

Projet de Traitement Numérique du Signal
Réalisation d'un modem de fréquence selon la recommandation V21 de
l'Union Internationale des Télécommunications (UIT)
Département Sciences du Numérique - Première année

1 Introduction

Un modem (acronyme pour modulateur/démodulateur) est utilisé, d'une part, pour transformer une information numérique à transmettre en un signal susceptible de se propager sur le support physique à traverser entre l'émetteur et le récepteur (modulation) et, d'autre part, pour retrouver, à partir du signal abîmé par le lien physique de communication, l'information binaire transmise (démodulation).

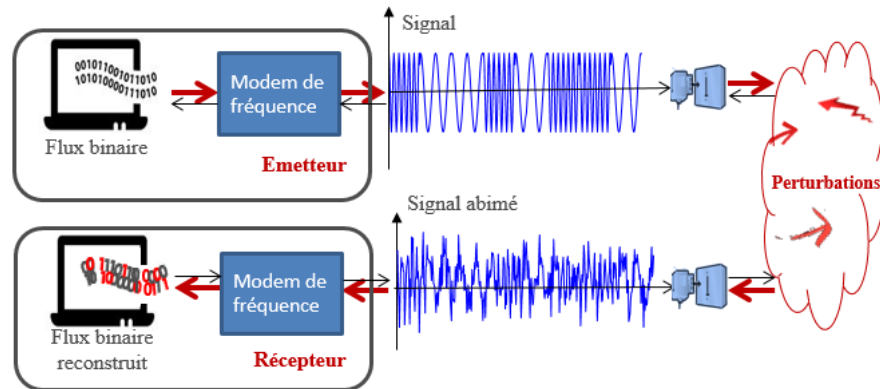


FIGURE 1 – Opération de modulation/démodulation

Le modem à implanter dans ce projet devra suivre la recommandation V21 du CCIT (Comité Consultatif International de Téléphonie et de Télégraphie, renommé depuis 1990 Union Internationale des Télécommunications (UIT)). Il s'agit d'un modem travaillant, avec pour support de communication la ligne téléphonique, à un débit maximal de 300 bits par secondes (également exprimé en bauds, en hommage à Emile Baudot (11 septembre 1845 - 28 mars 1903), célèbre ingénieur français ayant oeuvré dans les télécommunications. Ce modem est bien évidemment un vieux modem mais la technique de transmission utilisée, la modulation de fréquence numérique, reste présente dans de nombreux systèmes de transmissions à l'heure actuelle, dont, par exemple, l'internet des objets (technologie LoRa), le bluetooth ou bien la télémessure/télécommande par satellite. Elle présente, en effet, l'avantage de générer des signaux à enveloppes constantes, ce qui est très intéressant notamment en présence de non linéarités dans une chaîne de transmission (amplificateurs embarqués travaillant à saturation par exemple). Plusieurs versions de modulations de fréquence existent : FSK (Frequency Shift Keying), CP-FSK (Continuous Phase FSK), GMSK (Gaussian Minimum Shift Keying) ou plus généralement les CPM (Continuous Phase Modulation). La modulation de fréquence présente dans la recommandation V21 est une modulation de fréquence à 2 états (2-FSK). Les bits 0 et 1 à transmettre sont transformées en morceaux de cosinus de fréquences différentes pour former le signal transmis (voir figure 2).

