Rapport de projet – Traitement du signal

Florent Puy, Ewen Le Bihan

ENSEEIHT, département Sciences du Numérique

Table des matières

1 Introduction

Dans ce projet, nous implémentons un modem suivant les règles V21 de l'union internationale des télécommunications (UIT) en Matlab. Nous utiliserons la méthode de la modulation en fréquence numérique.

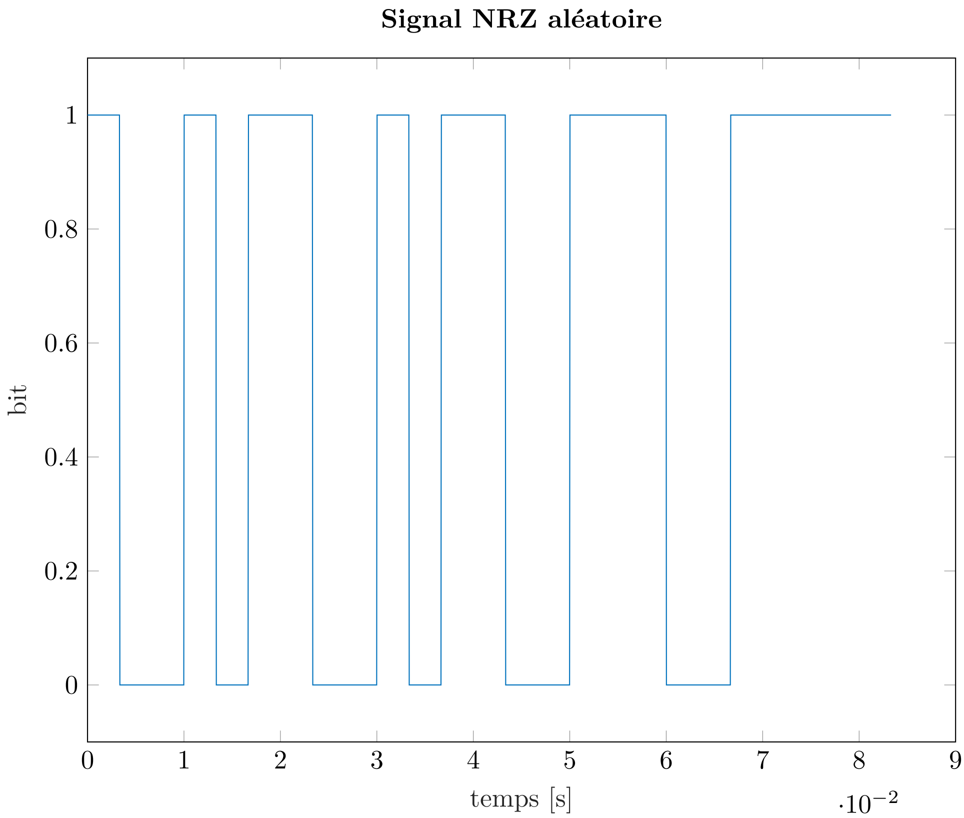
2 Modem en fréquence

Nous allons tout d'abord réaliser un signal NRZ binaire à partir duquel nous construirons ensuite un signal sinusoïdal modulé fréquence $F_0 = 6000\,\text{Hz}$ pour les bits 0 et de fréquence $F_1 = 2000\,\text{Hz}$ pour les bits 1.

Nous comparerons ensuite les densités spectrales de puissance théoriques et expérimentales du signal NRZ et du signal modulé en fréquence.

2.1 Génération d'un signal NRZ

On génère tout d'abord un signal NRZ prennant deux valeurs, 0 ou 1, générées aléatoirement d'une durée $T_s = 1/300s$. On effectue cela sur N_s périodes. Voici les résultats ainsi obtenus. Pour calculer le signal NRZ depuis un vecteur binaire de taille $1 \times N_{\text{bits}}$, fait le produit tensoriel de Kronecker entre le vecteur binaire et un vecteur comportant N_s fois le bit 1.

