



Réalisation d'un modem de fréquence selon la recommandation V21 de l'Union Internationale des Télécommunications (UIT)

Nom des auteurs

LAANAIYA Mahmoud

LE GRANCHE Oscar

Département Sciences du Numérique - Première année
2020-2021

Table des matières

1	Introduction	3
2	Modem de fréquence - Démodulation par filtrage	3
2.1	Construction du signal modulé en fréquence	3
2.1.1	Génération du signal NRZ	3
2.1.2	Génération du signal modulé en fréquence	3
2.2	Canal de transmission à bruit additif, blanc et Gaussien	4
2.3	Démodulation par filtrage	5
2.3.1	Synthèse du filtre passe-bas	5
2.3.2	Synthèse du filtre passe-haut	5
2.3.3	Filtrage, tracés, commentaires sur les résultats obtenus	5
2.3.4	Détection d'énergie	6
2.4	Application de la recommandation V21	8
3	Modem de fréquence V21 - Démodulateur FSK	8
3.1	Démodulateur FSK - Contexte de synchronisation idéale	8
3.2	Démodulateur FSK avec gestion d'une erreur de synchronisation de phase porteuse	9
3.2.1	Déphasage aléatoire	9
4	Conclusion	9
5	Références	11

Table des figures

1	Signal modulé en fréquence	3
2	NRZ	4
3	notre signal	4
4	Démodulation par filtrage.	5
5	notre signal	6
6	DSPx	6
7	notre passe bas	6
8	Sortie en basses fréquences	7
9	notre passe haut	7
10	sortie en passe haut	7
11	image reconstituée en passe bas avec bruit	8
12	image reconstituée en passe haut avec bruit	8
13	Démodulation FSK. Synchronisation supposée idéale.	9
14	image reconstituée en FSK	9
15	image confus	10
16	image reconstituée malgré le déphasage	10
17	Démodulation FSK - Gestion d'une erreur de phase porteuse.	10

1 Introduction

L'objectif de ce projet était de construire un signal modulé extrait des données des bits, puis le démoduler dans un premier temps par filtrage, puis par un démodulateur FSK. Les deux méthodes ont pour objectif de reconstituer les bits initiaux avec un taux d'erreur minimal.

2 Modem de fréquence - Démodulation par filtrage

2.1 Construction du signal modulé en fréquence

La première étape du projet consiste à réaliser la modulation de fréquence, i.e. transformer l'information binaire à transmettre en un signal modulé en fréquence (exemple sur la figure 16). Le signal modulé en fréquence $x(t)$ est généré de la manière suivante :

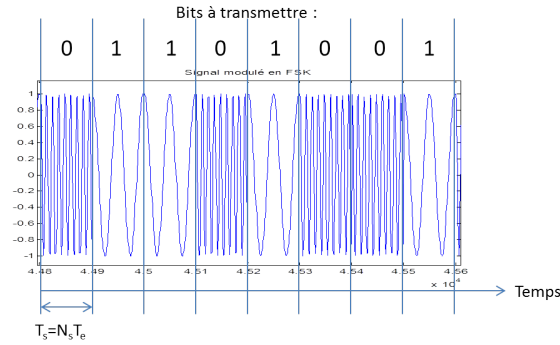


FIGURE 1 – Signal modulé en fréquence

$$x(t) = (1 - NRZ(t)) \times \cos(2\pi F_0 t + \phi_0) + NRZ(t) \times \cos(2\pi F_1 t + \phi_1) \quad (1)$$

où $NRZ(t)$ est un signal de type NRZ polaire formé à partir de la suite de bits à transmettre en codant les 0 et les 1 par des niveaux 0 et 1 de durée T_s secondes. ϕ_0 et ϕ_1 sont des variables aléatoires indépendantes uniformément réparties sur $[0, 2\pi]$ qui peuvent être obtenues sous matlab en utilisant $rand*2*pi$.

2.1.1 Génération du signal NRZ

1. On fait un load des bits générés par un des fichier `DonneesBinomei.mat`.
2. On génère le vecteur de taille de l'échantillon (`ones(1,Ns)`).
3. On obtient le NRZ par le produit de Kroneker entre le vecteur bits et le vecteur de taille de l'échantillon

2.1.2 Génération du signal modulé en fréquence

- On introduit quatre fréquences, deux pour la démodulation par fréquence (6000 Hz et 2000 Hz) et les autres pour la démodulation en FSK (1180 Hz et 980Hz).
- On génère le signal modulé qu'on va utilisé pour la démodulation par filtrage. (Voir l'équation(1)).
- On génère ensuite celui qu'on va utilisé pour la démodulation FSK noté x_{FSK} . (Voir l'équation (1)).

