

TP Couche PHY : GNU Radio

L'objectif du TP est de mettre en pratique la couche PHY en télécommunications radio. Nous appliquerons et analyserons quelques traitements fondamentaux de la couche PHY dans les systèmes numériques sans fil :

- modulations numériques en bande de base (ASK, FSK, PSK, QAM)
- transposition en fréquence
- filtrage
- (ré)échantillonnage
- synchronisation
- égalisation de canal

Vous utiliserez pour cela l'environnement de GNU radio **GNU Radio**))) (<https://www.gnuradio.org/about/>). Il s'agit d'un outil gratuit et open source, permettant de réaliser de la simulation de transmission radio et du développement logiciel.

Dans le cadre du TP, nous simulerons différents schémas de transmission numérique radio (ASK, FSK, PSK/QAM).

Préliminaires

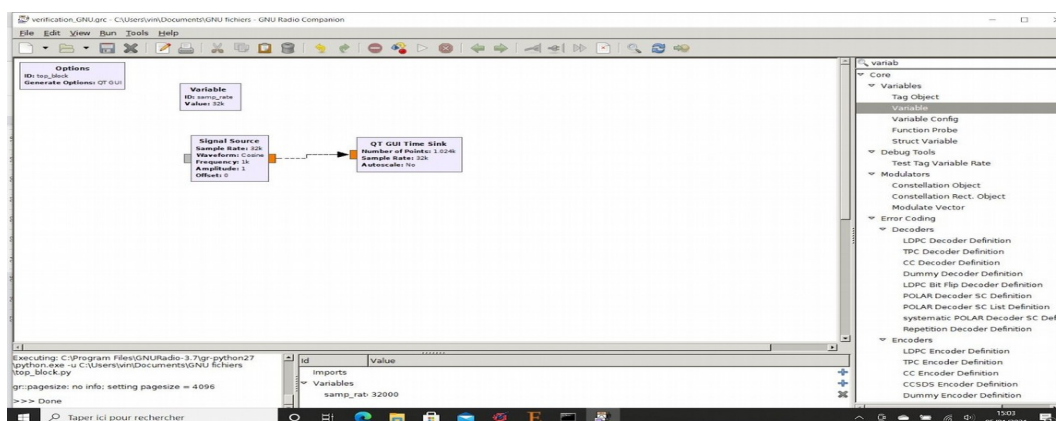
Installez *GNURadio* sur votre PC. Vous pourrez trouver les instructions sur ce lien : www.gcnddevelopment.com/gnuradio/downloads.htm

Initialement prévue et développée sous Linux, GNURadio propose des versions compatibles Windows et MAC.

> Pour les OS sous Windows, vous pouvez installer la version de GNURadio **gnuradio_3.7.13.5_win64**.

> Pour les PC sous MAC, vous avez la possibilité de l'installer en version native, ou de travailler sur une machine virtuelle.

Une fois l'installation réalisée, lancez et vérifiez le bon fonctionnement du logiciel en réalisant un schéma de simulation simple, avec un bloc "Signal Source" et une visualisation "QT GUI Time Sink" (oscilloscope).



Sous GNUradio, nous avons deux possibilités de framework : *QT GUI* et *WX GUI*. Ils font appel à deux bibliothèques distinctes d'interfaces graphiques (Qt et wxWidgets). Le mode *QT GUI* étant plus efficace et mieux documenté que celui de *WX GUI*, c'est celui que vous utiliserez.

> Le long du TP, il sera d'intérêt de visualiser le(s) signal(aux) dans les domaines fréquentiel (QT GUI Freq Sink), temporel (QT GUI Time Sink), et éventuellement le diagramme de constellation (QT GUI Constellation Sink), afin d'analyser / vérifier la cohérence du résultat obtenu.

Il faudra également penser à relever les paramètres utilisés pour les blocs de traitement.

Présentation du logiciel GNURadio

Familiarisez-vous avec les différentes fonctionnalités de GNURadio.

Attention : les flux numériques (liaisons entre "blocs") sont typés (byte, char, float, short, complex...). Il faudra veiller à la cohérence, et sélectionner le bon format dans les blocs, et si nécessaire, rajouter un bloc intermédiaire de conversion.