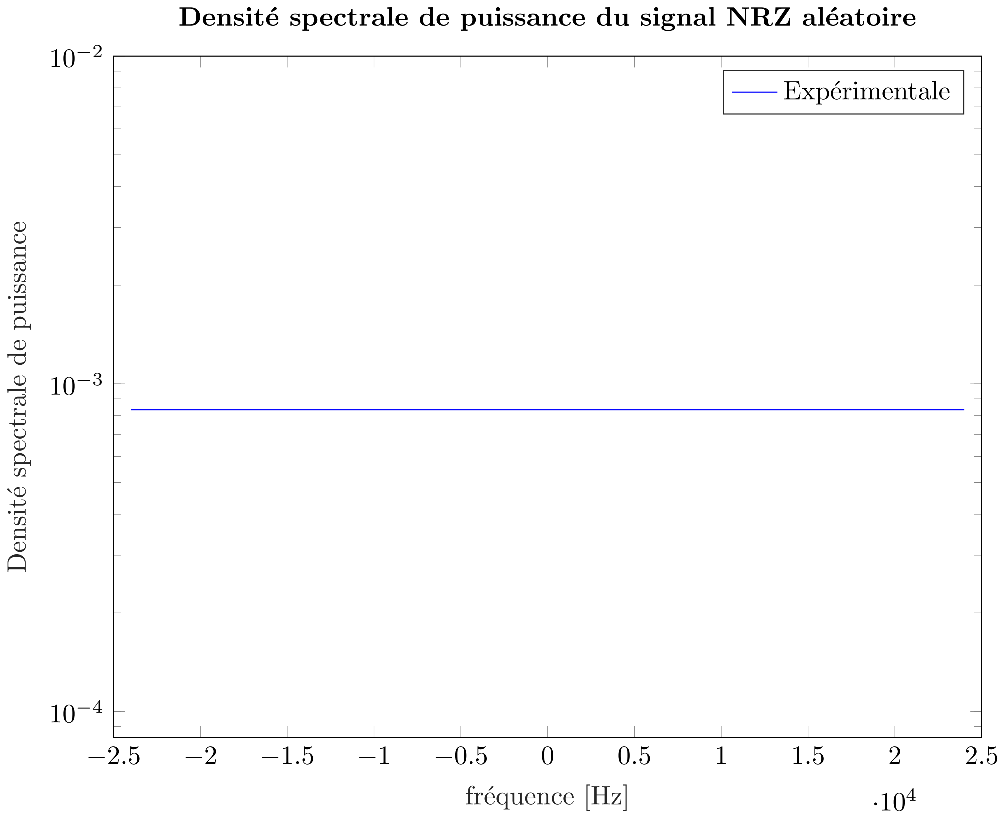


On calcule ensuite la densité spectrale de puissance de ce signal NRZ en utilisant la fonction *pwelch* de Matlab utilisant un périodogramme de Welch.

On calcule ensuite la densité spectrale théorique vue en cours d'un signal NRZ :

$$S_{\mathrm{NRZ}}(f) = \frac{1}{4} T_s \operatorname{sinc}^2(\pi f T_s) + \frac{1}{4}\delta(f)$$

On peut déormais comparer les densités spectrales de puissance théoriques et expérimentales :



2.2 Génération d'un signal modulé en fréquence

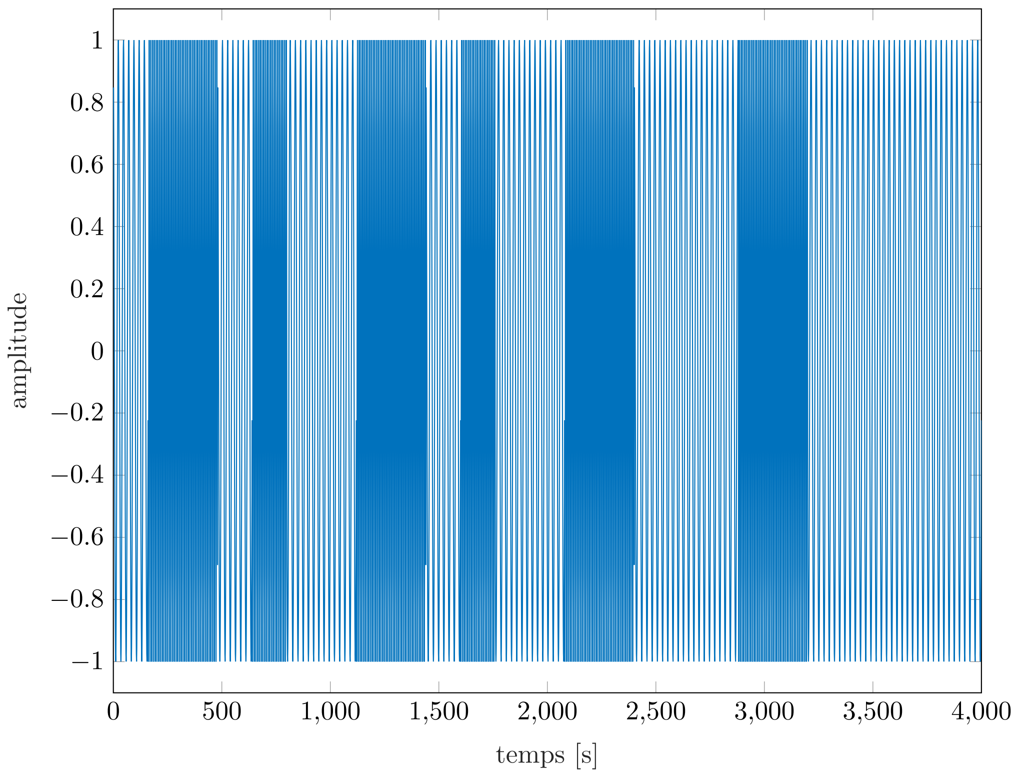
Pour construire notre signal modulé en fréquence nous allons nous baser sur le signal NRZ et sur une simple sinusoïdale. Lorsque le signal NRZ vaut 0, la fréquence de la sinusoïdale sera $F_0 = 1180\,\mathrm{Hz}$ et lorsqu'il vaut 1 la sinusoïdale sera de fréquence $F_1 = 980\,\mathrm{Hz}$. Au final, notre signal modulé suit la formule suivante :

$$\text{modulé}(t) = \mathrm{NRZ}(t)\cos(2\pi F_1 t + \phi_1) + (1 - \mathrm{NRZ}(t))\cos(2\pi F_0 t + \phi_1)$$

ϕ_1 et ϕ_2 étant des déphasages tirés aléatoirement dans $[0, 2\pi]$ et $\mathrm{NRZ}(t)$ le signal NRZ.

