

# Rapport de projet – Traitement du signal

Florent Puy, Ewen Le Bihan

ENSEEIH, département Sciences du Numérique

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>3</b>	<b>Modem en fréquence</b>	<b>1</b>
3.1	Génération d'un signal NRZ . . . . .	2
3.2	Signal NRZ . . . . .	2
3.3	Densité spectrale de puissance . . . . .	2
3.4	Génération d'un signal modulé en fréquence . . . . .	3
3.4.1	Modulation du signal NRZ . . . . .	3
3.4.2	Densité spectrale de puissance . . . . .	4
<b>4</b>	<b>Canal de transmission à bruit additif, blanc et Gaussien</b>	<b>5</b>
<b>5</b>	<b>Démodulation par filtrage</b>	<b>6</b>
5.1	Détection d'énergie . . . . .	8
5.2	Modification du démodulateur . . . . .	8
5.2.1	Modification du nombre de coefficients . . . . .	8
5.2.2	Utilisation des fréquences de la norme V21 . . . . .	9
<b>6</b>	<b>Démodulateur de fréquence adapté à la norme V21</b>	<b>10</b>
6.1	Contexte de synchronisation idéale . . . . .	10
6.2	Gestion d'une erreur de synchronisation de phase porteuse . . . . .	12
6.2.1	Image mystère . . . . .	13

## 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.

## 3 Modem en fréquence

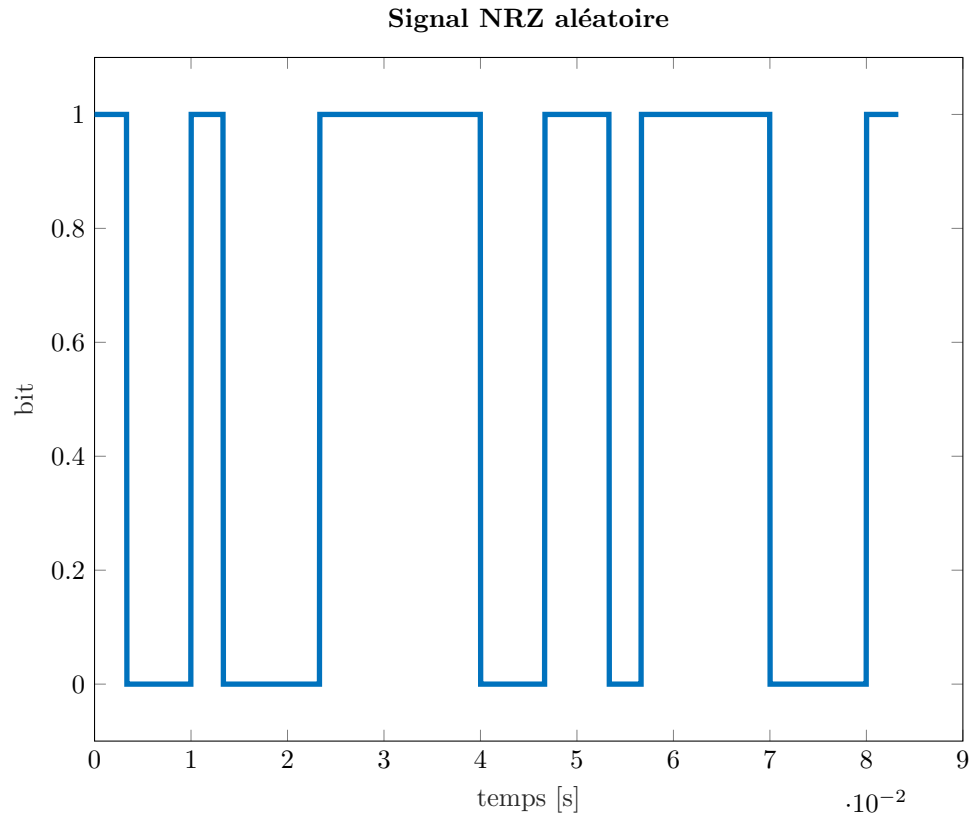
Nous allons tout d'abord réaliser un signal NRZ binaire à partir duquel nous construirons ensuite un signal sinusoïdal modulé fréquence  $F_0 = 6000 \text{ Hz}$  pour les bits 0 et de fréquence  $F_1 = 2000 \text{ Hz}$  pour les bits 1.

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

### 3.1 Génération d'un signal NRZ

### 3.2 Signal NRZ

On génère tout d'abord un signal NRZ prenant 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.