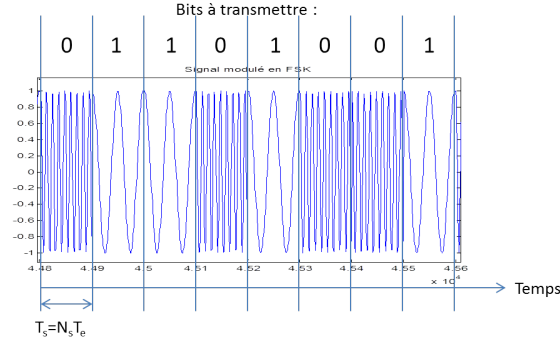


FIGURE 2 – Signal modulé en fréquence

Le modem V21 fonctionne en mode duplex. Nous simulerons dans ce projet la voie de transmission n°1 (appel). La fréquence moyenne nominale est de $F_c = 1080$ Hz. L'excursion de fréquence de $\pm\Delta f = \pm 100$ Hz. La fréquence la plus élevée ($F_0 = F_c + \Delta f = 1180$ Hz) devra correspondre à l'état binaire 0, tandis que la plus faible ($F_1 = F_c - \Delta f = 980$ Hz) devra correspondre à l'état binaire 1. La première étape consistera à former le signal 2-FSK à transmettre à partir d'un fichier d'information binaire (modulation). A partir d'un signal reçu abîmé par un canal à bruit additif blanc et Gaussien (canal AWGN : Additive White Gaussian Noise), il s'agira ensuite de mettre en place le modem de réception permettant de retrouver l'information binaire. Plusieurs versions du récepteur devront être implantées. Une première version procèdera par filtrage. Cette version nécessitant l'utilisation de fréquence suffisamment éloignées pour coder les bits 0 et 1, elle n'est pas utilisable dans le contexte de la norme V21 souhaité. On plantera donc une deuxième version de récepteur étant capable de travailler avec des fréquences proches (occupation spectrale plus faible pour le signal généré).

2 Information binaire à transmettre

Afin de tester vos codes vous utiliserez, dans un premier temps, une suite de bits à transmettre (bits 0 et 1 aléatoires et indépendants) générés grâce à la fonction *randi.m* de Matlab.

Vous pourrez ensuite appliquer vos codes à l'information qui vous sera fournie, par binôme, dans un fichier *donnees.mat*. Chaque fichier contiendra un message binaire à transmettre dans la variable *bits* que vous pourrez placer dans votre espace de travail en utilisant la commande *load donnees.mat*. Une fois transmis en utilisant le modem réalisé, vous accèderez en réception à un morceau d'image (voir comment par la suite). L'ensemble des morceaux d'images décodés dans votre groupe de TP et correctement repositionnés vous permettra de retrouver un lieu et un personnage.

3 Modem de fréquence - Démodulation par filtrage

3.1 Construction du signal modulé en fréquence

La première étape du projet va consister à réaliser la modulation de fréquence, i.e. transformer l'information binaire à transmettre en un signal modulé en fréquence (exemple sur la figure 2). Pour cela, on codera chacun des bits à transmettre par un morceau de cosinus durant T_s secondes, où $T_s = N_s T_e$, N_s étant le nombre d'échantillons distant de T_e formant chaque morceau de cosinus et $T_e = 1/F_e$ la période d'échantillonnage. Afin de pouvoir réaliser la démodulation par filtrage on utilisera un morceau de cosinus à la fréquence $F_0 = 6000$ Hz pour coder les bits 0 et un morceau de cosinus à la fréquence $F_1 = 2000$ Hz pour coder les bits 1. La fréquence d'échantillonnage F_e sera fixée à 48 kHz.

La figure 3 présente un exemple de signal NRZ généré à partir d'une information binaire à transmettre, avec $N_s = 3$ échantillons distants de T_e par niveau 0 ou 1 de durée T_s .

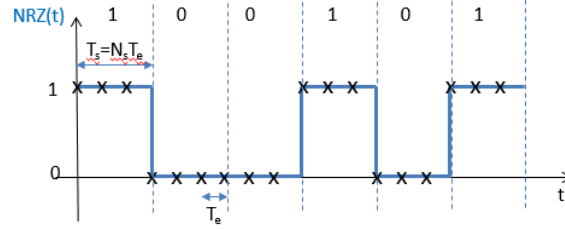


FIGURE 3 – Exemple de signal NRZ polaire

Le signal modulé en fréquence $x(t)$ sera alors généré de la manière suivante :

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

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`.

3.1.1 Génération du signal NRZ

1. Générez le signal NRZ à partir de la suite de bits à transmettre. Le nombre d'échantillons N_s à générer par niveau 0 ou 1 de durée T_s sera déduit du débit souhaité pour la transmission (300 bits par secondes).
2. Tracez le signal NRZ avec une échelle temporelle en secondes et vérifiez que le débit souhaité pour la transmission est le bon.
3. Estimez puis tracez la densité spectrale de puissance du signal NRZ, en utilisant un périodogramme. L'échelle fréquentielle devra être en Hz. La densité spectrale de puissance théorique d'un tel signal est donnée par (voir cours de télécommunications pour la démonstration) :

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

3.1.2 Génération du signal modulé en fréquence

1. A partir du signal NRZ précédent, et de la génération de deux cosinus numériques aux fréquences souhaitées, générez le signal modulé en fréquence $x(t)$ définie par l'équation 1.
2. Tracez le signal $x(t)$ avec une échelle temporelle en secondes.
3. Calculez la densité spectrale de puissance du signal modulé en fréquence $x(t)$ en fonction de celle du signal NRZ que l'on notera $S_{NRZ}(f)$.
4. Estimez puis tracez la densité spectrale de puissance du signal modulé en fréquence, en utilisant un périodogramme et avec une échelle fréquentielle en Hz. Commentez le résultat obtenu en le comparant à l'expression théorique précédente.