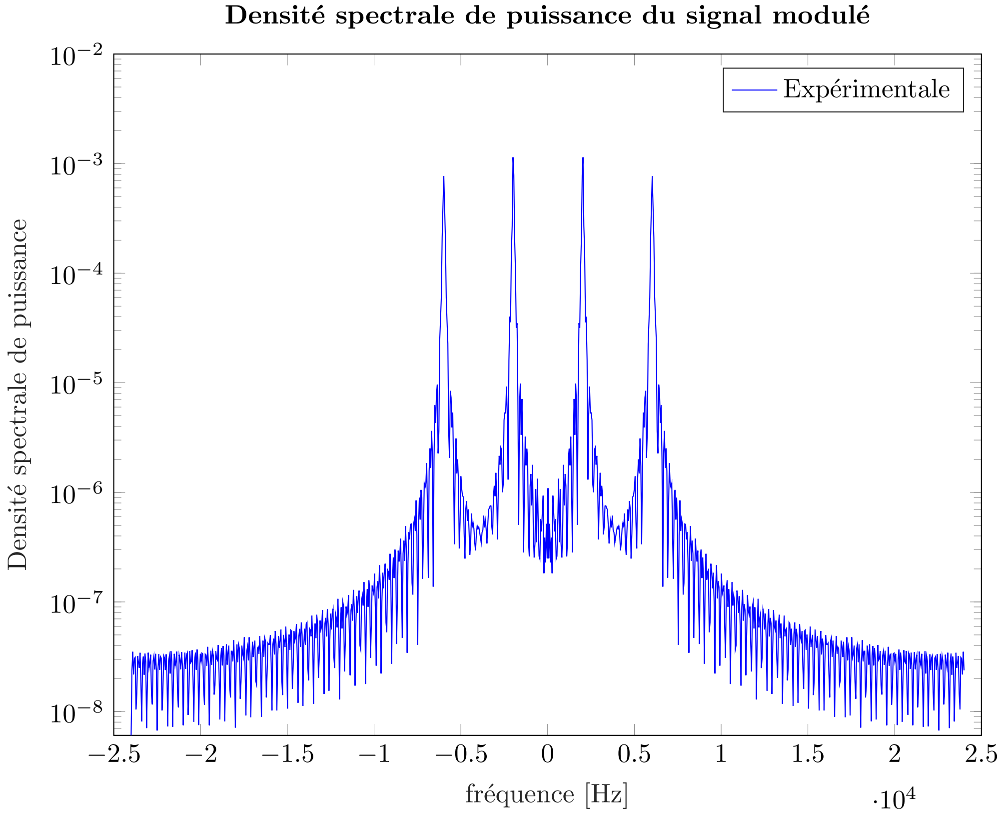
On obtient ainsi le signal suivant :

Modulation du signal NRZ aléatoire (bit 0 sur 6000Hz, bit 1 sur 2000Hz)



$$
\begin{aligned}
s_x(f) &= |\,\mathrm{TF}(x)(f)|^2 \\
&= |\mathrm{TF}(\mathrm{NRZ}(t)\cos(2\pi F_1 t + \phi_1)) + \mathrm{TF}((1 - \mathrm{NRZ}(t))\cos(2\pi F_0 t + \phi_0))|^2 \\
&= |\,\mathrm{TF}\,\mathrm{NRZ}(f) * \mathrm{TF}(\cos(2\pi F_1 t)\cos\phi_1 - \sin(2\pi F_1 t)\sin\phi_1) + \mathrm{TF}(1 - \mathrm{NRZ})(f) * \mathrm{TF}(\cos(2\pi F_0 t)\cos\phi_0 - \sin(2\pi F \\
&= |\,\mathrm{TF}\,\mathrm{NRZ}(f) * \frac{1}{2}\left(\cos\phi_1(\delta(f - F_1) + \delta(f + F_1)) - \frac{1}{i}\sin\phi_1(\delta(f - F_1) - \delta(f + F_1))\right) + \mathrm{TF}(1 - \mathrm{NRZ})(f) * \frac{1}{2}(\\
&= |\frac{1}{4}\left(\delta(f) + \mathrm{sinc}^2(\pi f T_s)\right) * \frac{1}{2}\left(\cos\phi_1(\delta(f - F_1) + \delta(f + F_1)) - \frac{1}{i}\sin\phi_1(\delta(f - F_1) - \delta(f + F_1))\right) + \left(\delta(f) - \frac{1}{4}\delta\right.
\end{aligned}
$$

On calcule ensuite la densité spectrale de puissance de ce signal NRZ en utilisant la fonction `pwelch` de Matlab utilisant un périodogramme de Welch. On obtient ceci :



3 Canal de transmission à bruit additif, blanc et Gaussien

Dans cette section, nous allons tenter de simuler un bruit blanc Gaussion que nous additionnerons à notre signal modulé en fréquence afin de modéliser le signal reçu par le modem. Le bruit simulé sera généré aléatoirement grâce au module `rand` de Matlab et sera de puissance σ^2 avec :

$$\sigma = \sqrt{\frac{S_{\text{module}}}{10^{\text{SNR}/10}}}$$

avec S_{module} représentant la densité spectrale de puissance du signal modulé en fréquence et SNR le rapport signal sur bruit (signal to noise ratio) que nous fixerons à 10 par la suite.

