes de codage et de décodage de canal.</b>	<b>16</b>
<b>V</b>	<b>Tâche 4 : Couche physique ADS-B Synchronisation en temps.</b>	<b>17</b>
1	Sous-tâche 4.1	17
2	Sous-tache 4.2	17
3	Sous-tache 4.3	19
4	Sous-tache 4.4	19
5	Sous-tache 4.5	19
<b>VI</b>	<b>Tâche 5 : Couche physique ADS-B Synchronisation en fréquence.</b>	<b>21</b>

<b>VII</b>	<b>Tâche 6 : Implémentation de la couche MAC ADS-B</b>	<b>22</b>
1	Objectifs	22
2	Structure des trames ADS-B	22
2.1	Le préambule	22
2.2	Format de la voie descendante	22
2.3	L'adresse OACI de l'appareil	22
2.4	Les données ADS-B	23
3	Sous-tâche 6.1	23
4	Sous-tâche 6.2 : Développement de la fonction MATLAB bit2registre	23
5	Sous-tâche 6.3 : Validation de la fonction bit2registre	23
<b>VIII</b>	<b>Tâche 7 : Amélioration de la fonction bit2registre</b>	<b>25</b>
1	Objectifs	25
2	Sous-tâche 7.1 : Intégration de la position au sol	25
3	Sous-tâche 7.2 : Intégration de la vitesse de l'avion en vol	25
4	Sous-tâche 7.3 : Calcul de la position en vol sans hypothèse de proximité	27
<b>IX</b>	<b>Tâche 8 : Traitement des Signaux Réels</b>	<b>29</b>
1	Objectifs	29
2	Méthodologie proposée	29
<b>X</b>	<b>Conclusion Générale</b>	<b>30</b>
<b>XI</b>	<b>Références</b>	<b>30</b>

# I Introduction

Le projet TS229 vise à mettre en pratique des concepts clés en communication numérique, codage de canal, et traitement du signal en développant un émetteur/récepteur de données ADS-B (Automatic Dependent Surveillance Broadcast) simulé sous MATLAB. Ce système permettra de collecter et décoder en temps réel les informations diffusées par les avions, notamment leur position, altitude et trajectoire.

Ce rapport décrit les étapes de réalisation, les tâches assignées, et les prérequis pour atteindre les objectifs. Le projet se divise en trois types de tâches : des tâches orientées application, des tâches dédiées à la signalisation et aux communications numériques, et d'autres axées sur la couche d'accès. Certaines tâches, optionnelles, apportent des améliorations à la simulation.

Pour bien saisir la portée de ce projet, il est important de comprendre le rôle de l'ADS-B, qui permet aux aéronefs de diffuser leur position grâce à des systèmes de positionnement par satellite, comme le GPS. Ce système, complémentaire aux radars traditionnels, assure la transmission d'informations de vol aux tours de contrôle et aux autres aéronefs.

L'ADS-B présente un avantage majeur par son coût réduit, car le traitement principal du signal est effectué à bord des avions, les récepteurs au sol restant passifs. Le projet se concentre spécifiquement sur le mode 1090 Extended Squitter (1090 ES), dominant en Europe, avec une fréquence de 1090 MHz et des messages étendus à 112 bits pour une meilleure précision.

L'objectif final est de concevoir une application comparable à celles existantes, telles que flightradar24, pour suivre en temps réel la position et la trajectoire des aéronefs, conformément aux standards ADS-B.

## II Tache 1 : Couche physique ADS-B Prise en main de la chaine de communication ADSB

Dans cette tâche, la modulation PPM (Pulse Position Modulation) est explorée dans le contexte du standard ADS-B. La première étape a consisté à démontrer comment le signal émis peut être réécrit sous une forme particulière. Ensuite, les signaux correspondant à une séquence binaire donnée ont été représentés graphiquement. Les résultats théoriques ont été confirmés par une simulation MATLAB. La probabilité d'erreur binaire (Pb) a été calculée pour la modulation PPM. Enfin, la courbe du taux d'erreur binaire (TEB) a été simulée pour différentes valeurs du rapport signal à bruit ( $E_b/N_0$ ). À travers cette tâche, une meilleure compréhension de la modulation PPM et de ses performances en communication numérique a été acquise.

### 1 Sous-tache 1.1

#### Expression de $s_l(t)$

L'expression de  $s_l(t)$  est donnée comme la somme des impulsions associées à chaque bit  $b_k$ , chaque impulsion étant décalée dans le temps par un multiple de la période  $T_s$  :

$$s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s)$$

Où :

- $b_k$  représente le  $k$ -ème bit (0 ou 1).
- $p_{b_k}(t)$  est l'impulsion associée à  $b_k$ , avec :

$$p_{b_k}(t) = \begin{cases} p_0(t) & \text{si } b_k = 0, \\ p_1(t) & \text{si } b_k = 1. \end{cases}$$

Cela signifie que  $s_l(t)$  est la somme des impulsions  $p_0(t)$  (pour les bits 0) et  $p_1(t)$  (pour les bits 1), chacune étant décalée par un multiple de la période  $T_s$ .

Donc pour une séquence  $\mathbf{b}$  telle que :

$$\mathbf{b} = [1, 0, 0, 1, 0]$$

on obtient :

$$s_l(t) = p_1(t) + p_0(t - T_s) + p_0(t - 2T_s) + p_1(t - 3T_s) + p_0(t - 4T_s)$$

#### Expression de $r_l(t)$

Nous calculons la convolution  $r_l(t)$  de  $p_1^*(-t)$  avec  $s_l(t)$ , où

$$s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s)$$

Cela peut s'écrire comme :

$$r_l(t) = \int_{-\infty}^{\infty} p_1^*(\tau) s_l(t - \tau) d\tau$$

Or

$$p_1(t)^* = p_1(t) = 1 \quad \text{si } t \in [0, 0.5]$$

Donc :

$$r_l(t) = \int_0^{0.5} 1 \cdot s_l(t - \tau) d\tau$$

On pose le changement de variable suivant :

$$u = t - \tau$$

$$du = -d\tau$$

Soit :

$$r_l(t) = \int_{t-0.5}^t s_l(u) du$$

### Expression de $r_m$

Considérons un signal continu  $r_l(t)$ . Le signal discrétisé  $r_m(t)$  peut être obtenu en échantillonnant  $r_l(t)$  à intervalles de  $\frac{T_s}{2}$ . On peut alors écrire :

$$r_m[n] = r_l\left(\frac{nT_s}{2}\right)$$

Le signal discrétisé  $r_m(t)$  peut être exprimé comme la convolution de  $r_l(t)$  avec un peigne de Dirac :

$$r_m[n] = \frac{2}{T_s} \sum_{k=-\infty}^{\infty} r_l\left(t - k\frac{T_s}{2}\right)$$

Soit :

$$r_m(t) = \frac{2}{T_s} \sum_{k=-\infty}^{\infty} \int_{t-k\frac{T_s}{2}-0.5}^{t-k\frac{T_s}{2}} s_l(u) du$$

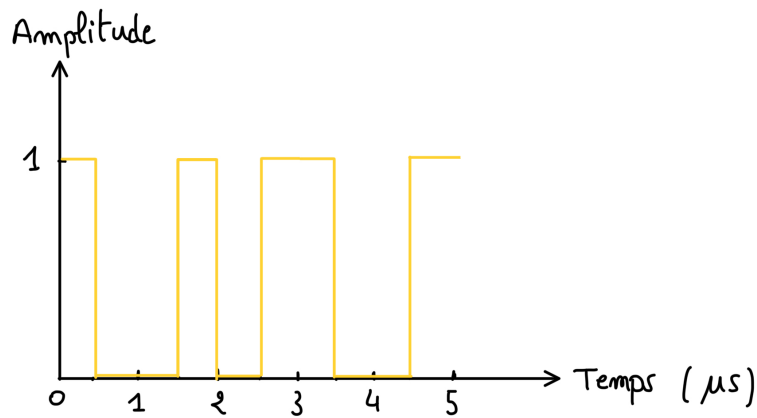


FIGURE 1 – Le signal  $s_l(t)$

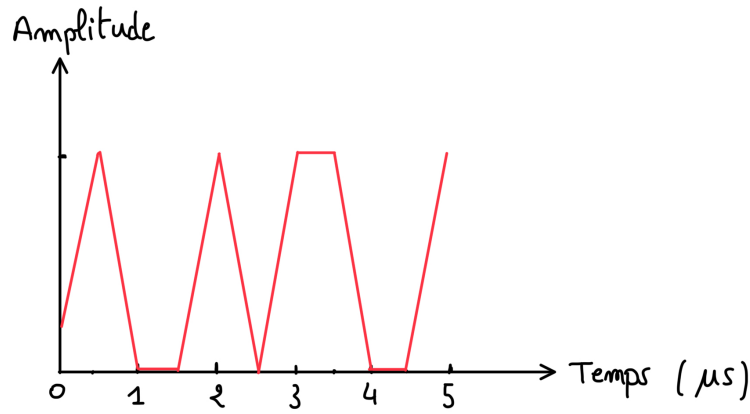


FIGURE 2 – Le signal  $r_l(t)$

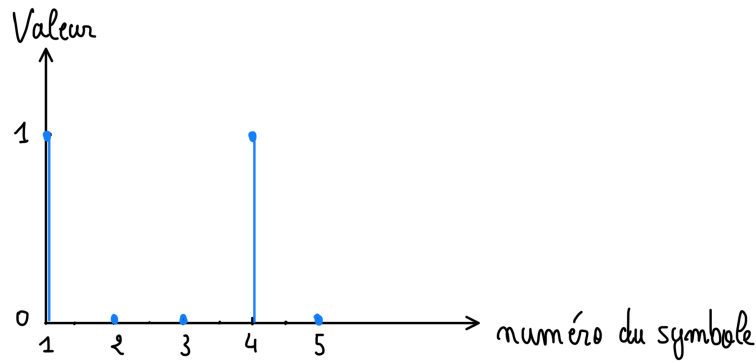


FIGURE 3 – Le signal  $r_m$

Le rôle du bloc décision est d'extraire les informations binaires à partir des observations analogiques du signal reçu en prenant une décision binaire basée sur les échantillons du signal filtré.

## 2 Sous-tache 1.2

Nous commençons avec l'expression suivante :

$$s_l(t) = \sum_{k \in \mathbb{Z}} P_{b_k}(t - kT_s)$$

Nous cherchons à la réécrire sous la forme :

$$s_l(t) = \sum_k u_k \cdot p_1 \left( t - k \frac{T_s}{2} \right)$$

Ensuite, nous décomposons la somme en deux composants représentant deux parties différentes de la séquence de pulsation. Cette expression devient alors :

$$s_l(t) = \sum_{k \in \mathbb{Z}} u_{2k} \cdot p_1(t - 2kT_s) + \sum_{k \in \mathbb{Z}} u_{2k+1} \cdot p_1 \left( t - (2k+1) \frac{T_s}{2} \right)$$

Les valeurs  $u_{2k}$  et  $u_{2k+1}$  correspondent aux symboles binaires  $b_k$ .

