
Communications numeriques

# Simulation d'un émetteur

Julien Choveton

Alexandra Abulaeva

# Contents


**1 Introduction** **2**

**2 Tache 1,Couche physique et chaine de communication ADS-B** **2**

2.1 Demonstation $s_l(t)$=0,5+$\sum_{k=-\infty}^{\infty} A_k(t - kTs)$ . . . . . 4

2.2 Implementation de la chaine de communication et tracé de signaux pour une entrée 5

2.3 Calcul de probabilité d'erreur binaire . . . . . 6

2.4 Taux d'erreur binaire . . . . . 7

**3 Tâche 2, Couche physique ADS-B Densité spectrale de puissance** **8**

3.1 Moment d'odre 1 . . . . . 8

3.2 Fonction d'autocorrélation . . . . . 8

3.3 DSP de $s_l(t)$ . . . . . 8

**4 Tache 3, Algorithme de codage et décodage CRC du canal** **9**

**5 Tache 4, Synchronisation en temps** **11**

5.1 Effet Doppler introduit par un avion . . . . . 11

5.2 Demonstation $|y(l)|^2$=$|sl(t - \delta_t)|^2$+$z_l$ . . . . . 11

5.3 Demonstration $\rho(\delta_t) <= 1$ . . . . . 12

5.4 Chaîne de communication prennant en compte les décalages fréquentiel et temporel 12

**6 Tache 6, Couche MAC ADS-B Implémentation de la couche MAC** **14**

**7 Tache 8, Couche physique et chaine de communication ADS-B** **14**

**8 Tache 9, Application Mise en place du temps réel** **16**

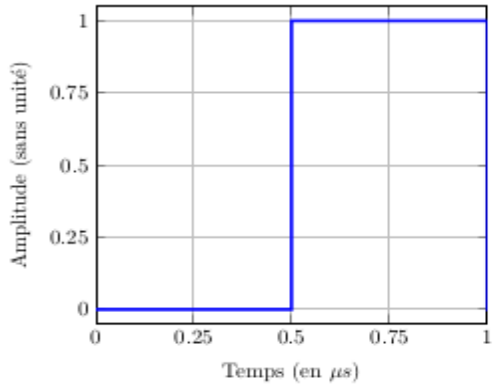
| Participation aux taches | Tache 1 | Tache 2 | Tache 3 | Tache 4 | Tache 6 | Tache 8 |
|---|---|---|---|---|---|---|
| Alexandra | 40 | 0 | 100 | 100 | 0 | 30 |
| Julien | 60 | 100 | 0 | 0 | 100 | 70 |

# 1 Introduction

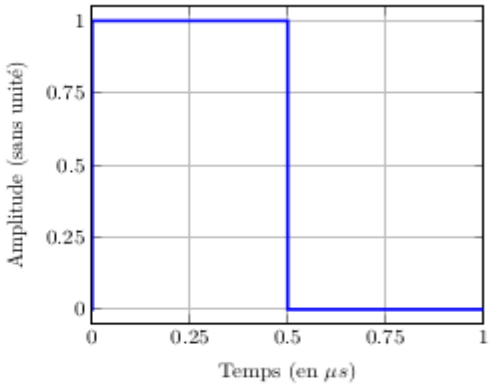
ADS-B,Automatic Dependant Surveillance - Broadcast, est un systeme permettant de contrôler l'etat du reseau aerien. Il a été proposé en complément des radars classiques. Les avions envoient leurs coordonnées GPS à intervalles de temps reguliers et des stations au sol peuvent recevoir ces données afin d'obtenir leurs trajectoires,positions. Ce système necessite uniquement une antenne au sol afin de recevoir les données,il s'agit de l'un de ces avantages par rapport aux systèmes classiques. Pour la transmission,on peut utiliser plusieurs liasons mais dans ce projet nous allons nous concentrer sur le 1090 ES. C'est à dire que la frequence porteuse des signaux est de 1090 MHz. Les signaux transmis sont autour de la porteuse à 1,09 GHz.

# 2 Tache 1,Couche physique et chaine de communication ADS-B

Dans le système ADS-B,la modulation utilisée est PPM,pulse position modulation. Elle encode les informations binaires avec les impulsions suivantes avec une periode symbole de $1\mu$s.




(a) Impulsion $p_0(t)$ encodant le bit 0




(b) Impulsion encodant $p_1(t)$ le bit 1

Afin d'etudier l'emission et la reception des signaux via la modulation/demodulation PPM,nous allons mettre en place la chaine de communication suivante. La demodulation est composée de la reception par un filtre recepteur,de l'echantillonage et de la decision.

