



TELECOMMUNICATIONS  
TS229  
RAPPORT

---

## Projet TS229 – Année 2021/2022

---

***Etudiants :***

Baudry Alexandre  
(abaudry10@gmail.com)  
Benjemaa Rayan  
(rayan.benjemaa@gmail.com)

***Professeurs :***

Guillaume Ferré  
Romain Tajan  
Malek Ellouze

27 octobre 2021

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Tâche 1 – Couche physique ADS-B Prise en main de la chaîne de communication ADSB</b>	<b>2</b>
2.1	Sous-tâche 1 - Théorie . . . . .	2
2.2	Sous-tâche 2/3/4 - Théorie/Matlab . . . . .	3
2.3	Sous-tâche 5 - Théorie . . . . .	4
2.4	Sous-tâche 6 . . . . .	5
<b>3</b>	<b>Tâche 2 - Couche physique ADS-B Densité spectrale de puissance</b>	<b>6</b>
3.1	Sous-tâche 1 - Théorie . . . . .	6
3.2	Sous-tâche 2 - Théorie . . . . .	7
3.3	Sous-tâche 3 - Théorie . . . . .	7
3.4	Sous-tâche 4 - Théorie . . . . .	8
3.5	Sous-tâche 5 - Matlab . . . . .	8
<b>4</b>	<b>Tâche 3 – Couche physique ADS-B Algorithmes de codage et de décodage de canal</b>	<b>9</b>
<b>5</b>	<b>Tâche 4 – Couche physique ADS-B Synchronisation en temps</b>	<b>9</b>
5.1	Sous-tâche 1 - Théorie . . . . .	9
5.2	Sous-tâche 2 - Théorie . . . . .	10
5.3	Sous-tâche 3 - Théorie . . . . .	10
5.4	Sous-tâche 4 - Matlab . . . . .	11
5.5	Sous-tâche 5 - Matlab . . . . .	11
5.6	Sous-tâche 6 - Matlab . . . . .	11
<b>6</b>	<b>Tâche 6 – Couche MAC ADS-B Implémentation de la couche MAC</b>	<b>12</b>
6.1	Sous-tâche 1 - Théorie . . . . .	12
6.2	Sous-tâche 2 - Matlab . . . . .	12
<b>7</b>	<b>Tâche 7 – Couche MAC ADS-B Implémentation de la couche MAC</b>	<b>12</b>
<b>8</b>	<b>Tâche 8 – Application Traitement de signaux réels</b>	<b>13</b>
<b>9</b>	<b>Conclusion</b>	<b>13</b>

## 1 Introduction

L'objectif final de ce projet est de simuler sous Matlab un émetteur/récepteur de données ADS-B (Automatic Dependent Surveillance Broadcast). Puis d'adapter notre récepteur afin d'être en mesure d'effectuer un décodage temps réel des avions survolant l'école.

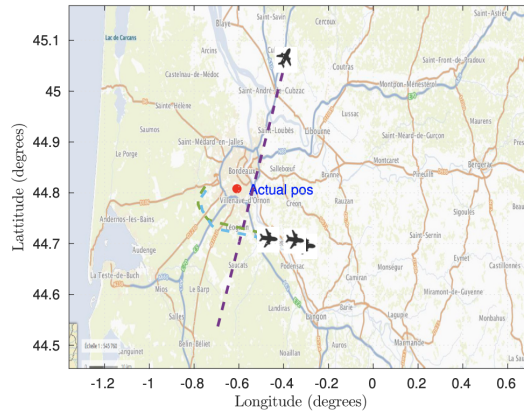


FIGURE 1 – Exemple de trajectoires décodées

## 2 Tâche 1 – Couche physique ADS-B Prise en main de la chaîne de communication ADSB

Dans cette première tâche on s'intéresse à une chaîne de communication transmettant un flux d'information binaires de type ADSB à l'aide d'une modulation dite en position d'amplitude (PPM - Pulse Position Modulation). La chaîne de communication complète à modéliser est la suivante :