3.2 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.

Vous simulerez ce canal de transmission en ajoutant au signal modulé en fréquence un signal gaussien généré grâce à la fonction *randn.m* de Matlab. Afin de générer N échantillons d'un signal gaussien de moyenne nulle et de puissance égale à σ^2 vous devez l'utiliser de la manière suivante $\sigma * \text{randn}(1, N)$.

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. La puissance d'un signal aléatoire $x(t)$ étant donnée par $E[|x(t)|^2]$, elle peut être obtenue sous Matlab en utilisant $P_x = \text{mean}(\text{abs}(x).^2)$, si x est le vecteur contenant les échantillons du signal. Vous pouvez, dans un premier temps, fixer le rapport signal sur bruit à 50dB, puis vous le ferez évoluer afin de visualiser son impact sur le taux d'erreur binaire obtenu pour la transmission et sur la qualité du morceau d'image restitué.

3.3 Démodulation par filtrage

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

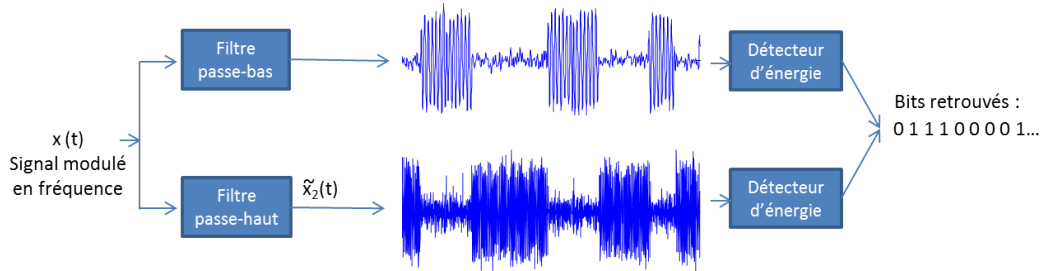


FIGURE 4 – Démodulation par filtrage.

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.

Remarque : un seul des deux filtres est bien entendu nécessaire pour démoduler votre signal. Néanmoins nous vous demanderons d'implanter les deux.

3.3.1 Synthèse du filtre passe-bas

Vous pouvez utiliser le filtre passe-bas réalisé lors du TP2, en adaptant ses paramètres afin de récupérer le signal permettant de retrouver les 1 émis.

3.3.2 Synthèse du filtre passe-haut

La réponse en fréquence d'un filtre passe-haut idéal est donné par : $H_{IPH}(\tilde{f}) = 1 - H_{IPB}(\tilde{f})$, où $H_{IPB}(\tilde{f})$ représente la réponse en fréquence du filtre passe-bas idéal de même fréquence de coupure

et $\tilde{f} = \frac{f}{F_c}$ la fréquence normalisée. Utilisez cette expression pour déterminer la réponse impulsionnelle idéale d'un filtre passe-haut : $h_{IPH}(n) = TFD^{-1} \left[H_{IPH}(\tilde{f}) \right]$. Une fois cette réponse impulsionnelle idéale obtenue, il vous restera à adapter ses paramètres afin de pouvoir l'implanter de manière à retrouver le signal permettant de récupérer les 0 émis en sortie du filtre.

3.3.3 Filtrage

Procédez aux filtrages (passe-bas et passe-haut) du signal reçu en utilisant la fonction *filter.m* de Matlab. Attention aux retards introduits par les filtres qui doivent être pris en compte afin de ne pas perdre d'information et de synchroniser correctement votre détecteur d'énergie (voir plus loin).

3.3.4 Tracés à réaliser

Attention les signaux devront être tracés avec une échelle temporelle en secondes et les densités spectrales de puissance avec une échelle fréquentielle en Hz. Pour chaque voie du récepteur :

1. Tracez la réponse impulsionnelle et la réponse en fréquence du filtre réalisé.
2. Tracez, sur un même graphique, la densité spectrale de puissance du signal modulé en fréquence reçu, $x(t)$, et la réponse en fréquences du filtre implanté. Commentez le résultat obtenu.
3. Tracez le signal en sortie du filtre ainsi que sa densité spectrale de puissance. Commentez le résultat obtenu.

3.3.5 Détection d'énergie

Afin de retrouver les bits émis à partir des signaux filtrés, on va utiliser un détecteur d'énergie.

1. Chaque signal filtré sera divisé en périodes temporelles de durée T_s secondes. Sur chaque tranche de signal, notée X et constituée de N_s échantillons : $X = \{x_1, \dots, x_{N_s}\}$, un calcul d'énergie devra être effectué et comparé à un seuil.