- **Les sources** (voir "waveform generators")
 - Signal source (sine/cosine/triangle....)
 - random source
 - file source, etc..
- **Les outils de visualisation** (voir "Instrumentation QT")
 - Constellation sink, Frequency sink, Time Sink...
- **Les filtres**
 - low-, high-, band-, reject-, pass filters...
- **Les blocs de conversion** (voir "type converters") **et de formattage** (voir "Byte operators")
 - char to float, float to complex, ...
 - pack/unpack k bits , ...
- **Les opérateurs mathématiques**
 - add,subtract, multiply, fft, ...
 - FreqXlating
- **Divers**
 - Rational resampler
 - throttle
 - variable
 - constant
 - QT GUI range (slider)

Partie 1. Modulation ASK

Vous allez simuler une modulation d'amplitude (ASK) en appliquant cela à un flux binaire aléatoire qui représentera votre source informative.

Attention : Veillez bien à ce que l'échantillonnage et le nombre d'échantillons par symbole soit cohérent pour la porteuse et pour le flux binaire aléatoire !

> Pour ajuster cela, utilisez les blocs suivants : “throttle” (systématiquement après chaque bloc *source*), et “repeat” (après le flux binaire). Contrôlez toujours sur l'oscilloscope QT GUI Time Sink.

Le signal informatif sera modulé sur une porteuse $f_p = 125$ KHz (vous pouvez utiliser un slider “QT GUI range”) et transposé à une fréquence RF : $f_{RF} = 1$ MHz.

Il faudra choisir judicieusement une fréquence d'échantillonnage adaptée.

Émission :

1. Réaliser la modulation ASK à 2 états d'amplitude (tout-ou-rien OOK), à l'aide des outils présentés, en bande de base.
2. Modifier légèrement votre framework pour le généraliser à une modulation M-ASK (toujours en bande de base). Vous pouvez utiliser le
3. Réaliser la transposition en fréquence autour de 1 MHz. Vous pouvez pour cela utiliser un multiplieur (« multiply ») suivi d'un filtre passe-bas, ou également le bloc « Freq Xlating ».

Réception :

4. Réaliser la transposition en bande de base, autour de f_p .
5. Il existe deux méthodes pour démoduler un signal ASK : la démodulation non-cohérente (par détection d'enveloppe) et la démodulation cohérente (avec PLL).
> Expliquer le principe de chacun de ses méthodes, en précisant les avantages de chacune.
6. Réaliser la démodulation non-cohérente. Pour cela, vous pourrez utiliser les blocs :
complex to mag^2 **binary slicer**
7. Réaliser la démodulation cohérente. On simulera une porteuse parfaitement restituée en utilisant un bloc de source sinusoidale.
8. Quels sont les avantages et inconvénients d'une modulation ASK ?

Partie 2. Modulation FSK

Dans cette partie, vous allez réaliser vous-même la modulation de fréquence (FSK) à l'aide d'opérateurs mathématiques. Cela permettra de bien maîtriser les opérations réalisées (plutôt que d'utiliser le bloc "FSK mod").

Afin de ne pas alourdir le framework, nous resterons en bande de base (pas de transposition en fréquence).

Émission :

1. Réaliser le framework pour effectuer une modulation 2-FSK. Prendre pour commencer, $f_1 = 2\text{kHz}$ et $f_2 = 4\text{ KHz}$. Comme précédemment, soyez vigilant sur l'échantillonnage des signaux informatif et de porteuse...
2. Proposer une mise en œuvre de la modulation M-FSK à l'aide d'opérateurs mathématiques/logiques simples.

Réception :

3. Il existe deux méthodes pour démoduler un signal FSK : la démodulation à quadrature (ou démodulation à coïncidence) et la démodulation cohérente (avec PLL).
> Expliquer le principe de chacun de ses méthodes.
4. Réaliser la démodulation à quadrature. Pour cela, vous pourrez utiliser les blocs :

freqXlating FIR filter

Quadrature demod

binary slicer

5. Quels sont les avantages et inconvénients de la modulation FSK ?

Partie 3. Modulations PSK et QAM

Vous allez maintenant réaliser les modulations PSK et QAM en restant en bande de base. Dans les deux cas, vous utiliserez le bloc "Constellation Modulator". Il faudra définir un bloc "Constellation Object" auquel le modulateur sera associé, pour spécifier le diagramme de constellation appliqué.

Vous réaliserez, comme précédemment, la simulation d'une transmission d'un flux binaire aléatoire.