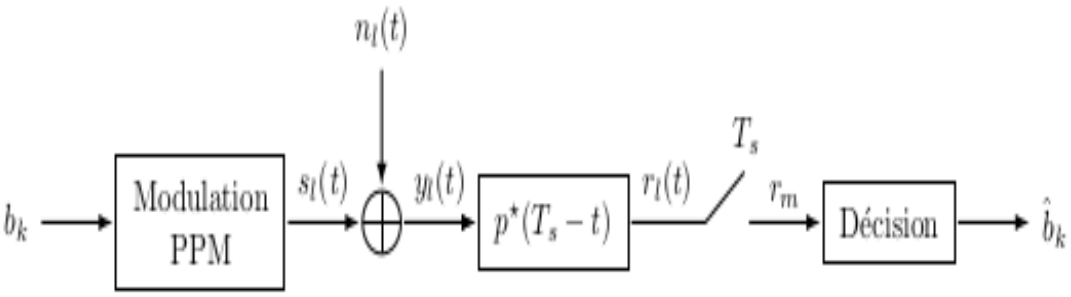
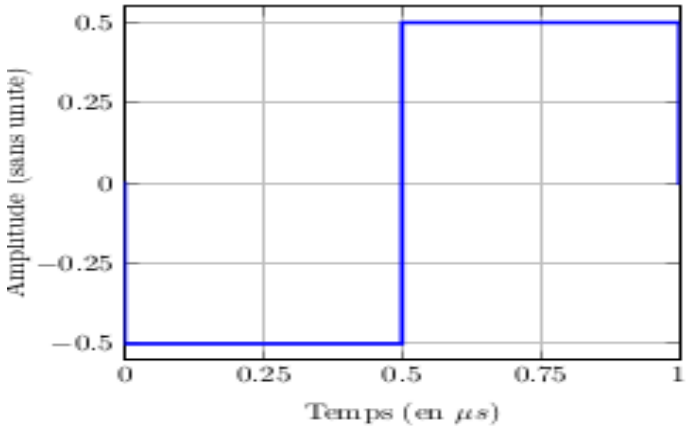



FIGURE 3 – Chaîne de communication complète, $p(t)$ étant donné en Figure 2.

Voici une representation graphique du filtre de reception,p$^*(Ts - t)$.




On va supposer ,pour les études theoriques, que: Le bruit $n_l$ suit une loi gaussienne de moyenne nulle et de variance $\sigma^2_{nl}$. et que les b$_k$ sont independants et uniformement distribuées.

D'apres le sujet,l'enveloppe complexe du signal envoyé s'ecrit:

$$s_l(t) = \sum_{k=-\infty}^{\infty} p_{bk}(t - kTs) \tag{1}$$

avec $p_{bk}(t) = p_o(t)$ si $b_k = 0$ ou $p_{bk}(t) = p_1(t)$ si $b_k = 1$

## 2.1 Demonstation $s_l(t)$=0,5+$\sum_{k=-\infty}^{\infty} A_k(t-kTs)$

Montrons que $s_l(t)$=0,5+$\sum_{k=-\infty}^{\infty} A_k(t-kTs)$ avec $A_k = 1$ si $b_k = 0$ ou $A_k = -1$ si $b_k = 1$

à l'aide des representations graphiques,on a pour tout k

$p_o$(t-kTs)=u(t-(k+1/2)Ts)-u(t-(k+1)Ts)

$p_1$(t-kTs)=u(t-kTs)-u(t-(k+1/2)Ts)

p(t-kTs)=0,5($p_o$(t-kTs)-$p_1$(t-kTs))

Partons du resultat,

$s_l(t)$=0,5+$\sum_{k=-\infty}^{\infty} A_k(t-kTs)$

=$\sum_{k=-\infty}^{\infty}(0,5(u(t-kTs)-u(t-(k+1)Ts))+A_k(t-kTs))$

On fait le cas pour $b_k$=1

$s_l(t)$=(0,5(u(t-kTs)-u(t-(k+1)Ts))-p(t-kTs))

=0,5((u(t-kTs)-u(t-(k+1)Ts))-$p_o$(t-kTs)+$p_1$(t-kTs))

=0,5((u(t-kTs)-u(t-(k+1)Ts))-u(t-(k+1/2)Ts)+u(t-(k+1)Ts)+u(t-kTs)-u(t-(k+1/2)Ts))

=2*0,5(u(t-kTs)-u(t-(k+1/2)Ts))

=$p_1$(t-kTs)

De même pour $b_k$=0

$s_l(t)$=(0,5(u(t-kTs)-u(t-(k+1)Ts))+p(t-kTs))

=0,5((u(t-kTs)-u(t-(k+1)Ts))+$p_o$(t-kTs)-$p_1$(t-kTs))

=0,5((u(t-kTs)-u(t-(k+1)Ts))+u(t-(k+1/2)Ts)-u(t-(k+1)Ts)-u(t-kTs)+u(t-(k+1/2)Ts))

=$p_0$(t-kTs)

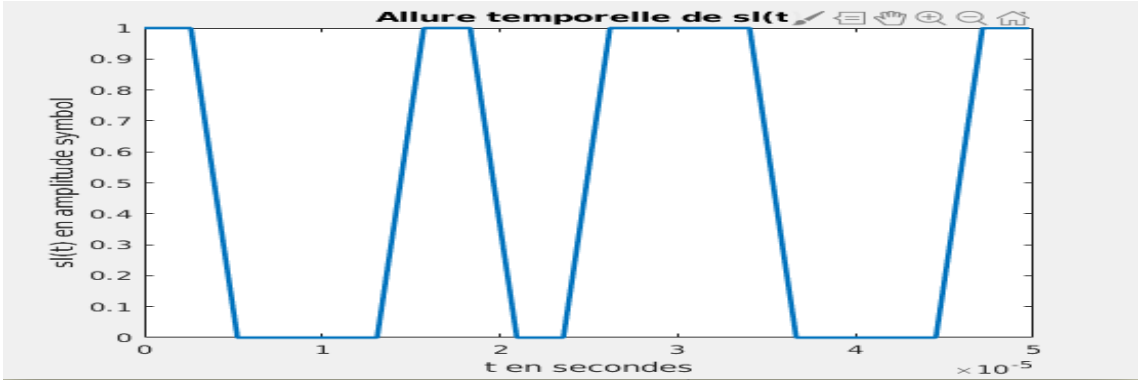
Ceci nous prouve que la relation est valable pour chaque echantillon.

