



## PROJET TS229

SIMULATION D'UN ÉMETTEUR / RÉCEPTEUR ADS-B ET DÉCODAGE  
TEMPS RÉEL À L'AIDE DE RADIO LOGICIELLE

---

# Rapport de projet

4 novembre 2025

---

*Rédigé par :*

Ben Amor Eya  
Mohamed Yassine Ait Mhand

*Encadrant :*

Romain Tajan

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Couche physique ADS-B</b>	<b>2</b>
2.1	Tâche 1 : Prise en main de la chaîne de communication ADSB . . . . .	2
2.1.1	Calcul Théorique . . . . .	3
2.1.2	Implémentation Matlab et Vérification . . . . .	5
2.2	Tâche 2 : Densité spectrale de puissance . . . . .	6
2.2.1	Calcul théorique . . . . .	6
2.2.2	Estimation et Analyse de la DSP par la méthode de Welch . . . . .	9
2.3	Tâche 3 : Algorithmes de codage et de décodage de canal . . . . .	10
2.4	Tâche 4 : Synchronisation en temps . . . . .	12
2.5	Tâche 5 : Synchronisation fréquentielle . . . . .	15
<b>3</b>	<b>Tâche 6 et 7 : Couche MAC ADS-B</b>	<b>16</b>
<b>4</b>	<b>Tâche 8 : Traitement de signaux réels ADS-B</b>	<b>18</b>
4.1	Sous-tâche 1 : Estimation du SNR et repérage du préambule . . . . .	18
4.2	Sous-tâche 2 : Décodage et affichage des trajectoires détectées . . . . .	20
4.3	Conclusion de la tâche 8 . . . . .	21
<b>5</b>	<b>Tâche 9 : Mise en place du temps réel</b>	<b>21</b>
<b>6</b>	<b>Conclusion et perspectives</b>	<b>22</b>

## 1 Introduction

Le système **ADS-B** (Automatic Dependent Surveillance – Broadcast) est une technologie essentielle de la surveillance aérienne moderne. Il permet à chaque avion de diffuser périodiquement sa position, sa vitesse et son identification, offrant ainsi une vision précise et en temps réel du trafic aérien.

Le projet **TS229** a pour objectif de simuler, sous **MATLAB**, une chaîne complète d'émission et de réception ADS-B. Cette étude couvre l'ensemble des étapes de la communication numérique : modulation PPM, ajout de bruit, filtrage, synchronisation et décodage.

Dans un second temps, la simulation est appliquée à l'analyse de **signaux réels** acquis à l'aide d'une **radio logicielle (SDR)**, afin d'évaluer la performance du récepteur et la qualité du canal.

Ce projet permet ainsi de relier les notions théoriques de transmission numérique à une application concrète de l'aéronautique : la réception et le décodage en temps réel des trames ADS-B.

## 2 Couche physique ADS-B

### 2.1 Tâche 1 : Prise en main de la chaîne de communication ADSB

La norme ADS-B repose sur une modulation de type **PPM** (Pulse Position Modulation).

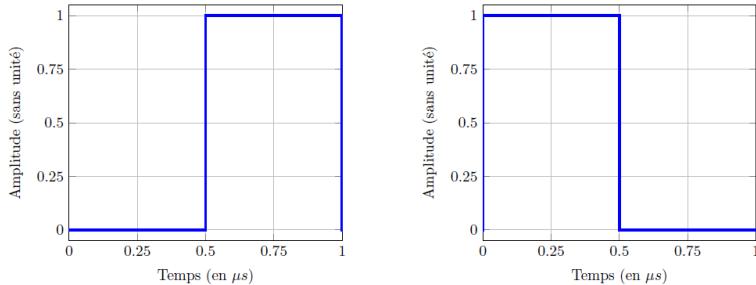


FIGURE 1 – Impulsions de base pour la modulation par position

Comme illustré à la figure 1, les bits sont représentés par des impulsions temporellement décalées .

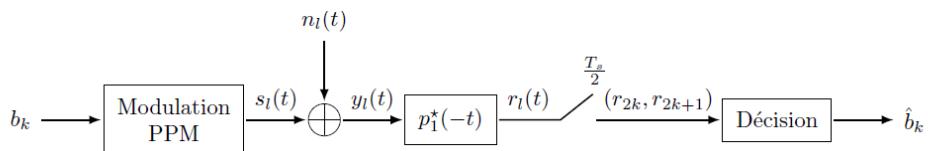


FIGURE 2 – Chaîne de communication ADS-B

On considère la chaîne présentée à la Figure 2, où :

- les bits d'entrée  $b_k$  sont indépendants et distribués uniformément sur  $\{0, 1\}$  ;
- le bruit en bande de base  $n_l(t)$  suit une loi normale centrée de variance  $\sigma_{n_l}^2$ , soit  $n_l(t) \sim \mathcal{N}(0, \sigma_{n_l}^2)$  ;
- la densité spectrale de puissance (DSP) bilatérale du bruit est constante et donnée par :

$$\Gamma_{n_l}(f) = \frac{N_0}{2}.$$

**Notations communes :** On fixe  $T_s = 1 \mu s$  et  $W = \frac{T_s}{2}$ . On note la porte causale de largeur  $W$  :

$$\Pi_W(t) = \begin{cases} 1, & 0 \leq t \leq W, \\ 0, & \text{sinon.} \end{cases}$$

Conformément à l'énoncé, les impulsions PPM binaire (causales) sont

$$p_0(t) = \Pi_W(t - W) \quad (\text{bit 0 : moitié tardive}), \quad p_1(t) = \Pi_W(t) \quad (\text{bit 1 : moitié précoce}).$$

Le signal émis s'écrit

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

où la forme d'onde élémentaire  $p_{b_k}(t)$  dépend du bit transmis :

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

L'énergie du gabarit  $p_1$  vaut

$$v_0 = \int_{-\infty}^{\infty} |p_1(t)|^2 dt = \int_0^W 1 dt = W = \frac{T_s}{2}.$$

### 2.1.1 Calcul Théorique

L'objectif de cette première tâche est de comprendre la modulation PPM (Pulse Position Modulation) utilisée dans la norme ADS-B, et de modéliser la chaîne de communication de la figure 2.

#### Sous-tâche 1 : Théorie – représentation de $s_l(t)$ , $r_l(t)$ et $r_m$

En considérant un bruit nul et une séquence binaire  $[1, 0, 0, 1, 0]$ , le signal émis s'écrit :

$$s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s), = \sum_{k \in \mathbb{Z}} \left[ (1 - b_k) \Pi_W(t - kT_s - W) + b_k \Pi_W(t - kT_s) \right],$$

où  $T_s = 1 \mu s$  et  $W = \frac{T_s}{2}$ .

Le filtre adapté au récepteur est construit à partir de  $p_1(t)$ , ce qui conduit à un signal en sortie de corrélation :

$$r_l(t) = (s_l * p_1^*(-t))(t) = \sum_{k \in \mathbb{Z}} \left[ (1 - b_k) (\Pi_W * \Pi_W)(t - kT_s - W) + b_k (\Pi_W * \Pi_W)(t - kT_s) \right].$$

Or, la convolution de deux portes rectangulaires  $\Pi_W$  donne une fonction triangulaire :

$$\Lambda_W(x) = (\Pi_W * \Pi_W)(x) = \begin{cases} W - |x|, & \text{si } |x| < W, \\ 0, & \text{sinon.} \end{cases}$$

Cette fonction  $\Lambda_W(x)$  possède une base  $2W = T_s$  et une hauteur maximale  $W$ .