- Par exemple, si  $b_k = 0$ , nous pouvons fixer  $u_{2k} = 0$  et  $u_{2k+1} = 1$ .

- Inversement, si  $b_k = 1$ , nous fixons  $u_{2k} = 1$  et  $u_{2k+1} = 0$ .

C'est une forme de modulation binaire, où nous représentons une valeur binaire comme une paire de symboles ou de valeurs qui modulent le signal.

La relation finale peut donc être exprimée comme suit :

$$b_k = \begin{cases} 0 & \text{si } (u_{2k}, u_{2k+1}) = (0, 1) \\ 1 & \text{si } (u_{2k}, u_{2k+1}) = (1, 0) \end{cases}$$

Ainsi,  $b_k$  est directement déterminé par les valeurs de  $u_{2k}$  et  $u_{2k+1}$ .

Cette relation est utilisée pour décoder les symboles binaires à partir des composants du signal  $u_k$ .

### 3 Sous-tache 1.3

#### Expression de $r_k = [r_{2k}, r_{2k+1}]$

Le vecteur  $r_k = [r_{2k}, r_{2k+1}]$  contient deux échantillons consécutifs du signal reçu. Ce vecteur est influencé par le bruit additif gaussien et par les impulsions  $p_1(t)$  associées aux bits émis  $b_k$ .

L'expression générale de ces échantillons est :

$$\begin{cases} r_{2k} = 0 + n_{2k} \\ r_{2k+1} = v_0 + n_{2k+1} = \int_{-\infty}^{\infty} |p_1(t)|^2 dt + n_{2k+1} \end{cases}$$

où  $n_{2k}$  et  $n_{2k+1}$  représentent les bruits blancs gaussiens.

#### Test de maximum de vraisemblance

Le test de maximum de vraisemblance est une méthode statistique permettant de déterminer la séquence binaire la plus probable en fonction du signal reçu, tout en tenant compte du bruit. Le critère de décision est de minimiser la distance entre le vecteur reçu  $r_k = [r_{2k}, r_{2k+1}]$  et deux hypothèses possibles concernant les valeurs des symboles  $b_k$  :

- L'hypothèse  $H_0 : b_k = 0$ , ce qui correspond à  $v_0[0, 1]$ , c'est-à-dire une impulsion  $p_0(t)$  suivie d'une impulsion  $p_1(t)$ .
- L'hypothèse  $H_1 : b_k = 1$ , ce qui correspond à  $v_0[1, 0]$ , c'est-à-dire une impulsion  $p_1(t)$  suivie d'une impulsion  $p_0(t)$ .

Le test de vraisemblance consiste alors à comparer les distances euclidiennes entre le vecteur  $r_k$  et ces deux hypothèses :

$$\|r_k - v_0[0, 1]\|_2^2 \leq \|r_k - v_0[1, 0]\|_2^2$$

Pour chaque hypothèse, calculons les distances :

-Hypothèse  $H_0 : b_k = 0$

$$\begin{aligned} \|r_k - v_0[0, 1]\|_2 &= \sqrt{(r_{2k} - 0)^2 + (r_{2k+1} - v_0)^2} \\ &= \sqrt{n_{2k}^2 + (v_0 + n_{2k+1} - v_0)^2} \\ &= \sqrt{n_{2k}^2 + n_{2k+1}^2} \end{aligned}$$

-Hypothèse  $H_1 : b_k = 1$

$$\begin{aligned} \|r_k - v_0[1, 0]\|_2 &= \sqrt{(r_{2k} - v_0)^2 + (r_{2k+1} - 0)^2} \\ &= \sqrt{(n_{2k} - v_0)^2 + (v_0 + n_{2k+1})^2} \\ &= \sqrt{n_{2k}^2 - 2n_{2k}v_0 + v_0^2 + v_0^2 + 2v_0n_{2k+1} + n_{2k+1}^2} \\ &= \sqrt{n_{2k}^2 + n_{2k+1}^2 + 2v_0^2 - 2n_{2k}v_0 + 2n_{2k+1}v_0} \end{aligned}$$

$$\text{-Si } v_0^2 - n_{2k}v_0 + n_{2k+1}v_0 > 0 \text{ alors : } \|r_k - v_0[0, 1]\|_2^2 < \|r_k - v_0[1, 0]\|_2^2$$



## Critère de décision

Pour prendre la décision, on compare les deux distances :

$$\begin{array}{ll} r_k = (0, 1) & \text{si } \|r_k - v_0[0, 1]\|_2^2 < \|r_k - v_0[1, 0]\|_2^2 \\ r_k = (1, 0) & \text{sinon.} \end{array}$$

## Interprétation

Ce test revient à déterminer si le signal reçu est plus proche du modèle correspondant à  $b_k = 0$  (hypothèse  $H_0$ ) ou à  $b_k = 1$  (hypothèse  $H_1$ ). Si la distance entre  $r_k$  et  $v_0[0, 1]$  est plus petite, on décide que  $b_k = 0$ , sinon, on décide que  $b_k = 1$ .

Ce test prend en compte la probabilité du bruit additif gaussien, de sorte que la décision minimise la probabilité d'erreur. Ce qui signifie que la décision est basée sur la comparaison entre les contributions des échantillons  $r_{2k}$  et  $r_{2k+1}$  par rapport à l'énergie  $v_0$  de l'impulsion  $p_1(t)$ .

### 4 Sous-tache 1.4

Le code Matlab de cette tâche a été déposé sur Thor.<sup>1</sup>

### 5 Sous-tache 1.5

Les trois figures représentent les différents graphes respectivement de  $s_l(t)$ ,  $r_l(t)$  et  $r_m(t)$  pour un signal d'entrée  $[1\ 0\ 0\ 1\ 0]$  et un bruit nul.

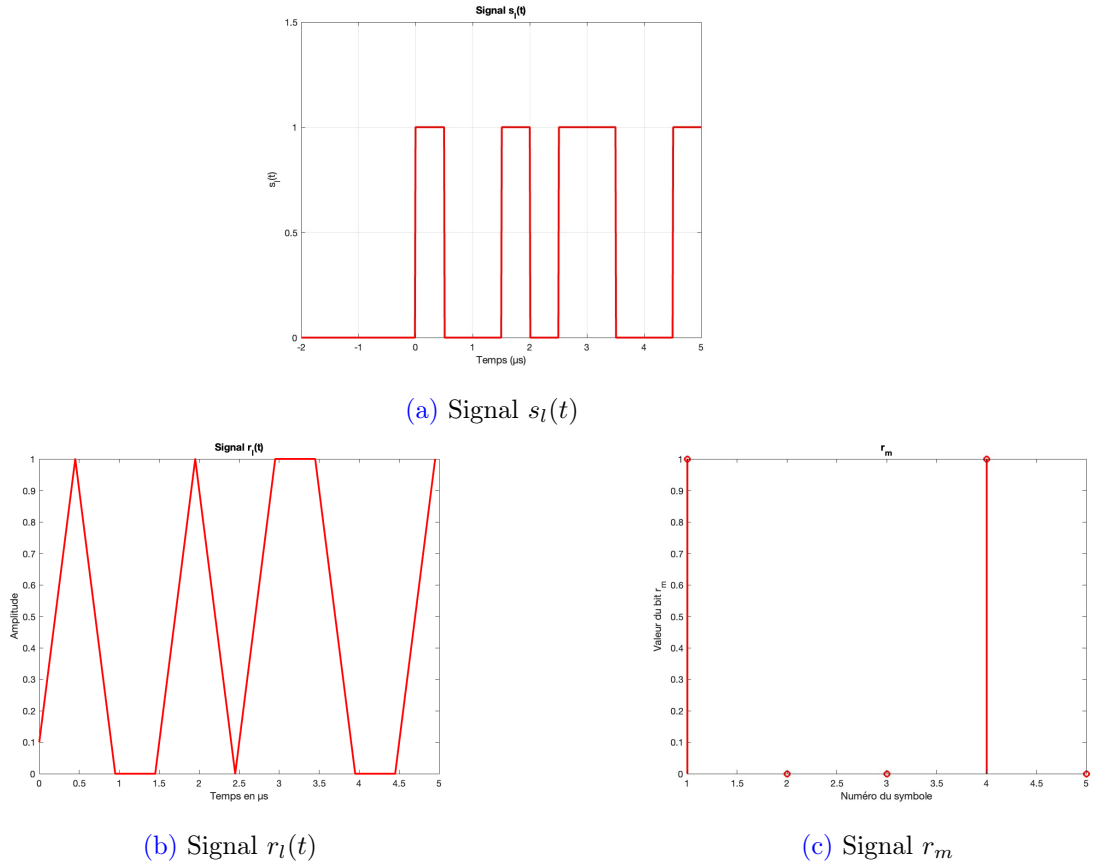


FIGURE 4 – Signaux théoriques  $s_l(t)$ ,  $r_l(t)$  et  $r_m(t)$  obtenus pour une entrée  $[1\ 0\ 0\ 1\ 0]$  et sans bruit.

La Figure 1a (signal  $s_l(t)$ ) présente un signal rectangulaire, ce qui est prévisible car  $s_l(t)$  est formé par la combinaison linéaire des signaux  $p_0(t)$  et  $p_1(t)$ , qui sont tous deux rectangulaires. Par exemple,

1. Fichier : Tache\_1.m

pour la séquence binaire [1 0 0 1 0], on peut l'exprimer comme suit :

$$s_l(t) = p_1(t) + p_0(t - T_s) + p_0(t - 2T_s) + p_1(t - 3T_s) + p_0(t - 4T_s).$$

Pour la Figure 1b (signal  $r_l(t)$ ), sa forme triangulaire est conforme à nos attentes, car  $r_l(t)$  résulte de la convolution du signal  $s_l(t)$  (lui-même étant une combinaison linéaire de fonctions porte) avec le filtre de mise en forme  $p(t)$  (lui aussi composé de deux fonctions porte). La convolution de deux fonctions porte donne une forme triangulaire, ce qui explique la présence d'une forme de scie dans le signal  $r_l(t)$ .

Enfin, la Figure 1c (signal  $r_m$ ) suggère que le bloc de décision permettra de retrouver les bits initialement transmis en début de la chaîne de communication. En effet, les valeurs de  $r_m$  à chaque intervalle  $kT_e/2$  correspondent aux bits d'entrée, où une valeur nulle correspond à un bit 0, et une valeur de 1 à un bit 1.

