



TELECOMMUNICATIONS

TS229

RAPPORT

Simulation d'un émetteur / récepteur ADS-B et décodage temps réel à l'aide de radio logicielle

Etudiants :

Kylian Ferron
Nicolas Bousquet

Professeurs :

Romain Tajan
Guillaume Ferré
Malek Ellouze

Octobre 2020

Table des matières

1	Introduction	2
2	Tâche 1	3
2.1	Sous-tâche 1	4
2.2	Sous-tâche 2	4
2.3	Sous-tâche 3	5
2.4	Sous-tâche 4	5
2.5	Sous-tâche 5	5
2.6	Sous-tâche 6	6
3	Tâche 2	6
3.1	Sous-tâche 1	6
3.2	Sous-tâche 2	7
3.3	Sous-tâche 3	7
3.4	Sous-tâche 4	7
3.5	Sous-tâche 5	8
4	Tâche 3	8
5	Tâche 4	9
5.1	Sous-tâche 1	9
5.2	Sous-tâche 2	9
5.3	Sous-tâche 3	10
5.4	Sous-tâche 4	11
5.5	Sous-tâche 5	11
5.6	Sous-tâche 6	11
6	Tâche 6	12
6.1	Sous-tâche 1	12
6.2	Sous-tâche 2	13
6.3	Sous-tâche 3	13
7	Tâche 7	14
7.1	Sous-tâche 1	14
7.2	Sous-tâche 2	14
8	Tâche 8	14
8.1	Sous-tâche 1	14
8.2	Sous-tâche 2	14
9	Conclusion	15

1 Introduction

Le projet TS229 est un regroupement de tâches ayant pour but final d'obtenir une application équivalente aux trackers en ligne tels que flightradar24 avec un rendu final représenté ci dessous. Pour ce faire, le projet se base sur une simulation d'un émetteur/récepteur de données ADS-B (Automatic Dependent Surveillance Broadcast) estimant les positions des appareils grâce aux techniques de positionnement par satellite et les diffusant (voir plus de documentations sur le git de Monsieur TAJAN). Cette simulation s'effectue sur MATLAB au fur et à mesure de plusieurs tâches qui sont spécifiées tout au long du rapport.

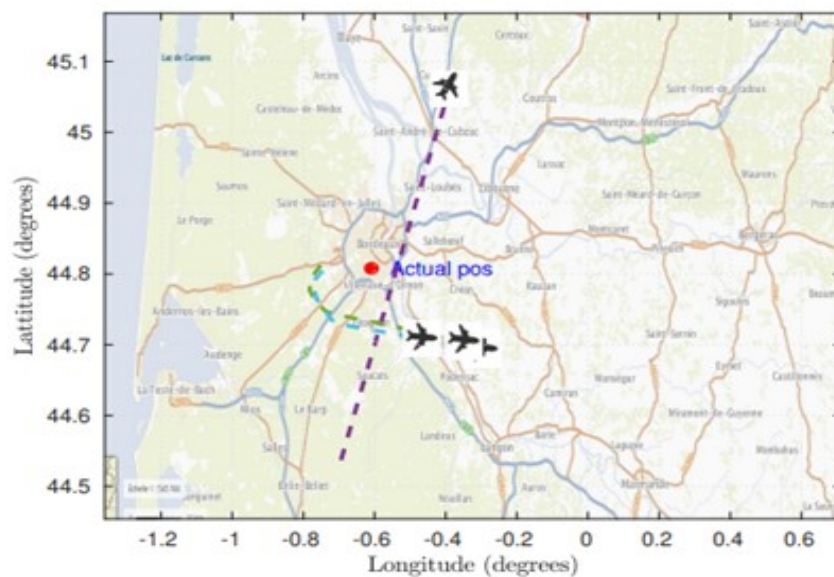


FIGURE 1 – Exemple de trajectoires décodées avec cette simulation

2 Tâche 1

Dans le cadre du standard ADS-B, les signaux transmis sont obtenus par une modulation dite en position d'amplitude (PPM - Pulse Position Modulation). Cette modulation PPM encode des informations binaires avec les impulsions suivantes :

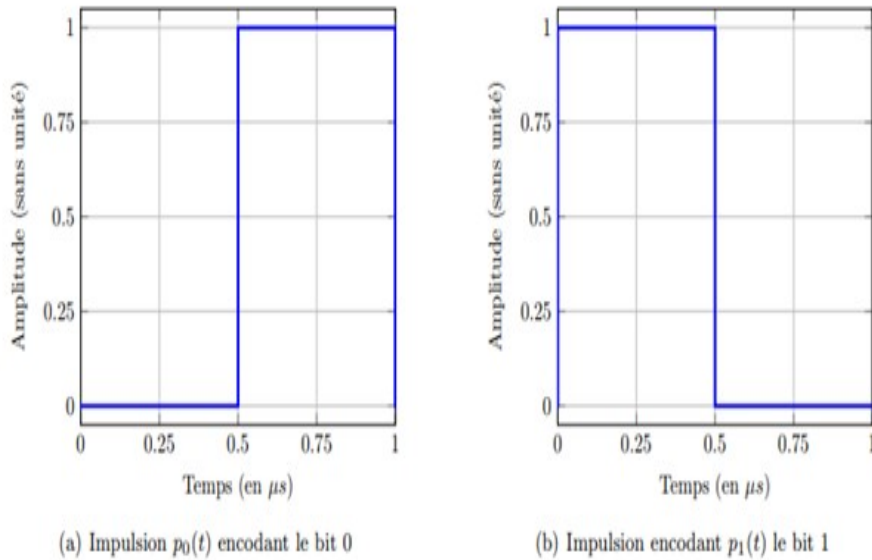


FIGURE 2 – Impulsions de base pour la modulation par position

L'enveloppe complexe du signal envoyé s'écrit donc :