$s_l(t)$=0,5+$\sum_{k=-\infty}^{\infty} A_k(t-kTs)$

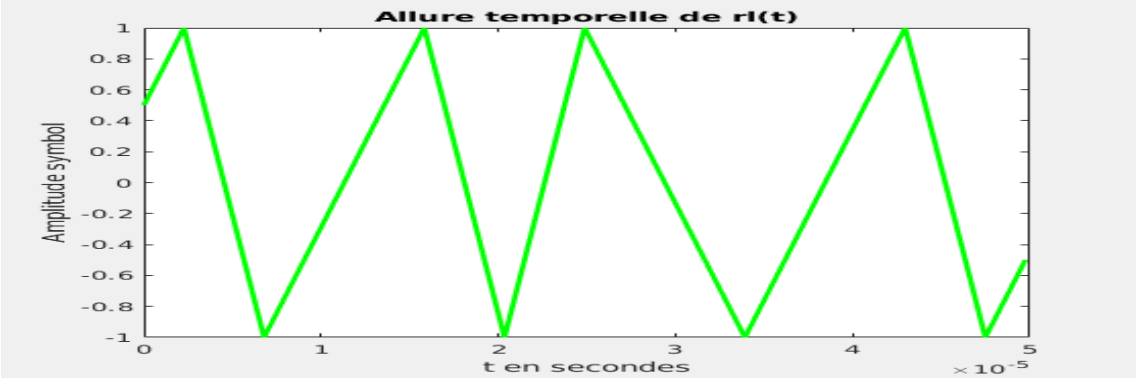
## 2.2 Implementation de la chaine de communication et tracé de signaux pour une entrée

A l'aide de l'implementation de la chaine de communication sur matlab,on peut obtenir les signaux suivants pour une entrée [1 0 0 1 0]:

sl(t),signal en sortie du bloc modulation PPM




rl(t),signal en sortie du filtre de reception




rm(t),signal echantilloné

A partir de ces graphiques et de notre implementation de la chaine de communication,on peut supposer que le bloc decision permet de retrouver,sans bruit additif, le signal mis en entré de la chaine via une relation/condition mathematiques. Il permet d'associer les symboles et les bits,de "repasser" des symboles aux bits. d'apres nos figures,nous avons remarqué que lorsque rm(t) est negatif,cela equivaut a un bit emis en entré de 1 et inversement lorsque rm(t) vaut une valeur positive,le bit emis en entré vaut 0.

C'est coherent avec la theorie de la partie precedente,puisque nous avons vu que lorsqu'on passe en entré un bit=0 on obtient un symbole A=1 et si on passe en entré un bit=1,on obtient un symbole A=-1.

## 2.3 Calcul de probabilité d'erreur binaire

En generale,lors d'une transmission de signal,des bruits se glissent dans les informations utiles à transmettre. On se propose de calculer la probabilité d'erreur binaire.

D'apres la chaine de communication,on a rl(t)=(sl(t)+nl(t))*p$^*(Ts-t)$

Donc rl(t)=$\sum(A_k v(t-kTs)+0,5\int p^*(Ts-t)+nl'(t)$

avec nl'(t)=nl(t)*p$^*(Ts-t)$ et v(t-kts)=p(t-kTs)*p$^*(Ts-t)$

De plus, $\int p^*(Ts-t)=0$,

finalement rl(t)=$\sum(A_k v(t-kTs)+nl'(t)$

Si on applique la formule du filtrage sur nl'(t)=nl(t)*p$^*(Ts-t)$, on obtient:

$\gamma_{nl'}=|P(f)|^2\gamma_{nl}$ avec $\gamma$ la densité spectrale de puissance et P(f) la transformée de fourier du filtre de reception.

En appliquant, la transformée de fourier inverse,on a $R_{nl'}(t)=R_{p*}(t)R_{nl}(t)$

puis en prenant t=0, $\sigma^2_{nl'}=E_b\sigma^2_{nl}$

De plus, $E_b=T_bP_{moy}$ avec $P_{moy}=\frac{\sigma_A^2\int p(t)^2}{T_s}$, $T_b=T_s$ et $\sigma_A^2=1$

*Donc* $E_b=\int p(t)^2=v(0)$

Du coup en remplaçant dans la formule du filtrage du bruit,on obtient $\sigma^2_{nl'}=v(0)\sigma^2_{nl}$