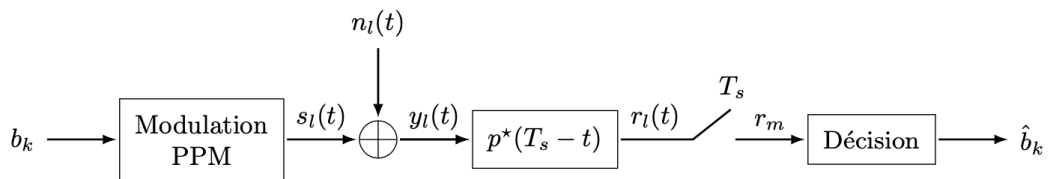


FIGURE 2 – Chaîne de communication complète

### 2.1 Sous-tâche 1 - Théorie

Dans cette première question théorique il s'agit de montrer que l'on peut réécrire le signal émis sous la forme :

$$s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) = 0$$

On repart pour cela de l'expression donnée du signal envoyé :

$$s_l(t) = \sum_{k \in \mathbb{Z}} p_{bk}(t - kt_s) = 0$$

Or  $p(t) = p_0(t) - 0.5 = -p_1(t) - 0.5$  soit  $p_{bk}(t) = (0.5 + A_k p(t))1_{[t - kT_s]}$   
on retrouve bien ainsi la formule attendue :

$$s_l(t) = 0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s) = 0$$

## 2.2 Sous-tâche 2/3/4 - Théorie/Matlab

On implémente dans cette sous-tâche la chaîne de communication précédente sous Matlab afin de visualiser l'allure des signaux  $s_l(t)$ ,  $r_l(t)$  et  $r_m(t)$ . On fait dans cette partie certaines hypothèses :

- Le bruit est nul
- Le signal binaire émis vaut  $[1,0,0,1,0]$
- Les signaux sont causaux

Après la modulation PPM on obtient le signal  $s_l(t)$  suivant :

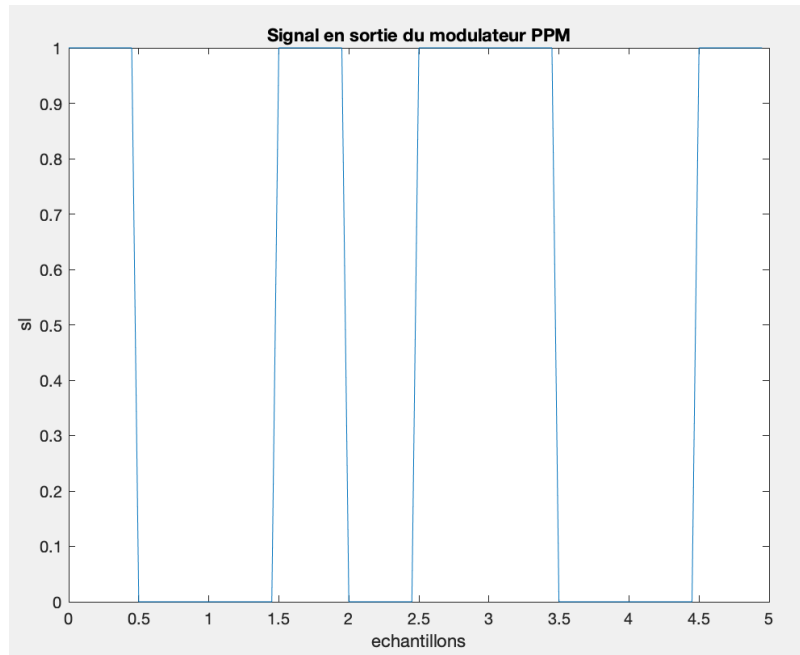


FIGURE 3 – Signal en sortie de modulation PPM

L'hypothèse du bruit nul nous permet d'affirmer que  $y_l(t) = s_l(t)$ . Puis par convolution avec le filtre adapté  $p * (-t)$  on obtient le signal  $r_l(t)$ . Enfin après échantillonnage au temps  $T_s$  on obtient le signal  $r_m(t)$ .

