

Compte rendu de projet - TS226

Simulation d'un émetteur / récepteur ADS-B

Maxime PETERLIN - maxime.peterlin@enseirb-matmeca.fr
Gabriel VERMEULEN - gabriel@vermeulen.email

ENSEIRB-MATMECA, Bordeaux

19 janvier 2014

Table des matières

Introduction	2
1 Étude théorique	2
1.1 Description de la chaîne de transmission	6
1.2 Etude du taux d'erreur binaire en fonction du SNR	9
1.3 Synchronisation en temps et en fréquence	9
1.4 Etude du taux d'erreur binaire en fonction du SNR avec synchronisation	10
2 Implémentation sous MATLAB	11
2.1 Bloc d'émission	11
2.2 Canal	11
2.3 Bloc de réception	11
2.4 Décodage ADSB	11
3 Résultats	13
Conclusion	14

Introduction

Le nombre d'avion dans le ciel ne faisant que croître, des systèmes de surveillance du réseau aérien ont été mis en place pour garantir en outre la sécurité. Ainsi un système appelé ADS-B (Automatic Dependent Surveillance Broadcast) a été développé permettant aux avions de broadcaster des informations de vol telles que leur vitesse, leur altitude et leurs positions GPS.

Ce projet a pour but de modéliser un émetteur ainsi qu'un récepteur de donnée ADS-B avec le logiciel MATLAB.

Une première partie permettra de simuler théoriquement la couche physique de l'ADS-B et de tester ses performances vis à vis de perturbations de type bruitage et désynchronisation. Ensuite l'implémentation des algorithmes sous MATLAB sera expliquée et enfin les résultats obtenus seront présentés.

1 Étude théorique

La modulation en position d'amplitude est utilisée pour la transmission de signaux ADS-B. On a alors l'enveloppe complexe du signal émis qui est la suivante :

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

avec $T_s = 1\mu s$ le temps de l'impulsion élémentaire et

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

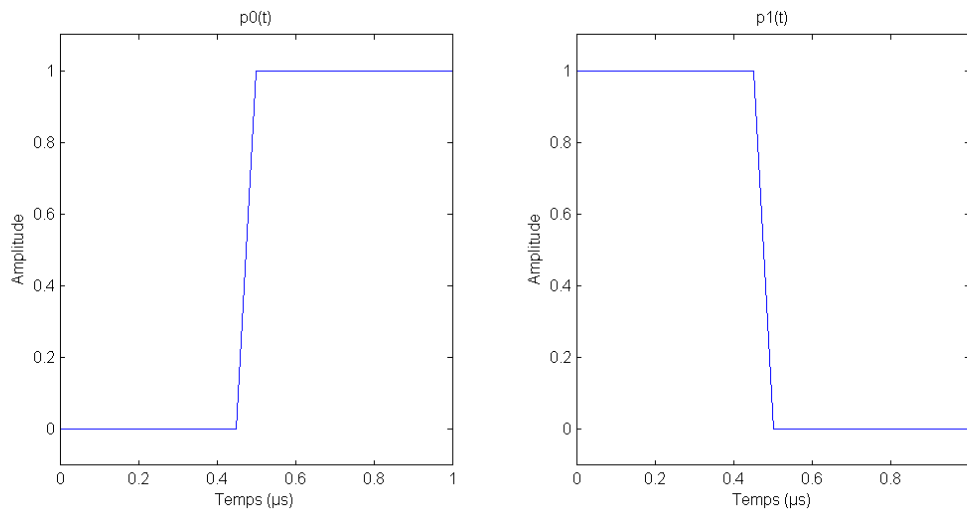


FIGURE 1 – Tracé des impulsions $p_0(t)$ et $p_1(t)$

s_l peut également s'exprimer en fonction des symboles émis A_k et de la forme d'onde biphasé donnée ci-dessous.