$$s_l(t) = \sum_{k \in \mathbb{Z}} p_{b_k}(t - kTs) \quad (1)$$

tel que Ts , le temps d'impulsion élémentaire et (incluant ensuite une forme d'onde biphasé)

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

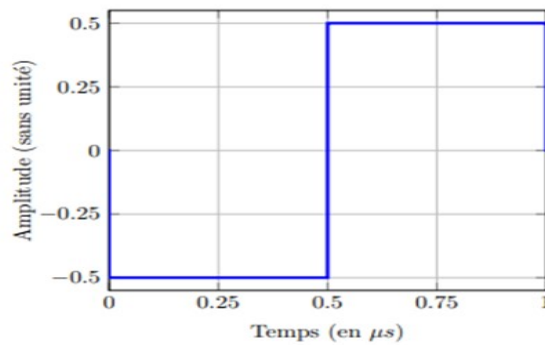


FIGURE 3 – Forme d'onde biphasé $p(t)$

2.1 Sous-tâche 1

Le signal émis peut se réécrire autrement. Tout d'abord, on remarque que :

$$p_{bk} = (A_k p + 0,5)1]t - kTs[\quad (2)$$

avec $A_k = 1$, si $b_k = 0$ et $A_k = -1$, si $b_k = 1$

On peut donc sortir tous les 0,5 sur (t-kTs) qui deviennent une constante et on obtient en remplaçant :

$$s_l(t) = 0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kTs) \quad (3)$$

2.2 Sous-tâche 2

Si on considère un bruit nul, des signaux causaux et un signal binaire [1,0,0,1,0], alors théoriquement on obtient $s_l(t)$, $r_l(t)$ et $r_m(t)$ de notre chaîne de communication (voir sous-tâche 3) tels que :

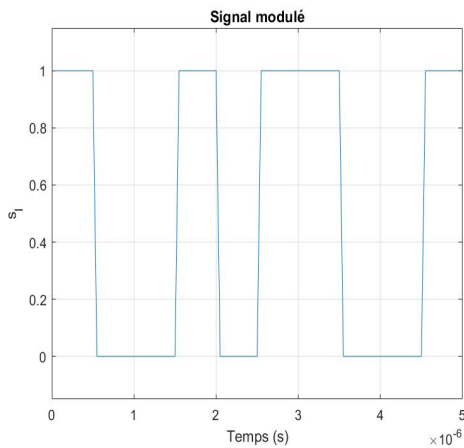


FIGURE 4 – $s_l(t)$

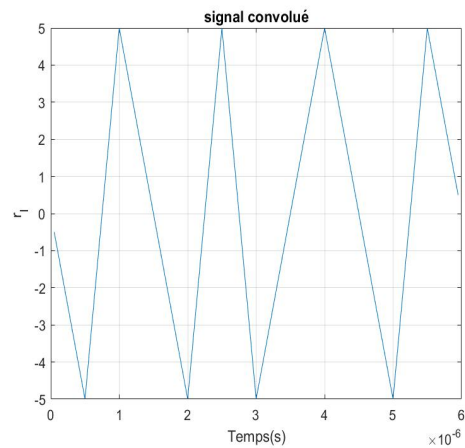


FIGURE 5 – $r_l(t)$

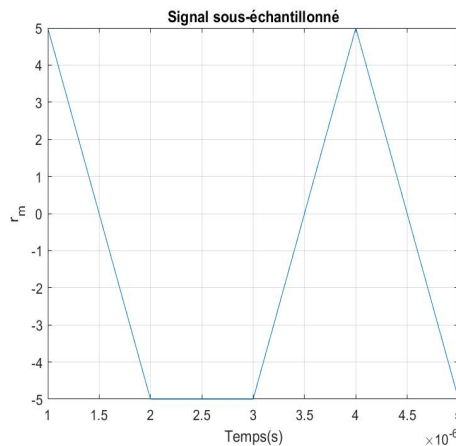


FIGURE 6 – $r_m(t)$

Si on observe r_m , on remarque que le bloc de décision est de la forme suivante :

$$\hat{b}_k = \begin{cases} 1 & \text{si } r_m(t) > 0 \\ 0 & \text{sinon} \end{cases} \quad (4)$$

2.3 Sous-tâche 3

On implémente ensuite la chaîne de communication suivante sur MATLAB en supposant une fréquence d'échantillonnage $f_e = 20$ MHz :

Hypothèses :

- Les b_k sont indépendants et distribués uniformément,
- Le bruit bande de base $n_l(t) \sim \mathcal{N}(0, \sigma_{n_l}^2)$ de densité spectrale de puissance (DSP) bilatérale $\Gamma_{n_l}(f) = \frac{N_0}{2}$,
- Le modèle bande de base de l'architecture de communication considérée (en excluant les parties codage et décodage de canal) est présenté sur la figure 3.