Or $\sigma_{nl}^2 = \frac{N_0}{2}$

On arrive a avoir la relation,$\sigma_{nl'}^2 = v(0)\frac{N_0}{2}$

Cela implique que $\frac{2}{N_0}=\frac{v(0)}{\sigma_{nl'}^2}$

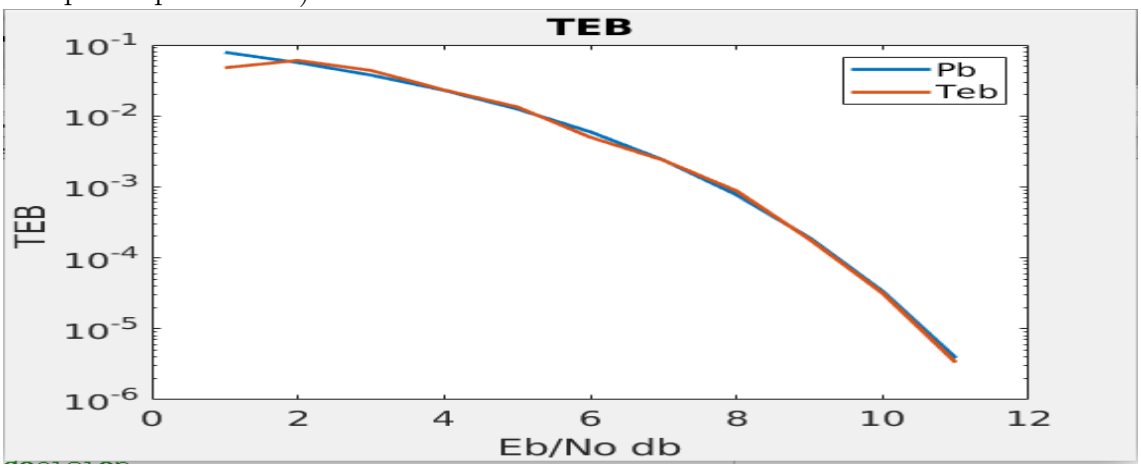
En multipliant par $E_b$, on obtient $\frac{2E_b}{N_0}=\frac{v(0)^2}{\sigma_{nl'}^2}$

Cela nous permet d'avoir la probabilité d'erreur binaire,$P_b$, en fonction de $\frac{E_b}{N_0}$

$P_b = Q(\frac{v(0)}{\sigma_{nl'}}) = Q(\sqrt{\frac{2E_b}{N_0}}) = \frac{1}{2}erfc(\sqrt{\frac{E_b}{N_0}})$

## 2.4 Taux d'erreur binaire

Le taux d'erreur binaire represente le rapport entre le nombre de bits reçus erronés et le nombre de bits total transmis,TEB=$\frac{bits_{erronés}}{bits_{transmis}}$. Nous avons tracé la probabilité d'erreur binaire(calculé dans la partie precedente) et le TEB.




En implémentant la chaîne de communication et le tracé du taux d'erreur binaire pour des erreurs suivant une loi gausienne. Afin de tracer le taux d'erreur binaire,nous avons varié la variance du bruit et reproduit la démodulation du signal en sortie du canal,nous avons sommé le nombre d'erreur et calculé le taux d'erreur binaire. Nous obtenons une courbe du taux d'erreur binaire qui se superpose avec la probabilité d'erreur binaire théorique.On peut interpréter que malgré que notre système soit expiremental,il reste proche du cas théorique puisque les courbes se superposent,notre système est donc robuste.

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

À travers cette tâche nous allons effectuer certains calculs théoriques, puis nous allons calculer grâce à la méthode de Welch la DSP expérimentale pour la comparer par la suite via un tracé avec la théorique.

## 3.1 Moment d'odre 1

Calculons tout d'abord le moment d'odre 1 du signal s $_{l(t)}$

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

le terme p ( t - k T s ) peut sortir de la somme car la fonction est déterministe,

de plus E $[A_k] = 0$.

## 3.2 Fonction d'autocorrélation

Calculons ensuite la fonction d'autocorrélation:

$$R_{s_l}(t,\tau) = E\left[s_l(t)s_l^*(t+\tau)\right] = E\left[\left(0,5 + \sum_{k\in Z} A_k p\left(t - kT_s\right)\right)\left(0,5 + \sum_{k'\in Z} A_{k'} p\left(t + \tau - k'T_s\right)\right)\right]$$
$$= E\left[0,25 + \sum_k \sum_{k'} A_k A_{k'} p\left(t - kT_s\right) p\left(t + \tau - k'T_s\right)\right.$$
$$\left. + 0,5 \sum_{k\in Z} A_k p\left(t - kT_s\right) + 0,5 \sum_{k'\in Z} A_{k'} p\left(t + \tau - k'T_s\right)\right]$$

Les deux termes en 0,5 sont nul car E $[A_k] = 0,$ il reste donc:

$= 0,25 + \sum_k \sum_{k'} E\left[A_k A_{k'}\right] p\left(t - kT_s\right) p\left(t + \tau - k'T_s\right)$

or si k=k' l'espérance du produit des Ak et Ak' vaux donc E $[A_{k^2}]$ on obtient donc :

$R_{s_l}(t,\tau) = 0,25 + \sum_k \sigma^2 p\left(t - kT_s\right) p\left(t + \tau - kT_s\right)$