Ainsi :

$$r_l(t) = \sum_{k \in \mathbb{Z}} \left[ (1 - b_k) \Lambda_W(t - kT_s - W) + b_k \Lambda_W(t - kT_s) \right].$$

Les échantillons de décision sont pris en deux instants par symbole (donc aux multiples de  $T_s/2$ ) :

$$t_{E,k} = kT_s - W \quad (\text{précoce}), \quad t_{L,k} = kT_s \quad (\text{tardif}),$$

ce qui permet de constituer les couples  $(r_{2k}, r_{2k+1})$  pour la prise de décision symbolique

$$(r_{2k}, r_{2k+1}) = \begin{cases} (0, W), & b_k = 0, \\ (W, 0), & b_k = 1. \end{cases}$$

### Sous-tâche 2 : Théorie – réécriture à pas $T_s/2$

On remarque que  $p_0(t) = p_1(t - W)$  avec  $W = \frac{T_s}{2}$ .  
Ainsi :

$$s_l(t) = \sum_{k \in \mathbb{Z}} \left[ b_k p_1(t - kT_s) + (1 - b_k) p_1(t - kT_s - W) \right] = \sum_{m \in \mathbb{Z}} u_m p_1\left(t - m \frac{T_s}{2}\right),$$

avec

$$(u_{2k}, u_{2k+1}) = (b_k, 1 - b_k).$$

En sortie du corrélateur, on obtient

$$r_m = W u_m,$$

### Sous-tâche 3 : Théorie – Décision ML

On regroupe les deux corrélations du symbole  $k$  dans le vecteur

$$\mathbf{r}_k = \begin{bmatrix} r_{2k} \\ r_{2k+1} \end{bmatrix} = \mathbf{s}_{b_k} + \mathbf{n}_k,$$

où, en absence de bruit,

$$\mathbf{s}_1 = \begin{bmatrix} v_0 \\ 0 \end{bmatrix}, \quad \mathbf{s}_0 = \begin{bmatrix} 0 \\ v_0 \end{bmatrix}, \quad \text{avec} \quad v_0 = \int_{-\infty}^{\infty} |p_1(t)|^2 dt.$$

Le bruit bande de base  $n_l(t)$  est gaussien blanc de DSP  $N_0/2$ . À la sortie des corrélateurs :

$$\mathbf{n}_k = \begin{bmatrix} n_{E,k} \\ n_{L,k} \end{bmatrix}, \quad n_{E,k}, n_{L,k} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \quad \sigma^2 = \frac{N_0}{2} v_0.$$

Sous ces hypothèses gaussiennes, le récepteur optimal au sens du **maximum de vraisemblance (ML)** choisit le symbole  $\mathbf{s}_{b_k}$  qui rend le vecteur reçu  $\mathbf{r}_k$  le plus probable. Comme la variance du bruit  $\sigma^2$  est identique pour les deux hypothèses ( $b = 0$  ou  $1$ ), le critère ML se simplifie et revient à minimiser la **distance euclidienne** :

$$\|\mathbf{r}_k - \mathbf{s}_0\|_2^2 \underset{b_k=1}{\underset{b_k=0}{\gtrless}} \|\mathbf{r}_k - \mathbf{s}_1\|_2^2$$

En développant :

$$\|\mathbf{r}_k - \mathbf{s}_0\|^2 - \|\mathbf{r}_k - \mathbf{s}_1\|^2 = 2v_0(r_{2k+1} - r_{2k}).$$

Ainsi le test ML est équivalent à la simple comparaison :

$$r_{2k} \underset{b_k=0}{\underset{b_k=1}{\gtrless}} r_{2k+1}$$

### Sous-tâche 7 : Théorie — Probabilité d'erreur binaire $P_b$ pour la PPM

Avec le test ci-dessus, supposons  $b_k = 1$  (cas symétrique pour  $b_k = 0$ ). Les sorties sont

$$r_{2k} = v_0 + n_{E,k}, \quad r_{2k+1} = 0 + n_{L,k},$$

avec  $n_{E,k}, n_{L,k} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ ,  $\sigma^2 = \frac{N_0}{2} v_0$ .

Une erreur se produit si  $r_{2k+1} > r_{2k}$ , soit

$$n_{L,k} - n_{E,k} > v_0.$$

Or  $n_{L,k} - n_{E,k} \sim \mathcal{N}(0, 2\sigma^2)$ , donc

$$P_e = P(n_{L,k} - n_{E,k} > v_0) = Q\left(\frac{v_0}{\sqrt{2\sigma^2}}\right) = Q\left(\frac{v_0}{\sqrt{N_0 v_0}}\right) = Q\left(\sqrt{\frac{v_0}{N_0}}\right).$$

Pour la PPM binaire, l'énergie par bit est précisément l'énergie de l'impulsion non nulle :

$$E_b = v_0.$$

D'où la forme finale classique

$$P_b = Q\left(\sqrt{\frac{E_b}{N_0}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right).$$

#### 2.1.2 Implémentation Matlab et Vérification

La chaîne de communication a été implémentée sous MATLAB avec un facteur de suréchantillonnage  $F_{se} = 20$  (soit  $f_e = 20$  MHz et  $T_s = 1 \mu\text{s}$ ).

Le récepteur applique un filtrage adapté et une détection « early/late » suivant la règle :

$$r_E > r_L \Rightarrow \hat{b}_k = 1, \quad \text{sinon } \hat{b}_k = 0.$$

La Figure 3 illustre les signaux simulés pour la séquence  $[1, 0, 0, 1, 0]$ , où l'on constate une parfaite correspondance entre les bits transmis et les décisions finales.

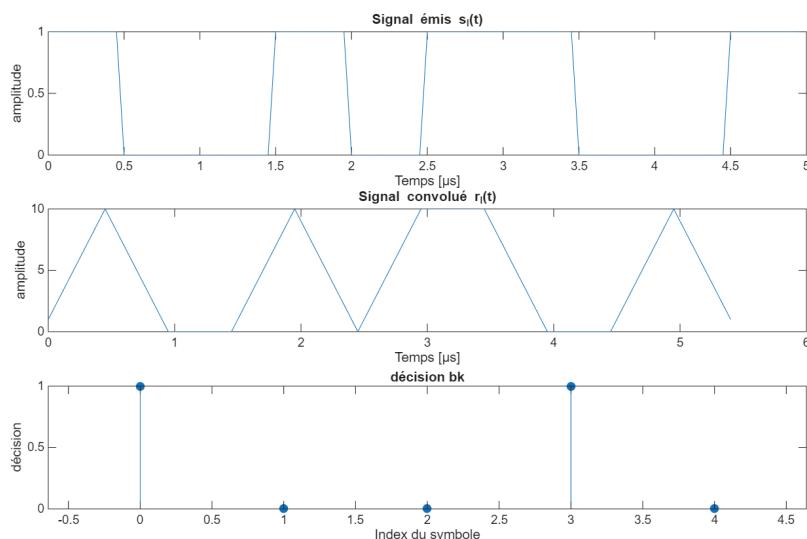


FIGURE 3 – Résultats de simulation :  $s_l(t)$ ,  $r_l(t)$  et décisions finales.

Pour un paquet de  $N_b = 1000$  bits, le TEB est estimé pour  $E_b/N_0 \in [0, 10]$  dB et comparé à la probabilité d'erreur théorique :

$$P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right).$$

La Figure 4 montre une superposition quasi parfaite entre les courbes simulée et théorique, validant la cohérence du modèle et la performance du récepteur.

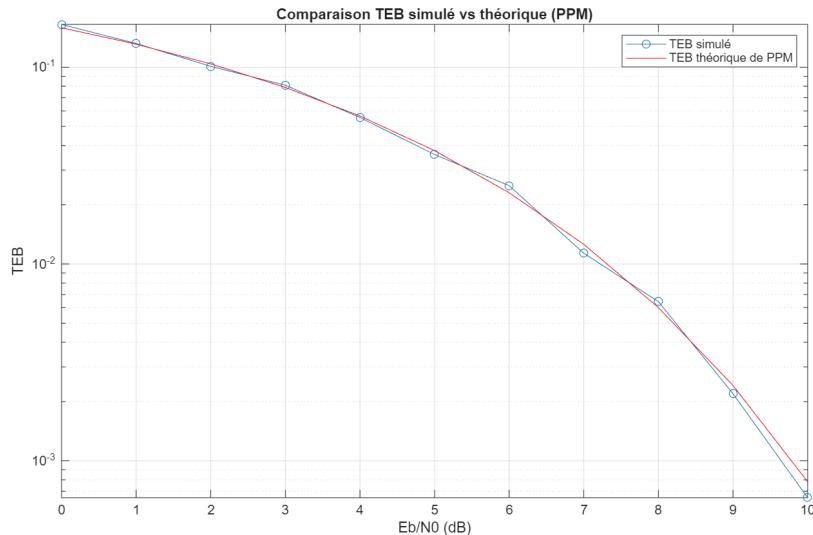


FIGURE 4 – Taux d'erreur binaire simulé vs théorique.