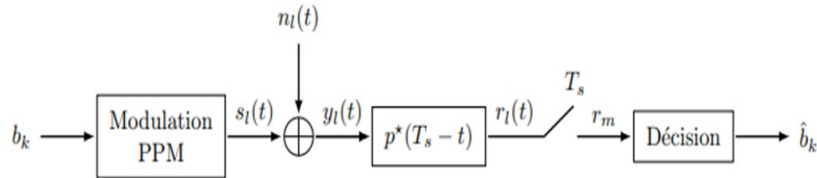


FIGURE 7 – Chaîne de communication complète

2.4 Sous-tâche 4

En parcourant la chaîne de communication, on obtient exactement les mêmes résultats pour $s_l(t)$, $r_l(t)$ et $r_m(t)$ (voir figure sous-tâche 2).

2.5 Sous-tâche 5

Pour la modulation PPM et le récepteur proposé durant cette tâche, la probabilité d'erreur binaire se trouve de la manière suivante (en reprenant les cours de communications numérique de Monsieur Ferre) :

Soit la constellation -1,1 et que chaque symbole est composé d'un seul bit ($P_b = P_s$). En considérant les probabilités d'obtenir A ou -A uniformes, on obtient que :

$$r_m(n) = -Av(0) + n'_l(t) \text{ avec } v \text{ le filtre de Nyquist à } T_s \text{ et } n'_l \text{ le bruit après la convolution} \quad (5)$$

Du fait que le seuil de décision soit à 0 dans notre cas, on :

$$P_b = \int_0^{+\infty} \frac{1}{\sqrt{2\pi}} \exp -\frac{z^2}{2} dx \text{ avec } z = \frac{x + Av(0)}{\sigma} \quad (6)$$

Avec $A = 1$ et sachant que $\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} \exp -t^2$, que le rapport est exprimé en dB dans erfc (ici rapport d'amplitude) alors on a :

$$P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \text{ avec } \frac{E_b}{N_0}, \text{ le rapport signal sur bruit} \quad (7)$$

2.6 Sous-tâche 6

Par la suite, on implémente sur Matlab, le calcul du ce taux d'erreur binaire pour le système proposé. Les bits sont générés aléatoirement et suivent une loi discrète uniforme (taille du paquet d'information binaire à $Nb = 1000 \text{ bits}$). On obtient le TEB en fonction du rapport signal bruit (pour des valeurs allant de 0 db à 10db) que l'on compare avec sa courbe théorique :

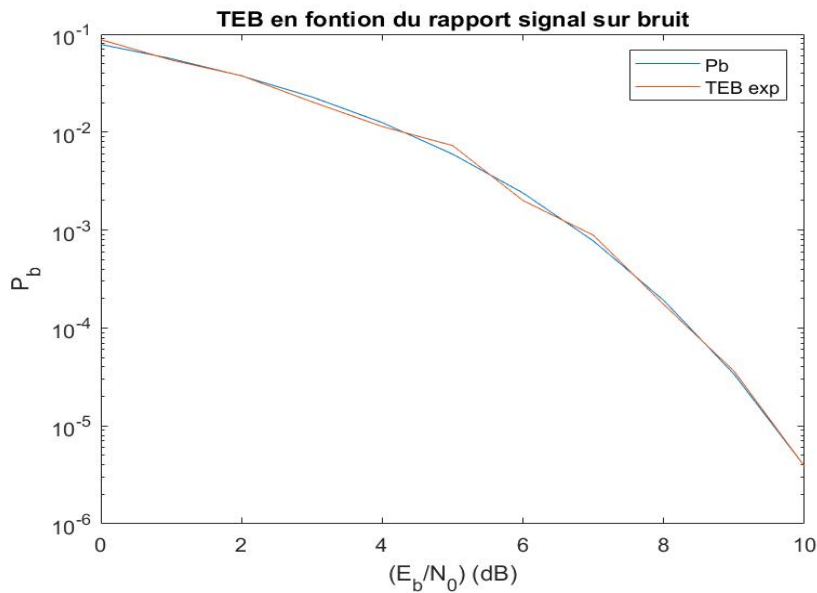


FIGURE 8 – TEB en fonction du rapport signal/bruit

3 Tâche 2

L'objectif de la seconde tâche est de calculer la densité spectrale de puissance de $s_l(t)$ du point de vue théorique et via une implémentation matlab.