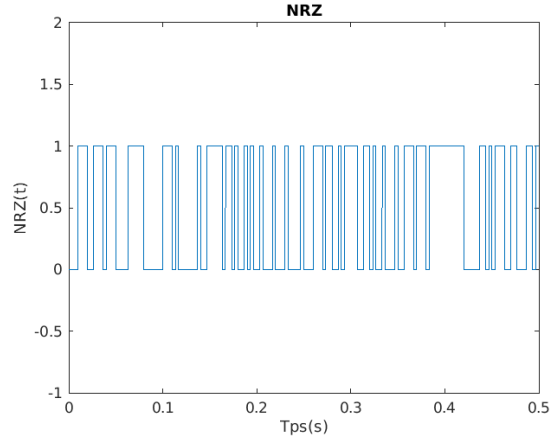
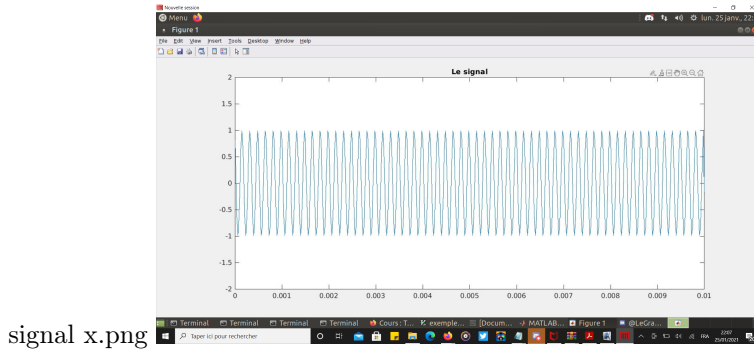


FIGURE 2 – NRZ



signal x.png

FIGURE 3 – notre signal

2.2 Canal de transmission à bruit additif, blanc et Gaussien

Nous allons considérer que le canal de propagation ajoute au signal émis un bruit que l'on suppose blanc et Gaussien et qui modélise les perturbations introduites.

La puissance du bruit Gaussien à ajouter devra être déduite du rapport signal sur bruit (SNR : Signal to Noise Ratio) souhaité pour la transmission donné en dB :

$$SNR_{dB} = 10 \log_{10} \frac{P_x}{P_b}$$

où P_x représente la puissance du signal modulé en fréquence et P_b la puissance du bruit ajouté. On génère le bruit

$$bruit = \sqrt{(SNR_{dB})} * randn(1, length(x))$$

On génère un signal bruité

$$x_{bruit} = x + bruit$$

Et on conserve un autre sans bruit

$$x_{sansbruit} = x$$

En effet, on va faire deux études, une avec un signal bruité, et l'autre avec un signal sans bruit.

2.3 Démodulation par filtrage

La figure 4 présente le récepteur implanté pour retrouver, à partir du signal modulé en fréquence bruité, le message binaire envoyé.

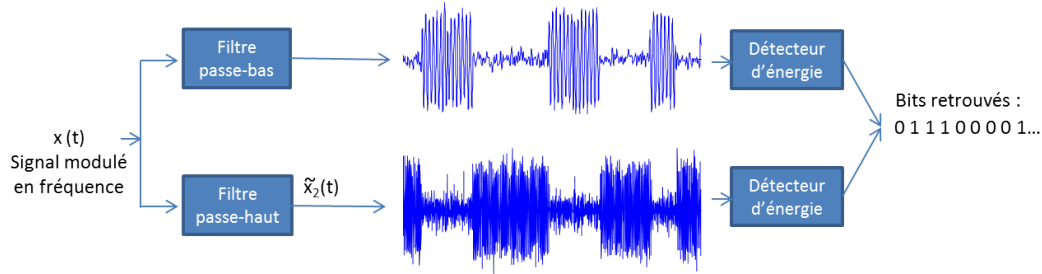


FIGURE 4 – Démodulation par filtrage.