## 2.2 Tâche 2 : Densité spectrale de puissance

### 2.2.1 Calcul théorique

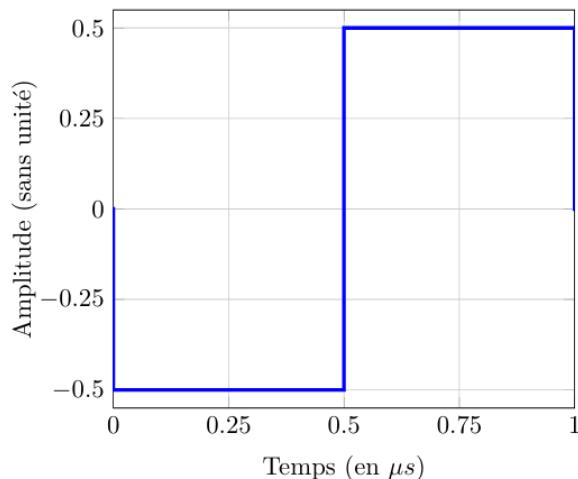


FIGURE 5 – Forme d'onde biphasé  $p(t)$

**Sous-tâche 1 — Réécriture biphasée de  $s_l(t)$  :** Nous souhaitons démontrer que le signal émis  $s_l(t)$ , défini par la Modulation PPM symbole par symbole, peut être réécrit sous une forme biphasée.

Le signal  $s_l(t)$  est initialement donné par :

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

Soit le coefficient biphasé  $A_k$ , lié au bit  $b_k$  par la relation :

$$A_k = 1 - 2b_k \Rightarrow \begin{cases} A_k = +1 & \text{si } b_k = 0 \\ A_k = -1 & \text{si } b_k = 1 \end{cases}$$

On remarque que la fonction d'impulsion biphasée  $p(t)$  est définie comme :

$$p(t) = 0.5 \cdot \mathbb{1}_{[0, T_s]}(t) - p_1(t) = p_0(t) - 0.5 \cdot \mathbb{1}_{[0, T_s]}(t)$$

On peut donc écrire :

$$\forall t : p_{b_k}(t) = A_k p(t) + 0.5 \cdot \mathbb{1}_{[0, T_s]}(t)$$

En substituant cette relation dans l'expression de  $s_l(t)$  :

$$\begin{aligned} s_l(t) &= \sum_{k \in \mathbb{Z}} p_{b_k}(t - kT_s) \\ &= \sum_{k \in \mathbb{Z}} [A_k p(t - kT_s) + 0.5 \cdot \mathbb{1}_{[0, T_s]}(t - kT_s)] \\ &= \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) + 0.5 \sum_{k \in \mathbb{Z}} \mathbb{1}_{[0, T_s]}(t - kT_s) \end{aligned}$$

Or

$$\sum_{k \in \mathbb{Z}} \mathbb{1}_{[0, T_s]}(t - kT_s) = 1 \quad \forall t \in \mathbb{R}$$

Alors, on obtient notre équivalence :

$$s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)$$

**Sous-tâche 2 — Moment d'ordre 1 (indépendant de  $t$ ) :**

Par linéarité :  $\mathbb{E}[s_l(t)] = \frac{1}{2} + \sum_k \mathbb{E}[A_k] p(t - kT_s) = \frac{1}{2}$   
or puisque les bits sont équiprobables alors  $\mathbb{E}[A_k] = 0$  Et donc

$$\boxed{\mathbb{E}[s_l(t)] = \frac{1}{2}.}$$

$\Rightarrow$  indépendant du temps

**Sous-tâche 3 — Autocorrélation  $R_{s_l}(t, \tau)$  :**

Nous partons de l'expression biphasée du signal  $s_l(t)$  :

$$s_l(t) = \frac{1}{2} + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)$$

$$\begin{aligned}
 R_{s_l}(t, \tau) &= \mathbb{E}[s_l(t) s_l(t + \tau)] = \mathbb{E} \left[ \left( \frac{1}{2} + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) \right) \left( \frac{1}{2} + \sum_{m \in \mathbb{Z}} A_m p(t + \tau - mT_s) \right) \right] \\
 &= \mathbb{E} \left[ \frac{1}{4} + \frac{1}{2} \sum_m A_m p(t + \tau - mT_s) + \frac{1}{2} \sum_k A_k p(t - kT_s) + \sum_k \sum_m A_k A_m p(t - kT_s) p(t + \tau - mT_s) \right]
 \end{aligned}$$

Par linéarité de l'espérance :

$$\begin{aligned}
 R_{s_l}(t, \tau) &= \frac{1}{4} + \frac{1}{2} \sum_m \mathbb{E}[A_m] p(t + \tau - mT_s) + \frac{1}{2} \sum_k \mathbb{E}[A_k] p(t - kT_s) \\
 &\quad + \sum_k \sum_m \mathbb{E}[A_k A_m] p(t - kT_s) p(t + \tau - mT_s)
 \end{aligned}$$

Or les coefficients  $A_k$  (codant les bits  $b_k$ ) sont supposés indépendants et équiprobables, ce qui implique :

- Moment d'ordre 1 :  $\mathbb{E}[A_k] = 0$ .
- Moment d'ordre 2 :  $\mathbb{E}[A_k A_m] = \delta_{k,m}$

Alors :

- Les deux sommes simples s'annulent (0).
- La double somme se réduit aux termes où  $k = m$  :

$$\sum_k \sum_m \delta_{k,m} p(t - kT_s) p(t + \tau - mT_s) = \sum_{k \in \mathbb{Z}} p(t - kT_s) p(t + \tau - kT_s)$$

Ainsi, la fonction d'autocorrélation est :

$$R_{s_l}(t, \tau) = \frac{1}{4} + \sum_{k \in \mathbb{Z}} p(t - kT_s) p(t + \tau - kT_s).$$

**Conclusion :** Cette fonction d'autocorrélation est périodique en  $t$  de période  $T_s$ . Le signal  $s_l(t)$  est donc une fonction *cyclostationnaire* de période  $T_s$ .

**Rq :** Si  $|\tau| \geq T_s$ , alors l'impulsion  $p(t)$  et son décalage  $p(t + \tau)$  ne se recouvrent jamais sur le même intervalle symbole, ce qui entraîne que le terme de sommation est nul pour tout  $t$ . Dans ce cas,  $R_{s_l}(t, \tau) = \frac{1}{4}$ .

**Sous-tâche 4 — Autocorrélation moyennée  $\tilde{R}_{s_l}(\tau)$  :**

$$\begin{aligned}
 \tilde{R}_{s_l}(\tau) &= \frac{1}{T_s} \int_0^{T_s} R_{s_l}(t, \tau) dt \\
 &= \frac{1}{T_s} \int_0^{T_s} \left( \frac{1}{4} + \sum_{k \in \mathbb{Z}} p(t - kT_s) p(t + \tau - kT_s) \right) dt \\
 &= \frac{1}{T_s} \int_0^{T_s} \frac{1}{4} dt + \frac{1}{T_s} \sum_{k \in \mathbb{Z}} \int_0^{T_s} p(t - kT_s) p(t + \tau - kT_s) dt
 \end{aligned}$$

Effectuons le changement de variable  $u = t - kT_s$  dans l'intégrale.

$$= \frac{1}{4} + \frac{1}{T_s} \sum_{k \in \mathbb{Z}} \int_{-kT_s}^{(1-k)T_s} p(u) p(u + \tau) du = \frac{1}{4} + \frac{1}{T_s} \int_{-\infty}^{\infty} p(u) p(u + \tau) du$$

Ainsi, la fonction d'autocorrélation moyennée est donnée par :

$$\tilde{R}_{sl}(\tau) = 0.25 + \frac{1}{T_s} R_p(\tau)$$

où  $R_p(\tau) = \int_{-\infty}^{\infty} p(t)p(t+\tau)dt$  la fonction d'autocorrélation de l'impulsion  $p(\cdot)$ .