3.1 Sous-tâche 1

Pour la formule du signal $s_l(t)$ voir tâche 1. On calcule le moment d'ordre 1 du signal tel que :

$$m_{s_l}(t) = E[s_l(t)] \quad (8)$$

$$m_{s_l}(t) = E[0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kTs)] \quad (9)$$

$$m_{s_l}(t) = 0,5 + E[\sum_{k \in \mathbb{Z}} A_k p(t - kTs)] \quad (10)$$

Or, on sait que $A_k = \pm 1$ avec la même probabilité donc $E[A_k] = 0$ donc $m_{s_l}(t)$ ne dépend pas du temps et :

$$m_{s_l}(t) = 0,5 \quad (11)$$

3.2 Sous-tâche 2

On calcule la fonction d'autocorrélation de ce signal telle que :

$$R_{s_l}(\tau) = E[s_l(t)s_l(t + \tau)] \quad (12)$$

$$R_{s_l}(\tau) = E[(0,5 + \sum_{k \in \mathbb{Z}} A_k p(t - kTs))(0,5 + \sum_{l \in \mathbb{Z}} A_l p(t + \tau - lTs))] \quad (13)$$

$$R_{s_l}(\tau) = 0,25 + \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} E[A_k A_l] p(t - kTs) p(t + \tau - lTs) \quad (14)$$

$$R_{s_l}(\tau) = 0,25 + \sum_{k \in \mathbb{Z}} p(t - kTs) p(t + \tau - kTs) \text{ car } E[A_k A_l] = \delta_{kl} \quad (15)$$

3.3 Sous-tâche 3

Le signal étant cyclo-stationnaire. On obtient sa DSP en calculant la transformée de Fourier de son autocorrélation moyennée :

$$\hat{R}_{s_l}(\tau) = \frac{1}{Ts} \int_a^b R_{s_l}(\tau) dt \quad (16)$$

$$\hat{R}_{s_l}(\tau) = \frac{1}{Ts} \int_a^b 0,25 + \sum_{k \in \mathbb{Z}} p(t - kTs) p(t + \tau - kTs) dt \quad (17)$$

$$\hat{R}_{s_l}(\tau) = 0,25 \frac{1}{Ts} \int_a^b p(t - kTs) p(t + \tau - kTs) dt = 0,25 + \frac{R_p}{Ts} \quad (18)$$

3.4 Sous-tâche 4

On en déduit au final sa DSP telle que :

$$\Gamma_{s_l}(f) = F(\hat{R}_{s_l})(f) = \frac{\delta(f)}{4} + \frac{|P(f)|^2}{Ts} \text{ d'après le théorème de Wiener-Khintchine} \quad (19)$$

Or,

$$p(t) = -\Pi_{\frac{Ts}{2}}(t - \frac{Ts}{4}) + \Pi_{\frac{Ts}{2}}(t - \frac{3Ts}{4}) \quad (20)$$

D'où,

$$P(f) = \frac{Ts}{2} \text{sinc}\left(\frac{Ts}{2}f\right) \left(\exp -j2\pi f \frac{3Ts}{4} - \exp -j2\pi f \frac{Ts}{4}\right) \quad (21)$$

$$|P(f)|^2 = T_s^2 \text{sinc}^2\left(\frac{Ts}{2}f\right) \text{sinc}^2\left(2\pi f \frac{Ts}{4}\right) = \frac{T_s^4 (\pi f)^2}{4} \text{sinc}^4\left(\frac{Ts}{2}f\right) \quad (22)$$

On obtient donc que :

$$\Gamma_{s_l}(f) = \frac{\delta(f)}{4} + \frac{T_s^3 (\pi f)^2}{16} \text{sinc}^4\left(\frac{Ts}{2}f\right) \quad (23)$$

3.5 Sous-tâche 5

En utilisant l'algorithme du périodogramme de Welch sans chevauchement et sans fenêtre de pondération, on implémente la fonction Mon_Welch pour obtenir la DSP de $s_l(t)$ et on compare avec sa courbe attendue théoriquement :

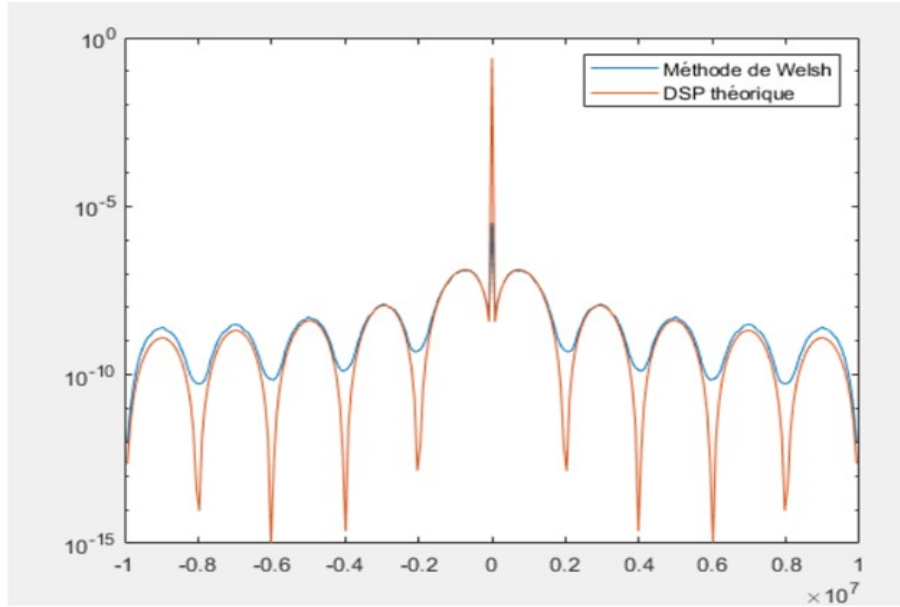


FIGURE 9 – Comparaison entre la DSP théorique du signal $s_l(t)$ et la DSP obtenue par la méthode Welch