## 3.3 DSP de s $_l(t)$

$\Gamma_{s_l}(f) = \mathcal{F}\left(\tilde{R}_{s_l}(\tau)\right)$

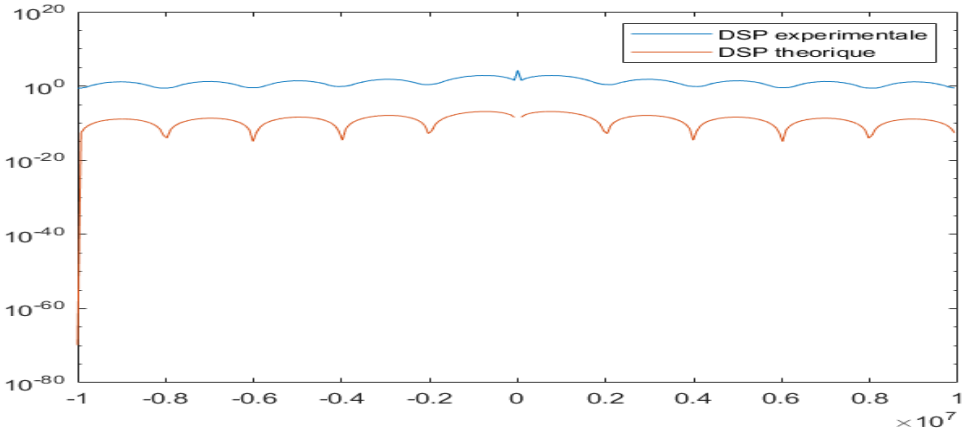
$= \int_{-\infty}^{+\infty} \tilde{R}_{s_l}(\tau) e^{-j2\pi f\tau} d\tau = 0.25\delta(f) + \frac{1}{T_s}|P(f)|^2 (1)$

Or nous avons $P(f)$ qui est égale à : $\int_{-\infty}^{+\infty} p(t)\exp(-j2\pi f\tau)dt$

soit $P(f) = -\,\mathrm{j}\exp(-j\pi f)\,\mathrm{sinc}(\pi f)\sin\left(\pi\frac{f}{2}\right)$

En remplacent P(f) dans l'équation 1 On obtient ainsi le DSP de $s_{l(t)}$ :

$\Gamma_{s_l}(f) = 0.25 * \delta(f) + \frac{1}{T_s} * \sin c^2(\pi f) * \sin^2\left(\pi \frac{f}{2}\right)$




Nous vous invitons a regarder le fichier Tache2.m pour voir l'ensemble du code implémenter. Nous avons un problème d'offset que nous n'avons pas réussi à résoudre lors du projet.

# 4 Tache 3, Algorithme de codage et décodage CRC du canal

Afin de transmettre des données ADS-B(utiles),nous avons besoin de les encapsuler et de préparer la trames pour la transmission.

Parmis les étapes nécessaires,on a le codage CRC pour l'émission et le décodage CRC pour la réception.

Le codage et décodage CRC permettent de s'assurer que la trame reçue est sans erreur afin d'éviter de démoduler de fausses données. En présence de bruit dans le canal, il suffit qu'un bit soit faussé dans la trame et qu'au moment de la démodulation,de fausses données soient transmises. En effet,le codage et décodage CRC permettent de détecter des erreurs dans la trame entière. Pour mettre en place la détection d'erreurs via la méthode CRC,l'émetteur et le récepteur doivent se mettre d'accord sur un polynôme générateur permettant de calculer/générer une suite de bits qui sera rajouter à la fin de la trame. A l'émission,à l'aide des fonctions,permettant le codage CRC,proposé par matlab nous avons générer un code qui est ajouté à la fin de chaque trame,ce code est composé de 24 bits(car notre polynôme générateur est de degré 24). A la réception, avant de démoduler la trame,

les fonctions CRC de matlab recalcule ce code CRC et le compare au code ajouté en fin de trame à l'émission. Si ces 2 codes sont identiques,la trame reçue est sans erreur et on peut passer à la démodulation. Dans le cas inverse,nous avons détecter au moins une erreur dans la trame et on ne peut pas traiter/démoduler cette trame. Il faut noter qu'une erreur peut être faite par le générateur ou par le détecteur de code CRC,au moment du calcul du code CRC(reste de la division de la trame par le polynôme générateur),donc l'erreur peut être parmis les bits de données,dans décapsulation ou encore dans le code CRC.

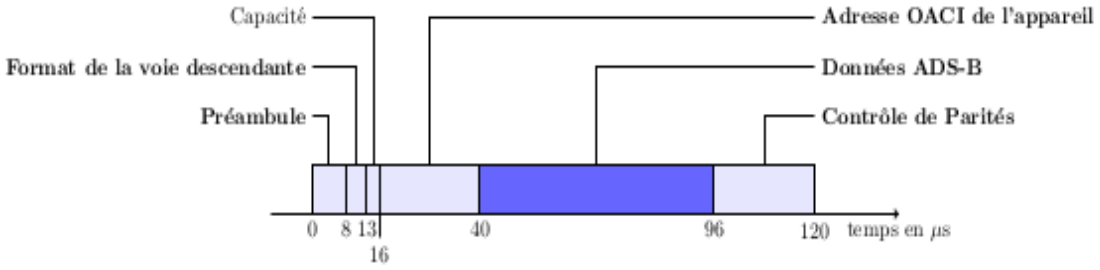



FIGURE 1 – Format d'une trame ADS-B

Dans cette partie,nous allons implémenter la même chaîne de communication que dans la tache 1 en ajoutant une partie codage CRC et une partie décodage CRC.
Voici la chaîne de communication implémentée.