**Sous-tâche 5 — DSP  $\Gamma_{sl}(f)$ .**

**Wiener–Khinchin :**  $F\{R_p(\tau)\} = |P(f)|^2$

Alors

$$\Gamma_{sl}(f) = \mathcal{F}\{\tilde{R}_{sl}(\tau)\} = \mathcal{F}\left\{\frac{1}{4} + \frac{1}{T_s} R_p(\tau)\right\} = \frac{1}{4} \delta(f) + \frac{1}{T_s} |P(f)|^2,$$

Avec  $p(t) = \frac{1}{2}\Pi_W(t) - \frac{1}{2}\Pi_W(t-W)$  et  $W = \frac{T_s}{2}$  :

$$P(f) = \frac{1}{2}W \operatorname{sinc}(fW) (e^{-j\pi fW} - e^{-j3\pi fW}),$$

d'où

$$|P(f)|^2 = \frac{W^2}{4} \operatorname{sinc}^2(fW) |1 - e^{-j2\pi fW}|^2$$

Or

$$|1 - e^{-j2\pi fW}|^2 = 4 \operatorname{sinc}^2(fW)$$

Alors

$$|P(f)|^2 = W^2 \operatorname{sinc}^2(fW) \sin^2(\pi fW).$$

Et donc finalement on obtient la formule de la DSP donnée par

$$\boxed{\Gamma_{sl}(f) = \frac{1}{4} \delta(f) + \frac{1}{T_s} W^2 \operatorname{sinc}^2(fW) \sin^2(\pi fW) = \frac{1}{4} \delta(f) + \frac{1}{T_s} \frac{\sin^4(\pi f \frac{T_s}{2})}{\pi^2 f^2}.}$$

### 2.2.2 Estimation et Analyse de la DSP par la méthode de Welch

**Objectif :** obtenir une estimation non paramétrique de la Densité Spectrale de Puissance (DSP) du signal  $s_l(t)$  en utilisant l'algorithme du périodogramme de Welch, qui consiste à découper le signal  $x[n]$  en  $L$  segments sans chevauchement, à calculer le périodogramme de chaque segment, et à moyennager ces périodogrammes pour réduire la variance de l'estimation de la DSP.

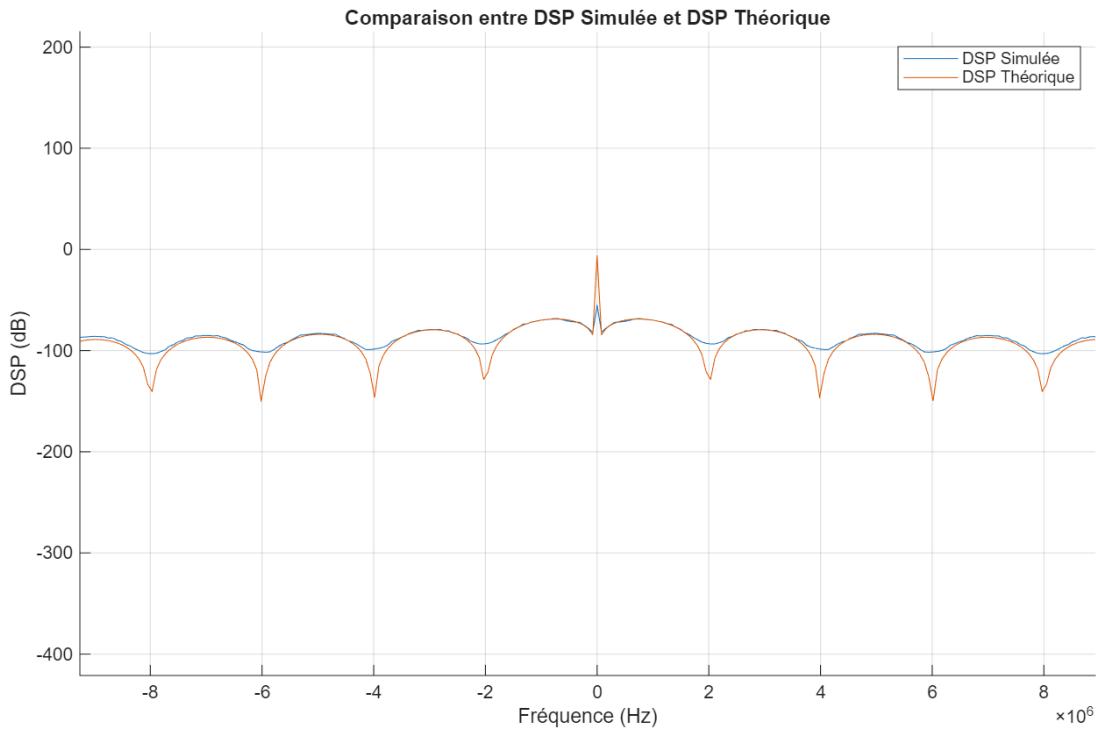


FIGURE 6 – Comparaison de la Densité Spectrale de Puissance (DSP) estimée par l'algorithme de Welch et de la DSP théorique du signal  $s_l(t)$ .

### Analysis

La Figure 6 présente la comparaison entre la densité spectrale de puissance (DSP) simulée à l'aide de la méthode de Welch et la DSP théorique du signal  $s_l(t)$ . On observe une très bonne concordance entre les deux courbes sur l'ensemble du spectre, confirmant la validité de l'expression analytique de  $\Gamma_{s_l}(f)$ .

La forme périodique de la DSP résulte de la structure de la modulation PPM, dont le spectre est caractérisé par des lobes principaux et secondaires liés à la durée  $T_s$  des symboles. On retrouve la structure spectrale attendue : un pic en  $f=0$  dû à la composante moyenne du signal, de légères différences en amplitude peuvent apparaître aux fréquences élevées en raison de la taille finie du signal et du fenêtrage implicite du périodogramme, mais la cohérence globale confirme la justesse de l'implémentation numérique.

### 2.3 Tâche 3 : Algorithmes de codage et de décodage de canal

Le standard ADS-B intègre un **codage de redondance cyclique (CRC)** pour la détection d'erreurs de transmission. Ce type de codage consiste à ajouter, à la fin de chaque message binaire, un ensemble de bits de redondance calculés à partir d'un **polynôme générateur**.

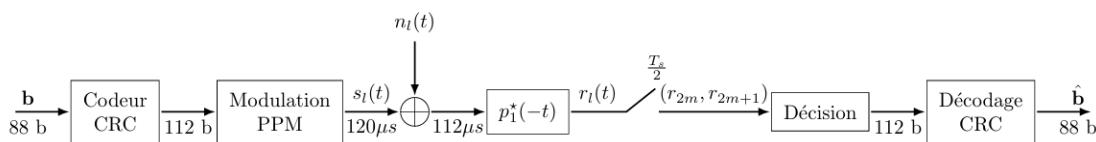


FIGURE 7 – Chaîne de communication avec codage et décodage CRC-24.

**Principe du codage CRC.** Le codage CRC repose sur une représentation polynomiale des mots binaires. Un message binaire de  $k$  bits est noté :

$$M = [m_{k-1} \ m_{k-2} \ \dots \ m_1 \ m_0],$$

et interprété comme un polynôme à coefficients binaires :

$$M(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \dots + m_1x + m_0.$$

On choisit un polynôme générateur  $p(x)$  de degré  $r$ . Le codeur multiplie  $M(x)$  par  $x^r$  (ce qui ajoute  $r$  zéros à la fin du message) puis effectue la division polynomiale par  $p(x)$ . Le reste de cette division, noté  $R(x)$ , constitue les bits de redondance :

$$R(x) = \text{reste}[x^r M(x) / p(x)].$$

Le mot codé émis est alors :

$$C(x) = x^r M(x) + R(x),$$

de sorte que  $C(x)$  est exactement divisible par  $p(x)$ .

À la réception, on vérifie si le mot reçu  $C'(x)$  est divisible par  $p(x)$  :

$$C'(x) = Q(x)p(x) + R'(x).$$

Si  $R'(x) = 0$ , le message est valide ; sinon, une ou plusieurs erreurs de transmission ont été détectées.

**Polynôme générateur utilisé :** Dans notre simulation, le polynôme générateur du codeur CRC est défini par :

$$p(x) = x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{10} + x^3 + 1.$$

Ce polynôme est celui spécifié dans la norme ADS-B , il possède un degré  $r = 24$  : le code CRC ajoute donc 24 bits de redondance à chaque trame, soit un taux de codage effectif :

$$R = \frac{k}{k+24}.$$

**Propriétés du CRC-24.** Le CRC-24 utilisé par l'ADS-B est capable de détecter :

- toutes les erreurs affectant un nombre impair de bits ;
- toutes les erreurs par rafale de longueur inférieure ou égale à 24 bits (puisque'un polynôme d'erreur de degré  $< 24$  ne peut être divisible par  $p(x)$ )

Il offre donc une très bonne capacité de détection pour un surcoût binaire modéré, et s'adapte bien aux longueurs de trames du protocole ADS-B (112 bits utiles + 24 bits de CRC).

### Implémentation et validation MATLAB

Les objets `comm.CRCGenerator` et `comm.CRCDetector` sont utilisés pour générer et vérifier les bits de redondance selon le polynôme spécifié précédemment.

La simulation consiste à :

- générer un mot aléatoire de 88 bits d'information ;
- ajouter 24 bits de redondance pour former une trame de 112 bits ;
- modular cette trame en PPM, puis appliquer un filtrage adapté et une décision bit-à-bit ;
- vérifier, en réception, si le mot reçu est divisible par le polynôme générateur  $p(x)$ .