On représente ces signaux ci-dessous :

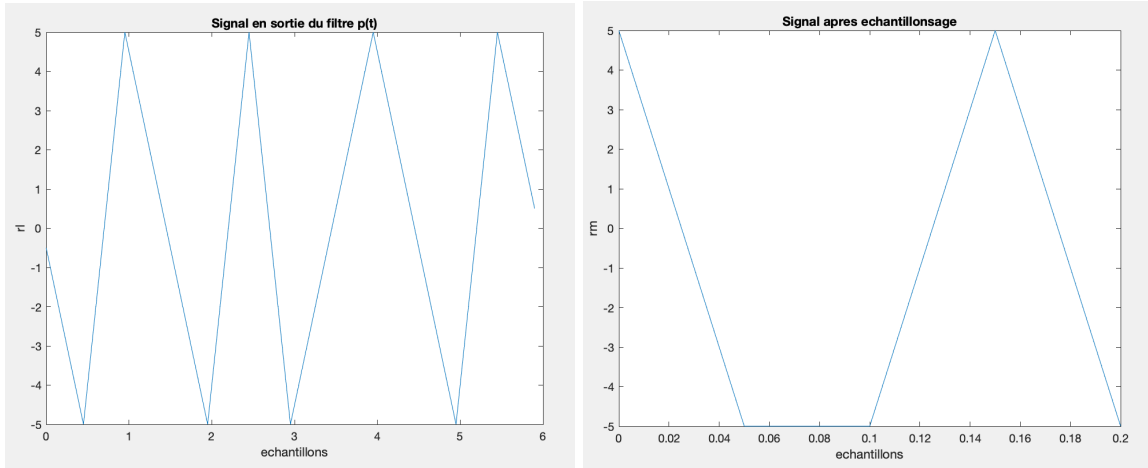


FIGURE 4 – Signaux  $r_l(t)$  et  $r_m(t)$

En fin de chaîne de communication le bloc de décision a pour rôle d'estimer le bit en sortie à partir du signal  $r_m(t)$ . Dans le cas de cette chaîne de communication il est de la forme :

$$\hat{b}_k(t) = \begin{cases} 1 & \text{si } r_m(t) > 0 \\ 0 & \text{sinon.} \end{cases}$$

Les bits estimés en fin de chaîne de communication sont bien identiques à ceux émis en absence de bruit, notre chaîne de communication fonctionne correctement .

### 2.3 Sous-tâche 5 - Théorie

On cherche à présent à calculer la probabilité d'erreur binaire  $P_b$  pour la modulation PPM et le récepteur proposé. La constellation utilisée pour notre mapping est  $(-1; 1)$ . On note  $S_k$  le k-ieme symbole décodé. La probabilité d'erreur symbole  $P_s$  est alors définie par :

$$P_s = \sum_{i \in \pm 1} P(A_k = i \cap A_k \neq i)$$

soit d'après la formule des probabilités conditionnelles :

$$P_s = P(A_k = +1)P(\text{Erreur}|A_k = +1) + P(A_k = -1)P(\text{Erreur}|A_k = -1)$$

or les  $b_k$  sont indépendants et distribués uniformément, et d'après le mapping d'une modulation PPM les symboles sont alors également distribués uniformément. On a donc :

$$P(A_k = +1) = P(A_k = -1) = \frac{1}{2}$$

d'où :

$$P_s = \frac{1}{2}P(\text{Erreur}|A_k = +1) + \frac{1}{2}P(\text{Erreur}|A_k = -1)$$

$$P_s = \frac{1}{2}(P(\text{decider} + 1|A_k = +1) + P(\text{decider} - 1|A_k = -1))$$

il faut donc déterminer  $P(\text{decider} + 1|A_k = +1)$  et  $P(\text{decider} - 1|A_k = -1)$ .  
On repart pour cela de la fonction  $r_m[n]$  :

$$r_m[n] = S_n R_g[0] + n'_l[n] \text{ (filtre de Niquist donc terme d'IES nul.)}$$

Avec  $n'_l[n]$  l'expression du bruit après convolution suivant une loi Normale de paramètres  $(0, E_{ga}\sigma_{nl}^2)$ . Nos deux symboles étant  $(-1, 1)$ , notre seuil de decision est  $\gamma = 0$ . on a donc :

$$P(\text{decider} + 1|A_k = +1) = P(\text{decider} - 1|A_k = -1) = \int_{\gamma=0}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{x+R_g[0]}{\sigma})^2} dx$$

On simplifie cette notation en faisant appel à la fonction :  $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$

On obtient alors par le changement de variable  $z = \frac{x+R_g[0]}{\sigma}$

$$P_s = Q\left(\frac{R_g[0]}{\sigma}\right)$$

avec :  $R_g[0] = E_b$  et  $\sigma = \sqrt{E_{ga}\sigma_{nl}^2}$ , où  $E_{ga} = E_b$  et  $\sigma_{nl} = \sqrt{\frac{N_0}{2}}$

$$\text{soit : } Q\left(\frac{R_g[0]}{\sigma}\right) = Q\left(\sqrt{2}\sqrt{\frac{E_b}{N_0}}\right)$$

