TP CHAINE DE TRANSMISSION OFDM

Introduction

Soit le système de transmission OFDM en bande de base représenté par la Fig. 1. Ce système transmet les symboles QPSK $\{a_0, a_1, \ldots, a_{N-1}\}$, $a_i \in \{(\pm 1 \pm j)/\sqrt{2}\}$, sur N sous-porteuses orthogonales. Après la modulation OFDM (IFFT), un intervalle de garde cyclique de longueur L est inséré. Ceci permet de s'affranchir des interférences entre symboles OFDM successifs lorsque la réponse impulsionnelle du canal $c(k)$ est nulle pour $k \geq L$. Le signal reçu $y(k)$ est également entâché d'un bruit additif blanc gaussien (BABG) complexe $n(t)$, de variance N_0.

Au récepteur, on supprime les L premiers échantillons correspondant à l'intervalle de garde cyclique avant démodulation OFDM (FFT). Les échantillons obtenus sont de la forme

$$Y_i = C_i a_i + N_i, \quad i = 0, \ldots, N-1,$$

où a_i est le symbole QPSK porté par la i-ème sous-porteuse, N_i est un BABG de variance N_0 et

$$\{C_i\}_{i=0}^{N-1} = FFT\{c(k), k = 0, \ldots, N-1\}.$$

Le canal de transmission introduit un décalage en fréquence Df (Hz) ainsi qu'un délai de propagation τ (s). Soit T_s la période d'échantillonnage, on peut écrire le délai de propagation sous la forme $\tau = (\theta + e)T_s$ avec

$$\theta = \left\lfloor \frac{\tau}{T_s} \right\rfloor$$
$$e = \frac{\tau - \theta T_s}{T_s}.$$

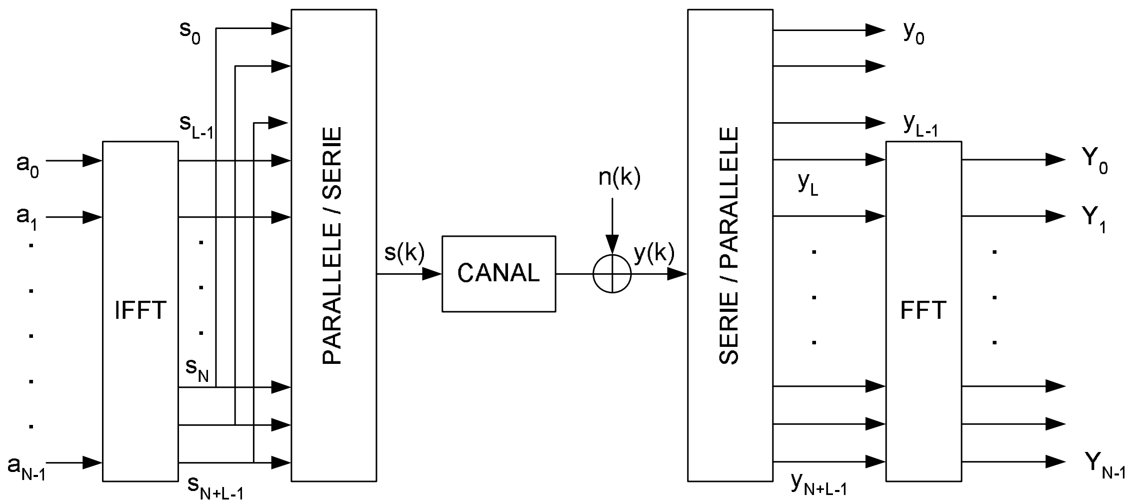

Fig. 1. Système de transmission OFDM à N sous-porteuses.

Ainsi, θ est le nombre entier d'échantillons retardés au récepteur et e désigne le retard fractionnaire. En supposant le retard fractionnaire nul, la sortie du canal devient

$$y(k) = e^{j2\pi DfkT_s} s(k-\theta) * c(k) + n(k)$$

Afin d'estimer les paramètres Df et θ, le récepteur utilise la métrique

$$P(d) = \sum_{m=0}^{L-1} y(d+m)y(d+m+N)^*,$$

où d est un nombre entier d'échantillons. On obtient alors les estimateurs suivants

$$\hat{\theta} = \arg\max_d |P(d)|$$

$$\hat{Df} = -\frac{\angle P(\hat{\theta})}{2\pi N T_s}.$$

Un script Python (`simu.py`) générant les échantillons $y(k)$ vous sera fourni pendant le TP. Le sous-programme Python `reponse_canal()` permet de générer la réponse impulsionnelle du canal $c(k)$ de longueur L échantillons. Lorsque le paramètre mode = 0, ce sous-programme rend le Dirac $c(k) = \delta(k)$. Lorsque le paramètre mode = 1, ce sous-programme rend un canal de Rayleigh d'étalement temporel T_m échantillons. Par défaut $T_m = 10$.

1. Vérifier que les paramètres choisis dans le script Pyton `simu.py` correspondent à une transmission selon la norme DVB-T en mode 2K étudiée en cours.
2. Tracer la réponse fréquentielle (module et phase) du canal quand le canal est un Dirac et pour le canal de Rayleigh. Pour le canal de Rayleigh, commenter l'allure de la réponse fréquentielle lorsque T_m varie.
3. Ecrire un sous-programme calculant $P(d)$, $d = 0, \ldots, N-1$. Tracer le module et la phase de $P(d)$ en fonction de d. Estimer les paramètres Df et θ. Quelle est la plage d'acquisition du paramètre Df?
4. Forcer le retard fractionnaire à $e = 0$ et $c(k) = \delta(k)$. Rajouter dans le corps du `script simu.py` une synchronisation en temps et en fréquence des symboles OFDM. La sortie du démodulateur (FFT) permet-elle de retrouver les symboles QPSK sans erreur de transmission ?
5. Même question quand e n'est pas forcé à 0.
6. On suppose maintenant que e n'est pas forcé à 0 et $c(k)$ est le canal de Rayleigh. Estimer la réponse fréquentielle du canal à l'aide des porteuses pilotes. On se servira du sous-programme `estimation_canal()`. Rajouter dans le corps du script `simu.py` une égalisation fréquentielle. La sortie du démodulateur (FFT) ne permet pas de retrouver les symboles QPSK sans erreur de transmission pour certaines sous-porteuses. Lesquelles?
7. Sans modifier les programmes utilisés à la question 5, remplacer le canal de Rayleigh par le Dirac avec e non forcé à 0. Cette fois la démodulation se passe sans erreur de transmission. Pourquoi? Estimer la valeur du retard fractionnaire.