Lorsque la transmission s'effectue sans bruit, le décodeur indique :

`aucune erreur détectée`  $\Rightarrow$  message validé.

Une altération d'un ou de plusieurs bits entraîne, dans la grande majorité des cas, la détection immédiate d'erreur ,ce qui garantit la fiabilité des trames transmises dans le système ADS-B.

## 2.4 Tâche 4 : Synchronisation en temps

On introduit désormais les **désynchronisations temporelle et fréquentielle** qui existent dans tout système réel de transmission.

Afin de permettre la détection précise du début de trame, un **préambule**  $s_p(t)$  connu du récepteur est ajouté en tête du signal émis. Ce motif, composé d'une alternance régulière de moitiés de symboles « 1 » et « 0 », sert de signature de synchronisation et facilite l'estimation du décalage temporel  $\delta_t$ .

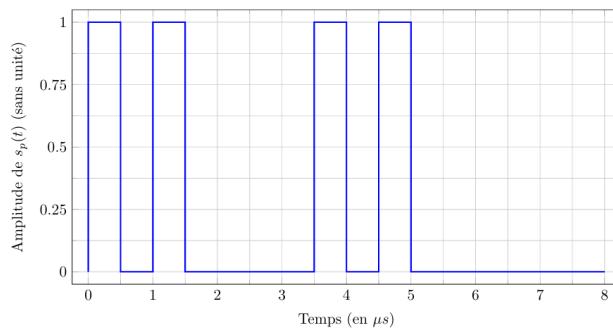


FIGURE 8 – Structure du préambule ADS-B utilisé pour la synchronisation.

Le choix de ce préambule n'est pas arbitraire : sa structure alternée génère un fort contraste temporel et une forme d'onde facilement identifiable par corrélation. En pratique, ce motif ne peut pas être produit accidentellement par la partie utile des données, ce qui garantit une détection non ambiguë du début de trame. De plus, sa richesse fréquentielle (due aux transitions rapides 1/0) le rend robuste face aux décalages Doppler modérés, améliorant ainsi la fiabilité de la synchronisation temporelle dans des environnements réels de transmission aéronautique.

La chaîne de communication complète est donc enrichie comme le montre la figure 9 :

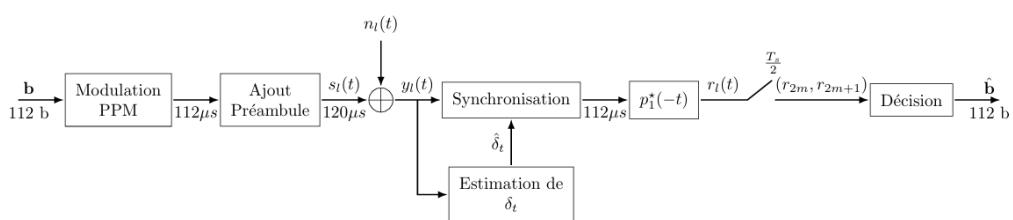


FIGURE 9 – Nouvelle chaîne de communication incluant la synchronisation temporelle et fréquentielle.

**Objectif :** introduire les désynchronisations temporelle et fréquentielle dans la chaîne de communication ADS-B, puis de concevoir un algorithme de **synchronisation temporelle** reposant sur le préambule  $s_p(t)$  des trames.

Le modèle du signal reçu en bande de base est :

$$y_l(t) = \alpha s_l(t - \delta_t) e^{-j(2\pi\delta_f t - \varphi_0)} + n_l(t),$$

où :

- $\alpha$  est un facteur d'atténuation (lié au canal),
- $\delta_t$  représente le décalage temporel (propagation),
- $\delta_f$  le décalage fréquentiel (effet Doppler),

- $\varphi_0$  le déphasage initial,
- $n_l(t)$  un bruit complexe gaussien additif.

### *Sous-tâche 1 : Estimation d'ordre de grandeur du décalage Doppler*

La fréquence porteuse des signaux ADS-B est  $f_c = 1090$  MHz. Pour un avion se déplaçant à  $v = 900$  km/h = 250 m/s, le décalage Doppler vaut :

$$\delta_f = \frac{v}{c} f_c$$

avec  $c = 3 \times 10^8$  m/s la vitesse de la lumière.

$$\Rightarrow \delta_f \approx \frac{250}{3 \times 10^8} \times 1090 \times 10^6 \approx 900 \text{ Hz.}$$

**Conclusion :** le décalage fréquentiel attendu est de l'ordre du kHz, ce qui correspond aux valeurs utilisées dans la simulation Matlab :

$$\delta_f \in [-1 \text{ kHz}, 1 \text{ kHz}].$$

### *Sous-tâche 2 : Bornes et cas d'égalité de la corrélation normalisée*

L'algorithme de synchronisation repose sur la corrélation normalisée :

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

**Démonstration :** D'après l'inégalité de Cauchy–Schwarz :

$$|\langle a, b \rangle|^2 \leq \langle a, a \rangle \langle b, b \rangle,$$

d'où :

$$|\rho(\delta'_t)| \leq 1.$$

**Cas d'égalité :** L'égalité est atteinte si et seulement si les deux signaux sont proportionnels :

$$y_l(t) = c s_p(t - \delta'_t)$$

avec  $c$  une constante

Dans ce cas,  $|\rho(\delta'_t)| = 1$ , et le maximum est obtenu pour  $\delta'_t = \delta_t$ .

### *Sous-tâche 3 : Invariance de la métrique vis-à-vis du facteur d'atténuation $\alpha$*

Considérons le signal :

$$y_l(t) = \alpha s_p(t - \delta_t) + n_l(t).$$

La corrélation normalisée s'écrit :

$$\rho(\delta'_t) = \frac{\alpha \int s_p(t - \delta_t) s_p^*(t - \delta'_t) dt}{\sqrt{|\alpha|^2 \int |s_p(t - \delta_t)|^2 dt} \sqrt{\int |s_p(t - \delta'_t)|^2 dt}}.$$

Le facteur  $|\alpha|$  apparaît à la fois au numérateur et au dénominateur ; il se simplifie.

$|\rho(\delta'_t)|$  ne dépend donc pas de  $\alpha$ .

#### *Sous-tâche 4 : Modification du test de décision pour éliminer $\alpha$ et $\varphi_0$*

En supposant que la synchronisation temporelle est correcte ( $\hat{\delta}_t = \delta_t$ ) et que  $\delta_f = 0$ , le signal en entrée du filtre adapté est :

$$y'_l(t) = \alpha e^{j\varphi_0} s_l(t) + n_l(t).$$

Le facteur complexe  $\alpha e^{j\varphi_0}$  modifie à la fois l'amplitude et la phase du signal reçu.

**Solution :** on considère la **magnitude** des corrélations pour la décision :

$$|r_{2k}| \underset{b_k=0}{\overset{b_k=1}{\gtrless}} |r_{2k+1}|.$$

L'introduction de la norme supprime donc toute dépendance à la phase et à l'amplitude du canal, et permet une détection robuste dans un contexte de transmission réelle.

#### *Sous-tâches 5 et 6 – Simulation et validation de la synchronisation*

La simulation Matlab reproduit les désynchronisations temporelle, fréquentielle et l'atténuation du modèle théorique. Un **préambule connu**  $s_p(t)$  est ajouté en tête de trame afin de permettre la détection du début de message. L'estimation du décalage temporel repose sur une **corrélation glissante normalisée** :

$$\rho(\delta'_t) = \frac{\sum_t y_l(t) s_p^*(t - \delta'_t)}{\sqrt{\sum_t |y_l(t)|^2} \sqrt{\sum_t |s_p(t)|^2}},$$

dont le maximum fournit  $\hat{\delta}_t$ .

Dans la simulation, les paramètres sont tirés aléatoirement dans les intervalles  $\delta_t \in [0, 100]$ ,  $\delta_f \in [-1, 1]$  kHz et  $\varphi_0 \in [0, 2\pi]$ . L'algorithme retrouve toujours la bonne valeur de  $\delta_t$ , résultat attendu dans un scénario idéal (sans bruit et avec un préambule bien distinct du signal utile).

#### *Sous-tâche 7 – Validation de la chaîne complète avec synchronisation*

Cette dernière étape consiste à valider l'ensemble de la chaîne de communication incluant les effets réalistes de désynchronisation et de bruit. Afin d'améliorer la stabilité de la détection, un **centrage** a été introduit dans la fonction de corrélation. Cet opération permet de supprimer la composante continue du signal qui stabilise et améliore la régularité de la courbe du TEB simulé.