En introduisant la fonction  $\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt$  on obtient :  $P_s = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$

Chaque symbole étant constitué d'un seul bit on a donc  $n_b = 1$  soit  $P_b = \frac{P_s}{n_b} = P_s$  et :

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

## 2.4 Sous-tâche 6

On génère à présent les bits aléatoirement et suivant une loi discrète uniforme, la taille du paquet d'information binaire est fixée et égale à  $N_b = 1000$  bits. On cherche maintenant à obtenir le Taux d'Erreur Binaire (TEB) en fonction du rapport signal à bruit  $\frac{E_b}{N_0}$  pour des valeurs allant de 0dB à 10dB. On implémente alors notre algorithme qui comptera le nombre de bits erronés par rapport au nombre de bits correctement reçus jusqu'à ce que le nombre d'erreur totale atteigne 100 (afin que notre TEB soit légitime).

Les courbes obtenues sont les suivantes, où  $P_b$  correspond à la probabilité d'erreur binaire théorique

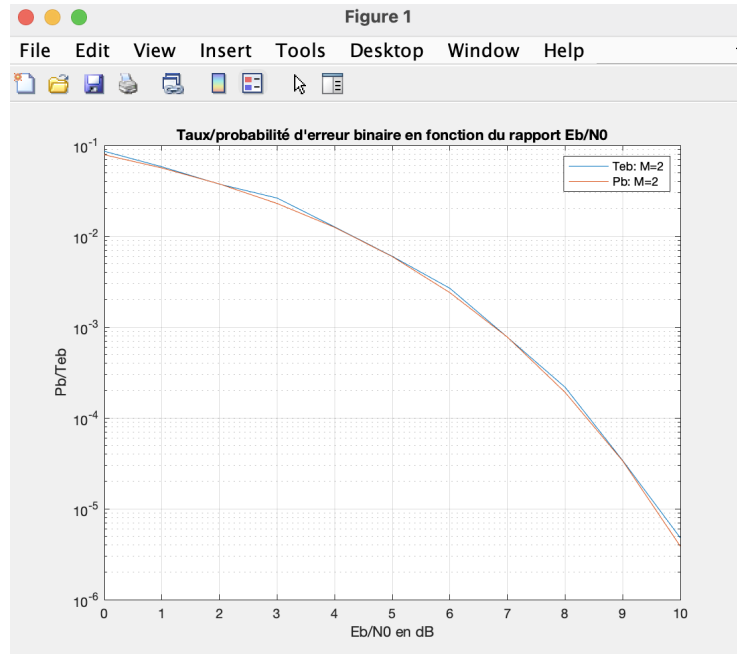


FIGURE 5 – TEB et  $P_b$  en fonction du rapport signal sur bruit

On obtient bien la superposition attendue de nos deux courbes.

### 3 Tâche 2 - Couche physique ADS-B Densité spectrale de puissance

Dans cette partie on se propose de calculer la densité spectrale de puissance d'un pont de vue théorique dans un premier temps puis expérimentalement à l'aide de Matlab.

#### 3.1 Sous-tâche 1 - Théorie

Dans cette première sous tâche nous allons calculer le moment d'ordre 1 du signal  $s_l(t)$ ,  $m_{s_l(t)} = \mathbb{E}[s_l(t)]$  et montrer que celui ci ne dépend pas de  $t$ .

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

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

Or ici  $A_k = \pm 1$  avec des probabilités identiques on a donc :

$$\mathbb{E}[A_k] = \sum_{A_k \in \pm 1} A_k p(A_k) = \frac{1}{2} - \frac{1}{2} = 0$$

On a donc :

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

Le moment d'ordre 1 ne depend donc pas de  $t$ .