## 6 Sous-tache 1.6

Le code Matlab de cette tâche a été déposé sur Thor.<sup>2</sup>

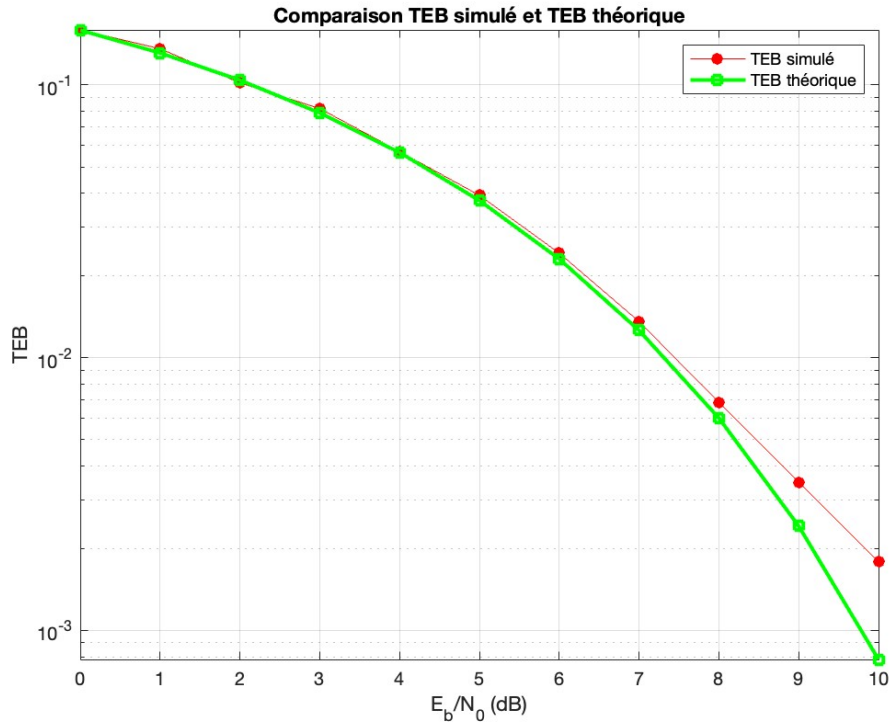


FIGURE 5 – TEB obtenu théoriquement et expérimentalement.

La figure 5 illustre les Taux d'Erreur Binaire (TEB) obtenus théoriquement et expérimentalement avec un vecteur d'entrée de 1000 bits générés uniformément. On remarque bien que les deux courbes se superposent quasiment. On en déduit que notre simulation est précise, ce qui est en accord avec nos attentes.

## 7 Sous-tache 1.7

On a l'expression de  $r_m$  :

$$r_m[n] = |u_n|.v[0] + n'_l[n]$$

Rq : Le processus aléatoire  $n'_l[n]$  conserve le caractère blanc du processus aléatoire source ( $n_l[n]$ ) car le filtre adapté  $g_a[n]$  est un filtre de Nyquist à  $F_{se}$ .

---

2. Fichier : Tache\_1.m

À partir de l'expression de  $r_m$  on va évaluer la probabilité d'erreur binaire. La probabilité d'erreur symbole notée  $P_s$  est définie ainsi :

$$P_s = \sum_{j \in \Omega_s} \mathbb{P}(\text{erreur} \mid j)$$

$$P_s = \sum_{j \in \Omega_s} \mathbb{P}(\text{erreur} \mid j) \cdot \mathbb{P}(j); \text{ avec : } \mathbb{P}(j) = \frac{1}{M}$$

$$P_s = \frac{1}{M} \sum_{j \in \Omega_s} \mathbb{P}(\text{erreur} \mid j).$$

On fait une erreur si le démodulateur numérique décide que le symbole le plus probable n'est pas  $j$ . Donc

$$\mathbb{P}(\text{erreur} \mid j) = \mathbb{P}(\text{décider } i \neq j \mid j)$$

Cette probabilité dépend de l'ensemble  $\Omega_s$  et donc de  $M = 2^{n_s}$  et de la famille de modulation numérique choisie. Nous allons donc évaluer ces probabilités d'erreur sur l'exemple de  $\Omega_s$  suivant :

$$\Omega_s \in \{\pm A\}$$

Donc pour une 2-PAM :

$$P_s = \frac{1}{M} \sum_{j \in \Omega_s} \mathbb{P}(\text{erreur} \mid j).$$

$$P_s = \frac{1}{2}(\mathbb{P}(\text{erreur} \mid -A) + \mathbb{P}(\text{erreur} \mid +A))$$

$$P_s = \frac{1}{2}(Q\left(\frac{Av[0]}{\sigma_{n_l}}\right) + Q\left(\frac{Av[0]}{\sigma_{n_l}}\right))$$

Donc :

$$P_s = Q\left(\frac{Av[0]}{\sigma_{n_l}}\right) = P_b$$

Lorsqu'on représente l'évolution de  $P_b$  en fonction du SNR, on utilise généralement le  $SNR = \frac{E_b}{N_0}$  avec :  $E_b = P_{moy}T_b = A^2E_g$ .

on a :

$$\frac{E_b}{N_0} = \frac{A^2E_g^2}{2\sigma_{n_l}^2}$$

Donc :

$$P_b = Q\left(\frac{Av[0]}{\sigma_{n_l}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

et comme :  $Q(x) = \frac{1}{2}erfc(\sqrt{\frac{E_b}{N_0}})$

Alors :

$$P_b = \frac{1}{2}erfc(\sqrt{\frac{E_b}{N_0}})$$

### III Tache 2 : Couche physique ADS-B | Densité spectrale de puissance

Dans cette tâche, l'objectif était de calculer la densité spectrale de puissance d'un signal, en utilisant à la fois une approche théorique et une implémentation Matlab. Les sous-tâches comprenaient le calcul du moment d'ordre 1 du signal, la détermination de la fonction d'autocorrélation, le calcul de la densité spectrale de puissance, et enfin, la mise en œuvre de l'algorithme du périodogramme de Welch en utilisant Matlab pour obtenir une estimation expérimentale de la DSP.

#### 1 Sous-tache 2.1

D'après l'expression (1) du sujet, on a :  $s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s)$  où  $p_{b_k}$  est défini comme suit :

$$p_{b_k} = \begin{cases} p_0(t), & \text{si } b_k = 0 \\ p_1(t), & \text{si } b_k = 1 \end{cases}$$

Montrons que  $s_l(t)$  peut s'écrire de la forme  $s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)$ .

Par définition de  $p(t)$ ,  $p_0(t)$  et  $p_1(t)$  peuvent s'écrire de la forme :

$$p_0(t) = 0.5 \Pi_{T_s}(t - \frac{T_s}{2}) + p(t) \text{ et } p_1(t) = 0.5 \Pi_{T_s}(t - \frac{T_s}{2}) - p(t),$$

la porte de largeur  $T_s$  vient du fait que  $p(t)$  est définie sur  $1\mu s$ .

En introduisant  $A_k$  défini comme suit :

$$A_k = \begin{cases} 1, & \text{si } b_k = 0 \\ -1, & \text{si } b_k = 1 \end{cases}$$

On trouve :  $p_{b_k}(t) = 0.5 \Pi_{T_s}(t - \frac{T_s}{2}) + A_k p(t)$

En sommant sur  $k \in \mathbb{Z}$ , il arrive :

$$\begin{aligned} s_l(t) &= \sum_{k \in \mathbb{Z}} (0.5 \Pi_{T_s}(t - kT_s - \frac{T_s}{2}) + A_k p(t - kT_s)) \\ s_l(t) &= 0.5 \underbrace{\sum_{k \in \mathbb{Z}} \Pi_{T_s}(t - kT_s - \frac{T_s}{2})}_{=1} + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) \end{aligned}$$

$$\text{d'où, } \boxed{s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)} \quad (1)$$

#### 2 Sous-tache 2.2

Calculons le moment d'ordre 1 de  $s_l$ ,  $m_{s_l}(t)$ , et montrons qu'il ne dépend pas du temps.

D'après l'équation 1,  $s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)$ ,

d'où, par définition de  $m_{s_l}(t)$  :

$$\begin{aligned} m_{s_l}(t) &= E[s_l(t)] = E \left[ 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) \right] \\ &= 0.5 + E \left[ \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) \right] \\ &= 0.5 + \sum_{k \in \mathbb{Z}} E[A_k] p(t - kT_s) \end{aligned}$$

$$\text{D'où, } m_{s_l}(t) = 0.5 + \sum_{k \in \mathbb{Z}} p(t - kT_s) E[A_k] \quad (2)$$

Or,  $E[A_k] = 0$ , d'où  $m_{s_l}(t) = 0.5$ .

Ainsi, le moment d'ordre 1 de  $s_l$  ne dépend pas de  $t$ .

Et,  $m_{s_l}(t) = 0.5$

### 3 sous-tâche 2.3

Calculons la fonction d'autocorrélation de  $s_l(t)$  définie par  $R_{s_l}(t, \tau) = E[s_l(t)s_l^*(t + \tau)]$ .

On a :

$$\begin{aligned} R_{s_l}(t, \tau) &= E[s_l(t)s_l^*(t + \tau)] & s_l^*(t + \tau) &= s_l(t + \tau) \text{ car } s_l(t) \in \mathbb{R} \\ &= 0.25 + \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} E[A_i A_j] p(t - iT_s) p(t - jT_s + \tau) \\ &= 0.25 + \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} R_A(i - j) p(t - iT_s) p(t - jT_s + \tau) \quad \text{en posant } E[A_i A_j] = R_A(i - j) \end{aligned}$$

d'où ,  $R_{s_l}(t, \tau) = 0.25 + \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} R_A(i - j) p(t - iT_s) p(t - jT_s + \tau)$  (3)

### 4 sous-tâche 2.4

Calculons  $\tilde{R}_{s_l}(\tau)$ . On a par définition :

$$\begin{aligned} \tilde{R}_{s_l}(\tau) &= \frac{1}{T_s} \int_0^{T_s} R_{s_l}(t, \tau) dt \\ &= \frac{1}{T_s} \int_0^{T_s} \left( 0.25 + \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} R_A(i - j) p(t - iT_s) p(t - jT_s + \tau) \right) dt \\ &= 0.25 + \frac{1}{T_s} \int_0^{T_s} \sum_{i \in \mathbb{Z}} \sum_{d \in \mathbb{Z}} R_A(d) p(t - iT_s) p(t - iT_s + dT_s + \tau) dt \quad \text{en posant } d \leftarrow i - j \\ &= 0.25 + \frac{1}{T_s} \sum_{i \in \mathbb{Z}} \sum_{d \in \mathbb{Z}} R_A(d) \int_0^{T_s} p(t - iT_s) p(t - iT_s + dT_s + \tau) dt \\ &= 0.25 + \frac{1}{T_s} \sum_{d \in \mathbb{Z}} R_A(d) \sum_{i \in \mathbb{Z}} \int_{-(i-1)T_s}^{-iT_s} p(u) p(u + dT_s + \tau) du \quad \text{en posant } u \leftarrow t - iT_s \\ &= 0.25 + \frac{1}{T_s} \sum_{d \in \mathbb{Z}} R_A(d) \int_{-\infty}^{+\infty} p(u) p(u + dT_s + \tau) du \\ &= 0.25 + \frac{1}{T_s} \sum_{d \in \mathbb{Z}} R_A(d) R_p(-\tau - dT_s) \quad \text{où } R_p \text{ est la fonction d'autocorrélation du filtre } p(t). \end{aligned}$$

d'où,  $\tilde{R}_{s_l}(\tau) = 0.25 + \frac{\sigma^2}{T_s} R_p(\tau)$  avec  $\sigma^2$  la variance de  $A$  (4)

## 5 sous-tâche 2.5

Calculons  $\Gamma_{sl}(f)$ .