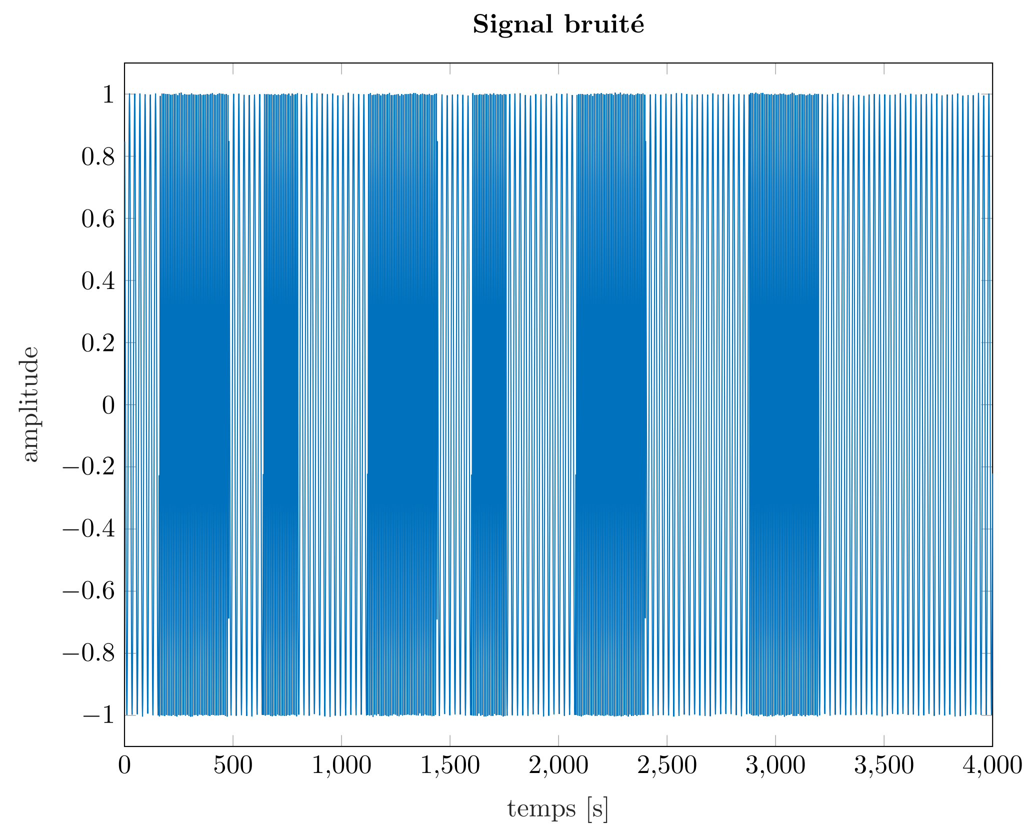


FIGURE 1 – Signal bruité

4 Démodulation par filtrage

On souhaite désormais reconstituer le signal de départ. Pour cela, nous allons procéder à un filtrage passe bas d'une part et passe haut d'autre part, avec une fréquence de coupure $F_c = \frac{F_0+F_1}{2}$. Nous ferons ensuite passer chacun des signaux filtrés par un détecteur d'énergie qui permettra de reproduire de signal binaire initial de manière fidèle.

Filtre passe haut Pour le passe haut, nous allons utiliser un filtre de réponse impulsionnelle suivante :

$$h_{\text{haut}}(t) = \frac{2F_c}{F_e}\,\text{sinc}(2F_c t)$$

et donc

$$H_{\text{haut}}(f) = \text{TF}(h_{\text{haut}}(t))$$

Filtre passe bas

$$H_{\text{bas}}(f) = 1 - H_{\text{haut}}(f)$$

Les réponses des filtres sont les suivantes :

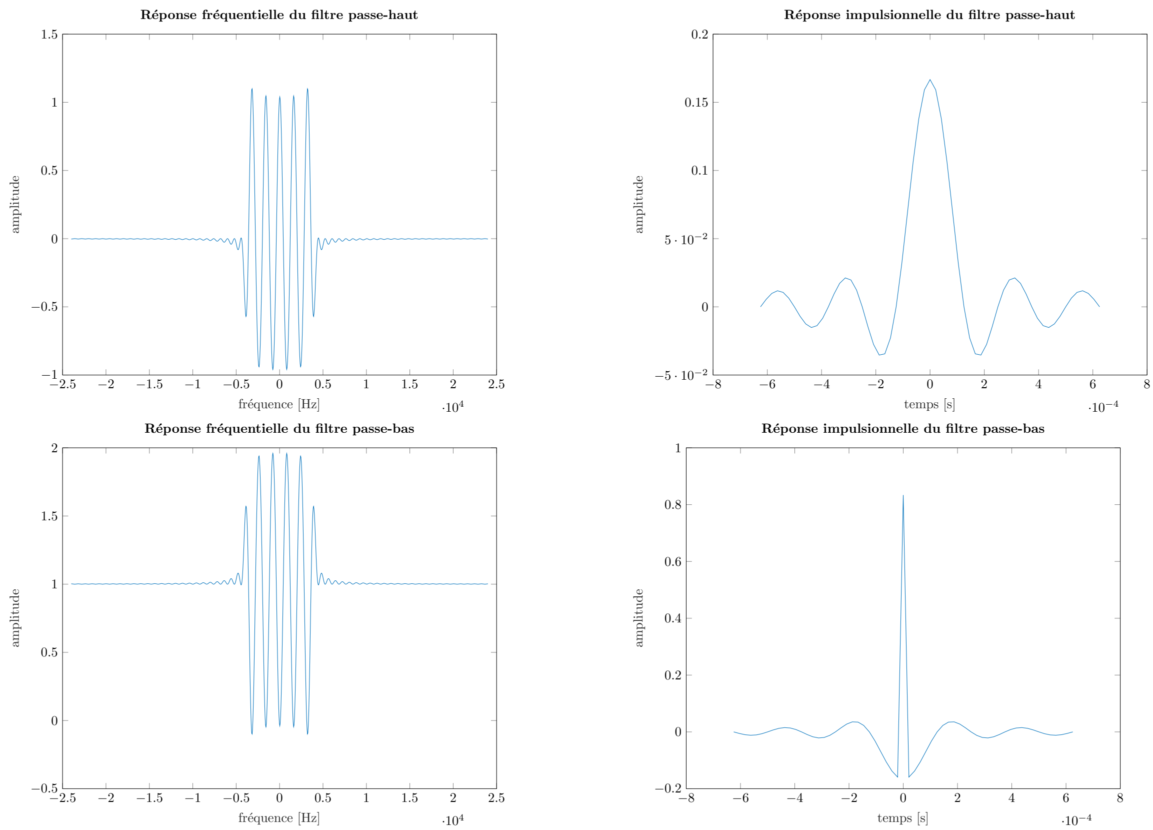
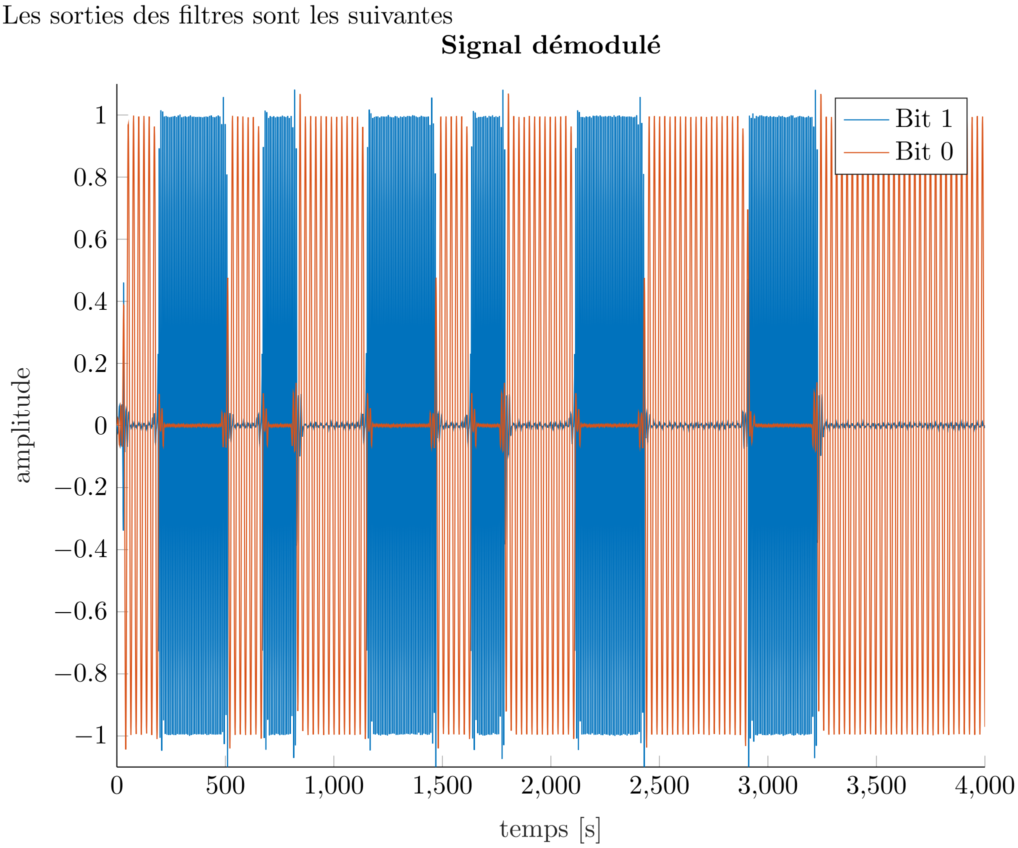