Si on considère, par exemple, le signal filtré par le passe-bas, on aura la condition suivante :

$$\text{Si } \sum_{n=1}^{N_s} x_n^2 > K \text{ alors bit 1 sinon bit 0}$$

où K représente le seuil qui sera fixé expérimentalement après observation de l'énergie calculée sur les différentes périodes T_s de signal.

2. Une fois le message binaire retrouvé vous pourrez :
 - Calculer le taux d'erreur binaire obtenu : nombre de bits erronés sur nombre de bits transmis. Bien entendu ce taux d'erreur binaire dépendra du rapport signal sur bruit considéré dans votre transmission, ce que vous pourrez constater en le modifiant.
 - Utiliser la fonction *reconstitution_image.p* qui vous est fournie, sur la suite binaire retrouvée en réception (notée *suite_binaire_reconstruite*), de la manière suivante :
`code_reconstitution_image;`
`reconstitution_image(suite_binaire_reconstruite);`
`which_reconstitution_image;`
 afin d'afficher le morceau d'image qui vous a été fournie sous forme binaire dans le fichier *donnees.mat*. L'ensemble des morceaux d'images décodés dans votre groupe de TP et correctement repositionnés vous permettra de retrouver une image complète faisant référence à un lieu et à un personnage. Qui et où ?

3.4 Application de la recommandation V21

Testez la chaîne de transmission implantée en utilisant les fréquences préconisées dans la recommandation V21, à savoir $F_0 = 1180$ Hz et $F_1 = 980$ Hz.

- Retrouvez-vous un taux d'erreur binaire nul lorsqu'il n'y a aucun bruit (ou pour un rapport signal sur bruit suffisamment élevé) ?
- Est-il possible de modifier les filtres (fréquence de coupure, nombre de coefficients, fenêtre de troncature) pour retrouver un taux d'erreur binaire nul lorsqu'il n'y a aucun bruit (ou pour un rapport signal sur bruit suffisamment élevé) ? Si oui réalisez cette nouvelle implantation, si non expliquez ce qui vous en empêche.

4 Modem de fréquence V21 - Démodulateur FSK

4.1 Démodulateur FSK - Contexte de synchronisation idéale

Nous vous proposons d'implanter ici le modem qui fonctionne avec les fréquences préconisées dans la recommandation V21, à savoir $F_0 = 1180$ Hz et $F_1 = 980$ Hz, en supposant que la synchronisation entre l'émetteur et le récepteur est parfaite. L'émetteur reste le même à l'exception des fréquences F_0 et F_1 . Le canal de propagation reste le même (ajout d'un bruit additif blanc et gaussien sur le signal émis). Le récepteur, par contre, va devoir être modifié afin de démoduler correctement les données. La figure 5 présente le récepteur que vous devez implanter pour retrouver, à partir du signal modulé en fréquence suivant la recommandation V21, le message binaire envoyé.

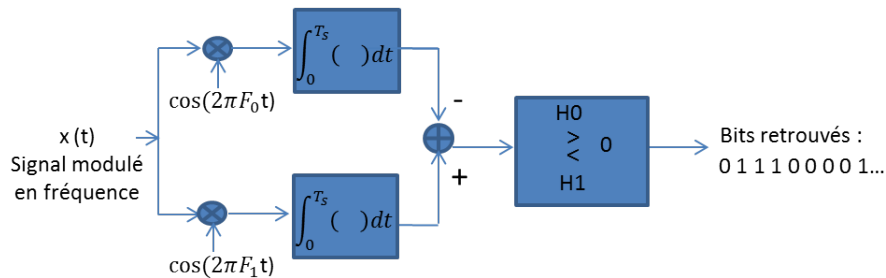


FIGURE 5 – Démodulation FSK. Synchronisation supposée idéale.

1. Expliquez le principe de fonctionnement de ce récepteur.
2. Implantez le démodulateur proposé et vérifiez qu'il fonctionne correctement. Vous devez retrouver un taux d'erreur binaire nul sans bruit, ou pour un rapport signal sur bruit suffisamment élevé, et un morceau d'image de bonne qualité si vous utilisez les données binaires fournies dans le fichier *donnees.mat*.

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

Le problème de la synchronisation entre l'émetteur et le récepteur est un problème important lorsque l'on réalise une transmission. Les deux doivent être parfaitement synchronisés en temps et en fréquence pour que le démodulateur idéal implanté précédemment fonctionne, ce qui en pratique n'est bien entendu pas possible.