On a :

$$\begin{aligned}\Gamma_{sl}(f) &= \mathcal{F}(\tilde{R}_{sl}(\tau)) \\ &= \int_{-\infty}^{\infty} \tilde{R}_{sl}(\tau) e^{-j2\pi f t} dt \\ &= \int_{-\infty}^{\infty} \left( 0.25 + \frac{\sigma^2}{T_s} R_p(\tau) \right) e^{-j2\pi f t} dt\end{aligned}$$

$$\text{D'où, } \quad \underline{\Gamma_{sl}(f) = \frac{1}{4}\delta(f) + \frac{\sigma^2}{T_s}|\Gamma_p(f)|^2} \quad (5)$$

Maintenant calculons  $\Gamma_p(f)$ . Sachant que la transformée de Fourier de la fonction porte  $\Pi_\theta(t)$  est donnée par  $TF(\Pi_\theta(t)) = \theta \text{sinc}(f\theta)$ , on a :

$$\begin{aligned}\Gamma_p(f) &= -\frac{1}{4}T_s \text{sinc}\left(\frac{fT_s}{2}\right) e^{-j2\pi f T_s \frac{1}{4}} + \frac{1}{4}T_s \text{sinc}\left(\frac{fT_s}{2}\right) e^{-j2\pi f T_s \frac{3}{4}} \\ &= \frac{1}{4}T_s \text{sinc}\left(\frac{fT_s}{2}\right) \left( -e^{-j\pi f T_s \frac{1}{2}} - e^{j\pi f T_s \frac{1}{2}} \right) \\ &= \frac{1}{4}T_s \text{sinc}\left(\frac{fT_s}{2}\right) e^{(-j\pi f T_s)} \left( -e^{j\pi f T_s \frac{1}{2}} + e^{j\pi f T_s \frac{3}{2}} \right) \\ \Gamma_p(f) &= -\frac{j}{4}T_s^2 \pi f \text{sinc}^2\left(\frac{fT_s}{2}\right) e^{(-j\pi f T_s)}\end{aligned}$$

$$\text{d'où, } \quad \underline{|\Gamma_p(f)|^2 = \frac{1}{16}T_s^4 \pi^2 f^2 \text{sinc}^4\left(\frac{fT_s}{2}\right)} \quad (6)$$

Ainsi, on a :

$$\boxed{\Gamma_{sl}(f) = \frac{\delta(f)}{4} + \frac{1}{16}T_s^3 \pi^2 f^2 \text{sinc}^4\left(\frac{fT_s}{2}\right)} \quad (7)$$

## 6 sous-tâche 2.6

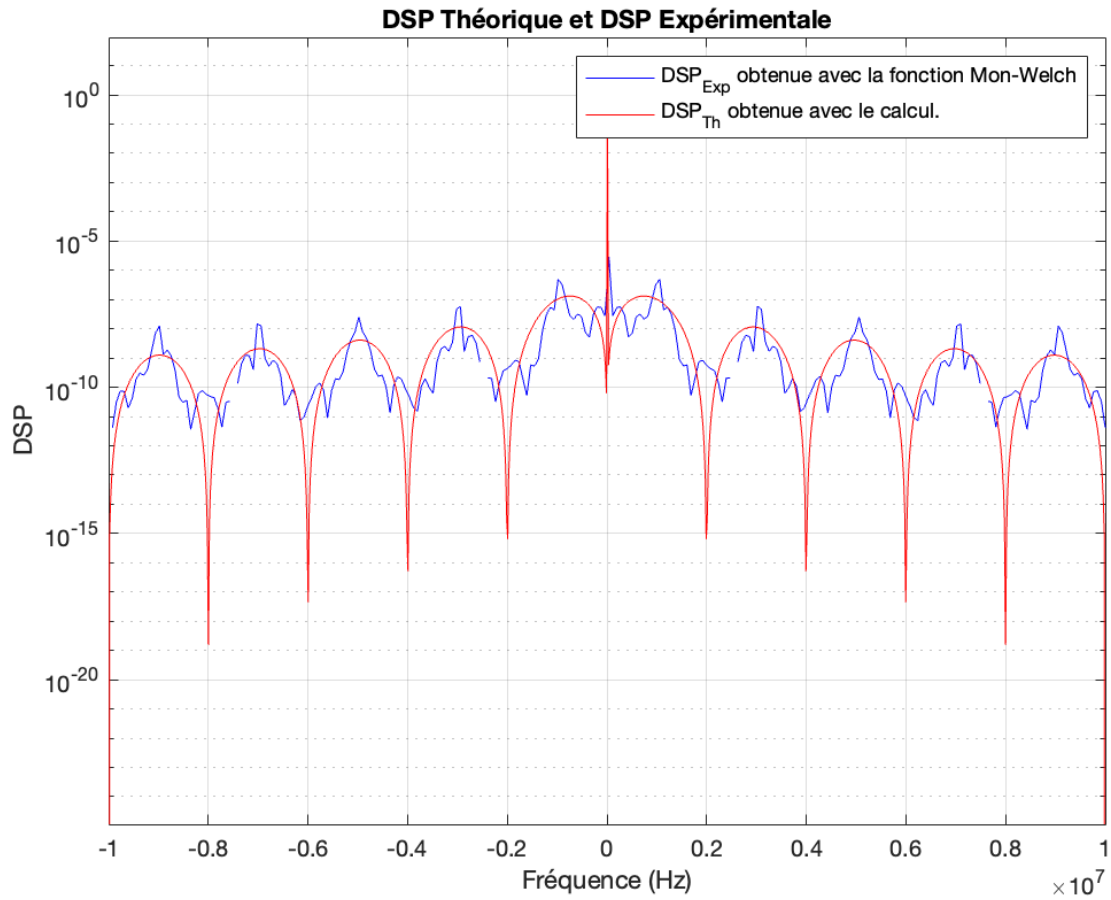


FIGURE 6 – DSP obtenue de différentes manières

Sur la [figure 6](#), on remarque que les 2 courbes de la densité spectrale de puissance calculée de 2 manières différentes se superposent. Par conséquent, je peux conclure que l'implémentation de l'algorithme de Welch, réalisée dans la fonction `Mon_welch`, est correcte et produit des résultats cohérents.<sup>3</sup>

---

3. Fichiers : `Mon_welch.m` et `Tache_2.m`

## IV Tache 3 : Couche physique ADS-B Algorithmes de codage et de décodage de canal.

L'objectif de cette section est d'implémenter le codeur et le décodeur de canal utilisés pour la transmission des signaux ADS-B. La structure des signaux ADS-B émis par les avions est illustrée à la figure 7. Il n'est pas nécessaire de comprendre précisément la signification des différentes parties de la trame ADS-B à ce stade. Ce travail sera abordé ultérieurement lors de l'étude de la couche MAC. Ici, nous nous concentrons sur l'observation qu'une trame ADS-B dure 120  $\mu$ s, ce qui correspond à 120 bits, et qu'elle inclut un contrôle de parité de 24 bits à la fin (voir Fig. 7).

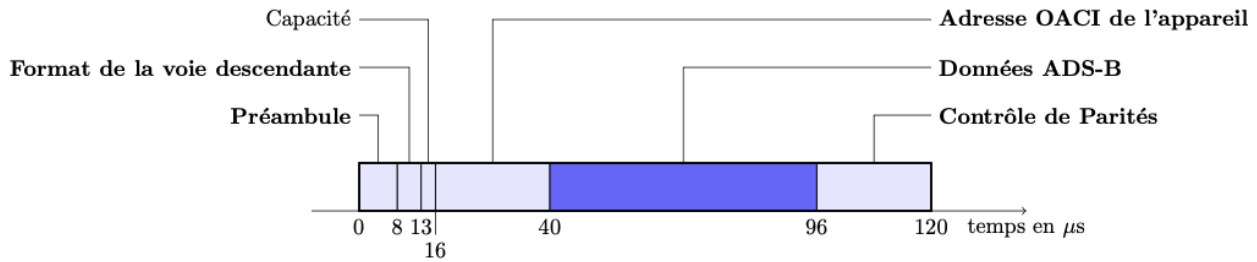


FIGURE 7 – DSP obtenue de différentes manières

Dans le cadre de ce projet, on a implémenté une chaîne de communication ADS-B intégrant le codage et le décodage de canal. Le codage de canal repose sur un Code à Redondance Cyclique (CRC) de 24 bits, dont le polynôme générateur est  $p(x) = x^{24} + x^{23} + \dots + x^{10} + x^3 + 1$ . Ce CRC permet d'ajouter une redondance au message composé de 88 bits d'informations utiles, afin de détecter les erreurs à la réception. Le message total transmis est donc de 112 bits : 88 bits d'information suivis de 24 bits de parité générés par le codeur CRC.

Pour mettre en œuvre cette chaîne de communication, on génère d'abord un paquet de données binaires aléatoires de taille fixe,  $N_b = 88$  bits. Ensuite, on ajoute les 24 bits de CRC pour former un message complet de 112 bits, qui est ensuite transmis via le canal. À la réception, on utilise un décodeur CRC pour vérifier l'intégrité du message. Les fonctions MATLAB 'crc.detector' et 'detect' permettent respectivement de configurer le détecteur CRC à partir du polynôme générateur, et de détecter la présence d'erreurs dans les données reçues.