### 3.2 Sous-tâche 2 - Théorie

Nous allons à présent calculer la fonction d'autocorrélation du signal  $S_l(t)$  :

$$R_{S_l}(t, \tau) = \mathbb{E}[S_l(t)S_l^*(t + \tau)]$$

$$R_{S_l}(t, \tau) = \mathbb{E}[(0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s))(0.5 + \sum_{l \in \mathbb{Z}} A_l p(t + \tau - lT_s))]$$

$$R_{S_l}(t, \tau) = 0.25 + \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} \mathbb{E}[A_k A_l] p(t - kT_s) p(t + \tau - lT_s)$$

Or : si  $k \neq l$ ,  $\mathbb{E}[A_k A_l] = \mathbb{E}[A_k] \mathbb{E}[A_l] = 0$  (par indépendance des  $A_k$ )

Et : si  $k = l$ ,  $\mathbb{E}[A_k A_l] = \mathbb{E}[A_k^2] = 1$

On peut donc simplifier notre double somme en une unique somme sur les indices  $k$  :

$$R_{S_l}(t, \tau) = 0.25 + \sum_{k \in \mathbb{Z}} p(t - kT_s) p(t + \tau - lT_s)$$

### 3.3 Sous-tâche 3 - Théorie

$s_l(t)$  étant cyclo-stationnaire de période de cyclostationarité  $T_s$ , On peut alors calculer sa Densité Spectrale de Puissance en calculant la transformée de Fourier de l'autocorrélation moyennée du signal  $s_l(t)$  calculons donc :

$$\hat{R}_{s_l}(\tau) = \frac{1}{T_s} \int_0^{T_s} R_{s_l}(t, \tau) dt$$

$$\hat{R}_{s_l}(\tau) = \frac{1}{T_s} \int_0^{T_s} 0.25 + \sum_{k \in \mathbb{Z}} p(t - kT_s) p(t + \tau - lT_s) dt$$

$$\hat{R}_{s_l}(\tau) = 0.25 + \frac{1}{T_s} \int_0^{T_s} p(t - kT_s) p(t + \tau - lT_s) dt$$

$$\hat{R}_{s_l}(\tau) = 0.25 + \frac{R_p(\tau)}{T_s}$$



### 3.4 Sous-tâche 4 - Théorie

On en déduit alors la DSP théorique de  $s_l(t)$  :

$$\Gamma_{s_l}(f) = TF(\hat{R}_{s_l}(\tau))$$

$$\Gamma_{s_l}(f) = \int_0^{T_s} \hat{R}_{s_l}(\tau) e^{-j2\pi\tau} d\tau$$

En repartant de l'expression de  $\hat{R}_{s_l}(\tau)$  on a :

$$\Gamma_{s_l}(f) = \frac{\delta(f)}{4} + \frac{|P(f)|^2}{T_s} \quad (1)$$

Or,

$$p(t) = -\Pi_{\frac{T_s}{2}}(t - \frac{T_s}{4}) + \Pi_{\frac{T_s}{2}}(t - \frac{3T_s}{4}) \quad (2)$$

D'où,

$$P(f) = \frac{T_s}{2} \text{sinc}(\frac{T_s}{2}f) (\exp -j2\pi f \frac{3T_s}{4} - \exp -j2\pi f \frac{T_s}{4}) \quad (3)$$

$$|P(f)|^2 = T_s^2 \text{sinc}^2(\frac{T_s}{2}f) \text{sinc}^2(2\pi f \frac{T_s}{4}) = \frac{T_s^4 (\pi f)^2}{4} \text{sinc}^4(\frac{T_s}{2}f) \quad (4)$$

On obtient donc que :

$$\Gamma_{s_l}(f) = \frac{\delta(f)}{4} + \frac{T_s^3 (\pi f)^2}{16} \text{sinc}^4(\frac{T_s}{2}f) \quad (5)$$

### 3.5 Sous-tâche 5 - Matlab

Dans cette sous-tâche nous allons implémenter un algorithme basé sur celui du périodogramme de Welch permettant d'obtenir une estimation de la DSP de  $s_l(t)$ . Pour une meilleure estimation nous avons utilisé un chevauchement de 66%. On obtient alors les courbes suivantes qui se superposent bien, notre estimation est donc correcte.

