

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).

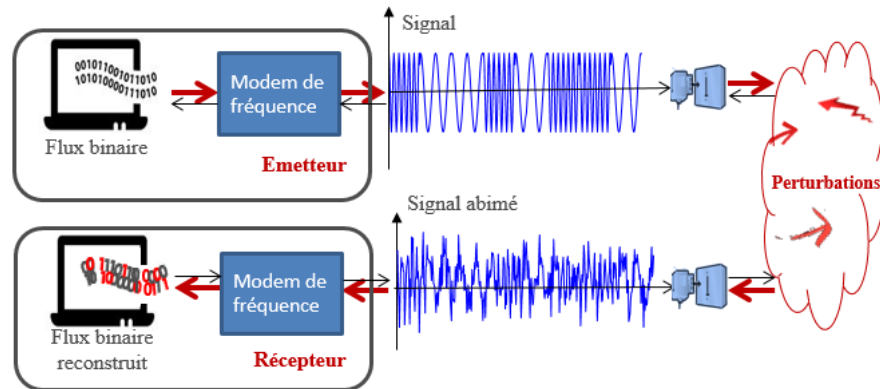


FIGURE 1 – Opération de modulation/démodulation

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 œuvré 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).

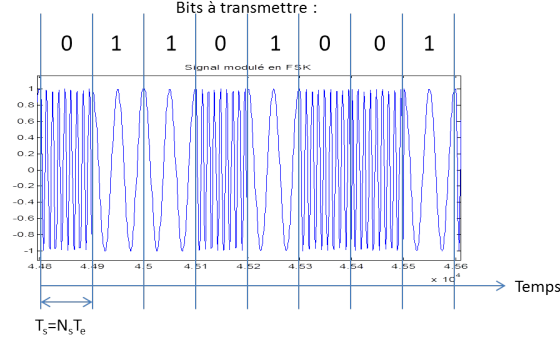


FIGURE 2 – Signal modulé en fréquence - Modulation 2-FSK

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) que vous devrez générer, il s'agira 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 capable de travailler avec des fréquences proches (occupation spectrale plus faible pour le signal généré). Une troisième version permettra de prendre en compte une erreur de synchronisation entre l'émetteur et le récepteur.

2 Information binaire à transmettre

Afin de mettre en place votre modem vous utiliserez, dans un premier temps, une suite de bits à transmettre (bits 0 et 1 aléatoires et indépendants) que vous générerez grâce à la fonction *randi.m* de Matlab.

Une fois que votre modem fonctionnera, vous pourrez utiliser le démodulateur implanté pour démoduler les signaux fournis dans les fichiers *fichieri.mat*, *i* allant de 1 à 6. Chaque fichier contient un signal à démoduler dans la variable *signal* que vous pourrez placer dans votre espace de travail en utilisant la commande *load fichieri.mat*. Chaque signal démodulé vous donnera accès à un morceau d'une image. L'ensemble des morceaux correctement repositionnés vous permettra de retrouver un lieu et un personnage dont vous nous donnerez une petite biographie...

3 Modem de 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. Les bits 0 seront codés par un morceau de cosinus à la fréquence F_0 , tandis que les bits 1 seront codés par un morceau de cosinus à la fréquence F_1 .

Le signal modulé en fréquence $x(t)$ sera 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 unipolaire 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 $rand*2*pi$.

Bien entendu les signaux générés seront des signaux numériques, échantillonnés à T_e (période d'échantillonnage). La durée T_s sera donc composée d'un certain nombre d'échantillons, que nous noterons N_s , distants de T_e : $T_s = N_s T_e$. La fréquence d'échantillonnage $F_e = 1/T_e$ sera fixée à 48 kHz.

La figure 3 présente un exemple de signal NRZ unipolaire 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 .

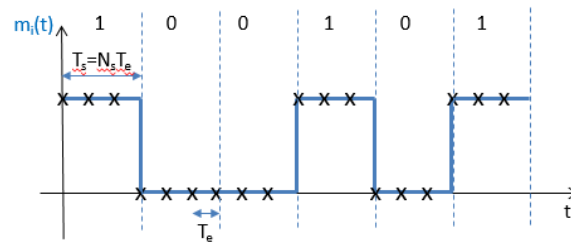


FIGURE 3 – Exemple de signal NRZ unipolaire

3.1 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 de la fréquence d'échantillonnage et du débit souhaités pour la transmission ($F_e = 48$ kHz, débit de 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. Votre tracé devra comporter des labels sur les axes (*xlabel.m*, *ylabel.m*) et un titre (*title.m*). Cela sera, bien entendu, également le cas pour toutes vos prochaines figures.
3. Estimez, puis tracez, la densité spectrale de puissance du signal NRZ, en utilisant, par exemple, un périodogramme de Welch. L'échelle fréquentielle devra être en Hz. Vous pouvez utiliser, pour cela, la fonction *pwelch.m* de Matlab de la manière suivante :
 $DSP = pwelch(NRZ, [], [], [], Fe, 'twosided');$
4. En les traçant sur une même figure, comparez la densité spectrale de puissance estimée à la densité spectrale de puissance théorique, qui vous est donnée ci-dessous pour un signal NRZ polaire (voir cours de télécommunications pour la démonstration) :

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

Afin d'identifier les deux tracés, votre figure devra comporter une légende (*legend.m*). Cela sera, bien entendu, également le cas pour toutes vos prochaines figures comportant plusieurs tracés.