Pour tester cette chaîne, on a vérifié deux situations. Dans le premier cas, sans erreur dans les 88 bits utiles, le décodeur CRC confirme que le message est intact. Dans le second cas, des erreurs sont introduites parmi les 88 bits d'information, et le décodeur signale correctement que le message contient des erreurs, prouvant ainsi le bon fonctionnement du système.

Le code Matlab de cette tâche a été déposé sur Thor.<sup>4</sup>

---

4. Fichiers : Codage\_CRC.m Decodage\_CRC.m Tache\_3.m



## V Tâche 4 : Couche physique ADS-B Synchronisation en temps.

Dans cette partie, nous abordons deux problèmes majeurs qui n'ont pas été pris en compte dans les tâches précédentes : la *synchronisation temporelle* et la *synchronisation fréquentielle*. Le signal ADS-B reçu subit un délai de propagation  $\delta_t$  et un décalage en fréquence  $\delta_f$ , dus à l'effet Doppler et aux imperfections des oscillateurs. Ces effets doivent être compensés pour une bonne réception des signaux. Le modèle en bande de base représentant le signal reçu est donné par l'équation suivante :

$$y_l(t) = \alpha s_l(t - \delta_t) e^{-j2\pi\delta_f t + j\phi_0} + n_l(t)$$

où  $\alpha$  représente l'atténuation du signal,  $\delta_t$  et  $\delta_f$  les désynchronisations temporelle et fréquentielle, et  $\phi_0$  un déphasage initial. Pour effectuer la synchronisation, on utilise un préambule  $s_p(t)$  de durée  $T_p = 8 \mu s$  inséré en tête de chaque trame ADS-B. Ce préambule est unique et permet de réaliser une *intercorrélation* pour estimer le délai et le décalage fréquentiel.

### 1 Sous-tâche 4.1

Les formules du décalage Doppler sont les suivantes :

— Pour une source se rapprochant :  $f_r = f_e \frac{V_{\text{onde}}}{V_{\text{onde}} - V_e}$ .

— Pour une source s'éloignant :  $f_r = f_e \frac{V_{\text{onde}}}{V_{\text{onde}} + V_e}$ .

On en déduit que la variation de fréquence  $\Delta f$  est donnée par :  $\Delta f = f_e \frac{V_e}{V_{\text{onde}}}$ .

En appliquant les valeurs numériques, on obtient :  $\Delta f = 1090 \times 10^6 \frac{900}{3.6 \times 10^8} \approx 908.33 \text{ Hz}$ .

En termes d'ordre de grandeur, cela correspond à :  $\Delta f = 10^3 \text{ Hz}$ .

Ainsi, l'ordre de grandeur du décalage Doppler pour un avion se déplaçant à 900 km/h est de l'ordre du **kHz**

### 2 Sous-tâche 4.2

Soit  $\rho(\delta'_t)$  la corrélation suivante :

$$\rho(\delta'_t) = \frac{\int_{\delta'_t}^{\delta'_t + T_p} y_l(t) s_p^*(t - \delta'_t) dt}{\sqrt{\int_0^{T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}}$$

La propriété  $|\rho(\delta'_t)| \leq 1$  est cruciale pour évaluer la qualité de la synchronisation dans les systèmes de communication. En effet, la corrélation  $\rho(\delta'_t)$  mesure l'alignement entre le signal reçu  $y_l(t)$  et le préambule connu  $s_p(t)$ . Si  $|\rho(\delta'_t)|$  est proche de 1, cela signifie que le signal est bien synchronisé, tandis qu'une valeur plus faible indique une désynchronisation. La borne  $|\rho(\delta'_t)| \leq 1$  normalise la fonction de corrélation, facilitant ainsi son interprétation : une valeur de 1 correspond à une synchronisation parfaite, tandis que des valeurs inférieures signalent un décalage.

Cette propriété aide également dans la prise de décision pour estimer le décalage temporel. En maximisant  $|\rho(\delta'_t)|$ , on trouve l'instant  $\delta'_t$  qui correspond à la meilleure synchronisation entre le signal reçu et le préambule. De plus, une valeur faible de  $|\rho(\delta'_t)|$  peut indiquer non seulement un problème de synchronisation, mais aussi des erreurs de transmission ou des interférences, offrant ainsi un critère pour juger de la qualité du signal global reçu.

Pour que l'estimation soit valide et interprétable, cette corrélation doit être normalisée. La normalisation s'assure que la valeur maximale possible est 1 (correspondant à une parfaite similitude entre le signal reçu  $y_l(t)$  et le préambule  $s_p(t)$ ). Si la corrélation dépassait 1, cela indiquerait un signal amplifié ou une mauvaise normalisation, ce qui ne correspond pas à un scénario physique réaliste.

Montrons que  $|\rho(\delta'_t)| < 1 \quad \forall \delta'_t \in \mathbb{R}$ .

On a :

$$\begin{aligned}
\rho(\delta'_t) &= \frac{\int_{\delta'_t}^{\delta'_t+T_p} y_l(t) s_p^*(t - \delta'_t) dt}{\sqrt{\int_0^{T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt}} \\
|\rho(\delta'_t)|^2 &= \frac{|\int_{\delta'_t}^{\delta'_t+T_p} y_l(t) s_p^*(t - \delta'_t) dt|^2}{\int_0^{T_p} |s_p(t)|^2 dt \cdot \int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt} \\
&\leq \frac{\int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt \cdot \int_{\delta'_t}^{\delta'_t+T_p} |s_p^*(t - \delta'_t)|^2 dt}{\int_0^{T_p} |s_p(t)|^2 dt \cdot \int_{\delta'_t}^{\delta'_t+T_p} |y_l(t)|^2 dt} && \text{d'après l'inégalité de Cauchy-Schwarz dans } \mathcal{L}^2 \\
&\leq \frac{\int_{\delta'_t}^{\delta'_t+T_p} |s_p^*(t - \delta'_t)|^2 dt}{\int_0^{T_p} |s_p(t)|^2 dt} \\
&\leq \frac{\int_0^{T_p} |s_p^*(v)|^2 dv}{\int_0^{T_p} |s_p(t)|^2 dt} && \text{en effectuant le changement de variable } v \leftarrow t - \delta'_t \text{ au numérateur}
\end{aligned}$$

donc  $|\rho(\delta'_t)|^2 \leq 1$

Finalement, on a  $\boxed{|\rho(\delta'_t)| < 1, \quad \forall \delta'_t \in \mathbb{R}}$

L'égalité dans l'inégalité de Cauchy-Schwarz est atteinte si et seulement si les fonctions  $y_l(t)$  et  $s_p^*(t - \delta'_t)$  sont proportionnelles, c'est-à-dire s'il existe une constante  $\lambda$  telle que :

$$y_l(t) = \lambda s_p^*(t - \delta'_t), \quad \text{pour tout } t \in [\delta'_t, \delta'_t + T_p].$$

Dans ce cas, on a  $|\rho(\delta'_t)| = 1$ .

### **Bon fonctionnement du récepteur en cas d'égalité :**

Supposons qu'on ait  $|\rho(\delta'_t)| = 1$ .

Alors, il existe une constante  $\lambda$  telle que :

$$y_l(t) = \lambda s_p^*(t - \delta'_t), \quad \text{pour tout } t \in [\delta'_t, \delta'_t + T_p].$$

Ainsi,  $y_l(t)$  est une version décalée (et éventuellement amplifiée) de  $s_p^*(t)$ . En d'autres termes, le récepteur reçoit un signal qui est parfaitement aligné avec le signal attendu, ce qui se traduit par une corrélation maximale.

### 3 Sous-tache 4.3

Reprenons le signal reçu donné par :

$$y_l(t) = \alpha s_p(t - \delta_t) e^{-j2\pi\delta_f t + j\phi_0} + n_l(t), \quad \text{pour } t \in [\delta_t, \delta_t + T_p]$$

où :

- $\alpha$  est un facteur d'amplification,
- $\delta_f$  est le décalage fréquentiel,
- $\phi_0$  est une phase initiale,
- $n_l(t)$  est le bruit.

Nous voulons montrer que la métrique proposée est invariante par rapport à  $\alpha$ .

En effet, lors de l'application de l'inégalité de Cauchy-Schwarz (voir sous-tache 4.2), on obtient une simplification de  $\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt$ . Ainsi, même si  $\alpha$  varie, on aura toujours  $|\rho(\delta'_t)| < 1, \forall \delta'_t \in \mathbb{R}$ .

La métrique est donc invariante lorsque  $\alpha$  varie.

### 4 Sous-tache 4.4

On souhaite l'étape de décision insensible à  $\alpha$  le facteur d'atténuation et à  $\phi_0$  la phase initiale.

On a les Hypothèses :

- $\delta_f = 0$ , donc il n'y a pas de décalage de fréquence.
- $\delta_t$  est correctement estimé, donc le signal reçu est bien synchronisé en temps.

#### Insensibilité à $\alpha$ :

Pour rendre l'étape de décision insensible à  $\alpha$ , il faut normaliser par  $\alpha$ , qui est un facteur d'atténuation du signal. Cela peut se faire en divisant la sortie de corrélation  $\rho(\delta_t)$  par la norme du signal reçu. Cette normalisation permet d'éliminer la dépendance à l'amplitude  $\alpha$ , laissant uniquement l'information de phase relative.

#### Insensibilité à $\phi_0$ :

L'effet de  $\phi_0$  est un déphasage global. Pour éliminer la dépendance à  $\phi_0$ , on peut utiliser uniquement la partie réelle de  $\rho(\delta_t)$  dans l'étape de décision, ou bien prendre le module au carré de  $\rho(\delta_t)$ .

En prenant le module de  $\rho(\delta_t)$ , on élimine l'impact de  $\phi_0$ , car  $\phi_0$  n'affecte pas le module.

L'étape de décision modifiée doit alors être basée sur le **module** de la corrélation, qui est insensible à la phase  $\phi_0$  et normalisé pour éliminer l'effet de l'atténuation  $\alpha$ .

### 5 Sous-tache 4.5

Dans cette sous-tache, l'objectif principal est d'implémenter un algorithme de synchronisation temporelle dans MATLAB. Le signal est modélisé avec un délai de propagation aléatoire  $\delta t$  et un décalage fréquentiel  $\delta f$ , tous deux uniformément répartis. Afin de compenser ces désynchronisations, un algorithme basé sur la corrélation est utilisé pour estimer  $\delta t$ .<sup>5</sup> La chaîne de communication complète (voir Fig. 8) inclut un préambule de synchronisation, et l'algorithme de corrélation permet d'estimer le délai de propagation, ce qui conduit à la synchronisation du signal reçu. Ce modèle tient également compte d'autres facteurs tels que l'ajout de bruit gaussien centré et d'un déphasage aléatoire.