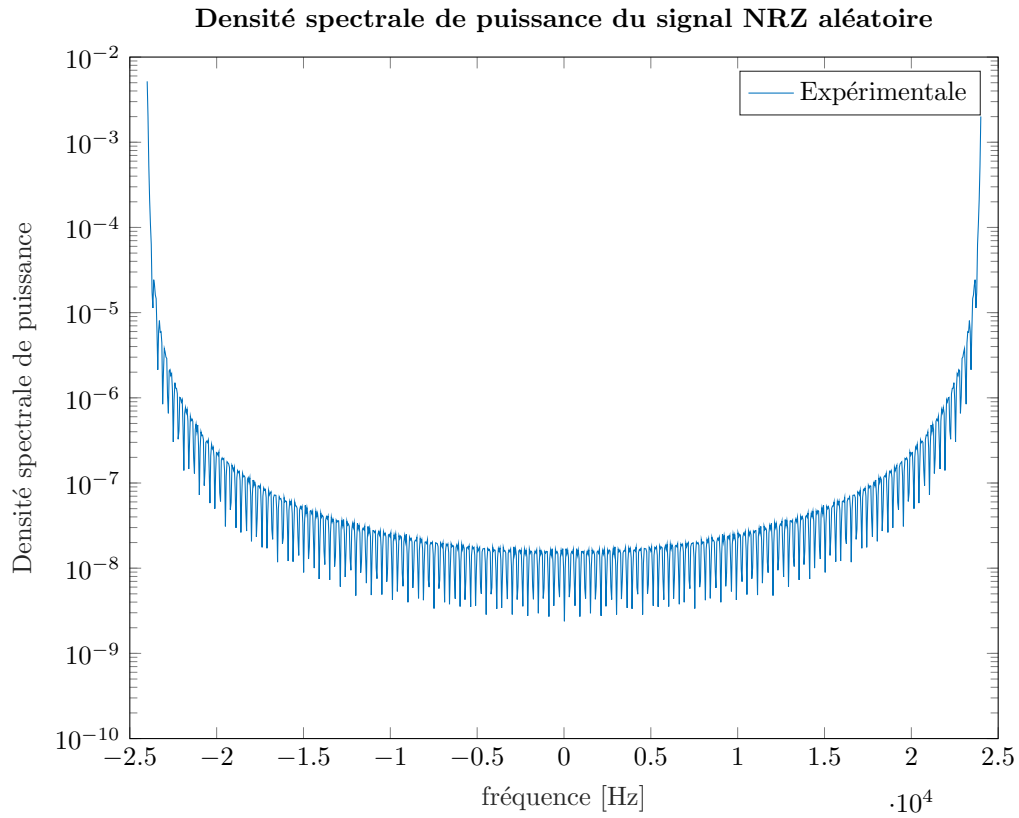
### 3.3 Densité spectrale de puissance

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

Puis la DSP théorique :

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

Voici la DSP expérimentale :



### 3.4 Génération d'un signal modulé en fréquence

#### 3.4.1 Modulation du signal NRZ

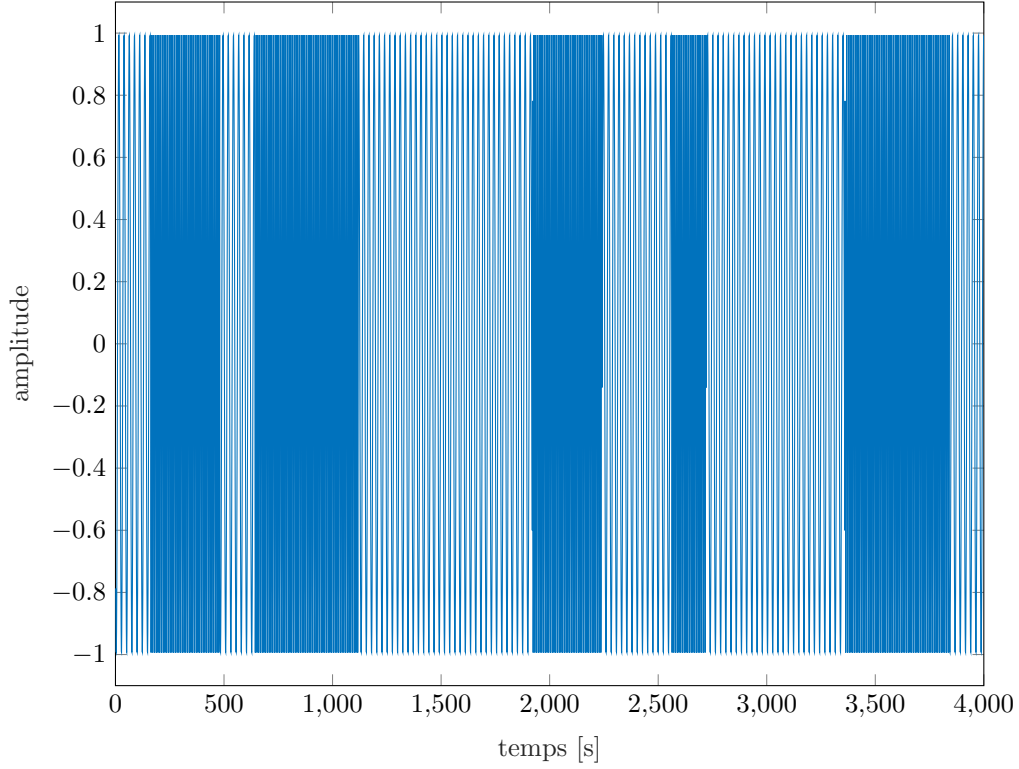
Pour construire notre signal modulé en fréquence nous allons nous baser sur le signal NRZ et sur une simple sinusoïdale. Lorsque le signal NRZ vaut 0, la fréquence de la sinusoïdale sera  $F_0 = 1180 \text{ Hz}$  et lorsqu'il vaut 1 la sinusoïdale sera de fréquence  $F_1 = 980 \text{ Hz}$ . Au final,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

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

$\phi_1$  et  $\phi_2$  étant des déphasages tirés aléatoirement dans  $[0, 2\pi]$  et  $\text{NRZ}(t)$  le signal NRZ.