4 Tâche 3

Cette tâche a pour but d'implémenter les codeurs et décodeurs dans le cadre de la transmission des signaux ADSB. On détaillera les différentes parties de la trame ADSB par la suite, cependant pour cette tâche, on peut souligner que cette trame dure $120\mu s$ et qu'elle possède un contrôle de parités de 24bits en fin de trame. Ces bits sont issus d'un CRC (Code à Redondance Cyclique) dont le polynôme générateur est le suivant :

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

Ces 24 bits servent à détecter les erreurs éventuelles parmi les 88 bits utiles de la trame en réception. Les algorithmes de codage et de décodage sont implémentés sous MATLAB avec l'aide des fonctions `crc.detector` et `detect`. On obtient maintenant une chaîne de communications telle que :

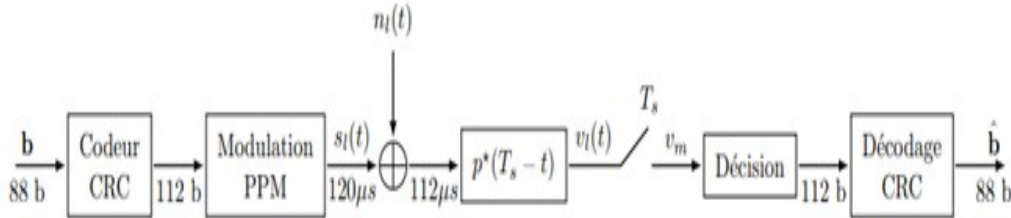


FIGURE 10 – Chaîne de communication complète

5 Tâche 4

Dans les tâches précédentes, deux défauts n'ont pas été pris en compte, à savoir : - une synchronisation temporelle, en effet le signal subit un délai de propagation δ_t - une synchronisation temporelle, en effet l'effet Doppler bien connue dû au mouvement de l'avion et les défauts d'oscillateurs locaux introduisent aussi un décalage en fréquence δ_f

On introduit donc le modèle en bande de base suivant pour les prendre en compte :

$$y_l = s_l(t - \delta_t) \exp -j2\pi\delta_f t + n_l(t) \quad (25)$$

5.1 Sous-tâche 1

Pour donner un exemple de l'ordre de grandeur du décalage de fréquence Doppler, on peut calculer celui pour un avion se déplaçant à 900km/h :

$$|f_r - f_e| = f_e \left(\frac{c}{c + v_e} - 1 \right) \quad (26)$$

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

Soit la chaîne de communications modifiée telle que :

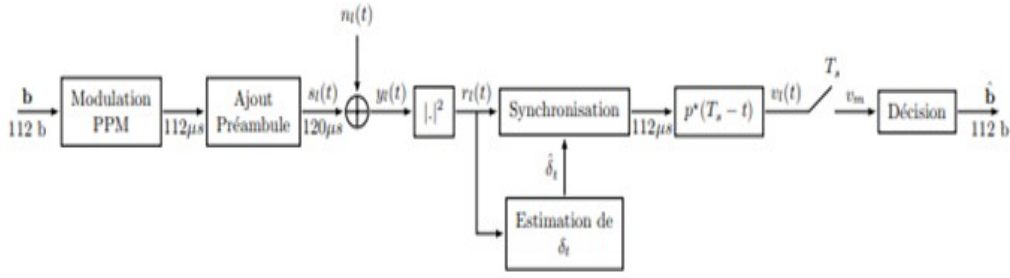


FIGURE 11 – Chaîne de communication complète avec l'estimation du décalage temporel et la synchronisation

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

$$|y_l(t)|^2 = y_l(t)y_l^*(t) = s_l^2(t - \delta_t) + s_l(t - \delta_t) \frac{1}{2} (Re(n_l(t)) \cos(2\pi\delta_f t) + Im(n_l(t)) \sin(2\pi\delta_f t)) + |n_l(t)|^2 \quad (27)$$

En identifiant $z_l(t)$, on remarque qu'il ne s'agit pas d'un bruit blanc gaussien et qu'il n'est pas indépendant de $s_l(t)$. Le fait de prendre le carré du module permet de retrouver une expression qui ne dépend pas de la fréquence enlevant ainsi le problème de synchronisation fréquentielle.