---

5. Fonction : Estimation\_delta\_t

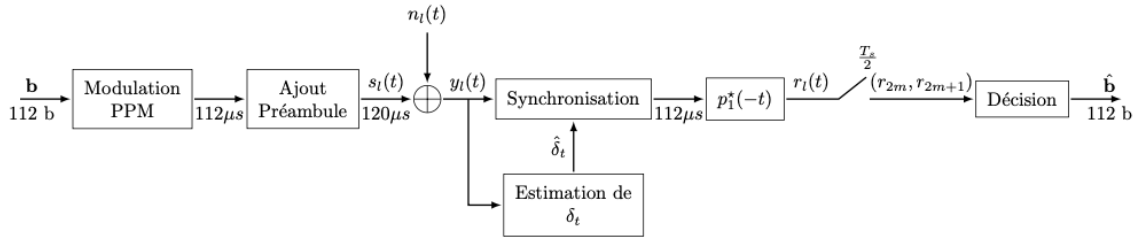


FIGURE 8 – Chaîne de communication complète

Une fois la synchronisation effectuée, le signal est convolué avec un filtre adapté et échantillonné pour extraire les bits transmis. L'estimation des bits se fait à l'aide de la méthode du maximum de vraisemblance, permettant de comparer les bits transmis aux bits estimés et de calculer le nombre d'erreurs. Le processus intègre une estimation du délai de propagation (voir Fig. 9), suivie d'une synchronisation efficace du signal (voir Fig. 10). Le code Matlab de cette tâche a été déposé sur Thor.<sup>6</sup>

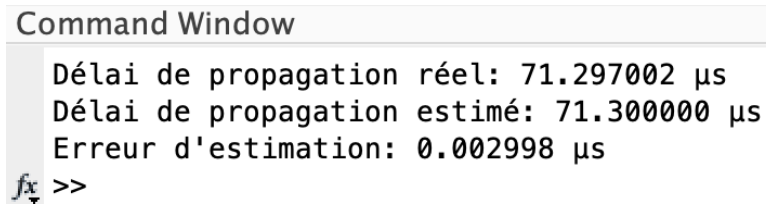


FIGURE 9 – Estimation du décalage temporel  $\delta_t$

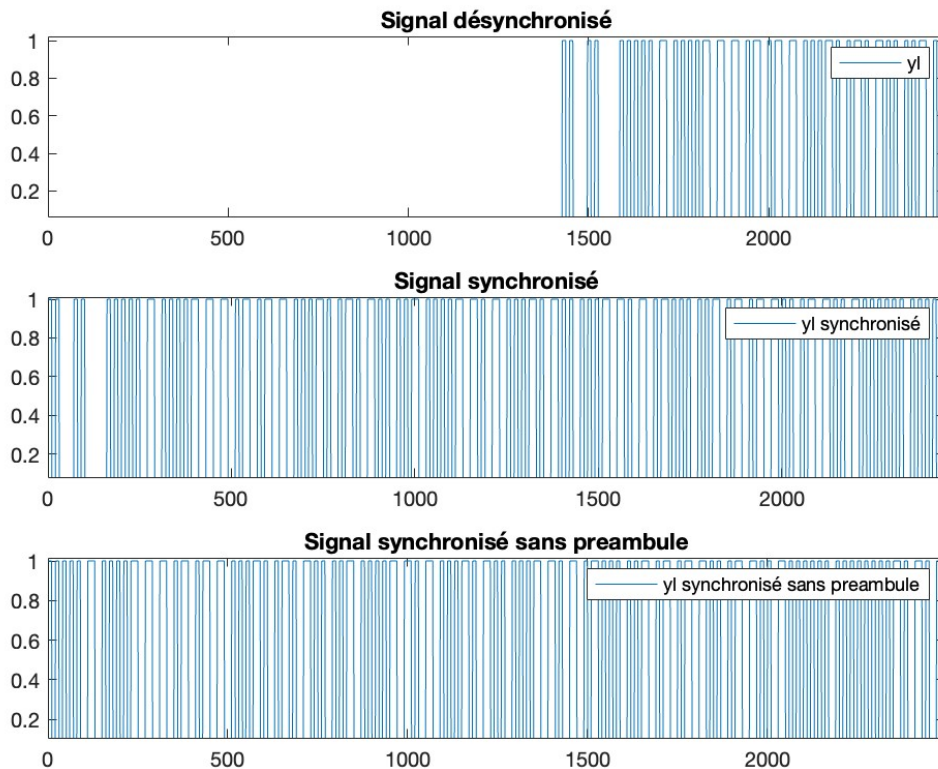


FIGURE 10 – Synchronisation temporelle du signal  $y_l(t)$  en fonction du temps  $t$

6. Fichier : Tache\_4.m

## VI Tâche 5 : Couche physique ADS-B Synchronisation en fréquence.

Dans cette tâche, nous allons nous concentrer sur la synchronisation fréquentielle en tenant compte de l'effet Doppler causé par le mouvement de l'avion et des défauts d'oscillateurs locaux, qui introduisent un décalage en fréquence  $\delta f$ . Nous allons améliorer le code MATLAB précédemment développé pour modéliser le signal reçu  $y_l(t)$  en utilisant l'équation suivante :

$$y_l(t) = s_l(t - \delta t)e^{-j2\pi\delta f t} + n_l(t)$$

où  $\delta t$  et  $\delta f$  représentent respectivement les désynchronisations temporelle et fréquentielle du signal.

L'objectif est d'intégrer ce décalage fréquentiel dans le traitement du signal afin de pouvoir estimer correctement  $\delta f$  dans un scénario sans bruit.

Pour valider notre approche, nous effectuerons une vérification en simulant la chaîne de communication sans bruit. En l'absence de bruit, nous devrions être en mesure d'estimer parfaitement le décalage fréquentiel  $\delta f$ . Nous procéderons par analogie avec la partie précédente, en codant une fonction nommée `Estimation_delta_f` qui estimera le décalage fréquentiel, puis nous synchroniserons le signal désynchronisé en utilisant cette estimation. Nous comparerons les valeurs estimées et réelles de  $\delta f$  pour assurer que notre algorithme de synchronisation fonctionne correctement dans des conditions idéales. Cependant, le travail de développement du code MATLAB n'a pas été complété en raison de quelques bugs à régler. <sup>7</sup>

---

7. Fichiers : `Tache_5.m` et `Estimation_delta_f.m`

## VII Tâche 6 : Implémentation de la couche MAC ADS-B

### 1 Objectifs

L'objectif de cette partie est de développer une fonction MATLAB capable de convertir un message binaire ADS-B en registre, afin de pouvoir en extraire des informations essentielles concernant les appareils émetteurs. Cette fonction permettra notamment de traiter des données de position et d'identification transmises dans les trames ADS-B, facilitant ainsi l'analyse et le suivi des appareils.

### 2 Structure des trames ADS-B

Les signaux émis par les appareils dans le cadre de l'ADS-B ont une durée totale de 120  $\mu\text{s}$  et sont composés des parties suivantes :

- le préambule, identique à celui utilisé dans la tâche 4,
- le format de la voie descendante (DF),
- la capacité (CA),
- l'adresse OACI (Organisation de l'Aviation Civile Internationale) de l'appareil,
- les données ADS-B,
- les bits de contrôle de parité.

La durée de chaque partie et leur position dans une trame ADS-B sont représentées dans la Figure 11.

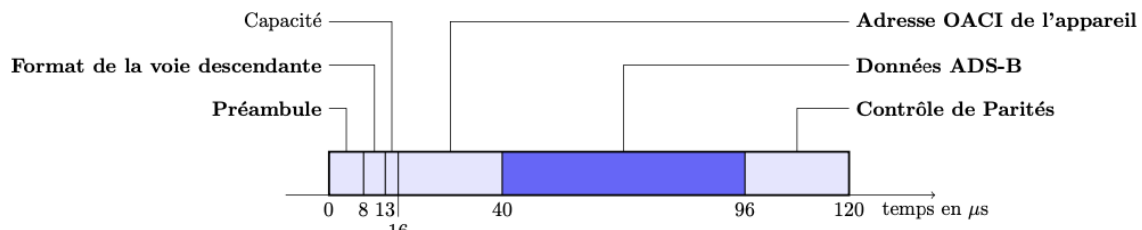


FIGURE 11 – Format d'une trame ADS-B

#### 2.1 Le préambule

Le préambule est utilisé pour la synchronisation temporelle et fréquentielle des signaux. Il est identique pour tous les types de signaux ADS-B et dure 8  $\mu\text{s}$ , comme décrit dans la tâche 4.

#### 2.2 Format de la voie descendante

Le format de la voie descendante, noté DF (Downlink Format), est codé sur 5 bits et indique le type de trame envoyée. Les valeurs spécifiques sont les suivantes :

- 11 : message d'acquiescement,
- 17 : message de type ADS-B,
- 18 : message de type TIS-B (Traffic Information Service – Broadcast),
- 19 : message de type ADS-B militaire (crypté).

Dans ce projet, seuls les messages avec  $DF = 17$  sont étudiés.

#### 2.3 L'adresse OACI de l'appareil

L'adresse OACI, codée sur 24 bits, identifie de manière unique chaque appareil tout au long de sa vie. La possibilité de décoder cette adresse est cruciale pour associer les données extraites aux appareils spécifiques, permettant d'accéder à des informations supplémentaires via des bases de données et des radars virtuels tels que [planefinder.net](http://planefinder.net) et [flightradar24.com](http://flightradar24.com).

## 2.4 Les données ADS-B

Pour les messages de type ADS-B ( $DF = 17$ ), les données occupent 56 bits et peuvent correspondre à divers types d'informations, comme la position au sol, la position en vol, l'identification de l'appareil, ou la vitesse. Le type de message est défini par le Format Type Code (FTC), un champ de 5 bits situé en tête des données.

### 3 Sous-tâche 6.1

En ADS-B, les valeurs de FTC (Format Type Code) servent à classer les différents types de messages émis par les aéronefs, notamment :

- Les valeurs de 9 à 18 et de 20 à 22 correspondent aux trames de position en vol (airborne position).
- Les valeurs de 1 à 4 sont utilisées pour les messages d'identification.

### 4 Sous-tâche 6.2 : Développement de la fonction MATLAB `bit2registre`

La fonction MATLAB `bit2registre` a été développée<sup>8</sup> pour convertir un vecteur binaire de 112 bits en un registre structuré contenant les informations pertinentes, telles que le format DF, l'adresse OACI, le code FTC, l'altitude, les coordonnées de latitude et de longitude, ainsi que les indicateurs `timeFlag` et `cprFlag`. Cette fonction ne retourne un registre que si le contrôle de CRC indique l'absence d'erreurs.