On obtient ainsi le signal suivant :

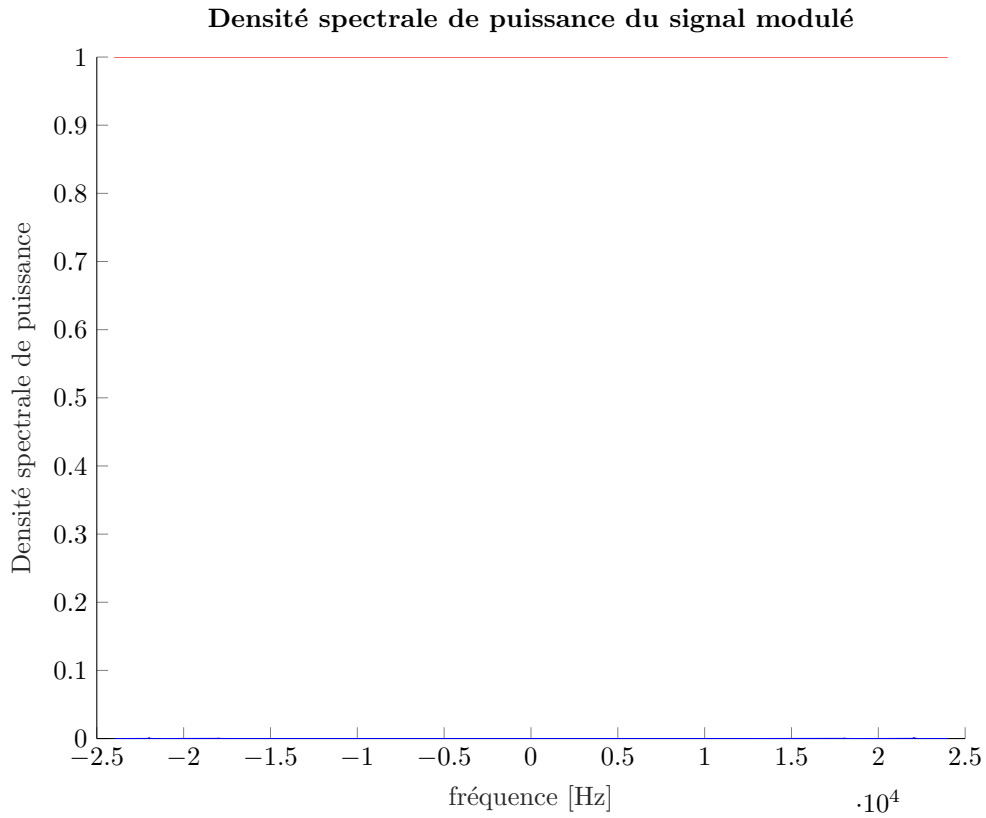
### Modulation du signal NRZ aléatoire (bit 0 sur 6000Hz, bit 1 sur 2000Hz)



#### 3.4.2 Densité spectrale de puissance

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

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 :



## 4 Canal de transmission à bruit additif, blanc et Gaussien

Dans cette section, nous allons tenter de simuler un bruit blanc Gaussien 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  $\sigma^2$  avec :

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

avec  $S_{\text{module}}$  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.