Un filtre passe-bas permet de filtrer les morceaux de cosinus à la fréquence $F_0 = 6000\text{Hz}$, tandis qu'un filtre passe-haut permet de filtrer les morceaux de cosinus à la fréquence $F_1 = 2000\text{Hz}$. Une détection d'énergie réalisée tous les T_s secondes permet de récupérer, à partir des signaux filtrés, les bits 0 et 1 transmis.

2.3.1 Synthèse du filtre passe-bas

On implémente un retard de zéros qu'on va l'associer à notre signal (avec et sans bruit), et on applique le filtrage avec un filtre passe-bas, qui est dans notre cas la transformée de Fourier inverse de la porte :

$$passe_{bas} = 2 \frac{Fc}{Fe} \frac{\sin(2 \frac{Fc}{Fe} t)}{2 \frac{Fc}{Fe} t}$$

On crée le signal filtré et on ne prend que les valeurs de 1 à $Ns * length_{bits}$

2.3.2 Synthèse du filtre passe-haut

On crée le filtre passe-haut à partir de celui du passe-bas.

$$H_{1_{ph}}(f) = 1 - H_{1_{pb}}(f)$$

Et on fait les mêmes étapes qu'avant. Sans oublier qu'on fait deux études en parallèle, une avec bruit et l'autre sans bruit.

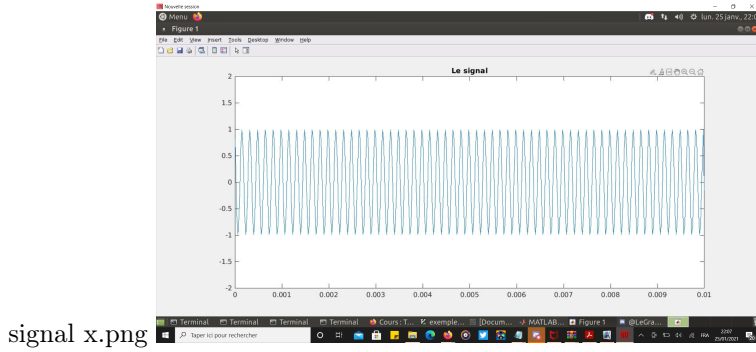
2.3.3 Filtrage, tracés, commentaires sur les résultats obtenus

On trace :
Le signal

Sa densité spectrale de puissance
Le filtre passe bas

La sortie en filtre passe bas

Le filtre passe haut
La sortie en filtre passe haut
Pour plus de tracés, Exécuter le programme Projet.m



signal x.png

FIGURE 5 – notre signal

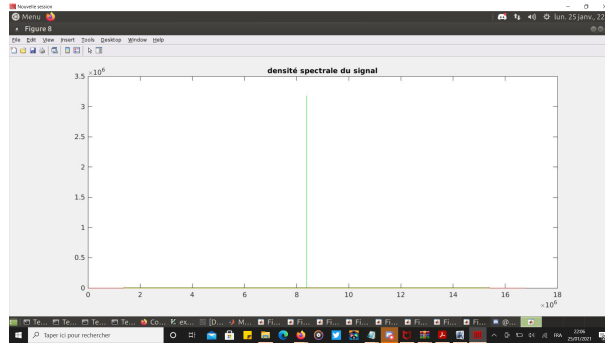
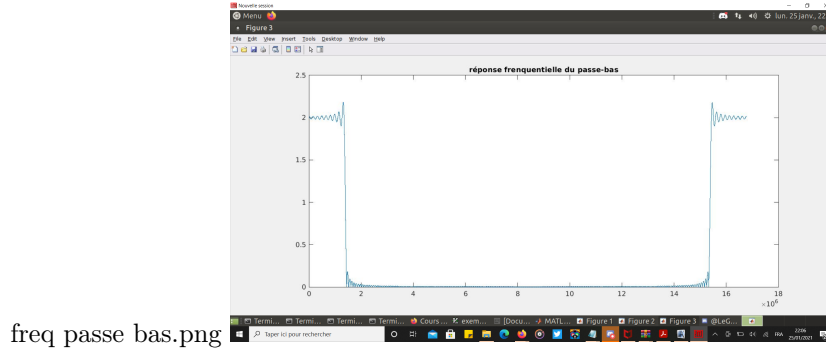


FIGURE 6 – DSPx



freq passe bas.png

FIGURE 7 – notre passe bas

2.3.4 Détection d'énergie

Afin de reconstituer l'image initiale, on fait une détection d'énergie.