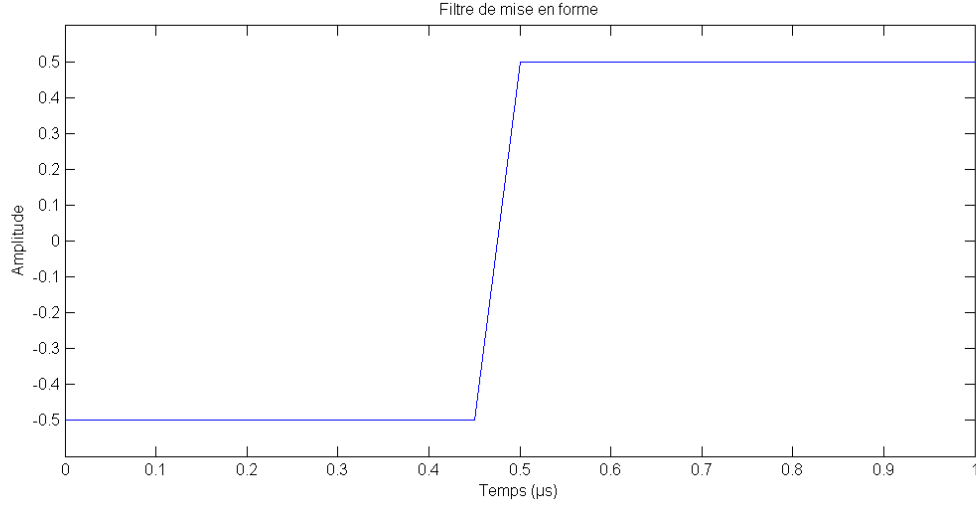


FIGURE 2 – Filtre de mise en forme

$$\begin{aligned}
 s_l(t) &= \sum_{k \in \mathbb{Z}} p_{b_k}(t_k T_s) \\
 &= \sum_{\substack{k \in \mathbb{Z} \\ b_k=0}} p_0(t - kT_s) + \sum_{\substack{k \in \mathbb{Z} \\ b_k=1}} p_1(t - kT_s) \\
 &= 0.5 + \sum_{b_k=0} p(t - kT_s) - \sum_{b_k=1} p(t - kT_s) \\
 s_l(t) &= 0.5 + \sum_{b_k=0} A_k p(t - kT_s) + \sum_{b_k=1} A_k p(t - kT_s)
 \end{aligned}$$

Finalement, on obtient :

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

avec

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

En réception, on aura les filtres adaptés p_0^* et p_1^* qui ont pour but de maximiser le SNR. De plus, afin de simplifier l'étape de décision et la rendre plus fiable, il faut supprimer l'interférence entre symboles, ce qui est rendu possible si les couples de filtres $(p_0(t), p_0^*(-t))$ et $(p_1(t), p_1^*(-t))$ respectent le critère de Nyquist :

$$\sum_{n=-\infty}^{\infty} P(f - \frac{n}{T_b}) = T_b$$

Avec $T_b = \frac{T_s}{2}$ le temps de l'impulsion et P la transformée de Fourier du filtre en réception.

Dans notre cas, les filtres en réception $p_0^*(-t)$ et $p_1^*(-t)$ sont, à un décalage temporel près, des portes de largeur $\frac{T_s}{2}$, ainsi leur transformée de Fourier est $\frac{T_s}{2} \text{sinc}(f \frac{T_s}{2})$. On remarque que pour $f = \frac{n}{T_b}$ avec $n \in \mathbb{Z}$ on

a $\text{sinc}(f \frac{T_s}{2}) = 0$. Ainsi, le critère de Nyquist est vérifié pour les deux filtres en réception.

$$\sum_{n=-\infty}^{\infty} P(f - \frac{n}{T_b}) = \frac{T_s}{2} \text{sinc}(0) = \frac{T_s}{2}$$

On veut à présent calculer la DSP du signal s_l à l'aide de l'autocorrélation moyennée de ce dernier. On commence par calculer le moment d'ordre 1 de s_l :

$$\begin{aligned} m_{s_l}(t) &= E[s_l(t)] = E[0.5 + \sum_{k \in \mathbb{Z}} A_k p(t - kT_s)] \\ m_{s_l}(t) &= 0.5 + \sum_{k \in \mathbb{Z}} \underbrace{E[A_k]}_{0.5 \cdot 1 + 0.5 \cdot -1} p(t - kT_s) \\ m_{s_l}(t) &= 0.5 \end{aligned}$$

On remarque que ce moment d'ordre 1 est indépendant du temps.

A présent nous allons calculer l'autocorrélation du signal s_l :

$$\begin{aligned} E[s_l(t)s_l^*(t - \tau)] &= \sum_k E[A_k^2 p(t - kT_s)p(t - \tau - kT_s)] \\ &= \sum_{k, k'} E[A_k p(t - kT_s) \cdot A_{k'} p(t - \tau - kT_s)] \end{aligned}$$

On sait que $p(t - kT_s) \neq 0$, si

$$0 < t - kT_s < T_s \Leftrightarrow \frac{t}{T_s} - 1 < k < \frac{t}{T_s}$$

Comme $k \in \mathbb{Z}$, on en déduit que $k = \left\lfloor \frac{t}{T_s} \right\rfloor$ Ainsi,

$$E[s_l(t)s_l^*(t - \tau)] = E[A_k]E[A_{k'}]p(t - \left\lfloor \frac{t}{T_s} \right\rfloor T_s)p(t - \tau - \left\lfloor \frac{t - \tau}{T_s} \right\rfloor T_s)$$

Finalement,

$$R_{s_l}(t, \tau) = \begin{cases} p(t - \left\lfloor \frac{t}{T_s} \right\rfloor T_s)p(t - \tau - \left\lfloor \frac{t - \tau}{T_s} \right\rfloor T_s), & \text{si } \left\lfloor \frac{t}{T_s} \right\rfloor = \left\lfloor \frac{t - \tau}{T_s} \right\rfloor \\ 0, & \text{si } \left\lfloor \frac{t}{T_s} \right\rfloor \neq \left\lfloor \frac{t - \tau}{T_s} \right\rfloor \end{cases}$$

Nous allons maintenant montrer que le signal est cyclo-stationnaire :

$$\begin{aligned} R_{s_l}(t + T_s, \tau) &= \begin{cases} p(t + T_s - \left\lfloor \frac{t + T_s}{T_s} \right\rfloor T_s)p(t + T_s - \tau - \left\lfloor \frac{t + T_s - \tau}{T_s} \right\rfloor T_s), & \text{si } \left\lfloor \frac{t + T_s}{T_s} \right\rfloor = \left\lfloor \frac{t + T_s - \tau}{T_s} \right\rfloor \\ 0, & \text{si } \left\lfloor \frac{t + T_s}{T_s} \right\rfloor \neq \left\lfloor \frac{t + T_s - \tau}{T_s} \right\rfloor \end{cases} \\ R_{s_l}(t + T_s, \tau) &= \begin{cases} p(t + T_s - T_s - \left\lfloor \frac{t}{T_s} \right\rfloor T_s)p(t + T_s - T_s - \tau - \left\lfloor \frac{t - \tau}{T_s} \right\rfloor T_s), & \text{si } \left\lfloor \frac{t}{T_s} \right\rfloor = \left\lfloor \frac{t - \tau}{T_s} \right\rfloor \\ 0, & \text{si } \left\lfloor \frac{t}{T_s} \right\rfloor \neq \left\lfloor \frac{t - \tau}{T_s} \right\rfloor \end{cases} \end{aligned}$$