Paramètres :

- Bloc "constellation modulator" : sps (sample per symbol)=4, excess bw=900m, codage différentiel
- $F_c = 20\text{ MHz}$
- $f_p = 2\text{ MHz}$

A) Transmission numérique idéale

1. Réaliser la modulation 4-PSK, en bande de base, avec les paramètres spécifiés.
2. Faire varier le nombre d'échantillons par symbole (entre 2 et 32). Qu'observez-vous sur le diagramme de constellation (QT GUI constellation sink) ?
3. Ajouter le rational resampler avec le bon paramétrage, et vérifier l'amélioration.
4. Réaliser la chaîne de réception, avec démodulation. Pour cela, il faudra utiliser le bloc "Constellation Receiver".

B) Prise en compte du canal de propagation

Intercaler un canal de simulation ("Channel model") entre l'émission et la réception.

1. Influence du bruit

Ajouter du bruit : faire varier de 10 m à 500 m. Comparer pour la modulation 16 QAM et 4PSK.

A partir de quel niveau de bruit ne peut-on plus distinguer les symboles ? (IES : Interférences entre Symboles)

2. Offset temporel entre les horloges de Tx et Rx

Par la suite, on gardera la modulation 8-PSK. Fixer le bruit à 20 mV.

Introduire un décalage temporel sur les horloges d'échantillonnage (via le paramètre "epsilon"). Cela modélise l'offset d'échantillonnage (ou SRO - sample rate offset) entre les instants d'échantillonnage du Tx et du Rx.

Faire varier epsilon de 1.0001 à 1.005.

- Observations ?

A partir de quelle valeur d'offset temporel a-t-on de l'IES ?

Pour réduire les IES causées par le bruit et l'offset temporel, on réalise une étape de resynchronisation (*Time recovery*) qui permet de déterminer le "meilleur moment" pour échantillonner et ségréguer les symboles.

Sous GNURadio, deux blocs permettent de réaliser cette fonction : "Time recovery MM" et "Polyphase clock sync".

- Résumer le fonctionnement de ces algorithmes de synchronisation

- Modifier le paramètre sur le modulateur, et ajouter un bloc permettant de synchroniser Tx et Rx (fonction Clock recovery)

Ces blocs de synchronisation intègrent également un filtre Root Raised Cosine, qui corrige le suréchantillonnage précédent (il faudra donc retirer le "rational resampler" précédent).

*Pour le bloc "Polyphase clock sync", il faudra spécifier dans le paramètre taps, une variable qui sera associée aux coefficients du filtre RRC générés par la fonction firdes : `firdes.root_raised_cosine(nfilt/2,nfilt/4,1.0/float(samp_per_sym),0.35,11*samp_per_sym*nfilt)`*

avec nfilt le nombre de filtres du RRC (prendre nfilt=32)

3. Offset fréquentiel

Par la suite, on fixera epsilon à 1.001.

Introduire un offset fréquentiel qui traduit le décalage entre les fréquences porteuse (fp) du Tx et du Rx (CFO - carrier freq offset).

Faire varier le taux d'offset fréquentiel de 1e-3 à 50e-3.

- Observations ?

A partir de quelle valeur d'offset temporel a-t-on de l'IES ?

On fixera par la suite un taux d'offset fréquentiel de 10e-3.

4. Multi-trajets

Maintenant, nous allons analyser l'influence des multi-trajets sur le signal reçu.

Pour se focaliser uniquement sur l'impact des multi-trajets, remettre provisoirement les autres paramètres du canal à 0 (bruit et offsets).

Pour cela, on introduira dans le bloc "Channel model" 4 multi-trajets (sur un modèle de 15 taps). Spécifier les "taps" suivants (amplitudes associées à la réponse impulsionnelle du canal simulé) :

1, 0, 0, $0.99+0.1j$, 0, 0, 0, 0, $0.97-0.2j$, 0, 0, $0.72-0.05j$, 0, 0, 0,

- Comparer le signal temporel reçu en sortie du canal et le signal avant canal.
- Modifier les taps et contrôler l'influence sur le signal après passage dans le canal.
Est-ce conforme à vos attentes ?

En télécommunications, pour corriger l'impact du canal, on applique un **égaliseur** (ou **filtre adapté au canal**).

Sous GNURadio, plusieurs blocs réalisent cette opération, les principaux étant : "CMA equalizer" (à module constant, uniquement adapté pour les modulations PSK) et "LMS DD equalizer" (basé sur un algorithme des moindres carrés, adapté aussi bien pour les modulations PSK que QAM)

- Expliquer brièvement le fonctionnement de ces algorithmes
- Tester ces deux égaliseurs, et comparer.

4. Conclusion