5.3 Sous-tâche 3

La synchronisation est réalisée à la réception en utilisant un signal s_p qui ne peut pas être présente dans le signal $s_l(t)$ appelé préambule et envoyé en entête des trames (voir la chaîne de communications sous tâche 1) tel que :

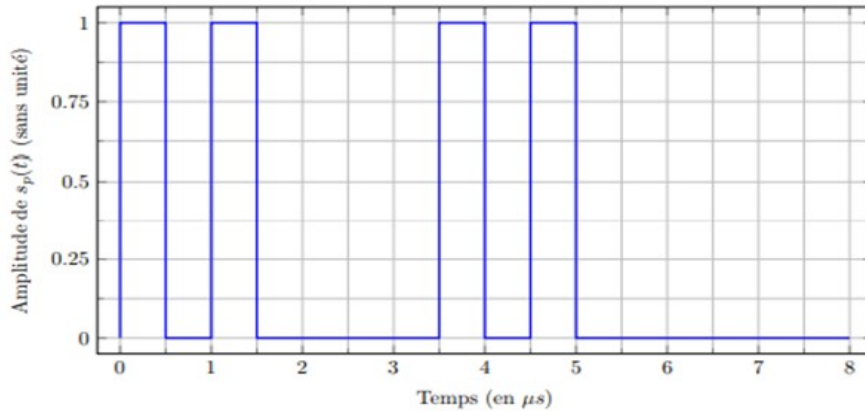


FIGURE 12 – Préambule $s_p(t) \text{ de } T_p = 8 \mu s$ débutant les trames ADSB-B

De ce fait, on utilise donc l'intercorrélation pour effectuer la synchronisation temps/fréquence avec pour $t \in [0, T_p]$, $s_l(t) = s_p(t)$ tel que :

$$y_l = s_p(t - \delta_t) \exp -j2\pi\delta_f t + n_l(t) \text{ pour } t \in [\delta_t, \delta_t + T_p] \quad (28)$$

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

$$p(\delta'_t) = \frac{\int_{\delta_t}^{\delta'_t+T_p} r_l(t) s_p * (t - \delta'_t) dt}{\sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_l(t)|^2 dt}} \quad (29)$$

On peut d'ailleurs montrer grâce à l'inégalité de Cauchy-Schwarz que $|p(\delta_t)| \leq 1$ pour tout δ'_t :

$$\left| \int_{\delta_t}^{\delta'_t+T_p} r_l(t) s_p * (t - \delta'_t) dt \right| \leq \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_l(t - \delta'_t)|^2 dt} \quad (30)$$

$$\left| \int_{\delta_t}^{\delta'_t+T_p} r_l(t) s_p * (t - \delta'_t) dt \right| \leq \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_l(t)|^2 dt} \quad (31)$$

D'où :

$$|p(\delta'_t)| = \left| \frac{\int_{\delta_t}^{\delta'_t+T_p} r_l(t) s_p * (t - \delta'_t) dt}{\sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_p(t)|^2 dt} \sqrt{\int_{\delta_t}^{\delta'_t+T_p} |s_l(t)|^2 dt}} \right| \leq 1 \quad (32)$$

avec une égalité seulement si s_p et r_l sont égales à un multiple près.

5.4 Sous-tâche 4

On modifie le code pour devenir cohérent avec la nouvelle chaîne de communications sachant que :

- δ_t est aléatoirement et uniformément réparti entre 0 et $100T_e$
- δ_f de même réparti entre $-1kHz$ et $1kHz$

5.5 Sous-tâche 5

En obtenant le décalage temporel δ_t , sans bruit dans la chaîne de communications, on remarque qu'il est parfaitement estimé.

5.6 Sous-tâche 6

On modifie encore le code MATLAB pour observer les performances du récepteur malgré les imperfections. On désynchronise aléatoirement chaque nouvelle trame émise et on calcule le TEB de la même manière que précédemment :

Avec la synchronisation, on a 3dB de perte environ.

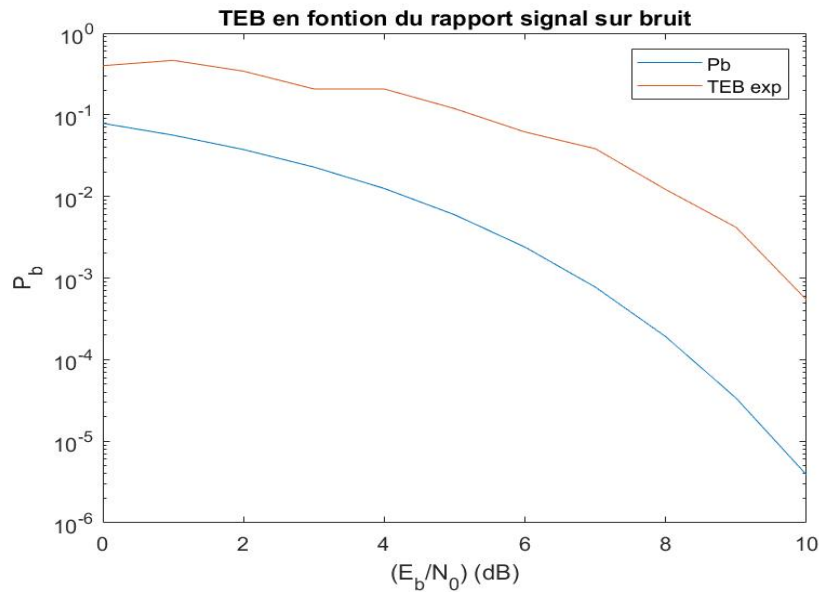


FIGURE 13 – TEB en fonction du rapport signal/bruit pour chaque désynchronisation

