IUT de Béziers	Travaux pratiques	R205
Réseaux et Télécoms		Signaux et systèmes de ...
Université de Montpellier	n°1 : GnuRadio	

Sources : Y. Haddab (IUT Béziers), E. Monier et J-P. Barbot (ENS Paris-Saclay)

1. Introduction

GNU Radio est une boîte à outils open-source pour la construction d'émetteurs/récepteurs radio logiciels. Contrairement aux systèmes classiques, ce ne sont pas des équipements matériels fixes qui réalisent les opérations à appliquer sur les signaux, mais un logiciel fortement modulable grâce à une programmation graphique par blocs.
En fait, GNU Radio est un environnement de traitement du signal qui englobe des centaines de blocs de traitement du signal et de communications numériques, tous développés en C++ ou Python. Il permet d'émettre et de recevoir des signaux physiques à l'aide d'interfaces matérielles appelées carte ou clé SDR (software defined radio). A l'IUT nous possédons par exemple des « cartes » Adalm Pluto et des clés USB SDR.

Les TP que nous réaliserons avec GNU Radio s'appuie sur la distribution Dragon OS qui intègre tous les composants pour faire fonctionner GNU Radio et les équipements SDR. La machine virtuelle utilisée pour ce TP est récupérable pour travailler à la maison.

2. Objectif

L'objectif de ce TP est de vous familiariser avec l'environnement GNU Radio dans le cadre de l'étude des fonctions de transfert. Dans les TP suivants nous utiliserons des cartes SDR pour émettre et recevoir des signaux et étudier les canaux de transmission.

3. Mise en place

- Vous travaillerez en binôme.
- Prenez un PC du chariot n°7, connectez vous au réseau wifi puis démarrez la machine virtuelle DragonOS. Vous pouvez alors la mettre en plein écran.
- Démarrez *GNU Radio Companion*, vous le trouverez dans le menu *Programmation*.
- Récupérez sur moodle le premier programme : « *premier-montage* ».

Le bloc « Options » définit certains paramètres de la simulation. Le bloc « variable » permet de fixer le « pas de la simulation », ici le pas est de 1/32000ème de seconde soit 32000 pas par seconde. Cela signifie qu'on ne peut raisonnablement pas étudier des signaux de fréquences plus grandes que 3200Hz. Le bloc « Low-pass filter taps » définit les paramètres du système.

Le bloc « Throttle » permet de préciser que le temps de la simulation est le temps réel, sinon la simulation se fait au maximum de la puissance de l'ordinateur.

Exercice

Dans cet exercice vous devez vous familiariser avec l'utilisation des blocs « générateur de signaux » (Signal Source) et « oscillo » (QT GUI Time Sink). Pour ce dernier on apprendra à faire des mesures préciser, à afficher l'un ou l'autre des signaux, à zoomer, à dézoomer...

1. Dessinez le système étudié par cette simulation.
2. Relevez l'amplitude du signal de sortie pour une fréquence d'entrée de 100Hz, 1000Hz, 3000Hz.
3. Mesurez la période du signal et vérifiez que la fréquence correspondante est bien celle imposée par la source.
4. Pourquoi le signal à 3000Hz est-il un peu déformé ?
5. Mesurez le déphasage entrée/sortie à 100Hz et à 1000Hz.
6. Donnez la valeur de la fonction de transfert complexe $\underline{H}(\omega)$ à 100Hz et à 1000Hz.

4. Linéarité

Dans cette partie vous allez vérifier que le système étudié est bien linéaire.

Pour ce faire vous devez vérifier que :

- Si s(t) est la sortie pour une entrée e(t) alors $\alpha.s(t)$ est la sortie pour une entrée $\alpha.e(t)$
- Si $s_1(t)$ et $s_2(t)$ sont les sorties pour les entrées $e_1(t)$ et $e_2(t)$ alors $s_1(t)+s_2(t)$ est la sortie pour une entrée $e_1(t)+e_2(t)$

Dans GNU Radio le bloc : Math Operator/Multiply Const permet de multiplier un signal par une constante et le bloc Math Operator/Add permet d'ajouter deux signaux.

1. Modifiez le montage pour simuler deux systèmes identiques. Le premier ayant pour entrée $e_1(t)=e(t)$, le second $e_2(t)=\alpha.e(t)$. Visualisez simultanément $\alpha.s_1(t)$ et $s_2(t)$.
2. Pour plusieurs fréquences et plusieurs types de signaux vérifiez que $\alpha.s_1(t) = s_2(t)$. Donner les preuves dans votre compte rendu.
3. Concevez et mettez en œuvre le montage qui permet de vérifier la seconde formule de la linéarité. Faites l'expérience et donner des preuves dans votre compte rendu.

5. Réponse impulsionnelle

La réponse impulsionnelle d'un système caractérise entièrement sa fonction de transfert.
Sous GNU Radio, par défaut, il n'y a pas de signal impulsionnel. Néanmoins il est facile d'en produire un qui « ressemble » (impulsion très courte et d'aire unité).

1. Chargez le schéma *rep-impulsion* (voir moodle).
2. Expliquez comment le schéma proposé produit une impulsion.
3. Quelle est sa durée ?
4. Quelle est son aire ?
5. Relevez la réponse impulsionnelle du système étudié.
6. Vérifiez que le système est bien causal.
7. Estimez rapidement l'aire de la réponse impulsionnelle.

6. Réponse indicielle

1. Réalisez un schéma permettant de mesurer la réponse indicielle du système.
2. Relevez la réponse indicielle.
3. Quel est le temps de réponse du système : 90 % de la valeur finale atteinte ?
4. Faites le lien entre réponse indicielle et réponse impulsionnelle.

7. Réponse harmonique = régime sinusoïdal

Le fichier *rep_harmonique* permet d'étudier simplement la fonction de transfert en régime sinusoïdal car elle permet de régler en direct la fréquence de la source.

1. Mesurez les paramètres de la fonction de transfert complexe pour les points suivants : 10Hz, 30Hz, 100Hz, 300Hz, 800Hz, 900Hz, 1000Hz, 1100Hz, 1200Hz, 3000Hz
2. Tracez le gain en fonction de la fréquence. En général on le trace en dB (G_{dB}=20 log G).
3. Tracez le déphasage en fonction de la fréquence.
4. Pourquoi ce système est-il appelé filtre passe-bas ?