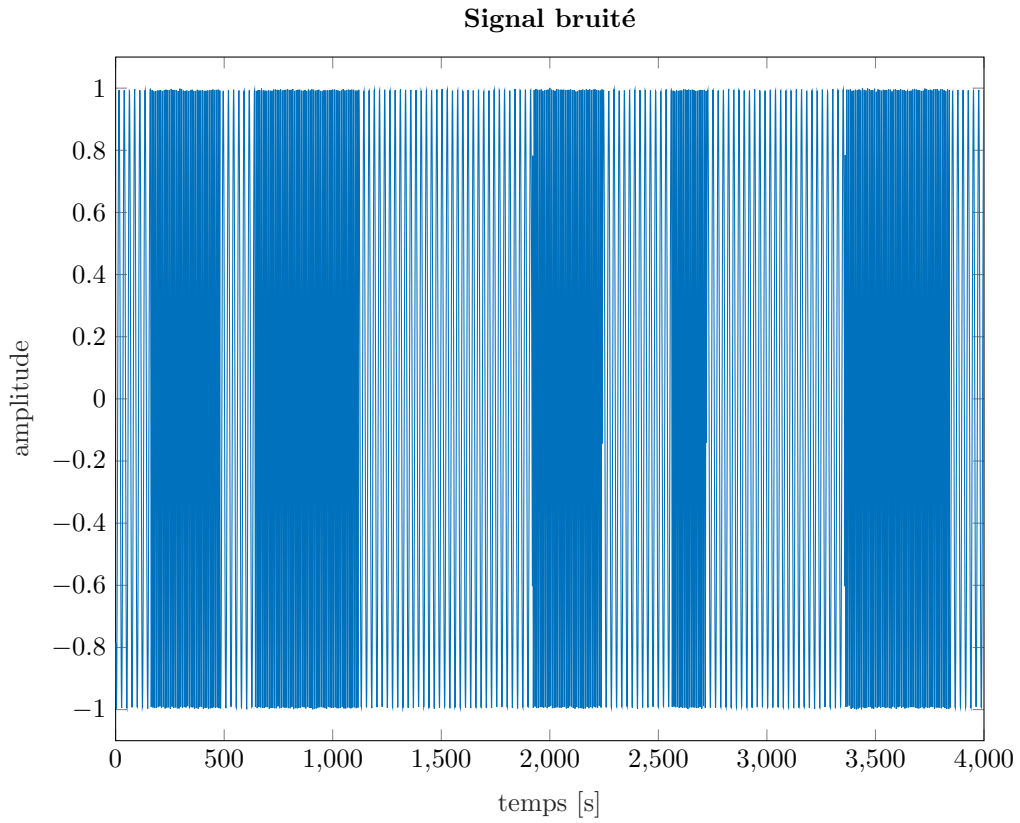


FIGURE 1 – Signal bruité

## 5 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**

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

Les réponses des filtres sont les suivantes :