On a alors :

$$R_{s_l}(t + T_s, \tau) = R_{s_l}(t, \tau)$$

De plus, on a vu que le moment d'ordre 1 est indépendant du temps, le signal s_l est bien cyclo-stationnaire.

Nous pouvons à présent calculer l'autocorrélation moyennée du signal s_l . Ainsi, pour $\tau \in [0, T_s]$, on a :

$$\begin{aligned}\tilde{R}_{s_l}(\tau) &= \frac{1}{T_s} \int_0^{T_s} R_{s_l}(t, \tau) dt \\ &= \frac{1}{T_s} \int_0^{T_s} p\left(t - \left\lfloor \frac{t}{T_s} \right\rfloor T_s\right) p\left(t - \tau - \left\lfloor \frac{t - \tau}{T_s} \right\rfloor T_s\right) dt\end{aligned}$$

Si $\tau < \frac{T_s}{2}$,

$$\begin{aligned}\tilde{R}_{s_l}(\tau) &= \frac{1}{T_s} \int_0^{\frac{T_s}{2}} \frac{1}{4} dt - \frac{1}{T_s} \int_{\frac{T_s}{2}}^{\frac{T_s}{2} + \tau} \frac{1}{4} dt + \frac{1}{T_s} \int_{\frac{T_s}{2} + \tau}^{T_s} \frac{1}{4} dt \\ &= \frac{1}{4} - \frac{3\tau}{4T_s}\end{aligned}$$

Si $\frac{T_s}{2} < \tau < T_s$,

$$\begin{aligned}\tilde{R}_{s_l}(\tau) &= -\frac{1}{T_s} \int_{\tau}^{T_s} \frac{1}{4} dt \\ &= \frac{\tau}{4T_s} - \frac{1}{4}\end{aligned}$$

En résumé, on a pour $\tau \geq 0$

$$\tilde{R}_{s_l}(\tau) = \begin{cases} \frac{1}{4} - \frac{3\tau}{4T_s}, & \text{si } 0 \leq \tau < \frac{T_s}{2} \\ \frac{\tau}{4T_s} - \frac{1}{4}, & \text{si } \frac{T_s}{2} \leq \tau < T_s \\ 0, & \text{sinon} \end{cases}$$

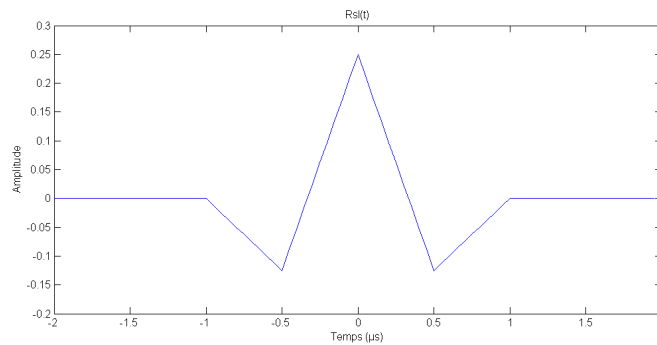


FIGURE 3 – Fonction d'autocorrélation $R_{s_l}(\tau)$

Calculons à présent la densité spectrale de puissance Γ_{s_l} du signal s_l . On a

$$p(t) = -\frac{1}{2}\Pi_{\frac{T_s}{2}}(t - \frac{T_s}{4}) + \frac{1}{2}\Pi(t - \frac{3T_s}{4})$$

Ainsi, la transformée de Fourier s'exprime de la manière suivante :

$$\begin{aligned} P(f) &= \frac{1}{2} \frac{T_s}{2} \text{sinc}(f \frac{T_s}{2}) e^{-j2\pi f \frac{T_s}{4}} + \frac{1}{2} \frac{T_s}{2} \text{sinc}(f \frac{T_s}{2}) e^{-j2\pi f \frac{3T_s}{4}} \\ &= \frac{T_s}{4} \text{sinc}(f \frac{T_s}{2}) [e^{-j2\pi f \frac{3T_s}{4}} - e^{-j2\pi f \frac{T_s}{4}}] \end{aligned}$$

La DSP a alors l'expression suivante :

$$\begin{aligned} \Gamma_{s_l} &= |P(f)|^2 = \frac{T_s^2}{16} \text{sinc}^2(f \frac{T_s}{2}) [-2j \sin(\pi f \frac{T_s}{2})]^2 \\ &= \frac{T_s^2}{16} \text{sinc}^2(f \frac{T_s}{2}) [4 \sin^2(\pi f \frac{T_s}{2})] \end{aligned}$$

$$\boxed{\Gamma_{s_l} = \frac{T_s^2}{4} \text{sinc}^2(f \frac{T_s}{2}) \sin^2(\pi f \frac{T_s}{2})}$$

1.1 Description de la chaîne de transmission

Pour simuler la modulation PPM, un message binaire aléatoire est initialement généré. Ce dernier est ensuite transformé en symbole appartenant à l'ensemble $\{-1, 1\}$ puis est sur-échantillonné. Enfin on applique le filtre de mise en forme ainsi qu'un décalage d'amplitude de +0.5.