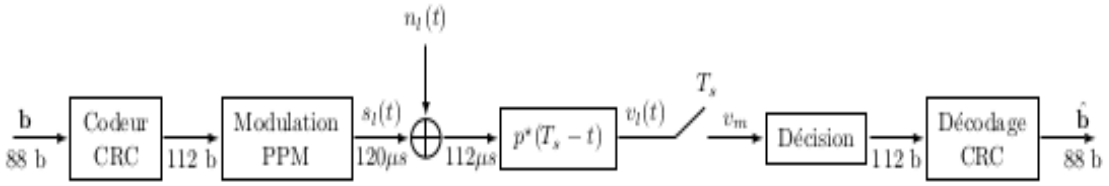



FIGURE 2 – Chaîne de communication

Après avoir implémenté cette chaîne,on peut remarquer qu'on a aucune information(données utiles/code CRC) sur l'erreur commise en présence de bruit. En l'absence de bruit,nous recevons une trame sans erreur.

# 5 Tache 4, Synchronisation en temps

Lors d'une reception de trames emises par de sources réelles,il faut prendre en compte certains points:effet Doppler introduit par le mouvement de la source et delai de propagation de la trame jusqu'au recepteur. Dans notre cas,les sources seront les avions envoyant des trames ADS-B et notre recepteur l'antenne.

## 5.1 Effet Doppler introduit par un avion

L'effet Doppler est le decalage de frequence d'une onde lorsque la distance emetteur-recepteur varie,donc lorsqu'un avion se deplace dans la zone de detectabilité de l'antenne. Si on suppose que l'avion se deplace sur la meme droite que l'antenne, on peut admettre que:

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

avec $f_{rec}$, frequence de l'onde reçu et $f_{em}$, frequence de l'onde reçu par la source.

$v_{em}$ vitesse de la source,$v_{rec}$ vitesse du recepteur et c la celerité de l'onde.

Or notre recepteur,l'antenne est statique,on peut donc dire :

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

En prenant c=$3.10^8 m/s$ et $v_{rec} = 900.10^3 m/s$, on trouve un rapport de 1 entre $f_{rec}$ et $f_{em}$.

L'effet doppler $\delta_f = f_{rec} - f_{em}$ tend vers 0 .

## 5.2 Demonstation $|y(l)|^2$=$|sl(t-\delta_t)|^2$+$\mathbf{z}_l$

$|y(l)|^2$=$|sl(t-\delta_t)\exp(-j2\pi\delta_f t) + n_l(t)|^2$

$=|sl(t-\delta_t)\cos(-j2\pi\delta_f t) + jsl(t-\delta_t)\sin(-j2\pi\delta_f t) + n_l(t)|^2$

$=(sl(t-\delta_t)\cos(-j2\pi\delta_f t) + n_l(t))^2 + (jsl(t-\delta_t)\sin(-j2\pi\delta_f t))^2$

$=(sl(t-\delta_t)^2 + n_l(t)^2 + 2n_l(t)sl(t-\delta_t)\cos(-j2\pi\delta_f t)$

=(sl(t-$\delta_t$)$^2$ + $z_l(t)$

avec $z_l(t) = n_l(t)^2 + 2n_l(t)sl(t-\delta_t)\cos(-j2\pi\delta_f t)$

$z_l$ est un bruit blanc gaussien qui depend de $s_l(t)$.

Afin d'eviter de prendre en compte le decalage frequentiel $\delta_f$, on peut prendre $|y(l)|^2$ puisqu'il est composé de $(sl(t-\delta_t)^2$ et de $z_l(t)$, un bruit blanc gaussien. De plus nous avons vu que

l'effet doppler tend vers 0.

## 5.3 Demonstration $\rho(\delta_t) <= 1$

$\rho(\delta'_t) = \frac{\int_{Tp}^{Tp+\delta'_t} s_p(t-\delta'_t) r_l(t)}{\int_{Tp}^{Tp+\delta'_t} |r_l(t)|^2 \int_0^{Tp} |sp(t)|^2}$

Or d'apres l'inegalité de Cauchy-schwarz:

$\int_{Tp}^{Tp+\delta'_t} s_p(t-\delta'_t) r_l(t) <= \int_{Tp}^{Tp+\delta'_t} |r_l(t)|^2 \int_{Tp}^{Tp+\delta'_t} |sp(t-\delta'_t)|^2$

De plus,sp(t) n'est pas nul uniquement sur l'intervalle [0,Tp]

$\int_{Tp}^{Tp+\delta'_t} s_p(t-\delta'_t) r_l(t) <= \int_{Tp}^{Tp+\delta'_t} |r_l(t)|^2 \int_0^{Tp} |sp(t)|^2$

Donc on peut dire:

$\rho(\delta_t) <= 1$

De plus,à partir de cette formule on peut admettre que le debut su signal rl(t) ressemble le plus au sp(t), signal du preambule, lorsque $\rho(\delta_t)$ est maximal. Ainsi le decalage temporel trouvé est $\delta_t$. En absence de bruit, $\rho(\delta_t) = 1$.

## 5.4 Chaîne de communication prennant en compte les décalages fréquentiel et temporel

En adaptant notre code de la tache 1: ajout du préambule et implémentions du bloc synchronisation,on obtient cette chaîne de communication.