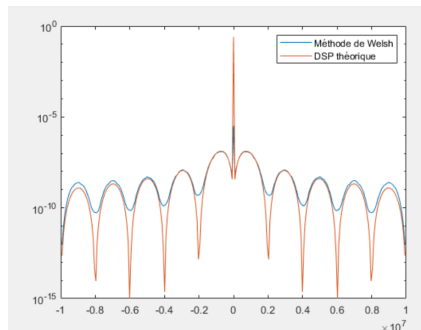


FIGURE 6 – TEB et  $P_b$  en fonction du rapport signal sur bruit

## 4 Tâche 3 – Couche physique ADS-B Algorithmes de codage et de décodage de canal

Pour cette tâche ci on considère à présent la chaîne de communication suivante :

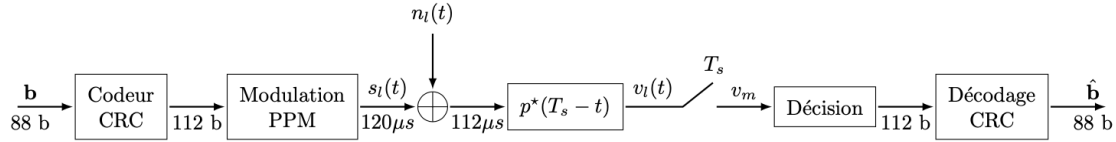


FIGURE 7 – Chaîne de communication tâche 3

Au cours de cette tâche, nous avons implémenté des algorithmes de codage et de décodage de Canal toujours dans le cadre de la transmission des signaux ADS-B. En effet, on nous indique dans cette partie la présence de 24 bits de parités sont ajoutés dans la trame issus d'un code CRC. Nous avons pu utiliser cette méthode de détection d'erreur à l'aide de fonctions Matlab prévu à cet effet. Nous tenons à préciser que nous avons utilisé des fonctions en plus de celles données dans le sujet (crc.generator et generate) afin de pouvoir utiliser correctement la fonction detect conseillée. Avant d'intégrer le codeur et décodeur de canal dans la chaîne de communication nous avons vérifié son bon fonctionnement de notre algorithme à l'aide du code de Test fourni par les professeurs.

## 5 Tâche 4 – Couche physique ADS-B Synchronisation en temps

Dans cette tâche l'objectif est de prendre à présent en compte le délai de propagation  $\delta(t)$ , et l'effet Doppler introduit par le mouvement de l'avion ainsi que les défauts d'oscillateurs locaux qui introduisent un décalage en fréquence  $\delta(f)$ . Pour cela on utilise le modèle en bande de base suivant :

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

On cherche alors à mettre en place un algorithme de synchronisation temporel permettant de compenser ce délai.

### 5.1 Sous-tâche 1 - Théorie

Dans cette sous tâche on s'intéresse à l'ordre de grandeur du décalage de fréquence Doppler d'un avion se déplaçant à 900km/h. On utilise pour cela la formule suivante :

$$|f_r - f_e| = f_e \left( \frac{c}{c + v_e} - 1 \right) \quad (6)$$

avec  $c$ , la vitesse de la lumière,  $v_e$ , la vitesse de l'avion,  $f_r$ , la fréquence reçue et  $f_e$ , la fréquence émise.

On retrouve  $|f_r - f_e| = 908,3 \text{ Hz}$ , soit un décalage d'environ 1kHz.

## 5.2 Sous-tâche 2 - Théorie

On considère désormais l'architecture de communication présentée sur la figure 7. On cherche à déterminer  $|y_l(t)|^2$  :

On montre que  $|y_l|^2 = y_l(t)y_l^*(t) = s_l^2(t - \delta_t) + z_l(t)$  :

$$|y_l(t)|^2 = s_l^2(t - \delta_t) + s_l(t - \delta_t) \frac{1}{2} (Re(n_l(t)) \cos(2\pi\delta_f t) + Im(n_l(t)) \sin(2\pi\delta_f t)) + |n_l(t)|^2 \quad (7)$$

Par identification on observe que  $z_l(t)$  n'est donc pas un bruit blanc gaussien, de plus il dépend de  $s_l(t)$ . Prendre le carré du module de l'enveloppe complexe du signal reçu permet de se séparer du décalage en fréquence puisqu'il est apparenté à un bruit.