La structure de registre adoptée est la suivante :

```
registre = struct('adresse', [], 'format', [], 'type', [], 'nom', [], ...  
    'altitude', [], 'timeFlag', [], 'cprFlag', [], 'latitude', [], 'longitude', []);
```

Cette approche facilite l'initialisation et l'extraction des champs de registre pour les différents types de messages.

### 5 Sous-tâche 6.3 : Validation de la fonction `bit2registre`

Pour vérifier le bon fonctionnement de la fonction `bit2registre`, un fichier de test MATLAB, `Tache_6.m`, a été créé. Ce fichier utilise des données de trame ADS-B stockées dans `adsb_msgs.mat`, contenant plusieurs vecteurs binaires de 112 bits, chacun représentant une trame distincte. En appliquant la fonction `bit2registre` sur chaque trame, les informations pertinentes ont pu être extraites. Les coordonnées de latitude et de longitude obtenues ont ensuite été tracées pour visualiser la trajectoire de l'appareil, confirmant la validité de l'algorithme d'extraction des données, comme illustré dans la Figure 12.

Les résultats obtenus sont présentés dans les figures ci-dessous, illustrant les différentes positions et identifications des appareils décodés à partir des messages ADS-B. Les données affichées confirment la précision des informations extraites, en particulier pour la localisation géographique et les identifiants uniques de chaque appareil, ce qui démontre l'efficacité du traitement appliqué par la fonction `bit2registre`.

---

8. Fichier : `bit2registre.m`

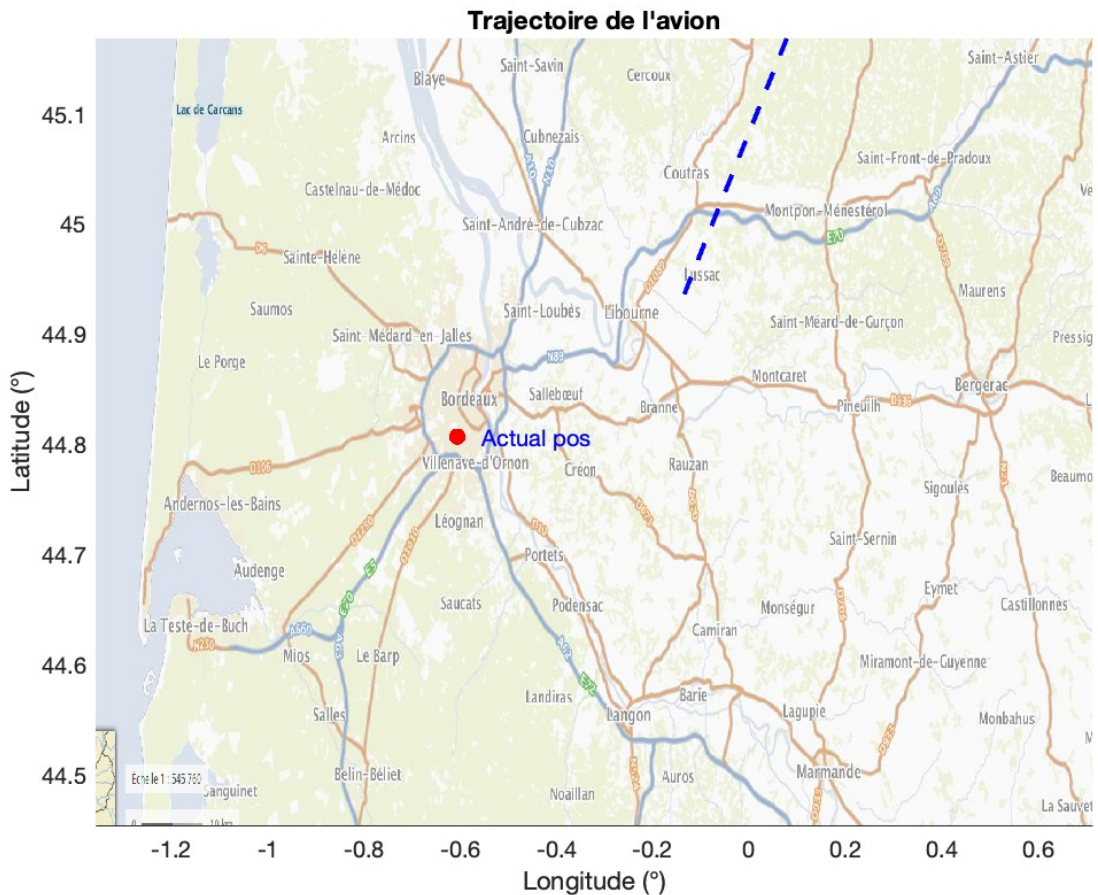


FIGURE 12 – Trajectoire de l'appareil obtenue à partir des données extraites

```

##### Début du Message #####
Aucune erreur détectée. Le message est intègre.
Format: 17(Trame ADSB)
Adresse de l'avion: 3420CA
CPR Flag: 1
Time Flag: 0
Type du message ADSB: 12 qui correspond à un:
##### Message de position en vol #####
Statut: 0
Type de l'antenne: 0
Altitude: 34950 ft
Latitude: 44.9267°
Longitude: -0.14343°
##### Fin du Message. #####
fx >>

```

FIGURE 13 – Décodage d'un message de position de vol

```

##### Début du Message #####
Aucune erreur détectée. Le message est intègre.
Format: 17(Trame ADSB)
Adresse de l'avion: 3420CA
CPR Flag: 0
Time Flag: 0
Type du message ADSB: 4 qui correspond à un:
##### Message d'identification #####
Catégorie de l'appareil: 0
Nom de l'appareil: H P WFVC
##### Fin du Message. #####
fx

```

FIGURE 14 – Décodage d'un message d'identification



## VIII Tâche 7 : Amélioration de la fonction bit2registre

### 1 Objectifs

L'objectif de cette tâche est d'améliorer la fonction `bit2registre` développée en Tâche 6, qui prend un vecteur de 112 bits en entrée et renvoie un registre de données extrait des trames ADS-B. Les améliorations apportées visent à enrichir le registre en incluant de nouvelles informations, telles que la position au sol et la vitesse en vol de l'avion.

### 2 Sous-tâche 7.1 : Intégration de la position au sol

Pour cette sous-tâche, la fonction `bit2registre` a été modifiée et désormais s'appelle `bit2registre2` pour inclure la position au sol de l'avion dans les données extraites. En se basant sur les informations de l'Annexe et de la tâche précédente, j'ai intégré un nouveau champ **positionSol** pour indiquer si l'information concerne une position au sol. J'ai d'abord initialisé la structure **registre** pour y inclure ce champ, ainsi que d'autres attributs pertinents. Ensuite, j'ai modifié la logique de décodage pour vérifier le type de message reçu en utilisant le champ FTC. Si le message correspond à une position au sol, j'ai défini l'altitude à zéro et j'ai extrait les valeurs de latitude et longitude à partir des bits appropriés, conformément à la documentation fournie. Pour ce faire, j'ai utilisé une fonction `decoder_lat_lon_sol`, que j'ai définie pour traiter les messages de position au sol selon le format CPR (Coordinated Position Reporting). J'ai également mis en place des vérifications pour stocker les coordonnées extraites dans des vecteurs afin de les tracer ultérieurement sur une carte. Ces modifications permettent une gestion plus efficace des données de localisation, assurant ainsi une analyse précise des positions d'atterrissage et de vol des aéronefs. Le résultat obtenu en exécutant le fichier **Tache\_8.m** se trouve dans la figure 15 ci-dessous :

```
%% Début du Message %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Aucune erreur détectée. Le message est intègre.
Format: 0
Adresse de l'avion: CEE0C2
CPR Flag: 1
Time Flag: 0
Type du message ADSB: 8 qui correspond à un:
##### Message de position au sol #####
Altitude de l'aéroport: 166 ft
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
  "Indicateur de Mouvement: 1"
Statut: 3
Latitude: 44.7261°
Longitude: -1.312°
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin du Message. %%%%%%%%%%
```

FIGURE 15 – Décodage d'un message de position de sol

### 3 Sous-tâche 7.2 : Intégration de la vitesse de l'avion en vol

En parallèle avec l'intégration de la position au sol, la fonction a été ajustée pour inclure la vitesse en vol de l'avion. Selon la documentation fournie : [ICAO Doc 9871, Technical Provisions for Mode S Services and Extended Squitter - ISBN 978-92-9231-117-9](#) de l'organisation internationale d'aviation civile, pages **153/154**, en exploitant les informations fournies dans les trames ADS-B, il est possible d'extraire des données relatives à la vitesse de l'avion lorsqu'il est en déplacement aérien. Le registre mis à jour contient désormais les champs nécessaires pour décrire la vitesse de l'appareil, permettant ainsi une vue plus complète de l'état en vol de celui-ci. Le résultat obtenu en exécutant le fichier **Tache\_8.m** se trouve dans la figure 16 ci-dessous :

```

##### Début du Message #####
Aucune erreur détectée. Le message est intègre.
Format: 13
Adresse de l'avion: BD8A2F
CPR Flag: 0
Time Flag: 1
Type du message ADSB: 19 qui correspond à un:|
##### Message de la vitesse en vol #####
Indice intent Change: 1
ifr_cap_flag: 1
Catégorie de la précision de navigation pour la vitesse: 5
Direction de la vitesse Est/West: Est
Vitesse Est/West: 42kt
Direction de la vitesse Nord/Sud: Sud
Vitesse Nord/sud: 559kt
Source du taux vertical (GNSS/Baro): GNSS
Signe du taux vertical (up/down): down
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"

  "Bits réservés: 0"   "Bits réservés: 1"

  "Bits réservés: 0"   "Bits réservés: 1"

Bit de différence de signe: Below baro alt.
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"

  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 24.8015 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 24.8015 ft"

Statut: Cap magnétique indisponible.
Type IAS/TAS de la vitesse: TAS
Vitesse en vol: 559kt
Source du taux vertical (GNSS/Baro): GNSS
Signe du taux vertical (up/down): down
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 0 ft/min"
  "Taux vertical: 63.8745 ft/min"
  "Taux vertical: 0 ft/min"

  "Bits réservés: 0"   "Bits réservés: 1"

  "Bits réservés: 0"   "Bits réservés: 1"

Bit de différence de signe: Below baro alt.
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 24.8015 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 0 ft"
  "Différence de hauteur géométrique par rapport à l'altitude barométrique: 24.8015 ft"
##### Fin du Message. #####

```

FIGURE 16 – Décodage d'un message de vitesse

#### 4 Sous-tâche 7.3 : Calcul de la position en vol sans hypothèse de proximité

Le calcul de la position aérienne des avions à partir des messages ADS-B repose sur le Compact Position Reporting (CPR), qui encode les latitudes et longitudes avec un nombre limité de bits. L'encodage CPR est conçu pour fonctionner en deux formats : even (pair) et odd (impair), qui permettent de calculer une position précise en utilisant deux messages consécutifs provenant de l'avion.