1. Introduisez, dans le modem implanté précédemment, une erreur de phase porteuse, c'est-à-dire utilisez en émission et en réception des cosinus de phases différentes. Votre modem continue-t-il à fonctionner ? Si ce n'est pas le cas expliquez pourquoi.
2. Afin que votre modem puisse continuer à fonctionner en présence d'une erreur de phase porteuse, celui-ci doit être modifié. La figure 6 présente un démodulateur permettant de s'affranchir de problèmes de synchronisation de phase entre les oscillateurs d'émission et de réception.
 - (a) Expliquez le principe de fonctionnement de ce nouveau récepteur : en quoi la modification apportée permet-elle de retrouver l'information même s'il existe une erreur de phase entre les cosinus d'émission et les cosinus et sinus de réception ?
 - (b) Implantez ce nouveau démodulateur et testez-le en présence d'une erreur de phase entre les oscillateurs d'émission et de réception pour vérifier que le taux d'erreur binaire obtenu sans bruit est bien égal à 0.

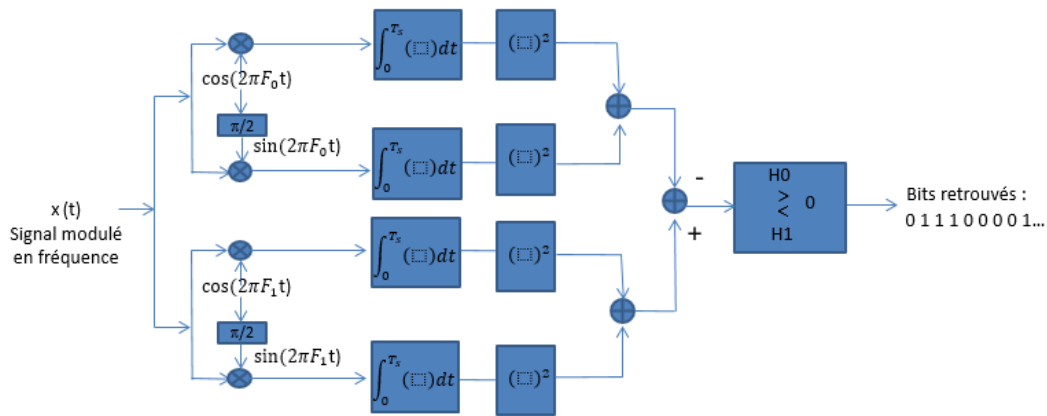


FIGURE 6 – Démodulation FSK - Gestion d'une erreur de phase porteuse.

5 Evaluation

Le travail sera réalisé en binôme et nous vous demanderons de déposer vos codes et un rapport au format pdf sur moodle, dans l'espace réservé à votre groupe de TP et prévu pour cela, avant le 22/01/2021.

5.1 Le rapport

1. Comme tout rapport, il devra comporter un sommaire, une introduction présentant les objectifs des TPs, une conclusion synthétisant les principaux résultats obtenus et une bibliographie comprenant les références éventuellement utilisées. On pourra y ajouter une table des illustrations.
2. Les équations devront être réalisées avec un éditeur d'équation.
3. Lorsque vous commentez une figure vous devez y faire référence dans votre texte : par exemple "comme le montre la figure 1, ..."
4. Tous vos tracés doivent comporter des labels sur les axes et un titre.
5. Si plusieurs courbes sont tracées sur la même figure, celle-ci devra comporter une légende permettant de les différencier.
6. Toutes vos explications/justifications/commentaires devront utiliser les bons termes techniques (provenant des cours/TDs/TPs, des livres/sites consultés et cités), pas d'à peu près. "En gros" est à proscrire...
7. Attention votre rapport doit être relu, éventuellement passé au correcteur orthographique et grammatical.

Un fichier exemple au format latex vous est fourni. Vous pouvez l'utiliser en le complétant, en le modifiant comme vous le souhaitez. Il vous donne quelques exemples pour réaliser un plan, l'écriture d'équations, l'insertion et l'appel à figure, l'appel à section ... Vous pouvez, par exemple, utiliser texmaker qui est gratuit pour l'éditer et le compiler.

5.2 Les codes

1. Vos codes doivent être commentés de manière suffisante et claire. Un nouvel utilisateur doit pouvoir comprendre ce que vous avez souhaité implanter.
2. Vous pouvez regrouper l'ensemble du code du projet dans un seul fichier .m. Si vous en fournissez plusieurs, un mode d'emploi devra être fourni dans le rapport pour savoir ce qui doit être lancé pour réaliser les différentes fonctions implantées.