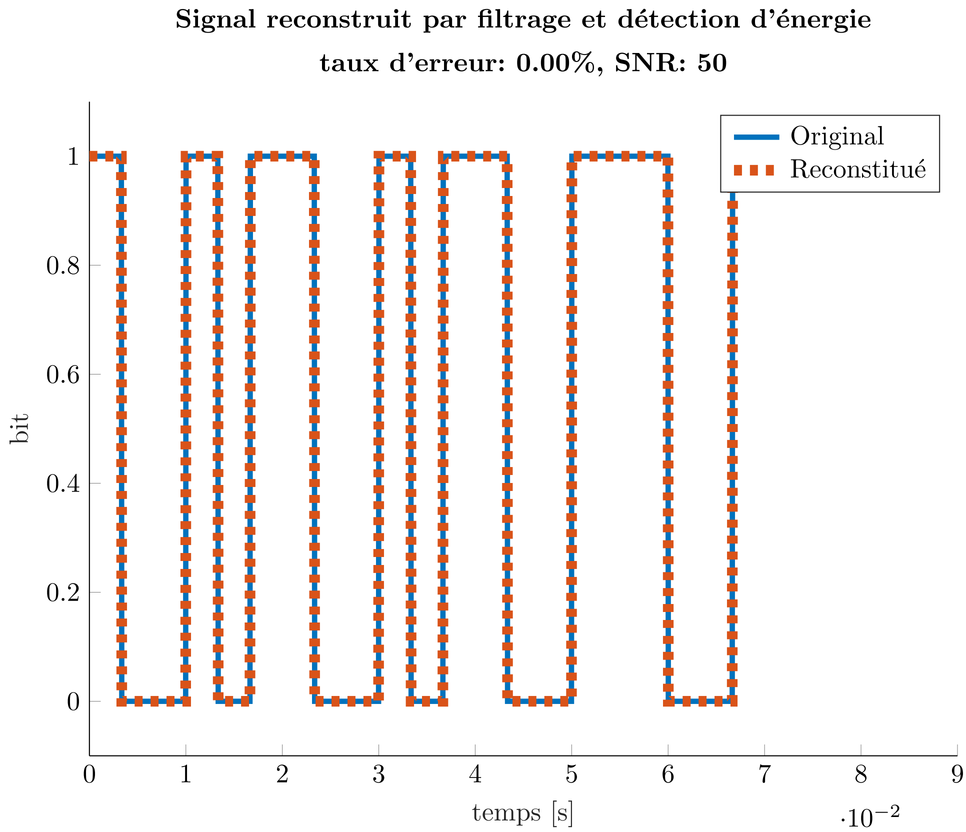
FIGURE 2 – Réponses des filtres

Les sorties des filtres sont les suivantes



4.1 Détection d'énergie

Avec ces signaux ainsi filtrés, nous désirons reconstituer le signal de base. Pour cela nous allons utiliser un détecteur d'énergie. Nous divisons nos signaux en périodes T_s et sur chaque période



nous calculons l'énergie suivant la formule suivante :

$$E = \sum_{i=1}^{N_s} x_n^2$$

Enfin, on compare cette énergie à un seuil K qu'on fixera à la moyenne des énergies du signal. Pour le signal en sortie du passe bas par exemple, si $E > K$ alors le signal reconstitué sera égal à 1 sur cette période T_s, sinon il sera égal à 0.

Voici les figures obtenues grâce à cette méthode.

4.2 Modification du démodulateur

4.2.1 Modification du nombre de coefficients

Avec 201 coefficients pour le filtre, le taux d'erreur est grand.