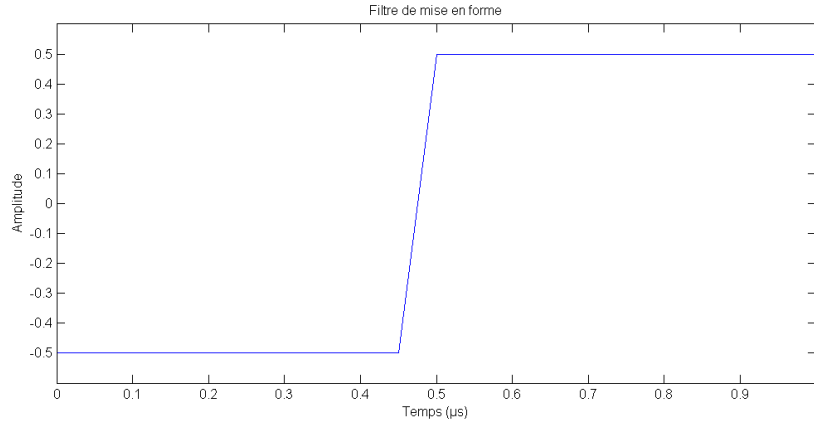


FIGURE 4 – Filtre de mise en forme

Le résultat obtenu est le signal $s_l(t)$ qui correspond à la sortie du bloc d'émission.

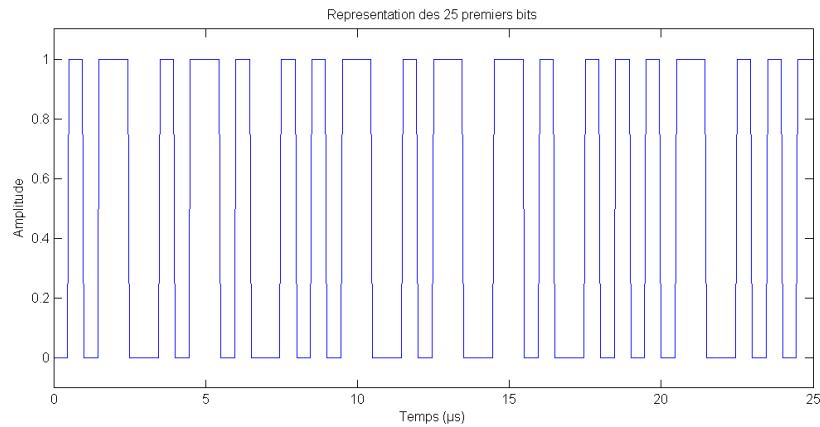


FIGURE 5 – Tracé de $s_l(t)$ pour les 25 premiers bits

Le diagramme de l'oeil nous permet de vérifier qu'il n'y a aucun défaut de synchronisation, de distorsion ou de gigue. Le tracé suivant est fait avant émission, il est donc parfait :

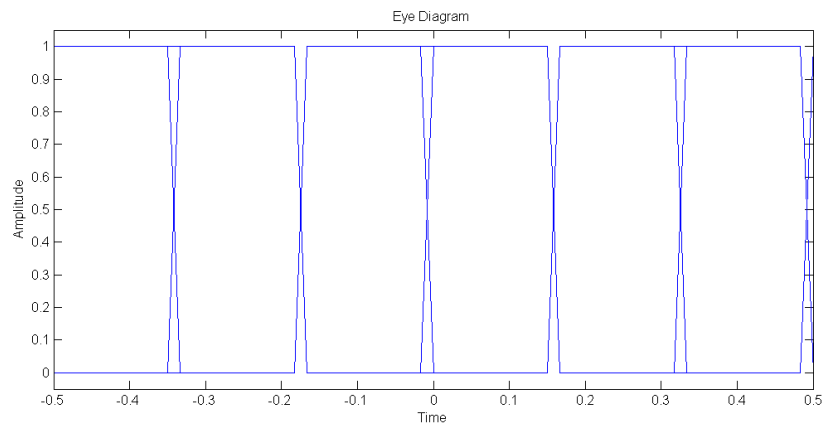


FIGURE 6 – Diagramme de l'oeil pour les 100 premiers bits

La densité spectrale de puissance du signal nous permet de voir la répartition fréquentielle de la puissance du signal. Un tracé théorique et réel est présenté sur la figure suivante. On peut remarquer, outre le décalage d'amplitude entre les deux courbes, le fait que le signal réel aléatoire résultant d'une seule réalisation engendre une courbe moins lissée que celle théorique.