Comme déjà dit on fait deux études (sans bruit et avec bruit).

On recrée le vecteur de la sortie du filtre avec le reshape, et puis on calcule la somme des carrés de ses éléments. Cette somme qu'on notera E_x va être comparée avec le seuil qui la convienne.

$$Seuil = \frac{(E_{max} - E_{min})}{2}$$

Si $E_x \leq Seuil$ alors on génère un bit égale à 0 si on travaille en passe-bas et 1 si on travaille en passe-haut.

Si $E_x > Seuil$ alors on génère un bit égale à 1 si on travaille en passe-bas et 0 si on travaille en passe-haut.

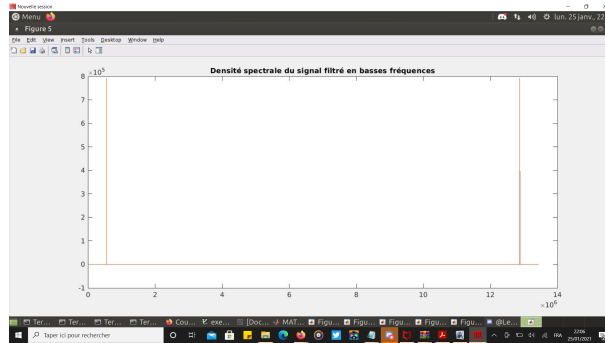
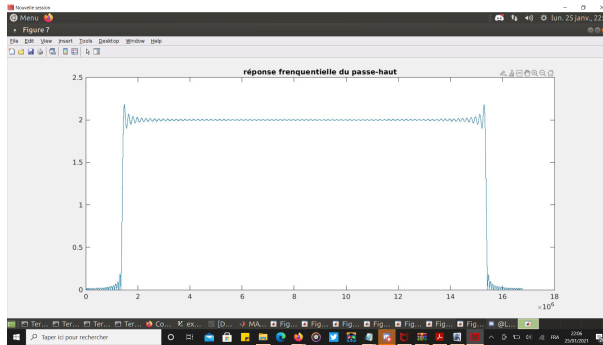


FIGURE 8 – Sortie en basses fréquences



freq passe haut.png

FIGURE 9 – notre passe haut

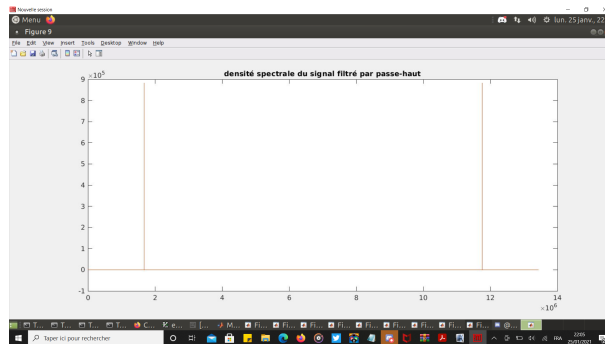


FIGURE 10 – sortie en passe haut

Cette suite des 0 et 1 nous permet de construire l'image souhaitée. On construit alors l'image depuis ces bits reconstitués dans tous les cas (passe-bas, passe-haut, sans bruit et avec bruit).

Taux d'erreur Pour le taux d'erreur, on calcule le nombre de bits erronées $\sum(|bits_i - E_x i|)$ et on divise cette quantité par la longueur du vecteur bits.

$$TauxDerreur = \frac{\sum(|bits_i - E_x i|)}{length_{bits}}$$

Remarque : Pour plus de tracés, exécuter le programme.