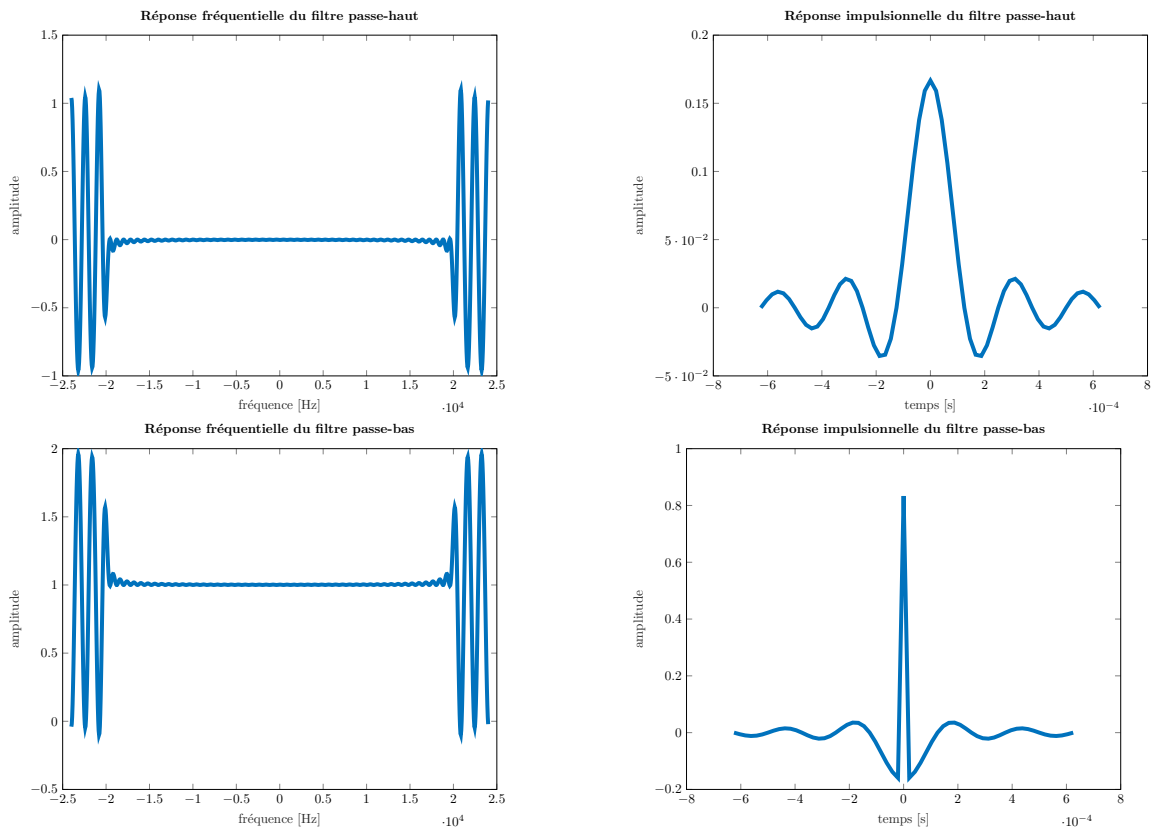


FIGURE 2 – Réponses des filtres

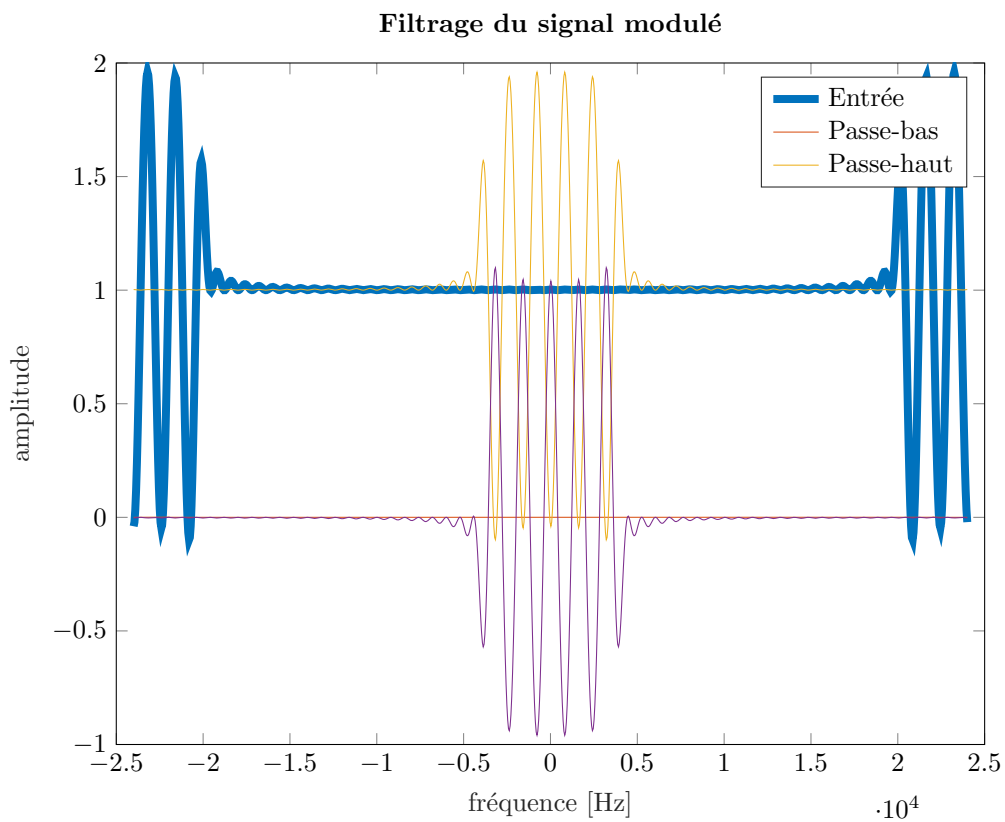
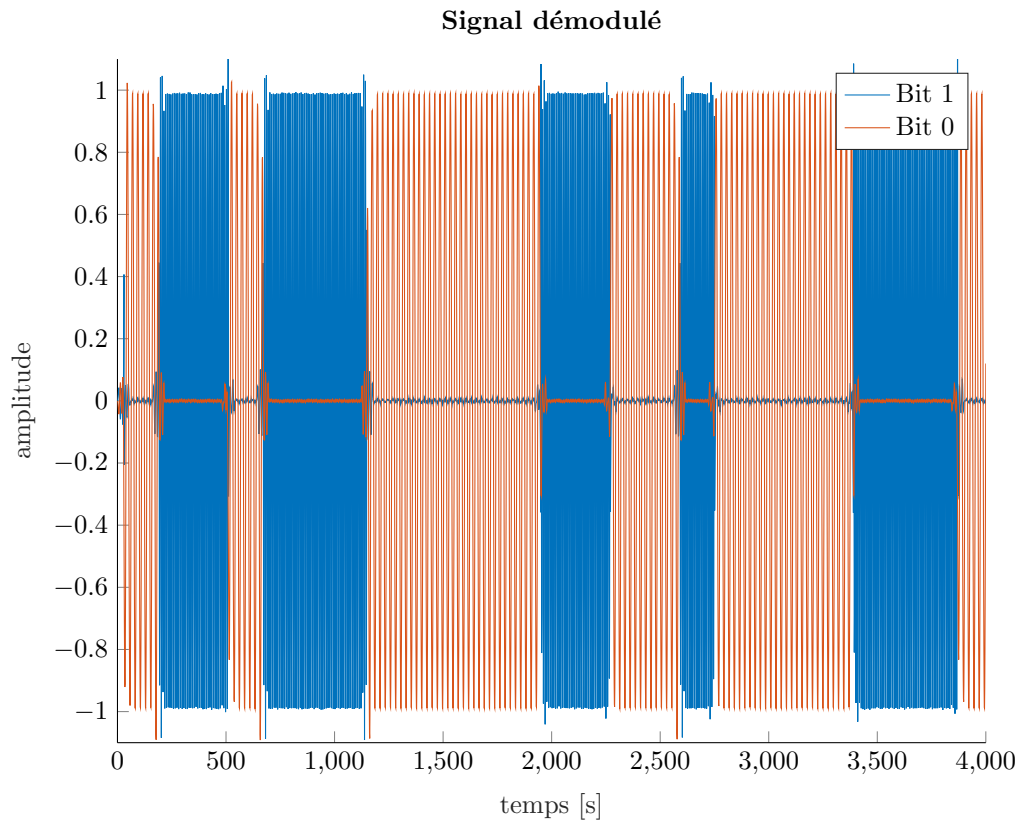


FIGURE 3 – Densités spectrales de puissance lors du filtrage

On remarque que le support de la DSP de l'entrée est quasi disjoint aux supports des réponses fréquentielles des filtres.