Une fois le début de trame estimé, le signal est recentré temporellement, puis soumis au filtre adapté et à la prise de décision *early/late* décrite précédemment. Le taux d'erreur binaire (TEB) est alors évalué pour différentes valeurs de  $E_b/N_0$  comprises entre 0 et 10 dB, et comparé à la courbe théorique :

$$P_b = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{2N_0}} \right).$$

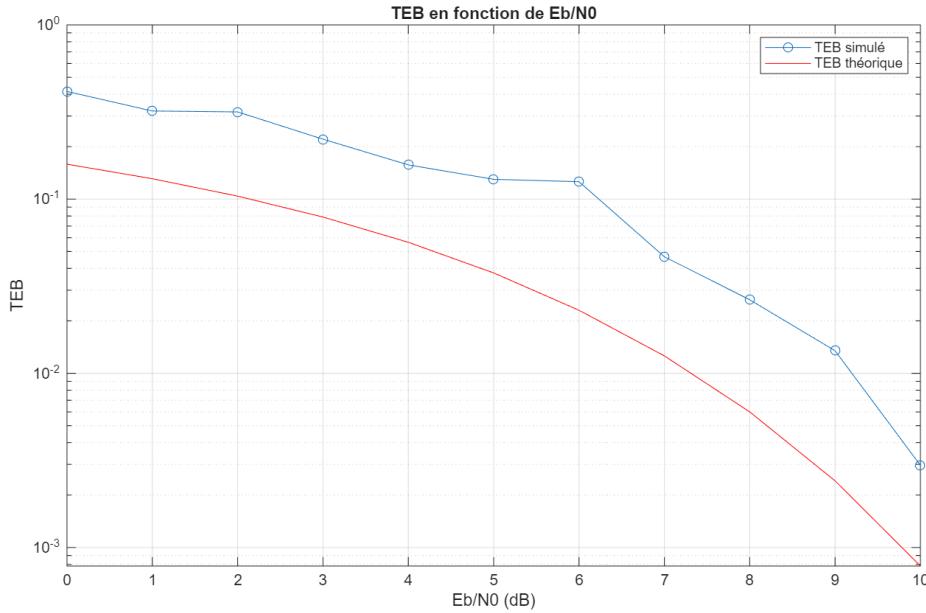


FIGURE 10 – Comparaison du TEB simulé et théorique après synchronisation.

**Analyse des résultats.** La Figure 10 montre que la courbe simulée suit globalement la tendance théorique, avec un écart (quelques décibels) dû aux effets résiduels de désynchronisation temporelle, de Doppler et de bruit de canal.

**Conclusion.** Cette simulation valide la robustesse de la méthode de synchronisation basée sur le préambule et confirme la cohérence globale du système ADS-B simulé : le récepteur parvient à estimer correctement le début de trame, à compenser le décalage temporel et à décoder les bits avec une performance proche de la limite théorique.

## 2.5 Tâche 5 : Synchronisation fréquentielle

Le décalage fréquentiel  $\delta_f$  résulte principalement de l'effet Doppler, dû au mouvement relatif entre l'émetteur et le récepteur, ainsi que des imprécisions des oscillateurs locaux. Dans le domaine complexe, ce décalage se traduit par une rotation progressive de la phase du signal reçu :

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

**Principe.** En supposant une synchronisation temporelle parfaite ( $\delta_t = 0$ ) et un signal sans bruit, la phase instantanée du signal s'écrit :

$$\phi(t) = \arg\{y_l(t)\} = -2\pi\delta_f t + \varphi_0.$$

On en déduit que la pente de cette phase est directement proportionnelle au décalage fréquentiel :

$$\frac{d\phi(t)}{dt} = -2\pi\delta_f.$$

Ainsi, une estimation de  $\delta_f$  peut être obtenue par une régression linéaire de la phase du signal sur le temps.

**Résultats et analyse.** La simulation Matlab intègre cette estimation de  $\delta_f$  dans la chaîne de réception complète. Après synchronisation temporelle et compensation fréquentielle, le signal corrigé est filtré et soumis à la détection *early/late*. Le taux d'erreur binaire est alors évalué pour différentes valeurs de  $E_b/N_0$  entre 0 et 10 dB, et comparé à la courbe théorique :

$$P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right).$$

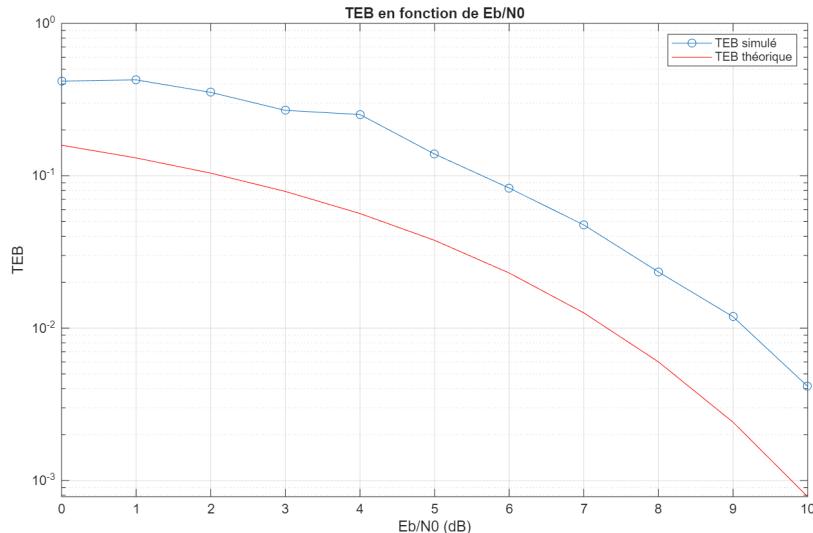


FIGURE 11 – Comparaison du TEB simulé et théorique après correction fréquentielle.

On observe que le **TEB simulé reste globalement proche de la courbe obtenue sans correction fréquentielle**. Cela indique que, dans les conditions simulées, le décalage Doppler introduit (de l'ordre du kHz) reste faible devant la fréquence d'échantillonnage, et son effet sur les performances est donc limité.

De plus, la simulation suppose une estimation de phase idéale, en **ignorant la présence de bruit sur les phases instantanées**. Le bruit ajouté étant gaussien centré (de loi  $\mathcal{N}(0, \sigma^2)$ ), il n'introduit pas de biais systématique dans l'estimation — celle-ci reste donc **non biaisée en moyenne**.

La méthode d'estimation fréquentielle reste néanmoins **cohérente et stable**.

### 3 Tâche 6 et 7 : Couche MAC ADS-B

La couche **MAC** (**Medium Access Control**) du système ADS-B définit la structure logique des trames transmises, leur identification ainsi que la protection des données grâce au **codage CRC-24** présenté dans la Tâche 3.

#### Structure générale d'une trame ADS-B

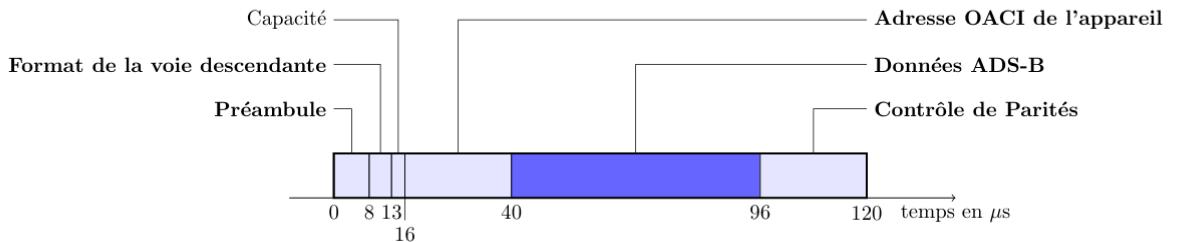


FIGURE 12 – Format général d'une trame ADS-B.

Chaque trame ADS-B comporte **120 bits**, organisés comme suit :

- **Préambule (8 µs)** : motif connu permettant la synchronisation temporelle et fréquentielle ;
- **DF (Downlink Format, 5 bits)** : indique le type de trame. Les valeurs courantes sont : DF=11 (*All-Call Reply*), DF=17 (**ADS-B**), DF=18 (*TIS-B*) et DF=19 (usage militaire) ;
- **CA (3 bits)** : sous-type de trame envoyée ;
- **AA (24 bits)** : adresse OACI unique de l'aéronef ;
- **Message utile (56 bits)** : informations ADS-B telles que l'identification, la position ou la vitesse de l'appareil ;
- **CRC-24 (24 bits)** : champ de contrôle d'erreur assurant la fiabilité des transmissions.