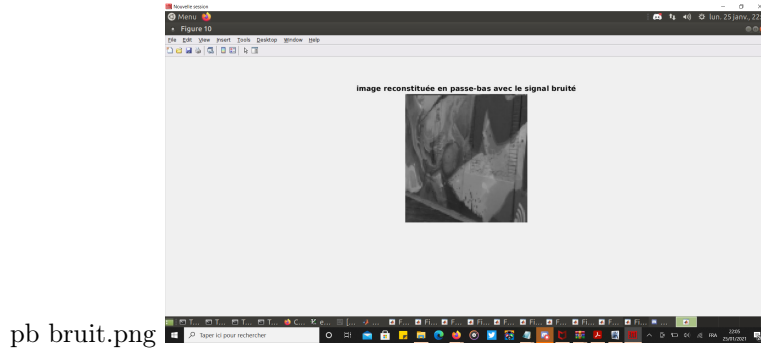


FIGURE 11 – image reconstituée en passe bas avec bruit

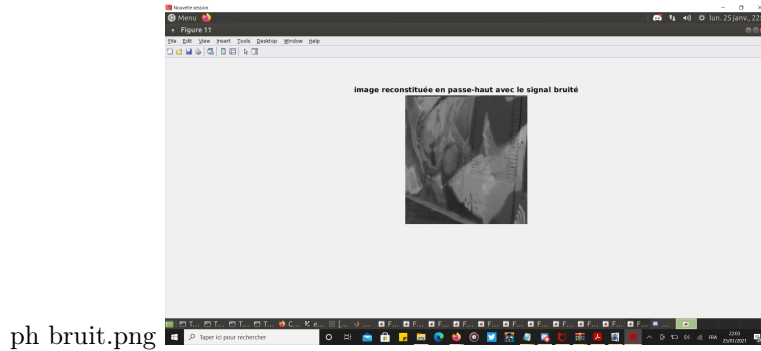


FIGURE 12 – image reconstituée en passe haut avec bruit

2.4 Application de la recommandation V21

Plus on augmente la fréquence, plus le taux d'erreur diminue.

3 Modem de fréquence V21 - Démodulateur FSK

3.1 Démodulateur FSK - Contexte de synchronisation idéale

La figure 13 présente le récepteur modifié implanté afin de retrouver, à partir du signal modulé en fréquence suivant la recommandation V21, le message binaire envoyé.

La méthode précédente a bien marché, dans cette étape on travaille avec les fréquences $F0 = 1180$ et $F1 = 980$, donc on utilise le signal définie précédemment x_{FSK} .

Ce dernier suit deux passages. Il se multiplie dans un premier temps avec un $\cos(2\pi F0t + \phi0)$ puis on intègre cette quantité grâce à la fonction définie sur MATALAB *trapz*.

En second on multiplie par $\cos(2\pi F1t + \phi1)$, et on intègre aussi.

On calcule H = la différence entre ces deux quantités et on génère un vecteur des bits sous la forme :

$$bits_{reconstitue} = H < 0$$

Ce vecteur de bits va nous permettre de reconstituer l'image.

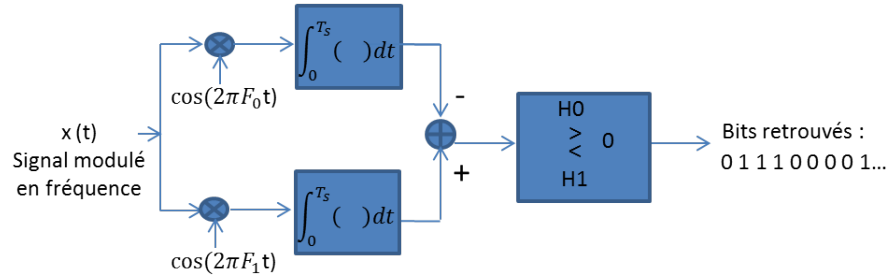


FIGURE 13 – Démodulation FSK. Synchronisation supposée idéale.

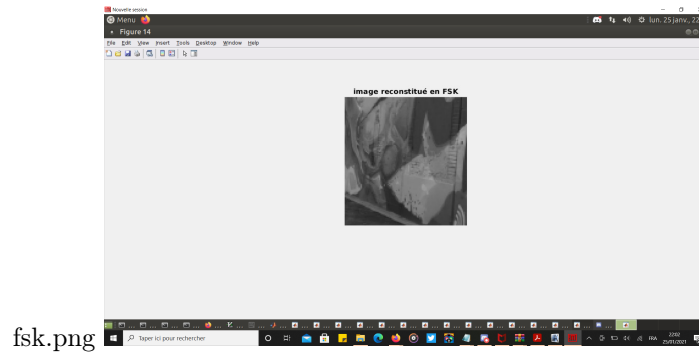


FIGURE 14 – image reconstituée en FSK

3.2 Démodulateur FSK avec gestion d'une erreur de synchronisation de phase porteuse

3.2.1 Déphasage aléatoire

On refait presque la même démarche, sauf qu'on ajoute deux déphasages aléatoires respectivement dans le premier cosinus et le deuxième.