## 5.3 Sous-tâche 3 - Théorie

On considère désormais l'architecture de communication suivante :

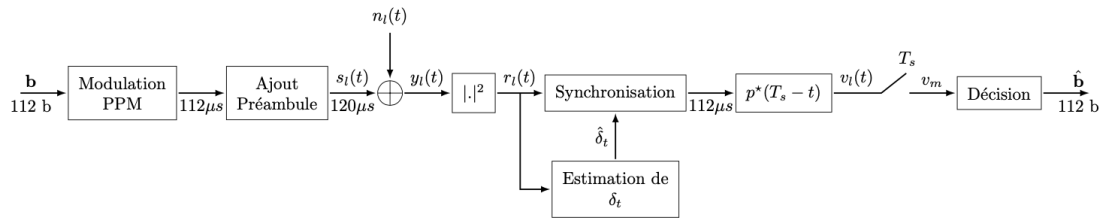


FIGURE 8 – Chaîne de communication tâche 4

On utilise l'intercorrélacion comme méthode pour effectuer la synchronisation temps/fréquence du signal. Cela se base sur l'ajout en début de trame d'un préambule  $s_p(t)$ . La synchronisation se fait en cherchant le maximum de la corrélation  $\rho(\delta'_t)$ . L'estimation de  $\delta_t$  permet d'effectuer la synchronisation temporelle.

$$y_l = s_p(t - \delta_t) \exp -j2\pi\delta_f t + n_l(t) \text{ pour } t \in [\delta_t, \delta_t + T_p] \quad (8)$$

La synchronisation se fait ensuite en cherchant le maximum de la corrélation suivante :

$$p(\delta'_t) = \frac{\int_{\delta'_t}^{\delta'_t + T_p} r_l(t) s_p * (t - \delta'_t) dt}{\sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |s_l(t)|^2 dt}} \quad (9)$$

On montre grâce à l'inégalité de Cauchy-Schwarz que  $|p(\delta_t)| \leq 1$  pour tout  $\delta'_t$  :

$$\left| \int_{\delta_t}^{\delta_t + T_p} r_l(t) s_p * (t - \delta'_t) dt \right| \leq \sqrt{\int_{\delta_t}^{\delta_t + T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta_t + T_p} |s_l(t - \delta'_t)|^2 dt} \quad (10)$$

$$\left| \int_{\delta_t}^{\delta_t' + T_p} r_l(t) s_p * (t - \delta_t') dt \right| \leq \sqrt{\int_{\delta_t}^{\delta_t' + T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta_t' + T_p} |s_l(t)|^2 dt} \quad (11)$$

D'où :

$$|p(\delta_t')| = \left| \frac{\int_{\delta_t}^{\delta_t' + T_p} r_l(t) s_p * (t - \delta_t') dt}{\sqrt{\int_{\delta_t}^{\delta_t' + T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta_t' + T_p} |s_l(t)|^2 dt}} \right| \leq 1 \quad (12)$$

Il y a égalité s'il y a relation de proportionnalité entre  $s_p$  et  $r_l$ , autrement dit lorsque  $s_p$  et  $r_l$  sont égaux à un facteur multiplicatif près. Ainsi une valeur de  $\rho(\delta_t)$  proche de 1 nous indiquera la présence du préambule à l'instant  $\delta_t$ .

#### 5.4 Sous-tâche 4 - Matlab

cf : *Synchronisation.m*

#### 5.5 Sous-tâche 5 - Matlab

Après implémentation du code Matlab on estime parfaitement le décalage temporel  $\delta_t$  pour notre chaîne de communication sans bruit.

#### 5.6 Sous-tâche 6 - Matlab

On génère à présent un décalage temporel aleatoire pour chaque trame émise et l'on trace l'évolution du taux d'erreur binaire en fonction du rapport  $\frac{E_b}{N_0}$ .

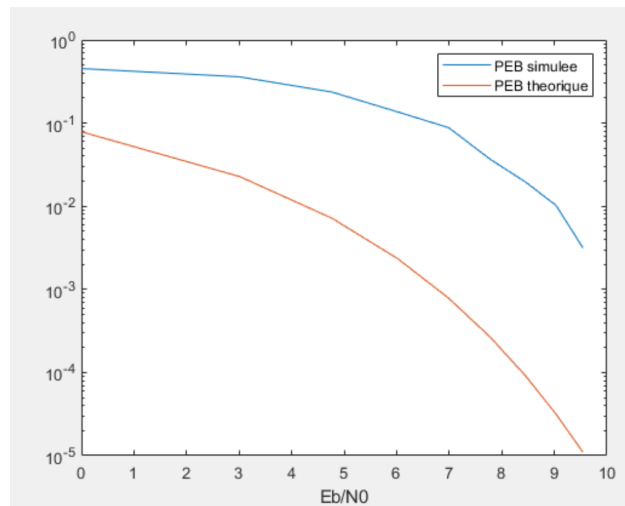


FIGURE 9 – Evolution du TEB en fonction du rapport  $\frac{E_b}{N_0}$

## 6 Tâche 6 – Couche MAC ADS-B Implémentation de la couche MAC

Cette partie traite de l'implémentation de la couche MAC. Nous avons créé une fonction qui convertit un message binaire en registre, afin de pouvoir l'utiliser lors de l'extraction de données sur les appareils concernés.