Dans le cas où l'on dispose uniquement d'un format CPR even (indicateur CPR à zéro), la position décodée est limitée à une précision locale. Cela signifie que l'on doit faire l'hypothèse que l'avion se trouve à proximité de la station de réception ou d'une position de référence connue. Cette hypothèse peut introduire des erreurs lorsque l'avion se trouve loin de la station de réception, car le système CPR présente une ambiguïté géographique sur une distance de plusieurs centaines de kilomètres (environ 180 nœuds nautiques). Par conséquent, se baser uniquement sur des messages even ne permet pas d'obtenir une position unambiguë à l'échelle globale.

Pour lever cette ambiguïté, le CPR utilise une technique basée sur la réception de deux messages CPR avec des formats différents : un message en format even et un autre en format odd, envoyés à quelques secondes d'intervalle (généralement moins de 10 secondes). La combinaison de ces deux messages permet de déterminer une position unique et précise à l'échelle globale.

Si l'intervalle est trop long (plus de 10 secondes), la précision de la position globale ne peut plus être garantie, car l'avion pourrait s'être déplacé de plus de 3 milles nautiques (environ 5.5 km), rendant ainsi la combinaison des messages imprécise.

Le code MATLAB de cette fonction qui améliore le calcul de la position en vol sans hypothèse de proximité se trouve dans le fichier : **decoder\_lat\_lon\_unambiguous.m** et est basé sur les **pages 52 et 53 section A.2.6.7 GLOBALLY UNAMBIGUOUS AIRBORNE POSITION DECODING** de la documentation : ICAO Doc 9871, Technical Provisions for Mode S Services and Extended Squitter - ISBN 978-92-9231-117-9 de l'organisation internationale d'aviation civile.

En voici un extrait (Figure 17) :

Given a 17-bit airborne position encoded in the “**even**” format ( $XZ_0, YZ_0$ ) and another encoded in the “**odd**” format ( $XZ_1, YZ_1$ ), separated by no more than 10 seconds (= 3 NM), the CPR algorithm shall regenerate the geographic position from the encoded position reports by performing the following sequence of steps:

- a) Compute  $Dlat_0$  and  $Dlat_1$  from the equation:

$$Dlat_i = \frac{360^\circ}{4 \cdot NZ - i}$$

- b) Compute the latitude index:

$$j = \text{floor} \left( \frac{59 \cdot YZ_0 - 60 \cdot YZ_1}{2^{17}} + \frac{1}{2} \right)$$

- c) Compute the values of  $Rlat_0$  and  $Rlat_1$  using the following equation:

$$Rlat_i = Dlat_i \cdot \left( \text{MOD}(j, 60 - i) + \frac{YZ_i}{2^{17}} \right)$$

Southern hemisphere values of  $Rlat_i$  will fall in the range from  $270^\circ$  to  $360^\circ$ . Subtract  $360^\circ$  from such values, thereby restoring  $Rlat_i$  to the range from  $-90^\circ$  to  $+90^\circ$ .

- d) If  $NL(Rlat_0)$  is not equal to  $NL(Rlat_1)$  then the two positions straddle a transition latitude, thus a solution for global longitude is not possible. Wait for positions where they are equal.
- e) If  $NL(Rlat_0)$  is equal to  $NL(Rlat_1)$  then proceed with computation of  $Dlon_i$ , according to whether the most recently received airborne position message was encoded with the even format ( $i = 0$ ) or the odd format ( $i = 1$ ):

$$Dlon_i = \frac{360^\circ}{n_i}$$

where  $n_i = \text{greater of } [NL(Rlat_i) - i] \text{ and } 1$ .

- f) Compute  $m$ , the longitude index:

$$m = \text{floor} \left( \frac{XZ_0 \cdot (NL - 1) - XZ_1 \cdot NL}{2^{17}} + \frac{1}{2} \right)$$

where  $NL = NL(Rlat_i)$ .

- g) Compute the global longitude,  $Rlon_0$  or  $Rlon_1$ , according to whether the most recently received airborne position message was encoded using the even format (that is, with  $i = 0$ ) or the odd format ( $i = 1$ ):

$$Rlon_i = Dlon_i \cdot \left( \text{MOD}(m, n_i) + \frac{XZ_i}{2^{17}} \right)$$

where  $n_i = \text{greater of } [NL(Rlat_i) - i] \text{ and } 1$ .

FIGURE 17 – Extrait de la section A.2.6.7 (Décodage sans l’hypothèse de proximité)

## IX Tâche 8 : Traitement des Signaux Réels

### 1 Objectifs

L'objectif principal de cette tâche est de traiter des signaux réels obtenus à partir de captures ADS-B en utilisant un buffer plus important que dans les analyses précédentes. Pour ce faire, il a été nécessaire d'adapter le code afin de gérer des enregistrements de signaux provenant du fichier `buffers.mat`, ce qui permet de traiter des données brutes en grand volume, capturées à une fréquence d'échantillonnage précise.

### 2 Méthodologie proposée

**Chargement et Préparation des Données** La première étape a consisté à charger les données du fichier `buffers.mat` et à analyser la structure du buffer pour adapter le traitement en conséquence. Des paramètres de traitement, tels que le temps d'échantillonnage ( $T_s$ ), la fréquence d'échantillonnage ( $F_e$ ) et la longueur de trame sans préambule, ont été définis pour effectuer des calculs précis et détecter efficacement les trames ADS-B présentes dans le buffer.

**Détection et Extraction des Trames** À partir de cette analyse, j'ai généré le préambule de référence (Voir Tache 4) pour les signaux ADS-B afin d'identifier les pics de corrélation dans le buffer. Un seuil de détection a été défini pour valider les corrélations significatives, selon un compromis entre le nombre de pics détectés et le nombre de faux vrais pics, permettant ainsi de repérer les indices de départ des trames ADS-B. Le code MATLAB de cette détection se trouve dans la fonction `Estimation_indices_max.m`. Ensuite, chaque trame détectée a été extraite et stockée pour un traitement individuel.

**Traitement et Décodage des Trames** Une fois les trames ADS-B extraites, un filtre adapté a été appliqué à chaque trame pour améliorer la détection des bits. J'ai ensuite échantillonné et décodé les signaux en appliquant une méthode de décision par maximum de vraisemblance, permettant d'estimer les bits de chaque trame. Les bits estimés ont été regroupés dans une matrice pour une analyse plus approfondie. Enfin, les informations de latitude et de longitude ont été extraites en utilisant la fonction codée précédemment à la tâche 6 et améliorée à la tâche 7 `bit2registre2.m`, qui, après avoir vérifié le CRC convertit chaque trame en registre avec des informations géographiques. Ces résultats constituent une étape clé pour le traçage de la trajectoire de l'avion en temps réel.

**Vérification et Résultats Préliminaires** Le nombre de trames détectées a été affiché pour évaluer l'efficacité de la détection, et les premières analyses montrent que les informations de position et de vitesse peuvent être extraites pour une traçabilité complète.

**Méthodologie de Traçage des Trajectoires** Bien que le traçage complet de la trajectoire de l'avion n'ait pas été achevé, la méthode prévue repose sur un processus de regroupement et d'organisation des données de position pour chaque avion détecté. En utilisant l'adresse unique de chaque avion, une liste de latitudes et de longitudes est créée pour suivre sa trajectoire.

Pour chaque avion identifié par une adresse unique, une liste de coordonnées géographiques (latitude et longitude) est stockée. Ce regroupement permet un aperçu visuel de la trajectoire de chaque avion et un suivi plus précis de sa position. L'étape de traçage finale consistera à afficher les trajectoires de tous les avions sur une carte pour une visualisation complète des parcours aériens.

## X Conclusion Générale

Ce projet visait à développer une chaîne de traitement complète pour l'analyse et la démodulation des signaux ADS-B, afin d'extraire et de visualiser des informations de positionnement et de suivi des avions. Diverses tâches ont été mises en œuvre pour accomplir cet objectif, chacune apportant des fonctionnalités et des améliorations spécifiques.

Les tâches successives ont couvert un ensemble complet de concepts et de techniques. Les premières tâches ont consisté en la prise en main de la chaîne de communication ADS-B en reproduisant et analysant des algorithmes de traitement du signal, de codage et de décodage des trames. À ce stade, nous avons pu identifier la structure fondamentale des messages ADS-B, explorer la densité spectrale de puissance, et travailler sur la mise en place d'algorithmes robustes pour le codage et le décodage de canal.

Les tâches suivantes se sont orientées vers la synchronisation en temps et en fréquence. Celles-ci étaient particulièrement complexes en raison des défis associés aux décalages temporels et fréquentiels, qui peuvent causer une perte d'information. L'implémentation d'algorithmes de correction a permis d'améliorer la précision de réception des messages en introduisant des méthodes de détection de préambule et de filtrage.

Par la suite, l'intégration de la couche MAC ADS-B et le développement de la fonction `bit2registre2` ont constitué une avancée importante. La fonction a été développée pour transformer des vecteurs de bits en informations décodées (adresse, format, type de message, position, etc.), et a ensuite été enrichie pour inclure des paramètres supplémentaires tels que la position au sol et la vitesse en vol. Cette fonction a également été ajustée pour pouvoir estimer la position d'un avion sans supposer de proximité avec la station de réception, ouvrant ainsi la voie à une meilleure estimation des positions dans des contextes variés.

Enfin, dans la tâche 8, le travail a été appliqué à des signaux ADS-B issus d'acquisitions réelles, nécessitant l'adaptation du code aux particularités des signaux collectés. La détection de préambule et l'extraction des messages ADS-B dans un environnement plus complexe ont représenté un défi technique, aboutissant à une méthode pour extraire, analyser, et regrouper les trajectoires des avions.

En conclusion, ce projet a permis d'explorer en profondeur le traitement des signaux ADS-B et d'appliquer des techniques avancées pour la synchronisation et le décodage. Les défis rencontrés — notamment ceux liés à la précision des algorithmes de synchronisation et à l'adaptation de la fonction de décodage — ont permis d'acquérir une expérience solide et un savoir-faire pratique dans le domaine des systèmes de communication aérienne.

## XI Références

- ICAO Doc 9871, *Technical Provisions for Mode S Services and Extended Squitter*, Order Number : 9871, ISBN 978-92-9231-117-9.