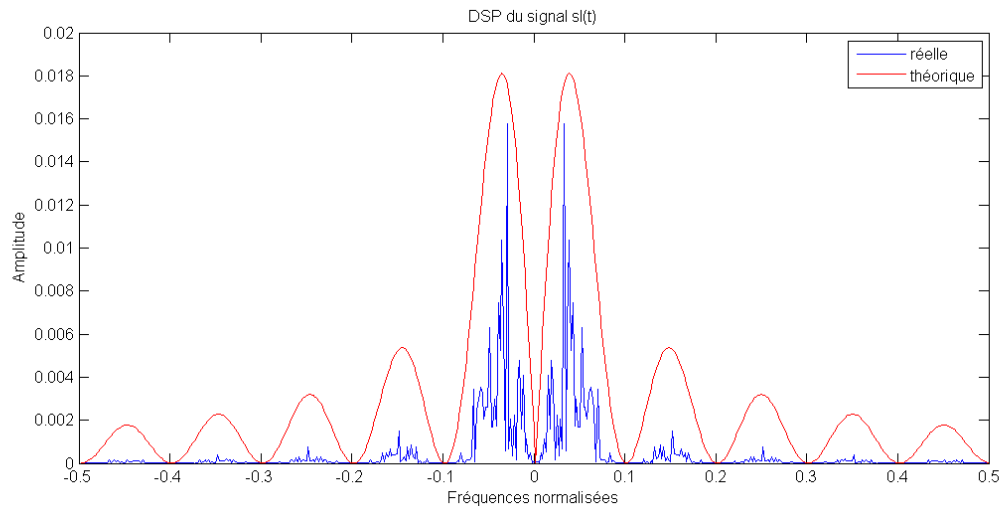


FIGURE 7 – Tracé théorique et réel de la DSP de $s_I(t)$

On considérera que le canal se comporte comme un filtre passe-tout. A la réception, le signal récupéré est constitué du signal émis auquel est ajouté un bruit de réception modélisé par un bruit blanc gaussien centré en 0. Tout d'abord un décalage d'amplitude de -0.5 est appliqué au signal. Ensuite ce dernier est filtré par un filtre adapté dont voici le tracé :

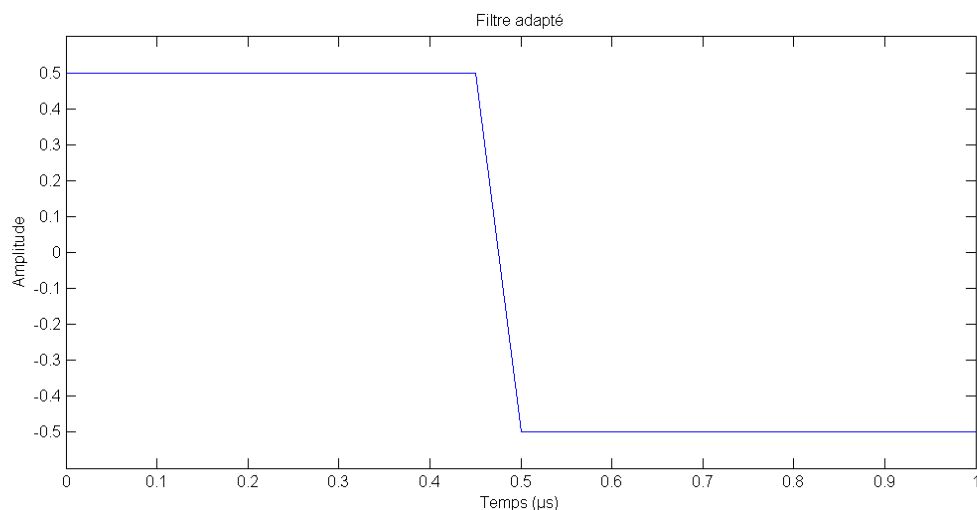


FIGURE 8 – Filtre adapté

Il ne reste qu'à sous-échantillonner au rythme du temps symbole et à effectuer la décision par rapport à l'amplitude du signal. Le message est ainsi récupéré.

1.2 Etude du taux d'erreur binaire en fonction du SNR

L'étude du lien entre le taux d'erreur binaire et du ratio signal sur bruit permet de mesurer les performances de la chaîne de transmission PPM malgré l'ajout d'un bruit blanc gaussien centré au signal émis. On peut observer un décalage entre la courbe théorique et la courbe réelle.

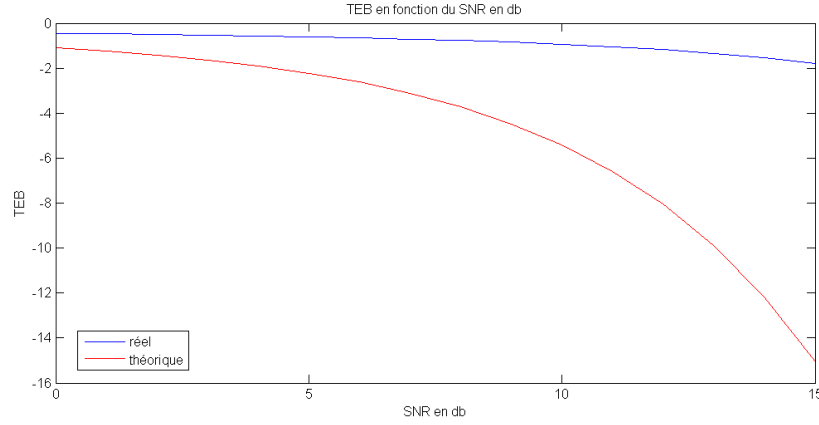


FIGURE 9 – Courbes réelles et théoriques du TEB en fonction du SNR en db

1.3 Synchronisation en temps et en fréquence

Afin de synchroniser temporellement et fréquentiellement le signal reçu (induit respectivement par un délais de propagation et par l'effet Doppler), on calcule le maximum de la corrélation suivante.