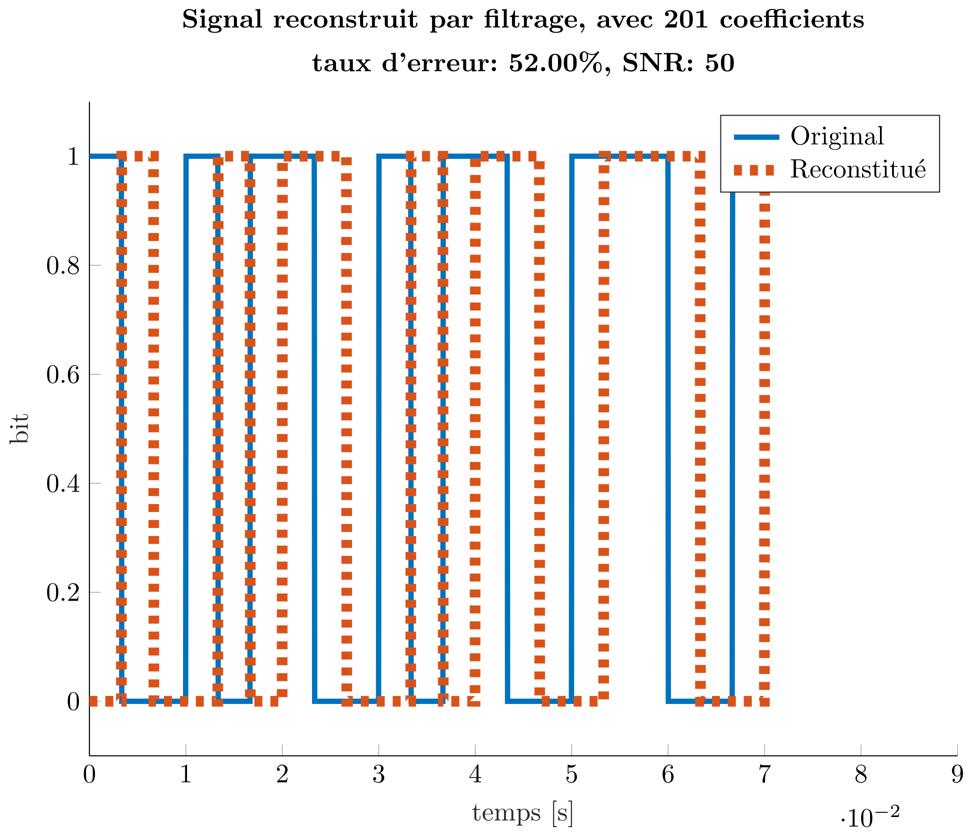


FIGURE 3 – Avec 201 coefficients

5 Démodulateur de fréquence adapté à la norme V21

Avec des fréquences porteuses beaucoup plus rapprochées, il devient difficile de démoduler en séparant le signal par filtrage haut et bas.

On observe donc un taux d'erreur non-nul.

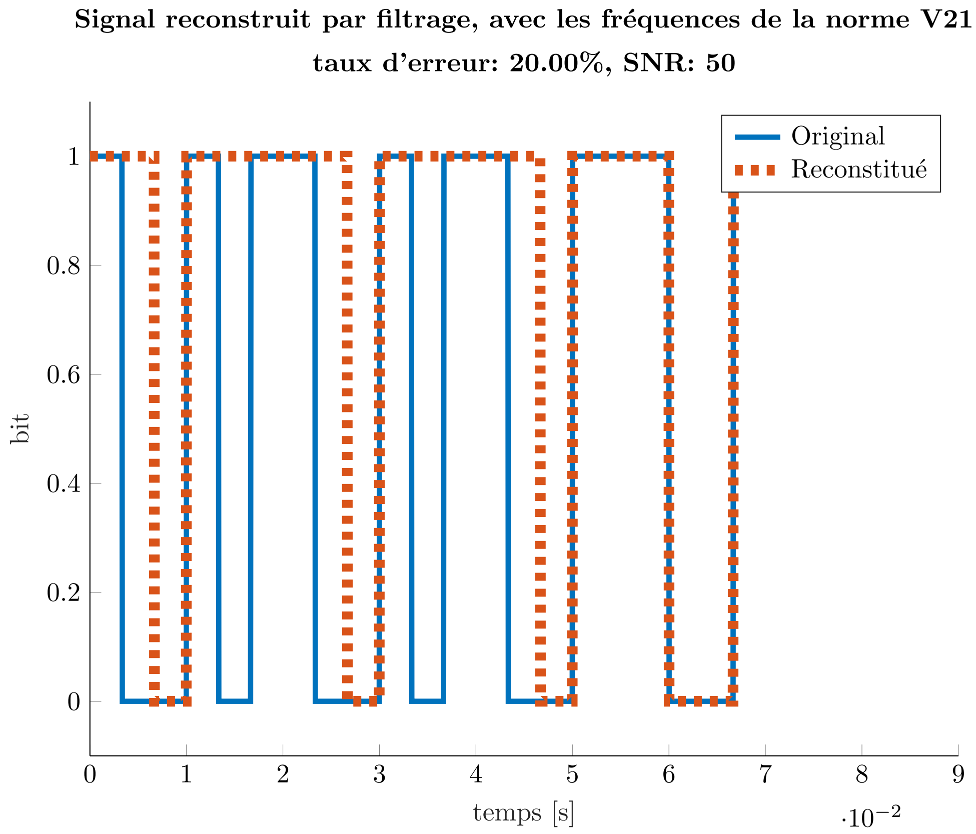


FIGURE 4 – Avec les fréquences de la norme V21

5.1 Contexte de synchronisation idéale

Nous allons ici introduire une nouvelle méthode de démodulation plus adaptée à la norme V21. La multiplication du signal avec le cosinus correspondant (par exemple) au bit 1 donne, en fonction du temps, une mesure de la synchronisation entre le signal et ce cosinus : plus le résultat est proche de 1, plus les signaux sont synchronisés à cet instant, et donc plus le signal est susceptible d'être un bit 1. On a :