### 6.1 Sous-tâche 1 - Théorie

Les valeurs de FTC (Format Type Code) correspondant à des trames de position en vol sont celles comprises entre 9 et 18 et celles de 20 à 22. Pour les messages d'identification, ce sont les valeurs comprises entre 1 et 4.

### 6.2 Sous-tâche 2 - Matlab

Nous avons implémenté la fonction `bit2registre` qui extrait les informations du vecteur binaire et renvoie la mis à jour seulement si le CRC ne détecte pas d'erreur. Pour cela, nous avons repris les codes CRC créés à la tâche 3 et construit une structure représentant un registre qui contient les différents champs caractéristiques d'un avion. Il a fallu distinguer les différents cas en fonction de la lecture du FTC. Une lecture très attentive du sujet et des annexes fournies était requise pour réaliser cette partie et de nombreux schémas nous ont permis de réussir cette tâche dont la difficulté était la distinction des différents cas.



FIGURE 10 – Trajectoire obtenue à partir d'une trame issue fichier *adsb<sub>m</sub>sgs.mat*

## 7 Tâche 7 – Couche MAC ADS-B Implémentation de la couche MAC

Le but de cette partie est de retravailler la fonction ajoutée à la tâche précédente qui convertit un vecteur de 112 bits et renvoie un registre en y incluant de nouvelles informations. Nous avons ajouté à notre fonction `bit2registre` le cas position au sol dont la FTC est comprise entre 5 et

8 en utilisant les fonctions développées précédemment. De plus, nous avons ajouté l'information liée à la vitesse de l'avion à l'aide de la documentation technique sur ADSB ( fournie sur la page Git de notre encadrant). Pour cette partie, nous n'avions pas de données fournies pour tester et comparer nos résultats. Cependant, l'encadrant a pu valider notre raisonnement sur cette tâche.

## 8 Tâche 8 – Application Traitement de signaux réels

On a mis à notre disposition un fichier `buffers.mat` contenant 9 enregistrements réalisés par l'antenne de l'enseirb. Ces enregistrements correspondent au  $y_l$  dans notre chaîne de communication. Nous avons implémenté une fonction `trame_detect` qui nous a permis de détecter si le début d'une trame correspondait bien au préambule. Pour cela, nous nous sommes basés sur un algorithme de recherche de synchronisation temporelle par maximum de corrélation liée un seuil de la fonction d'autocorrélation fixé. En cas de succès, la fonction renvoie la trame ADSB sans préambule. Sinon, elle renvoie un tableau dont la première case correspond au résultat de la corrélation et dont les autres cases sont remplies par des zéros. On peut donc affirmer que si la somme des éléments du tableau est inférieure à 1 alors aucune trame n'a été détectée. Une fois la trame obtenue en sortie de la fonction `trame_detect`, nous avons vérifié le CRC à l'aide des algorithmes implémentés précédemment. On a ensuite appliqué la fonction `bit2registre` sur la trame afin d'actualiser le registre. De plus, nous avons initialisé différentes variables pour comptabiliser les trames non considérées.

## 9 Conclusion

Pour conclure, ce projet est sûrement l'un des plus complets que nous ayons eu à réaliser. Les premières tâches ont permis de mettre en pratique un grand nombre de notions vues en première année et de revoir une partie des aspects théoriques fondamentaux en traitement de signal. La documentation et les codes de test fournis nous ont permis d'aborder des thématiques un peu plus éloignées, mais toutefois très intéressantes. Le contexte concret de ce projet (traçage d'avion survolant l'école), nous aura permis de nous projeter sur les réelles applications des algorithmes implémentés et sur l'intérêt de la communication numérique dans la vie courante. De plus, non réalisées par faute de temps, les tâches "bonus" semblaient apporter un aspect utilitaire à notre code qui se rapprochait de plus en plus à ce que l'on peut retrouver sur des sites de tracker en ligne.