3.2 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 $x(t)$ défini par l'équation (1) et échantillonné à T_e .
2. Tracez le signal $x(k)$ avec une échelle temporelle en secondes.

3. Calculez (théoriquement) la densité spectrale de puissance du signal modulé en fréquence $x(t)$ en fonction de celle du signal $NRZ(t)$ 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 ou un périodogramme de Welch et avec une échelle fréquentielle en Hz. Expliquez le résultat obtenu en vous aidant du calcul théorique réalisé précédemment.

4 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 et 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, par la suite, sur la qualité des morceaux d'image restitués. On pourra éventuellement tracer une courbe représentant le taux d'erreur binaire de votre transmission en fonction du rapport signal à bruit.

5 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.

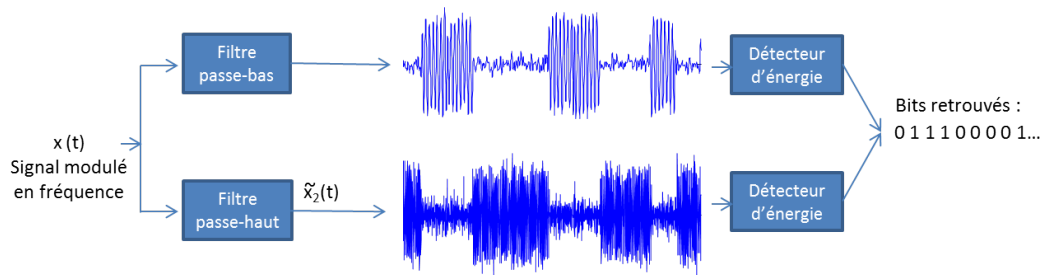


FIGURE 4 – Démodulation par filtrage.

Afin de faire fonctionner ce démodulateur vous prendrez $F_0 = 6000$ Hz et $F_1 = 2000$ Hz. Un filtre passe-bas permet de filtrer les morceaux de cosinus à la fréquence $F_0 = 6000$ Hz, tandis qu'un filtre passe-haut permet de filtrer les morceaux de cosinus à la fréquence $F_1 = 2000$ 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.

5.1 Synthèse du filtre passe-bas

Vous pouvez utiliser le filtre passe-bas réalisé lors du TP2 en adaptant sa fréquence de coupure et sa fenêtre de troncature afin de récupérer le signal permettant de retrouver les 1 émis, sachant que le nombre de coefficient du filtre devra être fixé à 61.

5.2 Synthèse du filtre passe-haut

La réponse en fréquence d'un filtre passe-haut idéal est donné par : $H_{I_{PH}}(\tilde{f}) = 1 - H_{I_{PB}}(\tilde{f})$, où $H_{I_{PB}}(\tilde{f})$ 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.

1. Utilisez cette expression pour déterminer la réponse impulsionnelle idéale d'un filtre passe-haut.
2. Une fois cette réponse impulsionnelle idéale obtenue, il vous restera à l'adapter afin de pouvoir implanter le filtre permettant de retrouver le signal permettant, à son tour, de récupérer les 0 émis en sortie du filtre. Le nombre de coefficients sera ici aussi fixé à 61.

5.3 Filtrage

Procéder aux filtrages (passe-bas et passe-haut) du signal reçu en utilisant la fonction *filter.m*. Que peut-on dire des 30 premiers points des signaux filtrés à l'aide de cette fonction ?

5.4 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. Et n'oubliez pas les labels sur les axes, les titres des figures, les légendes si besoin.

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 (entrée du démodulateur) 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.

5.5 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.

5.6 Modification du démodulateur

Modifier le démodulateur implanté

1. en passant le nombre de coefficients des filtres à 201. Retrouve-t-on les mêmes résultats que précédemment ? Si les résultats trouvés ne sont pas les mêmes :
 - (a) Expliquer pourquoi.
 - (b) Modifier votre code afin de retrouver les mêmes résultats quel que soit le nombre de coefficients utilisés dans les réponses impulsionnelles des filtres.
2. en considérant les fréquences de la recommandation V21, soit $F_0 = 1180$ Hz pour coder les bits 0 et $F_1 = 980$ Hz pour coder les bits 1.
 - 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 pourquoi.

6 Démodulateur de fréquence adapté à la norme V21

Nous vous proposons d'implanter ici le démodulateur qui fonctionne avec les fréquences préconisées dans la recommandation V21, à savoir $F_0 = 1180$ Hz pour coder les bits 0 et $F_1 = 980$ Hz pour coder les bits 1. 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 démodulateur, par contre, va devoir être modifié afin de démoduler correctement les signaux.