Les sorties des filtres sont les suivantes



## 5.1 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 :

## 5.2 Modification du démodulateur

### 5.2.1 Modification du nombre de coefficients

Avec 201 coefficients pour le filtre, le taux d'erreur est grand. Voici le signal démodulé qu'on obtient :



The plot shows two identical digital signals, 'Original' (solid blue line) and 'Reconstitué' (dashed orange line), plotted against time. The x-axis represents time in seconds, ranging from 0 to 9, with a multiplier of  $10^{-2}$  at the bottom right. The y-axis represents the bit value, ranging from 0 to 1. Both signals start at 1 bit. Transitions to 0 bit occur at approximately 0.3, 1.3, 2.3, 4.1, 5.3, 5.7, and 7.1 seconds. Transitions back to 1 bit occur at approximately 0.7, 1.7, 2.7, 4.3, 5.5, 5.9, and 8.1 seconds. The signals are perfectly aligned, indicating no loss or corruption in the reconstruction process.

### 5.2.2 Utilisation des fréquences de la norme V21

9

en utilisant ces fréquences une augmentation du taux d'erreur, il devient difficile de démoduler en séparant le signal par filtrage haut et bas.

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.

**Signal reconstruit par filtrage, avec les fréquences de la norme V21**  
**taux d'erreur: 0.00%, SNR: 50**

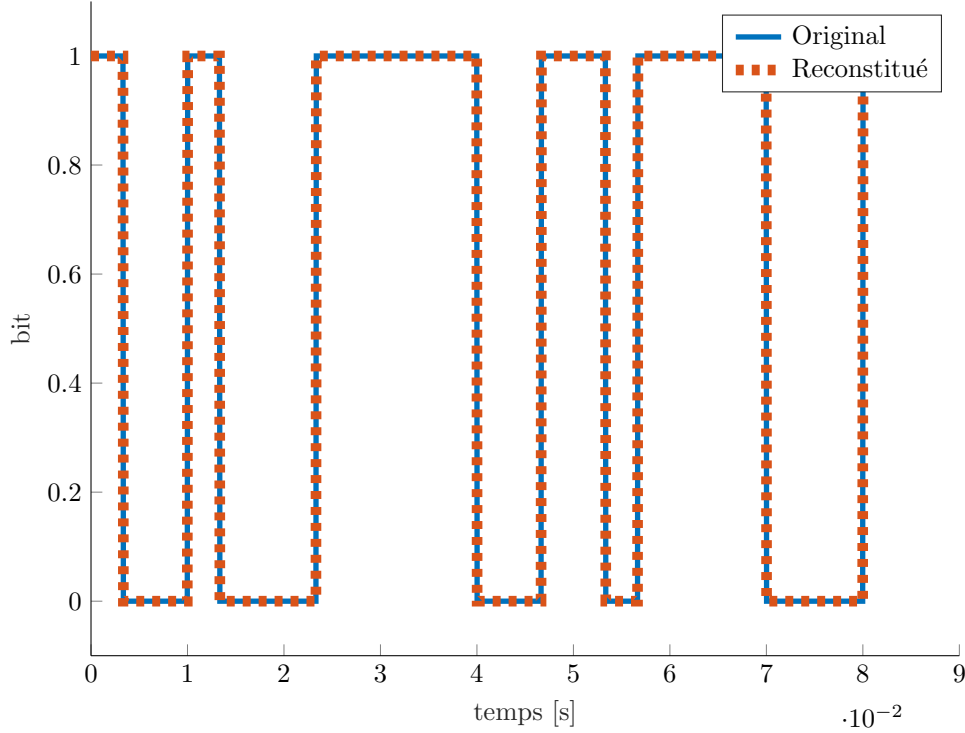


FIGURE 5 – Avec les fréquences de la norme V21

## 6 Démodulateur de fréquence adapté à la norme V21

Nous avons vu dans la partie précédente que la méthode par filtrage n'était pas optimale. Nous allons ici introduire une nouvelle méthode de démodulation plus adaptée à la norme V21.

### 6.1 Contexte de synchronisation idéale

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 :

$$\begin{aligned}
 \int_0^{T_s} \cos^2(2\pi F_0 t + \phi_0) dt &= \frac{1}{8\pi F_0} \sin(2(2\pi F_0 T + \phi_0)) + 4\pi F_0 T - \sin(2\phi_0) \\
 \int_0^{T_s} \cos^2(2\pi F_0 t + \phi_1) dt &= \frac{1}{8\pi F_0} \sin(2(2\pi F_0 T + \phi_1)) + 4\pi F_0 T - \sin(2\phi_1) \\
 \int_0^{T_s} \cos(2\pi F_0 t + \phi_0) \cos(2\pi F_0 t + \phi_1) dt &= \frac{1}{8\pi F_0} \sin(4\pi F_0 T + \phi_0 + \phi_1) \\
 &\quad + 4\pi F_0 T \cos(\phi_0 - \phi_1) - \sin(\phi_0 + \phi_1)
 \end{aligned}$$