6 Tâche 6

Dans cette partie, on cherche à convertir un message binaire en registre afin d'obtenir les données des appareils concernés. Les signaux émis par les appareils pour l'ADS-B ont une durée de 120 μ s. Ils sont constitués des parties suivantes : — le préambule (identique à celui de la tâche 4), — le format de la voie descendante, — la capacité, — l'adresse OACI (Organisation de l'Aviation Civile Internationale) de l'appareil, — les données ADS-B, — les bits de contrôle de parité. Pour plus de documentations, vous pouvez aller lire la documentation sur les trames ADSB sur le git de Romain Tajan.

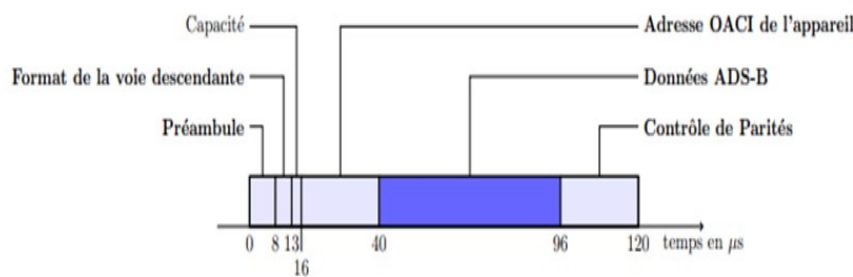


FIGURE 14 – Format d'une trame ADSB

6.1 Sous-tâche 1

Les registres comportant les informations des appareils sont de la forme suivante :

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

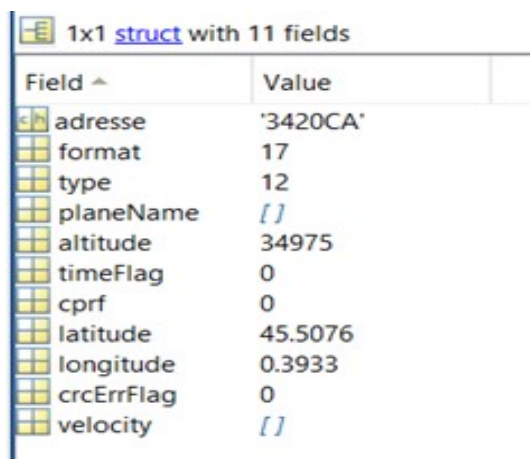
FIGURE 15 – Structure registre

Les premiers bits des données des trames ADSB correspondent au FTC (Format Type Code). En effet, tous les messages envoyés pour l'ADSB ne comportent pas systématiquement toutes les informations. C'est la détection du FTC qui permet de savoir s'il s'agit de position en vol ou de messages d'identifications par exemple. Voici les différentes valeurs de FTC :

- 1 à 4 : messages d'identifications
- 9 à 18 et 20 à 22 : trames de position en vol

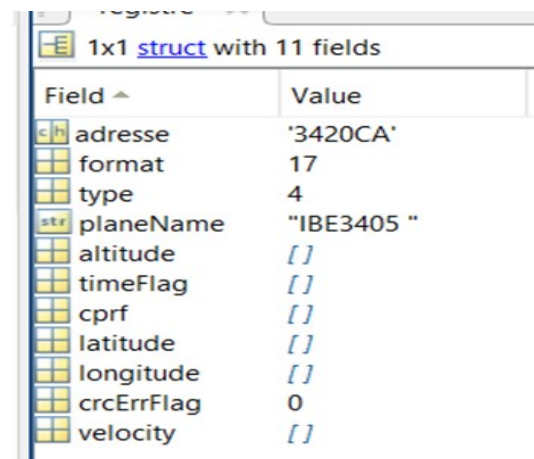
6.2 Sous-tâche 2

On code sur MATLAB la fonction bit2registre et toutes les autres fonctions annexes nécessaires pour extraire les informations et renvoyer le registre mis à jour. On a par exemple avec les datas de vérifications disponibles :



Field ^	Value
adresse	'3420CA'
format	17
type	12
planeName	[]
altitude	34975
timeFlag	0
cprf	0
latitude	45.5076
longitude	0.3933
crcErrFlag	0
velocity	[]

FIGURE 16 – registre rempli grâce à une trame de position en vol



Field ^	Value
adresse	'3420CA'
format	17
type	4
planeName	"IBE3405 "
altitude	[]
timeFlag	[]
cprf	[]
latitude	[]
longitude	[]
crcErrFlag	0
velocity	[]

FIGURE 17 – registre rempli grâce à une trame d'identification

6.3 Sous-tâche 3

En utilisant le fichier adsb_msgs_mat et en examinant chacune de ses trames, on peut afficher la trajectoire de l'avion considéré telle que :