6.1 Contexte de synchronisation idéale

La figure 5 présente le récepteur que vous devez implanter pour retrouver, dans un contexte de synchronisation idéale, à partir du signal modulé en fréquence suivant la recommandation V21, le message binaire envoyé.

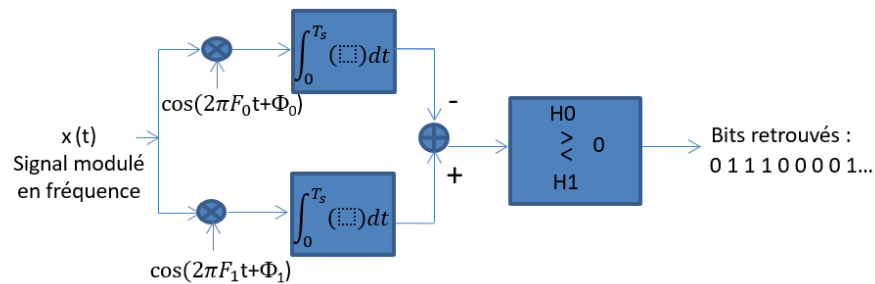


FIGURE 5 – Démodulation FSK. Synchronisation supposée idéale.

1. Expliquez le principe de fonctionnement de ce récepteur. Pour cela vous pouvez regarder quels sont les résultats de

$$\int_0^{T_s} \cos(2\pi F_0 t + \phi_0) \cos(2\pi F_0 t + \phi_0) dt$$

ou

$$\int_0^{T_s} \cos(2\pi F_1 t + \phi_1) \cos(2\pi F_1 t + \phi_1) dt$$

et de

$$\int_0^{T_s} \cos(2\pi F_1 t + \phi_1) \cos(2\pi F_0 t + \phi_0) dt$$

et les évaluer en utilisant les valeurs numériques des paramètres de la recommandation V21.

2. Implantez le démodulateur proposé et vérifiez qu'il fonctionne correctement. L'intégrale sur la durée T_s sera implantée en numérique comme une somme sur N_s échantillons. Vous devez retrouver un taux d'erreur binaire nul sans bruit, ou pour un rapport signal sur bruit suffisamment élevé.

6.2 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 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.

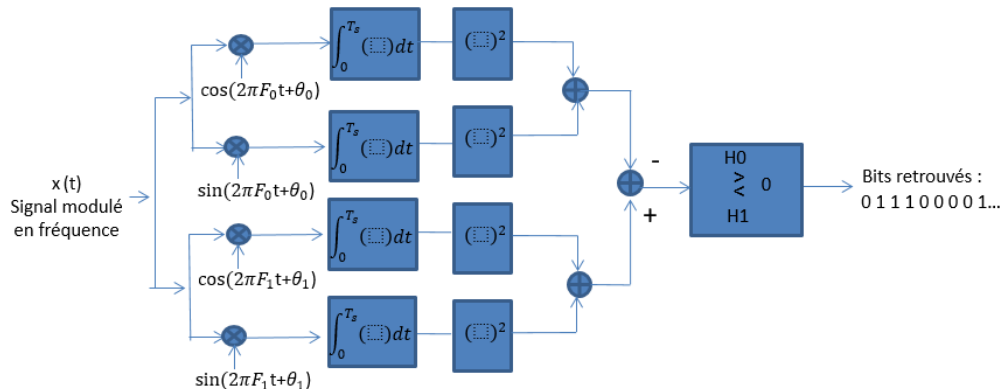


FIGURE 6 – Démodulation FSK - Gestion d'une erreur de phase porteuse.

- (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 et sinus 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.
- (c) Vous pouvez maintenant utiliser votre démodulateur sur les fichiers fournis afin de retrouver les 6 morceaux d'images qui, correctement assemblés, vous permettront de trouver un lieu et un personnage emblématiques de votre nouvelle vie post prépa/IUT/Licence... Pour cela utilisez les fichiers fournis *fichier_i.mat*, *i* allant de 1 à 6. Chaque fichier contient un signal à démoduler dans la variable *signal* que vous pourrez placer dans votre espace de travail en utilisant la commande `load fichier_i.mat`. Chaque signal démodulé

vous donnera accès à un des 6 morceaux de l'image mystère. Les paramètres qui ont été utilisés dans le modulateur sont bien entendu ceux de la norme V21.

7 Évaluation

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 14/01/2023 (date à préciser en fonction de celle prévue pour la remise des notes).

7.1 Le rapport

1. Comme tout rapport, il devra comporter un sommaire, une introduction présentant les objectifs du projet, 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 figures 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. **Il n'y a aucune obligation à l'utiliser, c'est une proposition pour ceux qui le souhaitent.** Dans tous les cas, votre rapport devra être rendu au format pdf.

7.2 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. Nous vous demanderons de réaliser au moins 3 fichier .m (modulateur/démodulateur par filtrage, modulateur/démodulateur V21, modulateur/démodulateur V21 avec gestion d'un problème de synchronisation de phase porteuse). Vos codes seront fournis avec un readme.txt indiquant la liste des codes fournis, leur fonction et ce qui doit être lancé pour réaliser les différentes fonctions implantées.