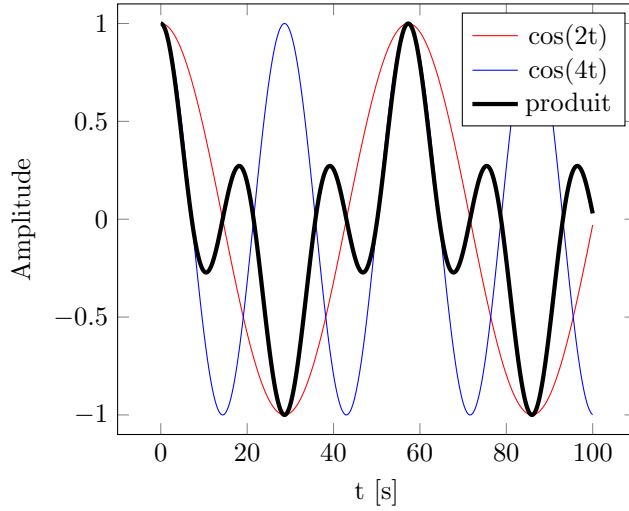


FIGURE 6 – Synchronisation entre deux cosinus de fréquences différentes

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 échantillonné.

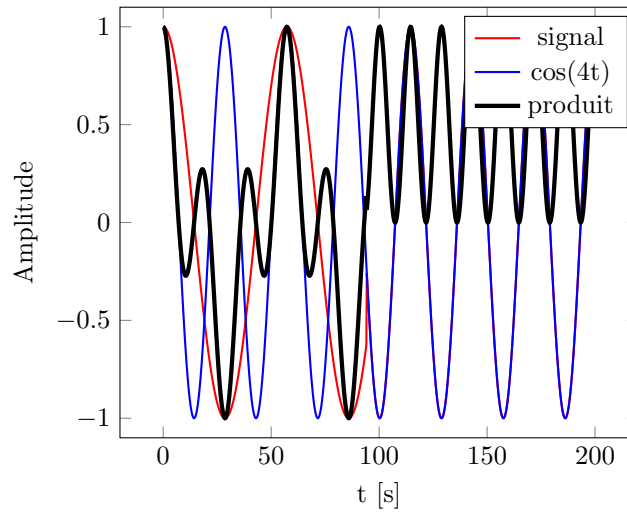


FIGURE 7 – Synchronisation entre le signal obtenu et le signal théorique d'un bit

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  $\text{sync}_b$  le produit pour le bit  $b$ .

Le signal démodulé avec cette méthode est le suivant :