$$\int_0^{T_s} \cos^2(2\pi F_0 t + \phi_0)\,\mathrm{d}t = \frac{1}{8\pi F_0}\sin(2(2\pi F_0 T + phi_0)) + 4\pi F_0 T - \sin(2\phi_0)$$
$$\int_0^{T_s} \cos^2(2\pi F_0 t + \phi_1)\,\mathrm{d}t = \frac{1}{8\pi F_0}\sin(2(2\pi F_0 T + phi_1)) + 4\pi F_0 T - \sin(2\phi_1)$$
$$\int_0^{T_s} \cos(2\pi F_0 t + \phi_0)\cos(2\pi F_0 t + \phi_1)\,\mathrm{d}t = \frac{1}{8\pi F_0}\sin(4\pi F_0 T + phi_0 + phi_1) + 4\pi F_0 T\cos(phi_0 - phi_1) - \sin(\phi_0 + \phi_1)$$

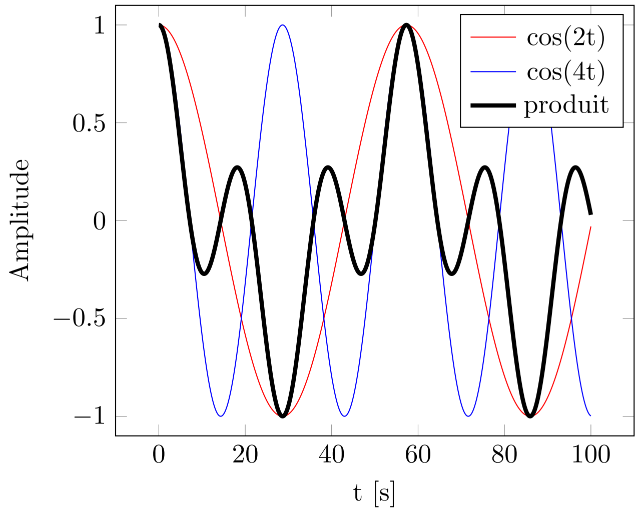


FIGURE 5 – Synchronisation entre deux cosinus de fréquences différentes

On fait ensuite une accumulation (une moyenne en quelque sorte) de ces mesures sur une période T_s en intégrant sur T_s, pour avoir une idée de la synchronisation avec le signal d'un bit 1 et d'un bit 0 à chaque instant échantilloné.

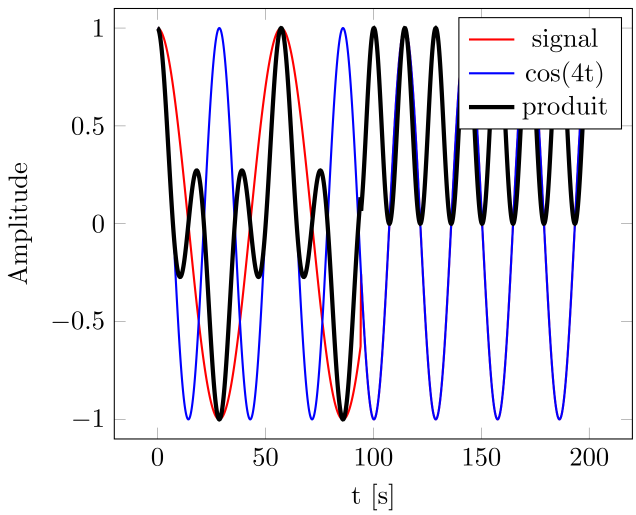


FIGURE 6 – Synchronisation entre le signal obtenu et le signal théorique d'un bit

Finalement, pour chaque échantillon temporel (de durée T_s), on effectue une comparaison :

$$\text{reconstitué}(kT_s) = \begin{cases} 1 & \text{si } \text{sync}_1(kT_s) - \text{sync}_0(kT_s) > 0 \\ 0 & \text{sinon} \end{cases}$$

En notant sync_b le produit pour le bit b.
Le signal démodulé avec cette méthode est le suivant :