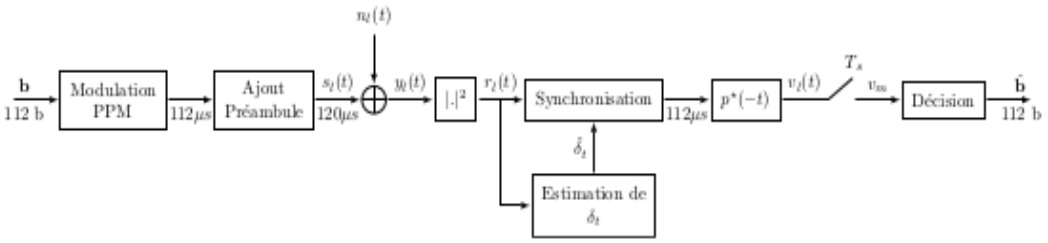



FIGURE 1 – Chaîne de communication complète

avec le signal preambule, sp(t) suivant:

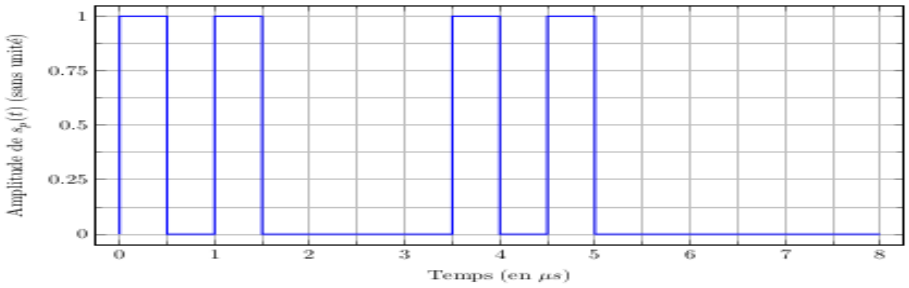
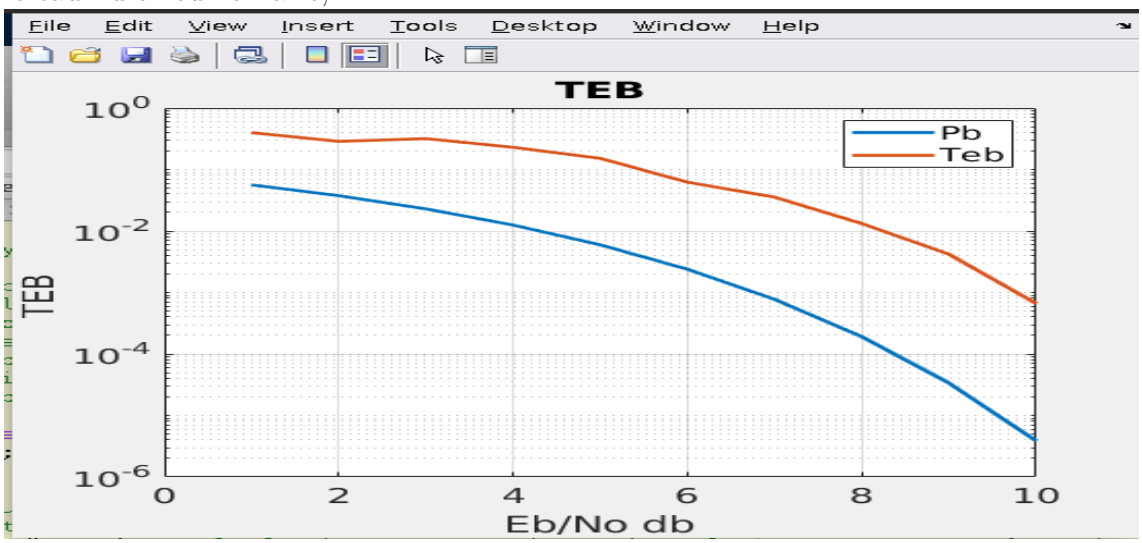



FIGURE 2 – Préambule $s_p(t)$ de $T_p = 8\ \mu s$ débutant les trames ADS-B

D'une maniere similaire à la tache 1,nous avons tracé la probabilité d'erreur théorique, Pb, et le taux d'erreur binaire, TEB.




Nous pouvons remarquer que les 2 courbes ne se superposent pas,cela est du aux erreurs de détermination du décalage temporel. Ces erreurs sont dues à la variance du bruit gaussien élevé qui modifie le préambule au sein de la trame et ainsi le bloc synchronisation a du mal à déterminer de façon juste le décalage temporel. On peut remarquer que pour une valeur de $10^{-3}$ en rapport de nombre de bits reçus faux et nombre totale de bits transmis,nous avons une perte de 3dB par rapport à la probabilité d'erreur théorique. On peut aussi dire qu'on arrivera pas à detecter certaines trames à cause de la robustesse de notre système.

# 6 Tache 6, Couche MAC ADS-B Implémentation de la couche MAC

Le code fonctionnel de la tâche 6 peut-être trouvée dans le fichier test Tache6.m, il permet d'exécuter le code qui est demandé dans la tâche 6 indépendamment du reste. Nous vous invitions à regarder en détail le code de bit2registre. Le prototype de la fonction Bit2registre a été modifié par rapport à celui énoncé dans le sujet, il suit celui qui a été donné dans le squelette du code. Nous avons donc 3 arguments en entrée, bitPacketCRC, refLon, refLat, et un argument en sortie, un registre. Le registre a aussi été légèrement modifié, en effet lors de l'exécution du programme qui fonctionne entièrement avec les fonctions en fonction.p nous avons observé que le registre de sortie avait été modifié. Nous avons donc réalisé les mêmes modifications, par exemple supprimer le champ de la trajectoire qui n'était plus nécessaire avec la présence de l'objet avion.m qui gère de façon autonome la trajectoire.

