Travaux pratiques sur l'identification et la linéarisation des amplificateurs de radiocommunications

1 Préambule

Le but de cette séance est d'améliorer les performances d'un amplificateur de puissance décrit sous Python. Pour cela, deux étapes sont nécessaires :

- le calcul de la prédistorsion (la fonction inverse) par Moindres Carrés Ordinaires (MCO),
- et l'application de cette fonction afin d'améliorer la linéarité.

Copier le répertoire *TpCSF* dans votre espace de travail. Il contient les fichiers suivants :

- *tpDPD.py* permettant la simulation du modèle non-linéaire de l'amplificateur,
- *commonFunctions.py* contenant les fonctions que nous développerons.
- un répertoire vide .*datas* dans lequel on stockera les fichiers de données d'entrée/sortie.

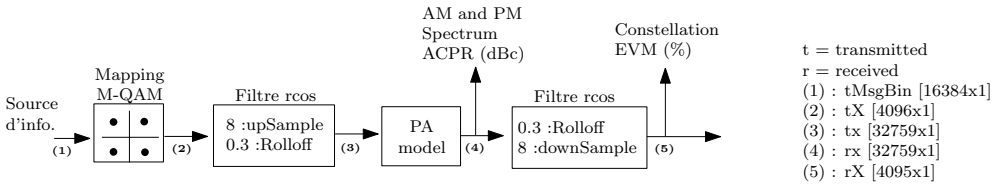


Figure 1 – Chaine de simulation 16-QAM

Simulez le fichier *tpDPD.py* (vous pouvez aussi travailler sous Jupyter Lab) dont le principe est sur la figure . Ce fichier permet d'appliquer une entrée 16-QAM à un modèle de PA non-linéaire de type Saleh et affiche les informations suivantes :

- les courbes AM/AM et AM/PM où l'on peut observer les non-linéarités de gain et de phase,
- le spectre où l'on peut observer les remontées spectrales sur les canaux adjacents, synonyme de mauvais *ACPR*,
- la constellation sur laquelle on remarque la dispersion des symboles et une rotation de phase (mauvais *EVM*),

L'objectif de cette séance est d'améliorer la linéarité de l'amplificateur afin d'avoir de meilleures réponses fréquentielles et temporelles.

2 Recherche des coefficients de la prédistorsion

Ouvrez le fichier *commonFunctions.py*. On va écrire la fonction *mco* où vous avez des commentaires qui vous guideront pour son écriture. Le but étant d'obtenir les paramètres d_i de la fonction inverse du $5^{\text{ème}}$ ordre décrite sous la forme $(2P+1=5)$:

$$tx = F(ty) = d_1 \cdot ty + d_3 \cdot |ty|^2 \cdot ty + d_5 \cdot |ty|^4 \cdot ty \tag{1}$$

Pour cela, on applique les MCO (voir le polycopié du cours). Le vecteur des paramètres à estimer est bien sûr :

$$\underline{\hat{\theta}} = \begin{bmatrix} d_1 & d_3 & d_5 \end{bmatrix}^T$$

Après identification, générer la sortie estimée $\underline{\hat{t}}_x$ et comparez-là avec la sortie réelle $\underline{t}_x$. Aussi, pour avoir un indicateur de la qualité de l'identification, créé une fonction *nmse()* qui permet de calculer le NMSE (Normalized Mean Squares Error) en dB. Un bon NMSE est souvent inférieur à -30 dB.

3 Tester la linéarisation

A présent, on va générer le signal de sortie de la DPD pour l'appliquer à l'amplificateur de puissance. Créez une copie du fichier *tpDPD.py* nommée *tpDPD2.py*. Dans ce fichier nous allons comparer l'amplificateur seul et l'amplificateur avec prédistorsion comme illustré sur la figure 2.

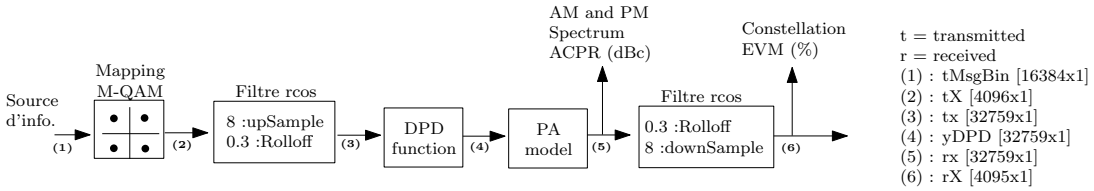


Figure 2 – Comparaison PA seul et avec prédistorsion simple

Comparez les constellations avec et sans prédistorsion et mettez les deux spectres sur la même figure afin de voir les améliorations en terme d'ACPR.
Calculer l'ACPR lower, l'ACPR upper et l'EVM (attention, il faut aligner tX et rX car il y a un échantillon de début en trop pour tX). Est-ce que ça passe les specs sachant que l'ACPR minimal est de -45dBc sur les deux bandes latérales et l'EVM est de 12.5%.

4 Généralisation du calcul de la prédistorsion

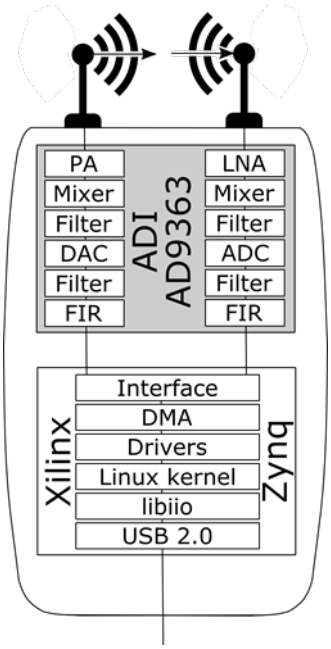
Dans la fonction *mco* précédente, l'ordre est fixé à 5 ce qui limite la taille de la fonction de prédistorsion. Généraliser ce programme à l'ordre $2P+1$ (le but étant toujours d'obtenir les paramètres d_i de la fonction inverse décrite sous la forme) :

$$tx = F(ty) = d_1 \cdot ty + d_3 \cdot |ty|^2 \, ty + d_5 \cdot |ty|^4 \, ty + \cdots + d_{2P+1} \cdot |ty|^{2P} \, ty \tag{2}$$

Tester les améliorations apportées par l'augmentation de l'ordre : 3, 5, 7, et 9.

5 Linéarisation du PA de la carte ADALM-Pluto

A présent, on change de contexte en travaillant directement sur la linéarisation de la carte ADALM-Pluto qui contient un tranceiver de Analog-Device. La chaine Tx de ce tranceiver contient un PA pouvant délivrer 0 dBm (1 mW), comme le montre la figure. Lorsque ce PA est utilisé à sa puissance maximale, on observe des remontées spectrales, synonyme de non-linéarités. On va tenter de les corriger.



Dans le répertoire *adalmPluto*, vous trouverez un fichier nommé *withoutDPD.py* qui réalise la transmission et la réception de données sous le standard *LTE/5G* à 20 MHz. Lisez-le bien et essayez de comprendre. L'exécution de ce programme permet de transmettre ce signal à une fréquence de 3.65 GHz. Il trace les données IQ, le spectre et les courbes AM/AM et AM/PM dynamiques. Le plus dure est déjà implémenté à savoir : la synchronisation des voies tx et rx et la correction de la rotation de phase du siganl rx.

Faites un *Enregistrez-sous* → *withDPD.py*

A présente, faites le nécessaire pour linéariser le PA. On ne s'intéressera qu'au spectre et à l'ACPR.