Les messages ADS-B de type **DF=17** sont les plus courants et véhiculent les données de position, d'identification et de vitesse. Leur structure interne dépend du **Format Type Code (FTC)**, qui précise la nature du message :

- FTC = 1–4 : identification (code d'appel *Call-Sign*) ;
- FTC = 5–8 : position au sol ;
- FTC = 9–18 : position en vol (altitude, latitude, longitude) ;
- FTC = 19 : vitesse sol et verticale.

Ces trames permettent de reconstituer en temps réel la trajectoire 3D de chaque avion, de garantir la séparation entre aéronefs et de faciliter la gestion du trafic aérien.

### Sous-tâches 2 et 3 – MATLAB : Fonction `bit2registre` et affichage de la trajectoire

Afin de valider le décodage des messages ADS-B, nous avons développé la fonction `bit2registre`, qui extrait les informations utiles contenues dans les 112 bits de données suivant le préambule.

Les trames réelles utilisées proviennent du fichier `adsb_msgs.mat`, contenant plusieurs messages échantillonnés à  $F_e = 4$  MHz et captés dans la zone de Bordeaux–Mérignac.

Le CRC-24 agit ici comme un filtre de fiabilité : seules les trames sans erreur sont retenues pour la reconstruction de trajectoire.

**Résultats et visualisation.** Les trames valides sont affichées avec leur contenu décodé (adresse, altitude, position, etc.), et les positions géographiques correspondantes sont stockées puis représentées sur une carte centrée sur Bordeaux–Mérignac.

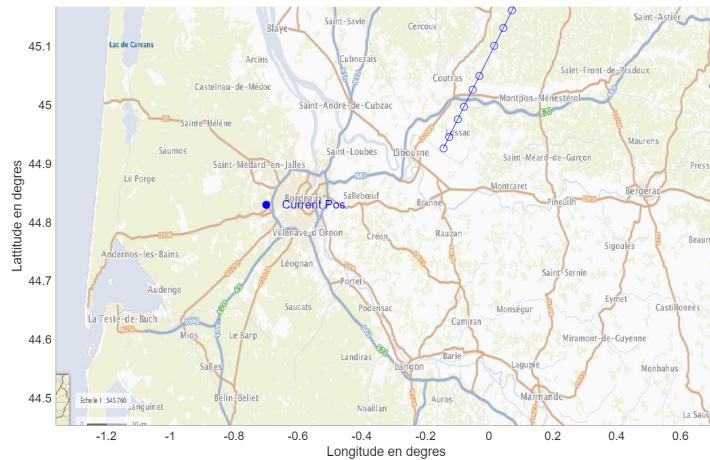


FIGURE 13 – Trajectoire reconstruite à partir des trames ADS-B décodées (zone de Bordeaux–Mérignac).

Le décodage a permis d’identifier **27 trames reçues**, dont **25 valides après vérification CRC**. Toutes proviennent du même aéronef, identifié par l’adresse OACI 3420CA . Les positions décodées indiquent un vol en croisière à environ **35 000 pieds**, évoluant progressivement de la latitude  $45.5^\circ$  à  $44.9^\circ$  et de la longitude  $0.39^\circ$  à  $-0.14^\circ$ , soit une trajectoire cohérente d’est en ouest au-dessus de la région de Bordeaux.

En complément, l’analyse des **trames de type 19** (messages de vitesse) a également été réalisée. Ces trames, extraites de plusieurs *buffers* de réception, ont permis d’estimer la **vitesse horizontale** des aéronefs. Les valeurs obtenues se situent majoritairement entre **420 et 480 nœuds (KT)**, soit environ **780 à 890 km/h**, ce qui correspond à la vitesse de croisière typique d’un avion de ligne.

## 4 Tâche 8 : Traitement de signaux réels ADS-B

Après avoir validé la chaîne de communication en simulation, cette tâche vise à exploiter des signaux ADS-B *réels* captés à l’aide d’une radio logicielle (SDR). L’objectif est d’analyser les buffers enregistrés et d’évaluer la qualité de réception par le calcul du rapport signal sur bruit (SNR).

### 4.1 Sous-tâche 1 : Estimation du SNR et repérage du préambule

Les fichiers de mesure `buffers.mat` contiennent  $N_b = 9$  buffers complexes, chacun correspondant à une durée de 0.5 s d’acquisition à une fréquence d’échantillonnage  $F_e = 4$  MHz. Chaque buffer peut contenir une ou plusieurs trames ADS-B, noyées dans le bruit de fond.

**Principe.** Pour chaque buffer, on calcule l’énergie instantanée du signal reçu :

$$E[k] = |y[k]|^2.$$

Les trames ADS-B apparaissent sous forme de pics d’énergie nettement supérieurs au bruit moyen. Le repérage du préambule repose sur une corrélation normalisée entre cette énergie et le motif connu  $s_p$  du préambule.

Dans la partie théorique, la synchronisation temporelle était définie de manière continue par :

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

Dans le cas des signaux réels acquis par SDR, on a remplacé cette intégrale par une **corrélation glissante discrète** calculée sur les échantillons  $E[k]$ .

### Implémentation :

- `synchro_tmp_adapte()` affine la détection du préambule de la nature discrète du signal et en supprimant les doublons trop proches (séparés de moins d'une trame de  $112 \times F_s$  échantillons) et ne garde que les maxima dépassant un seuil de corrélation fixé à 0.7.

**Résultats sur le buffer 2.** La figure 14 illustre la partie réelle du signal reçu, son énergie instantanée et le spectre estimé par la méthode de Welch. Une seule zone, autour de  $3.3 \times 10^5$  échantillons, présente une énergie significative, correspondant à la trame détectée.

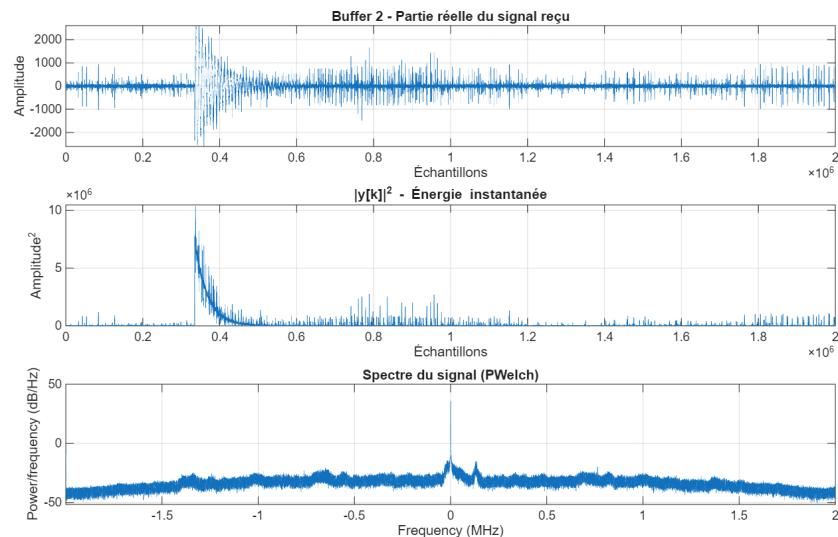


FIGURE 14 – Analyse du buffer 2 : partie réelle du signal, énergie instantanée et spectre (méthode de Welch).

Le repérage du préambule est illustré sur la figure 15. La ligne rouge indique la position estimée  $\hat{\delta}_t$ , correspondant à la zone de plus forte corrélation. On obtient un décalage temporel estimé de  $\hat{\delta}_t = 335\,091$  échantillons et un rapport signal sur bruit de  $SNR = 34.55$  dB.

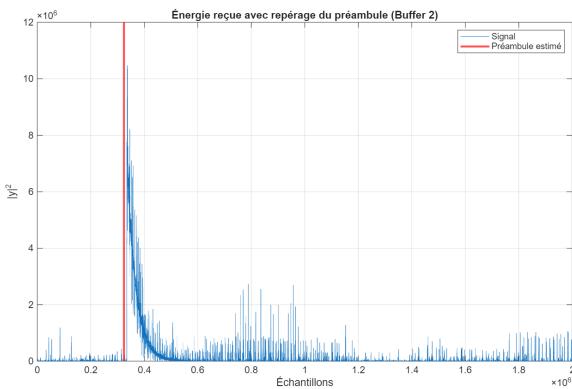


FIGURE 15 – Repérage du préambule sur l'énergie reçue (buffer 2). La ligne rouge correspond à la position estimée  $\hat{\delta}_t$ .