$$\rho(\delta'_t, \delta'_f) = \frac{\int_{\delta'_t}^{\delta'_t + T_p} y_l(t) s_p^*(t - \delta'_t) e^{j2\pi\delta'_f t} dt}{\sqrt{\int_0^{T_p} |s_p^2(t)| dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}} \quad (1)$$

Les décalage temporels et fréquentiels sont réalisés en prennant le couple $(\hat{\delta}_t, \hat{\delta}_f)$ tel que

$$(\hat{\delta}_t, \hat{\delta}_f) = \arg \max_{(\delta'_t, \delta'_f)} |\rho(\delta'_t, \delta'_f)|$$

Grâce au théorème de Cauchy-Schwarz, on montre que $|\rho(\delta'_t, \delta'_f)| \leq 1$. En effet,

$$\frac{\int_{\delta'_t}^{\delta'_t + T_p} y_l(t) s_p^*(t - \delta'_t) e^{j2\pi\delta'_f t} dt}{\sqrt{\int_0^{T_p} |s_p^2(t)| dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}} \leq \frac{\sqrt{\int_0^{T_p} |s_p^2(t)| dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}}{\sqrt{\int_0^{T_p} |s_p^2(t)| dt} \sqrt{\int_{\delta'_t}^{\delta'_t + T_p} |y_l(t)|^2 dt}} \leq 1$$

avec égalité lorsque $y_l = \lambda s_p$

Lorsqu'un avion se déplace à 900 km/h, son déplacement induit un décalage de fréquence perçu par le récepteur. Ce phénomène porte le nom d'effet Doppler. Il est possible d'estimer ce décalage de fréquence grâce à la formule qui suit :

$$f_{rec} = \frac{c - v_{rec}}{c - v_{em}} f_{em}$$

Avec f_{rec} la fréquence reçue, f_{em} la fréquence émise, v_{rec} la vitesse du récepteur, v_{em} la vitesse de l'émetteur et c la célérité de l'onde.

Ici sera considéré :

- $f_{em} = 1090 \text{ MHz}$
- $v_{rec} = 0 \text{ km.h}^{-1}$
- $v_{em} = 900 \text{ km.h}^{-1}$
- $c = 1\,079\,252\,848,8 \text{ km.h}^{-1}$

Finalement on obtient $f_{rec} = 1\,090\,000\,909 \text{ Hz}$, ce qui traduit un décalage maximum de **+909 Hz**.

1.4 Etude du taux d'erreur binaire en fonction du SNR avec synchronisation

Afin de gérer la synchronisation temporelle et fréquentielle, il faut modifier la chaîne de transmission.

Dans le bloc d'émission, il suffit d'ajouter le préambule au signal de sortie.

Avant d'ajout du bruit blanc gaussien centré au signal, il faut artificiellement désynchroniser le signal. Cela se fait en ajoutant des bits aléatoires avant le signal et en le multipliant par une exponentielle complexe.

Le bloc de réception se voit ajouter le calcul de la fonction (1) permettant d'estimer les deux paramètres δ_t et δ_f . Il devient alors possible de démoduler le signal, puis de calculer et de tracer la nouvelle courbe du TEB en fonction du SNR.

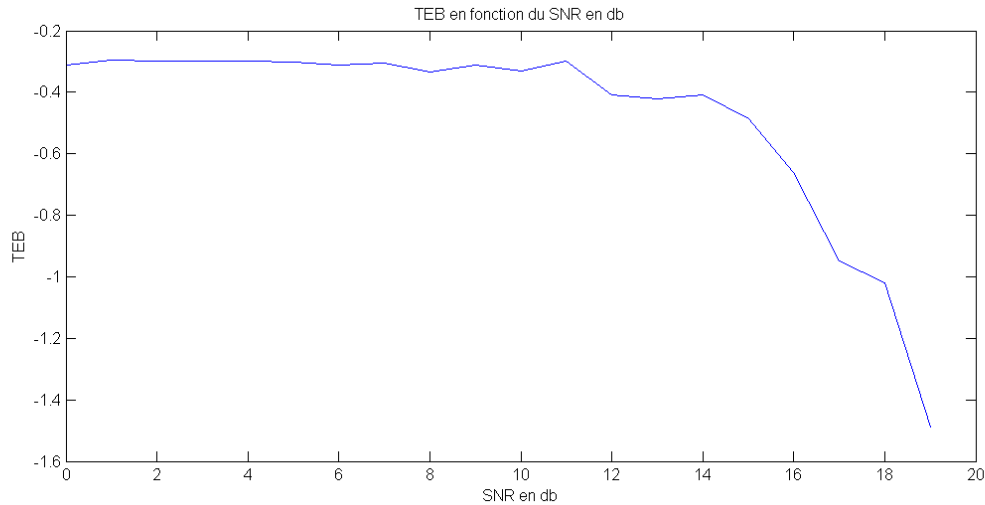


FIGURE 10 – Courbes réelles et théoriques du TEB en fonction du SNR en db

Il a été remarqué que la fonction établie pour estimer les paramètres δ_t et δ_f reste très sensible au bruit, ce qui peut donner des erreurs d'estimations. Lorsque c'est le cas, le TEB devient important.