Lors de la sous-tâches 3 nous avons décodé les informations présente dans toutes les colonnes de adsbmsgs. La trajectoire obtenue est différentes de celle représenté sur le sujet. Un grand nombre de groupe semble avoir obtenue la même trajectoire que nous. De plus lors de la tâche 8 nous obtenons bien les bonnes trajectoires ainsi que les avions.

Les valeurs de FTC qui correspondent à des trames de positions en vol sont comprises entre 5 et 8 inclus.

m

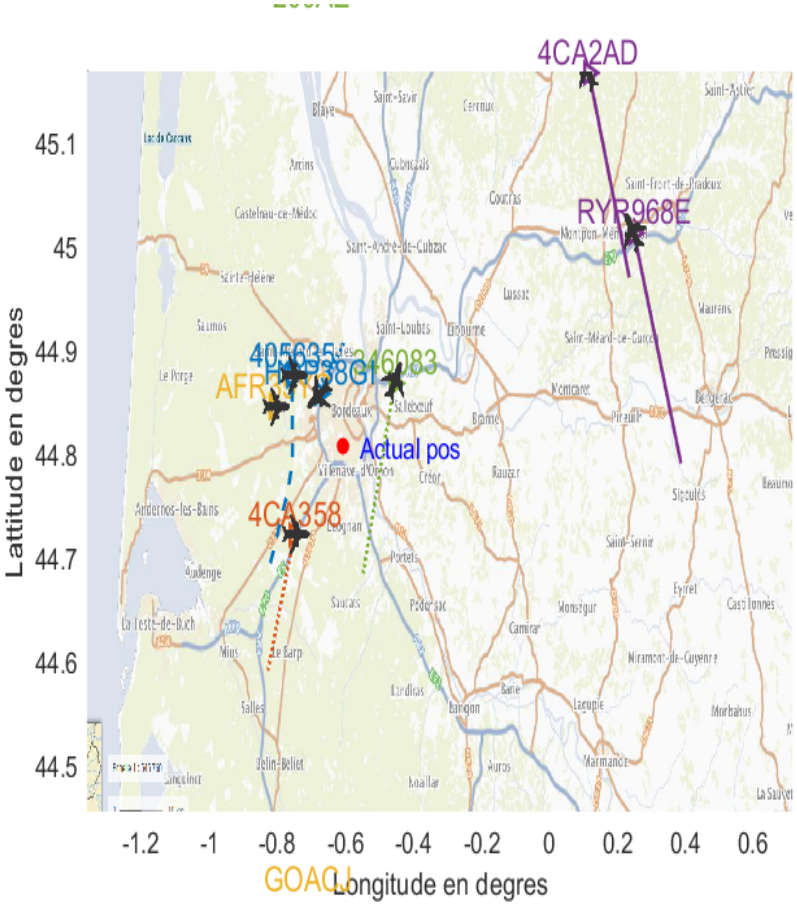
# 7 Tache 8, Couche physique et chaine de communication ADS-B

La tâche 8 représente la mise en commun de l'ensemble du travail qui a été fait précédemment. Lors de cette tâche, les différentes colonnes de buffer.mat vont être traitées pour en extraire les informations. Nous vous invitons à regarder le code présent dans le fichier Tache8.m. Pour

cette tâche voici les fonctions principales qui ont été utilisé: proces-buffer, synchro, bit2registre et update-liste-avion. Le choix des prototypes des fonctions et des variables de retour à été choisi en fonction de l'analyse des données retournée par l'ensemble des fonctions .p.

Process-buffer prend en entrée: un buffer brut, reflon, reflat, seuil-détection et Fse Process-buffer renvoie en sortie: liste-new-registre qui contient l'ensemble des messages adsb avec un crc correct, liste-corrVal contient la corrélation correspondante aux registres obtenue.

Update-liste-avion prend en entrée: une liste qui contient des objets avion, liste-new-registre, liste-corrVal, ainsi que d'autres grandeurs qui nous ont pas été utile. Update-liste-avion renvoie en sortie: la liste des avions qui ont été décoder à travers les différentes colonnes du buffer. La fonction synchro permet d'obtenir la corrélation entre une partie du signal et le préambule.




buffer.png

Le graphique obtenu coïncide avec le graphique fourni dans le sujet, tâche 8. Nous pouvons donc envisager que notre décodage de buffer.mat dans son intégralité semble correct.

Différentes optimisations ont été réalisées: le calcul de certains éléments redondant ont été

effectué en dehors de certaines boucles de calcule, écriture en vectoriel de certaine boucle for, l'ensemble de ces optimisations amène a un temps de décodage d'environs 5-7 second pour un échantillon de 2 millions (tracé des avions compris) soit pour une colonne du buffer. Notre programme met environ 40-50 secondes pour décoder l'ensemble du buffer, nous avons utilisé les fonctions Matlab tic et toc pour observer cela.

# 8 Tache 9, Application Mise en place du temps réel

Le code implémenté dans la tâche 8 permet normalement un décryptage en temps réel,nous avons développé l'ensemble du code pour que cela fonctionne ainsi.Nous n'avons malheureusement pas eu le temps lors des séances de tester correctement son fonctionnement vis-à-vis de notre progression. L'élément suivant, peu créé un dysfonctionnement: le temps d'analyse des échantillons se pourrait être trop long, il faudrait donc ne prendre qu'une certaine partie des valeurs renvoyées par le serveur.