**Signal reconstruit par démodulation FSK, avec une synchronisation idéale**  
**taux d'erreur: 0.00%, SNR: 50**

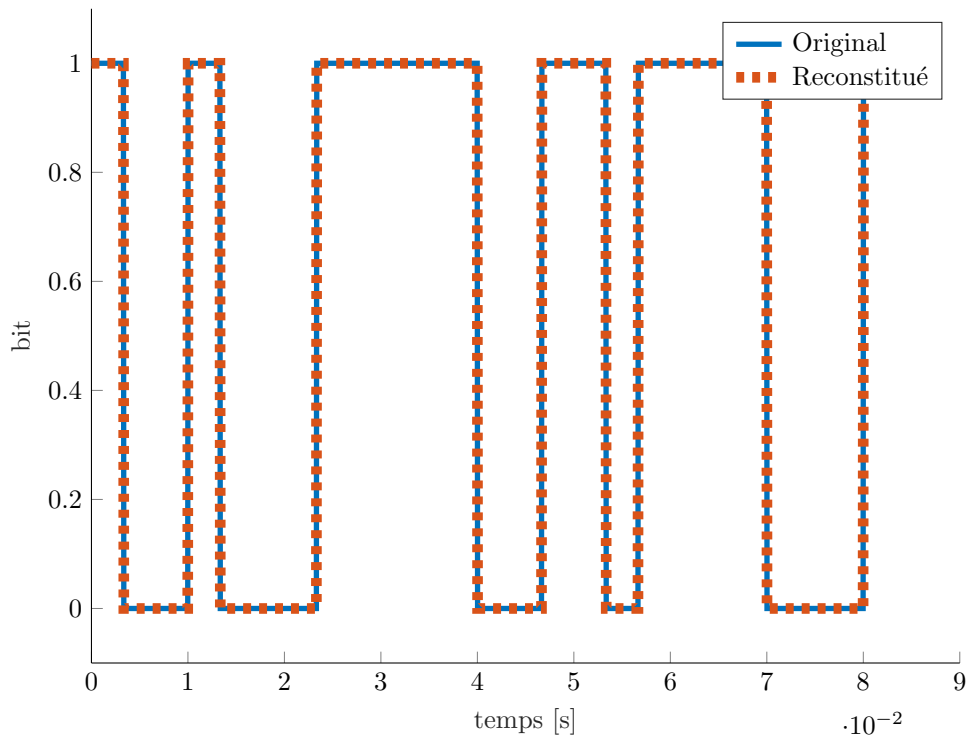


FIGURE 8 – Signal démodulé en supposant une synchronisation parfaite

## 6.2 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.

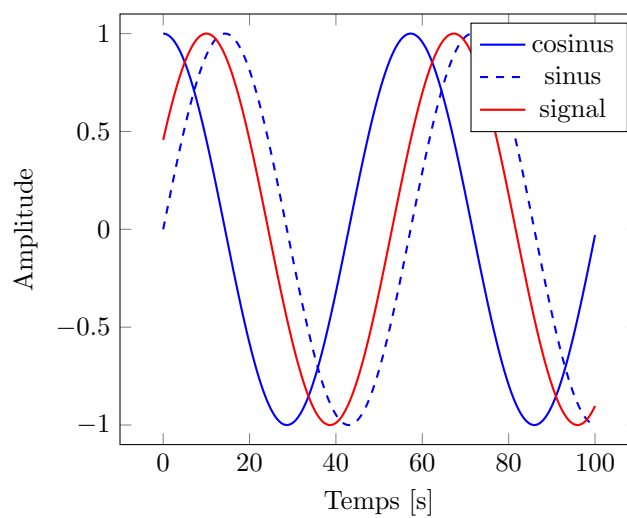
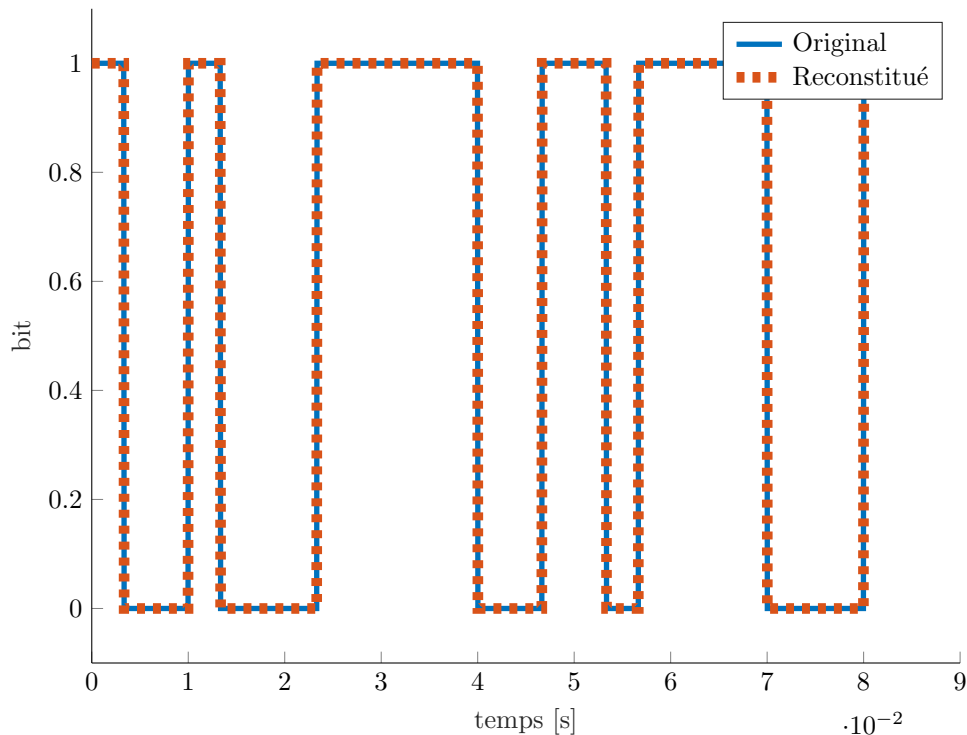


FIGURE 9 – Compensation d'un déphasage du signal d'entrée

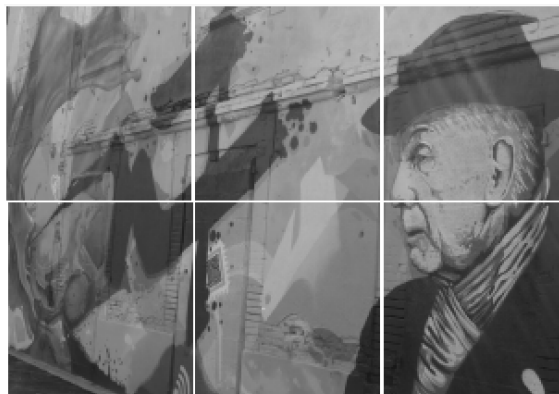
Le reste du processus reste le même qu'en 6.1.

taux d'erreur: 0.00%, SNR: 50



### 6.2.1 Image mystère

On peut désormais utiliser cette méthode de démodulation pour démoduler les images proposées. On divise les coefficients de la matrice par 255 et on assemble le puzzle pour obtenir l'image suivante :



Il s'agit du grand et fameux Charles Camichel! Normalien de bonne facture, il fut prof d'élec à l'université de Lille de 1895 à 1900. En 1907, à la suite d'une dépression causée par le mauvais temps du nord de la France, il s'installe à Toulouse et crée l'institut électrotechnique de Toulouse. Cette école évoluera jusqu'à avoir un nom beaucoup trop compliqué : école nationale supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications. Heureusement, le groupe centrale est en route pour venir secourir cette école en perdition en la renomant "Institut Centrale Toulouse", faisant payer ce changement de nom aux élèves en multipliant par quatre les frais d'inscription.

C'est une photo de l'ancien Foy' de l'ENSEEIH, situé très précisément aux coordonnées **43°36'06.7"N 1°27'18.3"E**