2 Implémentation sous MATLAB

2.1 Bloc d'émission

La génération aléatoire de message s'effectue simplement avec la fonction *randi* en précisant la taille de la matrice à générer, et l'ensemble $\{0, 1\}$ correspondant aux bits.

Ensuite l'association bits/symbole se fait avec un simple calcul mathématique et le sur-échantillonnage est effectué avec la fonction *upsample*.

Enfin, l'utilisation de la fonction *conv* permet d'appliquer le filtre de mise en forme au signal.

L'ajout du préambule se fait avec une concaténation de vecteurs.

2.2 Canal

Le canal est considéré comme un filtre passe-tout. Il ne change donc pas le signal émis.

Afin de désynchroniser artificiellement le signal, il faut ajouter au signal une séquence aléatoire de bits généré avec *randi*, et multiplier chaque termes par une exponentielle complexe, de fréquence aléatoirement déterminé par la fonction *randi*.

2.3 Bloc de réception

La première chose à faire est d'estimer les paramètres δ_t et δ_f . Pour ce faire, une boucle calcul pour un δ_f variant de -1000 Hz à 1000 Hz la convolution entre le signal multiplié par une exponentielle complexe de fréquence δ_f et le préambule. On recherche ensuite le maximum de cette convolution avec la fonction *max* ce qui nous permet d'obtenir δ_t et δ_f .

On peut ainsi compenser les décalages temporel et fréquentiel et sélectionner les $112 \times Fse$ points tra-
duisant une trame (Fse étant le ratio fréquence d'échantillonnage sur fréquence du débit symbole).

Il reste à utiliser la fonction *conv* pour appliquer le filtre adapté, puis à effectuer une simple condition *supérieure* à pour la décision et on obtient le message démodulé.

2.4 Décodage ADSB

Une fois que les buffers reçus ont été démodulés et synchronisés en temps et en fréquence, on peut extraire les trames ADS-B dont nous allons à présent expliquer le décodage que nous avons implémenté via la fonction *bit2registre*.

Les trames ont une longueur de 120 bits et possède la structure suivante :

- Préambule \rightarrow 8 bits
- Format de la voix descendante \rightarrow 5 bits
- Capacité \rightarrow 3 bits
- Adresse OACI \rightarrow 24 bits
- Données ADS-B \rightarrow 56 bits
- Contrôle de parités \rightarrow 24 bits

Tout d'abord, on vérifie que la trame est correcte grâce au CRC de polynôme générateur

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

On peut effectuer cette opération sur MATLAB à l'aide de la fonction *crc.detector* qui va générer le CRC à partir d'un polynôme générateur passé en argument, puis on vérifie que la trame est correcte grâce à *detect* qui nous renvoie le nombre d'erreurs trouvées.

S'il n'y a pas d'erreur, on récupère les 5 bits suivant le préambule afin de connaître le format de la voie descendante que l'on convertit en binaire à l'aide de *bi2de* (on ne traitera que les trames avec **DF = 17**).

Ensuite, on décode l'adresse OACI de l'appareil qui nous permettra de faire le lien entre les différentes trames reçues, car elle est propre à chaque avion (l'adresse est codée sur 24 bits et se trouve à la position 16 dans la trame). On la convertira tout d'abord en binaire, puis en hexadécimal à l'aide de la fonction *dec2hex* afin de faciliter sa lecture.

On termine par le décodage du message contenant les données transportées par la trame. Les 5 premiers bits du message permettent de connaître le format des données envoyées (FTC). On ne s'intéresse qu'aux valeurs de FTC comprises entre 1 et 4 (pour les messages d'identification) et entre 9 et 22 (en excluant la valeur 19 pour les messages de position en vol).

Si le FTC est compris entre 1 et 4, on commence par créer une matrice de taille 8x6 où les colonnes correspondent aux 6 bits encodant les caractères. Un tableau contenant la correspondance entre la valeur décimale de ces 6 bits et le caractère auxquels ils correspondent est créé afin de décoder à l'aide de *cells* le nom de l'avion.

Si le FTC est compris entre 9 et 22 (mais différent de 19), on extrait l'indicateur UTC qui en position 21 dans le message, ainsi que le bit suivant qui est l'indicateur CPR. Enfin, à l'aide d'une série de calculs complexes, on récupère l'altitude, la latitude et la longitude.

3 Résultats

La simulation du récepteur ADSB-B a permis d'étudier des données réelles obtenu à l'aide d'un récepteur.

Tout d'abord le code a été testé sur des trames déjà démodulées. Ainsi on peut observer par exemple le décollage d'un avion :