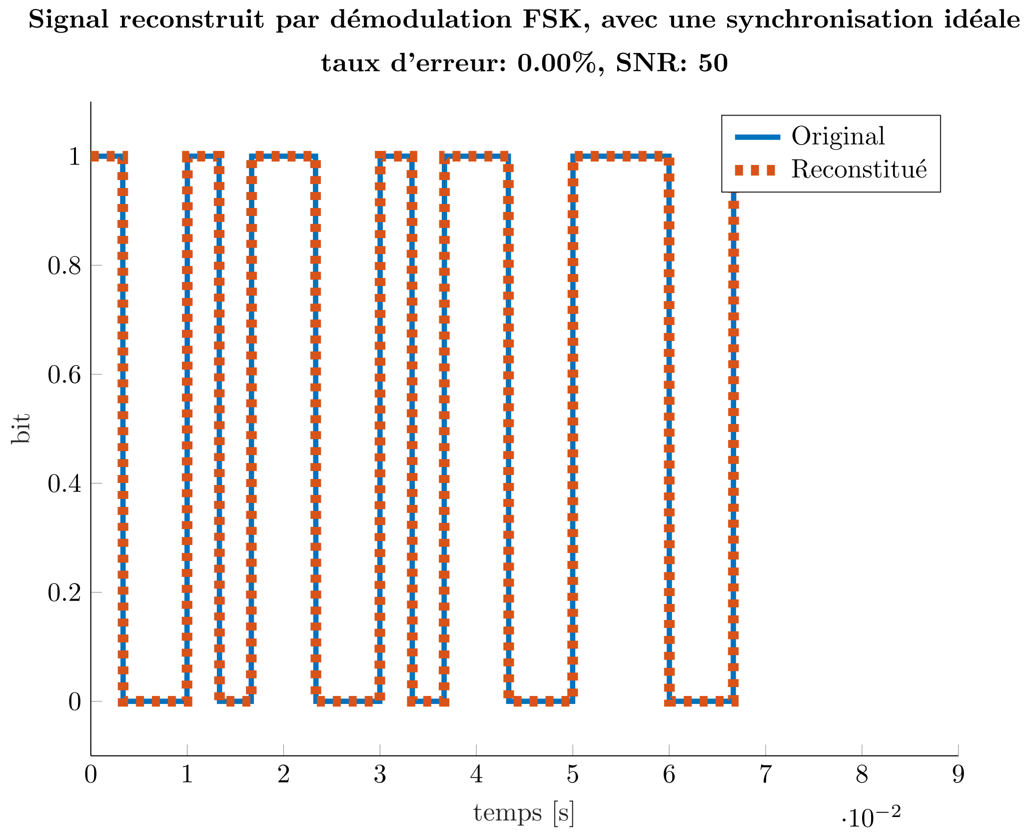


FIGURE 7 – Signal démodulé en supposant une synchronisation parfaite

5.2 Gestion d'une erreur de synchronisation de phase porteuse

Un sinus est déphasé d'un quart de phase, comparé à un cosinus de même fréquence et même déphasage.

En rajoutant ces mesures de désynchronisation, on prend en compte les signaux d'entrée qui seraient déphasés : si le signal n'est pas synchronisé avec le cosinus à cause d'un léger déphasage, l'ajout d'une mesure de synchronisation avec ce même cosinus, mais déphasé de $\frac{\pi}{2}$ compensera la faible valeur de synchronisation.

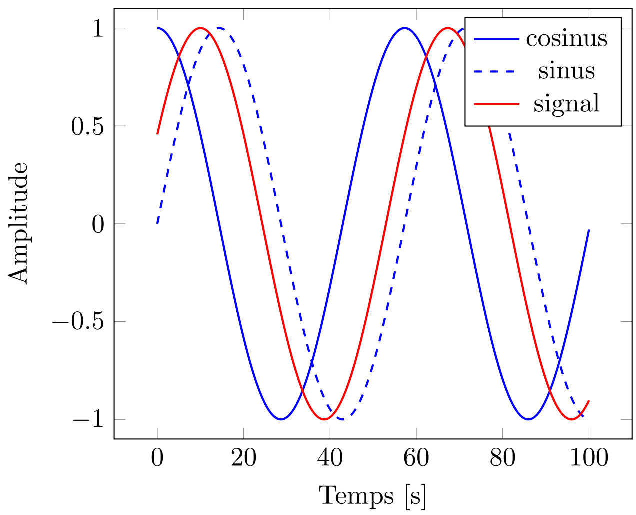
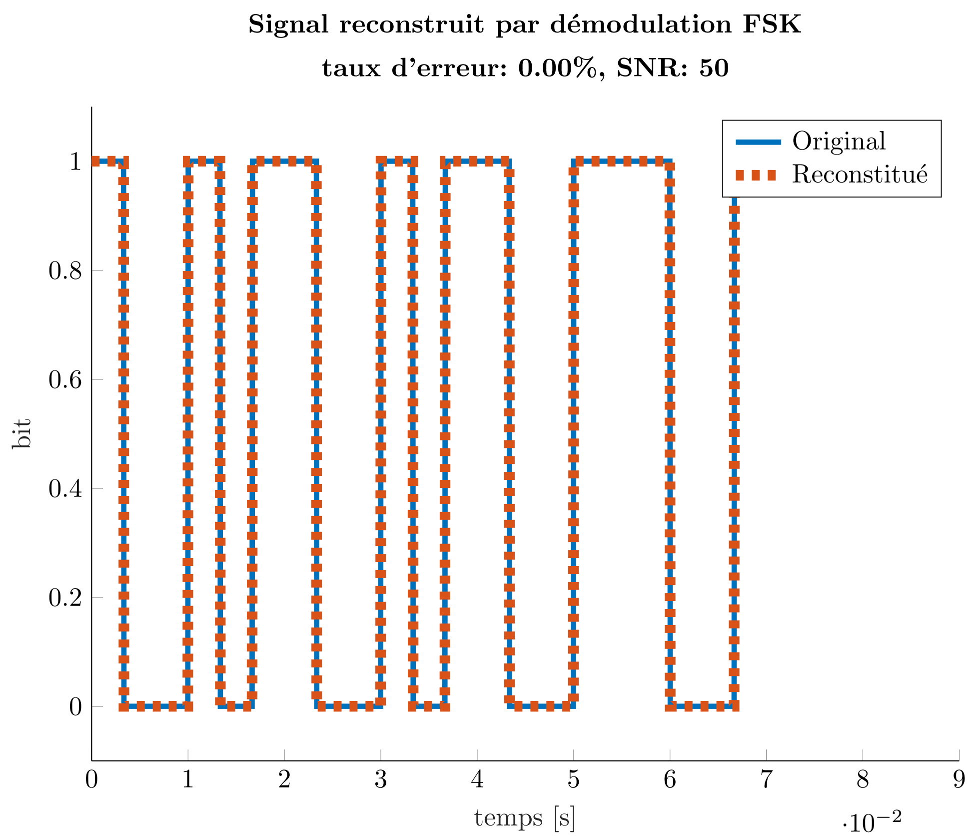


FIGURE 8 – Compensation d'un déphasage du signal d'entrée

Le reste du processus reste le même qu'en 5.1



Signal reconstruit par démodulation FSK
taux d'erreur: 0.00%, SNR: 50
Original
Reconstitué
bit
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4
5
6
7
8
9
temps [s]
$\cdot 10^{-2}$