**Interprétation.** L'analyse du buffer 2 montre une détection nette du préambule et un SNR élevé ( $\approx 35$  dB), confirmant la présence d'une trame ADS-B bien reçue dans ce segment de signal. La position estimée du préambule ( $\hat{\delta}_t = 335,091$  échantillons) coïncide parfaitement avec la zone de forte énergie visible sur la courbe, ce qui valide le bon fonctionnement de la synchronisation temporelle sur les données réelles.

#### 4.2 Sous-tâche 2 : Décodage et affichage des trajectoires détectées

Après la détection du préambule, les signaux identifiés dans les buffers ont été entièrement décodés à l'aide de la chaîne MATLAB développée précédemment. Chaque trame extraite a été filtrée, démodulée en PPM, puis analysée par la fonction `bit2register` pour récupérer les informations ADS-B (adresse, position, altitude).

**Résultats.** Le traitement des neuf buffers SDR a permis d'identifier plusieurs trames valides issues d'aéronefs distincts. Les positions associées ont permis de reconstituer plusieurs trajectoires dans la région de Bordeaux et Bergerac (figure 17).

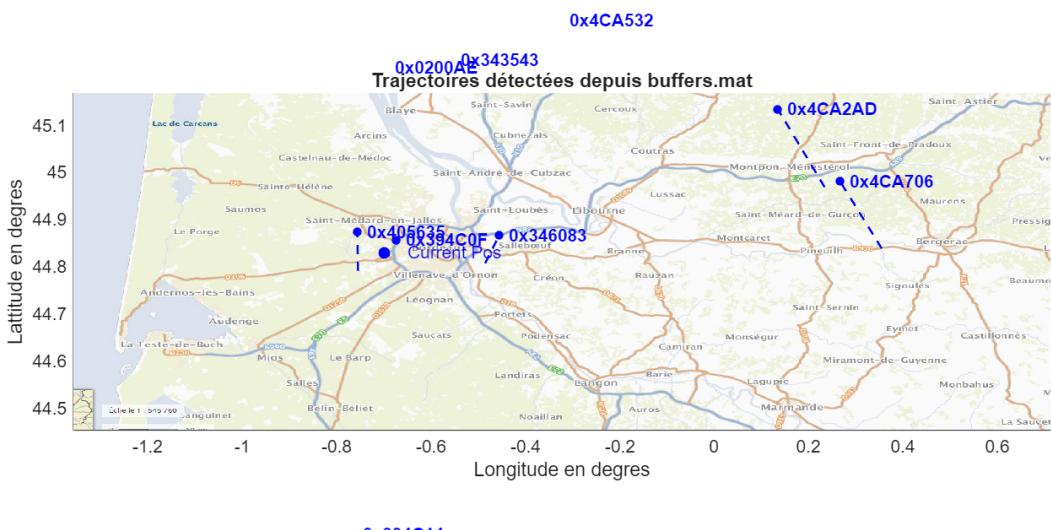


FIGURE 16 – Trajectoires détectées à partir des signaux réels ADS-B captés par SDR (zone Bordeaux–Bergerac).

### 4.3 Conclusion de la tâche 8

L'exploitation de signaux réels captés par radio logicielle a confirmé la robustesse des algorithmes de synchronisation, de démodulation et de décodage développés. Malgré la présence de bruit et d'effets de canal, plusieurs aéronefs ont pu être identifiés et localisés, validant ainsi le bon fonctionnement de la chaîne complète de réception ADS-B.

## 5 Tâche 9 : Mise en place du temps réel

**Objectif.** L'objectif de cette tâche était de mettre en relation l'ensemble des éléments développés dans les tâches précédentes afin de réaliser un décodage en temps réel des signaux ADS-B et d'afficher les trajectoires des avions reçus par la radio logicielle.

**Réalisation.** En s'appuyant sur le code Matlab élaboré lors de la tâche 8, nous avons configuré un récepteur ADS-B en temps réel à partir des données issues de la radio logicielle. Lors de la mise en place, nous avons rencontré un léger problème : la visibilité était limitée aux avions situés au sud-est de Talence, probablement en raison de l'orientation et de la position de l'antenne.

**Résultats.** Le traitement des données issues de la radio logicielle (SDR) permet d'afficher la trajectoire des avions détectés, principalement dans la zone sud-est de Talence. Les tests ont été réalisés entre 14h et 15h30, période correspondant à un pic d'activité aérienne durant laquelle plus de quarante avions ont été identifiés simultanément. Ce résultat illustre la bonne sensibilité du récepteur ADS-B ainsi que l'efficacité du décodage en temps réel.

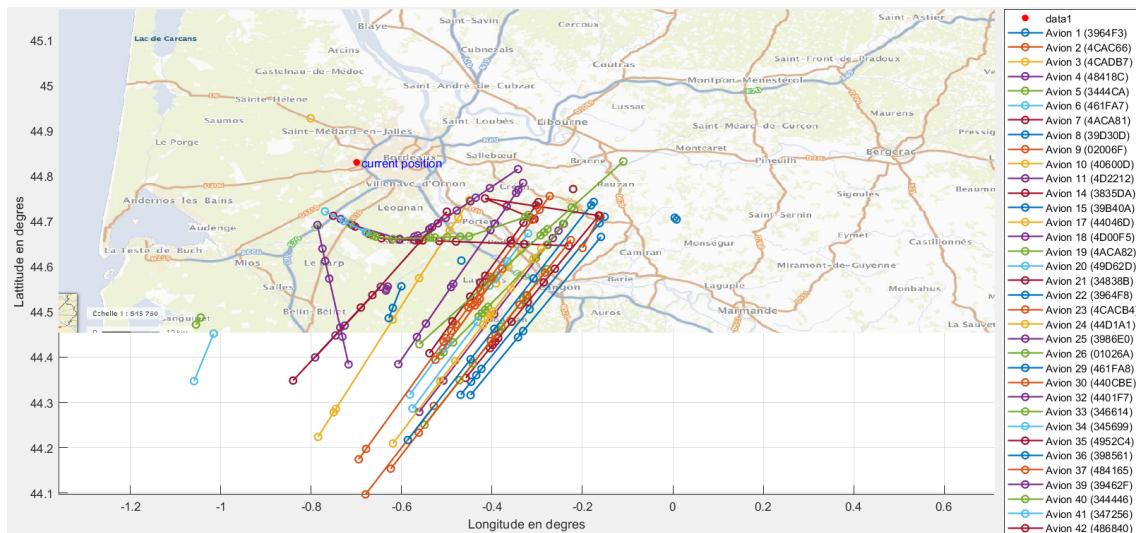


FIGURE 17 – Trajectoires détectés (Talence – 3 novembre 2025, 14h–15h30)

**Analyse comparative.** La session d'acquisition a eu lieu le **3 novembre 2025** entre **14 h 00 et 15 h 30**. Durant cette période :

- le récepteur SDR a **détecté et décodé 42 aéronefs distincts** ;
- les données de référence indiquent qu'environ **48 aéronefs** sont passés au-dessus de la zone sud-est de Talence , selon les données publiques de surveillance aérienne

**Interprétation.** L'écart observé ( $\sim 10\%$ ) peut s'expliquer par :

- la **perte de certaines trames ADS-B** dues au bruit radiofréquence ou à la superposition de transmissions simultanées ;
- la **variabilité du SNR** en fonction de la distance et de l'altitude des appareils.

Malgré ces limitations, le taux de détection reste très satisfaisant : environ **90 % du trafic réel** a été capté et correctement décodé, confirmant la validité du système en conditions réelles.

## 6 Conclusion et perspectives

Le projet a permis de réaliser une chaîne complète de réception ADS-B, de la simulation jusqu'au traitement de signaux réels. Les étapes de synchronisation, démodulation et décodage ont été validées expérimentalement, permettant l'identification et la localisation de plusieurs aéronefs à partir de données SDR.

Pour aller plus loin, il serait intéressant :

- d'affiner la détection en présence de fort bruit ou de collisions de trames ;
- d'intégrer une correction de fréquence prenant en compte les effets Doppler (important) et le déphasage introduit par le bruit ;
- de développer une interface graphique interactive de suivi en temps réel des aéronefs détectés.