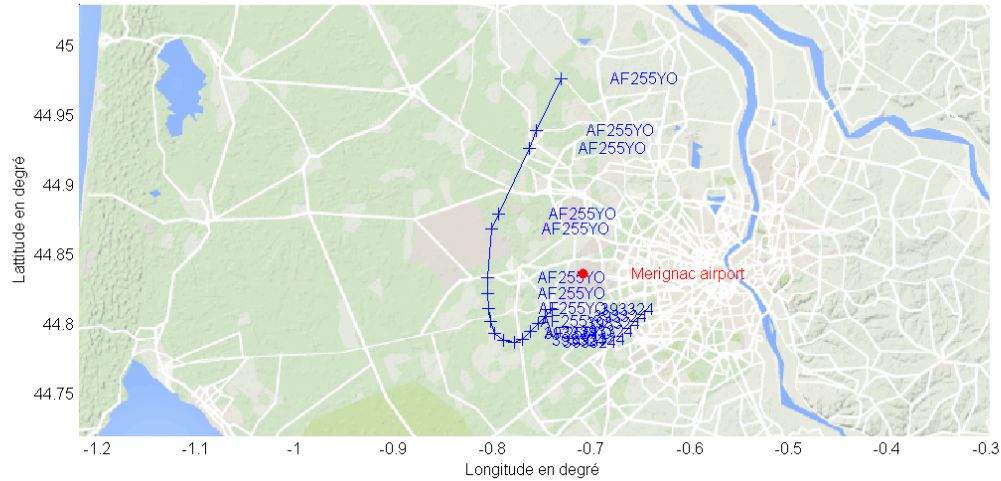


FIGURE 11 – Données provenant du fichier *trames_20141120.mat*

Par la suite, il a fallu intégrer la démodulation ce qui a permis de traiter un flux réel de données provenant d'un récepteur radio. Le flux de données est sous forme d'un buffer de 2 millions de point correspondant à 0.5 seconde de temps de réception.

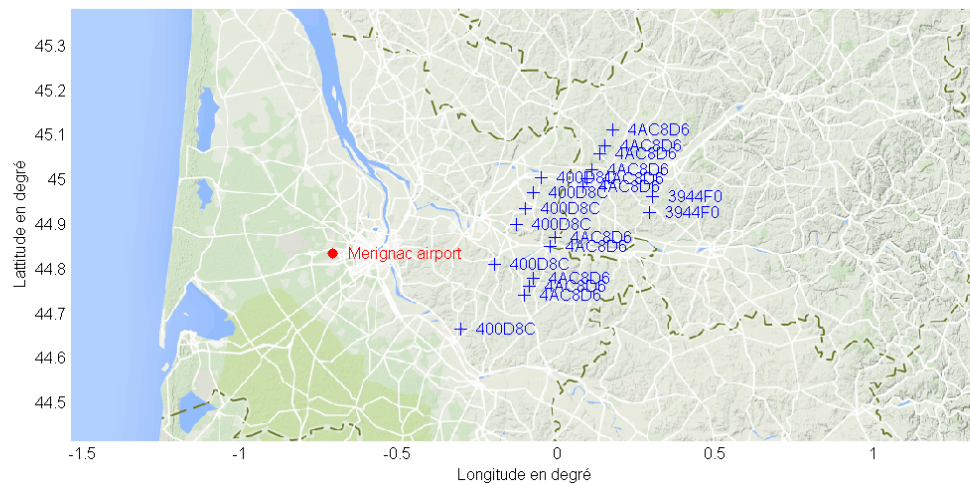


FIGURE 12 – *list_cplx_buffers2.mat* démodulé et décodé

Enfin le code a été testé en temps réel. Si les performance restent limité concernant le temps d'exécution, il a quand même été possible de récupérer quelques données :

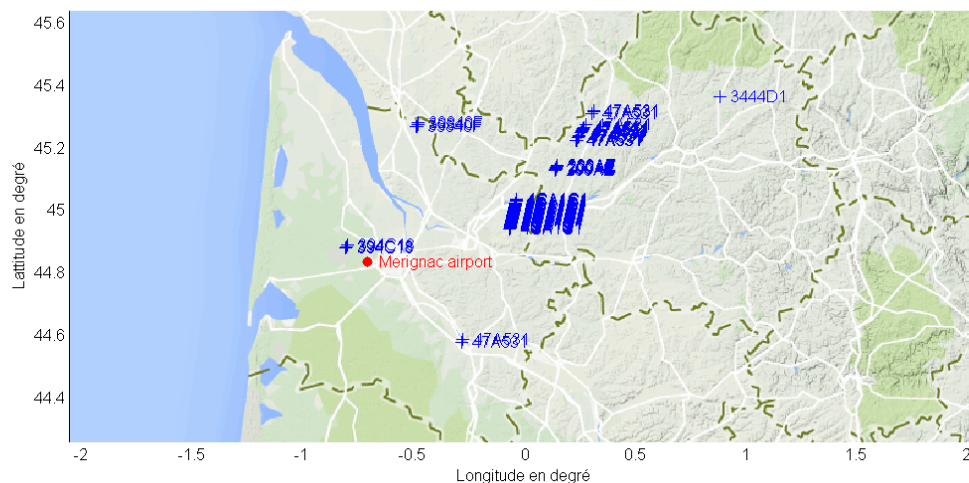


FIGURE 13 – Données traitées en direct

Conclusion

Ce projet nous aura permis d'étudier et de mettre en oeuvre une chaîne de communication numériques en mettant l'accent sur la partie réception. L'étude du protocole ADS-B, dont nous avons à présent une bonne compréhension, a notamment été un plus du fait de son utilisation répandue dans le secteur de l'aéronautique.

L'une des principales difficultés rencontrées a été la synchronisation temporelle et fréquentielle du signal reçu, étape cruciale du décodage des trames ADS-B.

Cependant, nous avons finalement réussi à décoder des buffers de trames en temps réel, par contre les calculs étaient trop lents par rapport au flux d'arrivée des buffers. Une grande partie d'optimisation pourrait encore être faite à ce niveau afin d'améliorer les performances de l'estimation de la position des trames qui est l'étape demandant le plus de calculs.