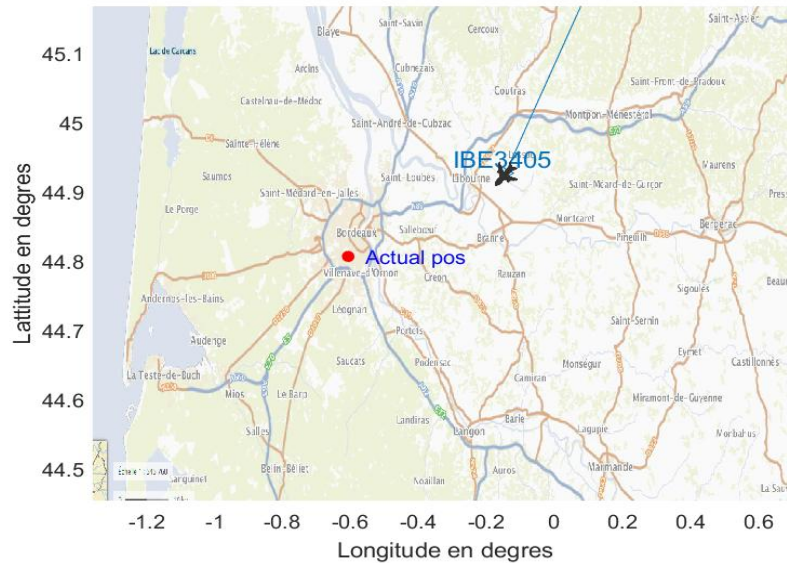


FIGURE 18 – Trajectoire de l'avion à partir des informations extraites de adsb_msg_mat

7 Tâche 7

7.1 Sous-tâche 1

Cette tâche permet simplement d'ajouter de nouvelles informations grâce à la lecture des trames, on inclut ici la position au sol.

7.2 Sous-tâche 2

Et dans cette partie, la vitesse de l'avion en vol en kt.

8 Tâche 8

8.1 Sous-tâche 1

On récupère les enregistrements de buffers.mat issus de signaux réels contenant 9 enregistrements fait par l'antenne de l'école. Chacun de ces enregistrements correspondent à y_i dans notre chaîne de communication complète. Ainsi pour extraire les données envoyés par les avions il nous faut grâce à notre fonction de synchronisation, récupérer les débuts de trame (tâche 4) pour ensuite les démoduler (tâche 1) et enfin les décoder (tâche 6).

8.2 Sous-tâche 2

On vérifie les avions obtenus à partir de ce fichier et on observe leur trajectoire :

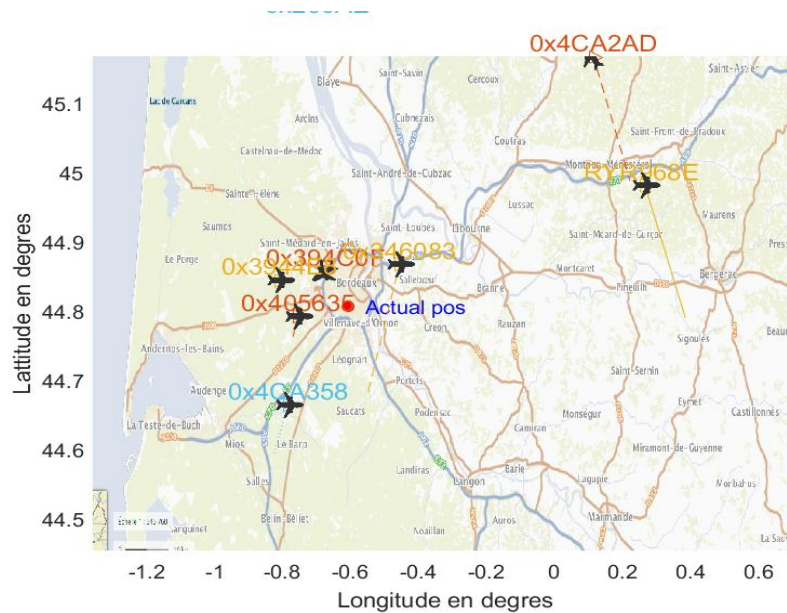


FIGURE 19 – Trajectoire des l’avion à partir des informations extraites de buffer.mat

9 Conclusion

Comme vous l'aurez remarqué ce projet s'est effectué au fil de nombreuses étapes sachant que la 1,2,3,4,6 et 8 étaient les tâches principales et que le reste dont la 7 était une tâche optionnelle. Nous avons donc réussi à parvenir au bout de la tâche 8 qui nous a permis d'observer les résultats de notre travail, ce qui a été très satisfaisant. Ce projet nous a permis de mieux comprendre l'intérêt d'une modulation, d'une convolution ou d'une intercorrélacion puisque ces concepts ont été utilisés à des fins concrètes. De plus, les recherches et la documentation sur les trames ADS-B nous ont apporté de vraies connaissances sur leur décryptage. Entre l'application des concepts vu l'année dernière et les recherches à effectuer, ce projet a atteint son terme au niveau des tâches principales demandées.