On obtient une image non claire, parfois claire, parfois confus (selon le déphasage, s'il est par exemple de $\frac{\pi}{2}$ l'image est invisible).

Solution Alors on reconstitue nos bits grâce au schéma de la figure 17.

4 Conclusion

Les deux méthodes ont réussies, ils ont presque la même démarche : Reconstituer les bits puis visualiser les images. La différence c'est la méthode de la reconstitution de ces bits.

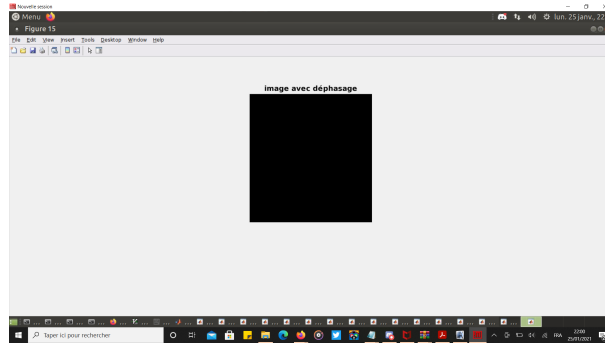
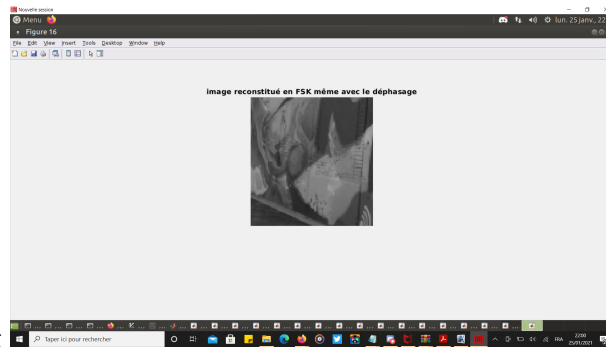


FIGURE 15 – image confus



deph m9ad.png

FIGURE 16 – image reconstituée malgré le déphasage

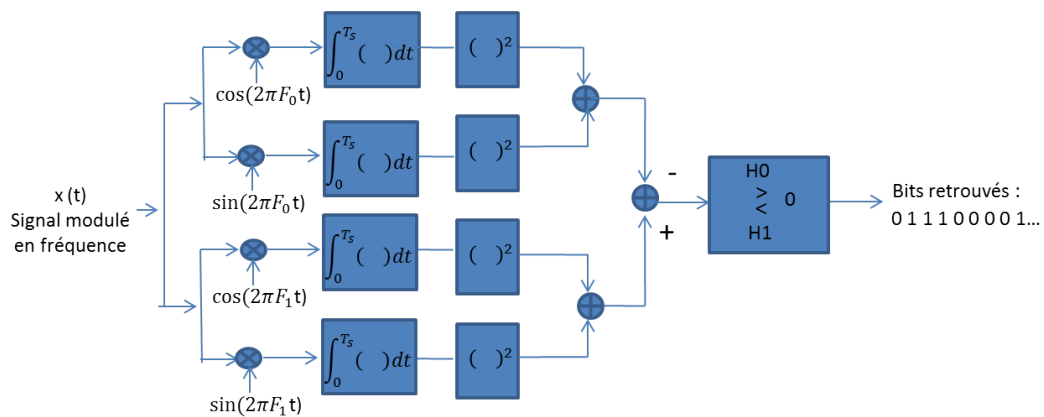


FIGURE 17 – Démodulation FSK - Gestion d'une erreur de phase porteuse.

La deuxième méthode prend en compte le déphasage qui peut être généré à cause du bruit physique probable entre l'émission et la réception du signal.

On avait un problème dans la démodulation FSK, car on avait implémenter un code de grande complexité, il nous a pris presque 20 min d'exécution. On a remédié à ce problème avec les reshape et l'opérateur .* dans MATLAB.

La deuxième méthode nous paraît plus efficace, car elle prend en compte le risque de déphasage.

5 Références

Nos références :

Wikipedia : Frequency shift keying

[http ://thomas.perso.enseeiht.fr/](http://thomas.perso.enseeiht.fr/)

UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS