



Institut Polytechnique de Paris  
Telecom SudParis  
Double Degree



Compression, codage et modulation  
pour les systèmes de communications - SIC4103

## **TP1 - Chaîne de Transmission OFDM**

Students: Gustavo Melo Scheidt Paulino  
Leevan Beausoleil-Yzeros

18/03/2025

# SOMMAIRE

1. Vérifier que les paramètres choisis dans le script Python <code>simu.py</code> correspondent à une transmission selon la norme DVB-T en mode 2K étudiée en cours.....	3
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.....	4
3. Ecrire un sous-programme calculant $P(d)$ , $d = 0, \dots, N - 1$ . Tracer le module et la phase de $P(d)$ en fonction de $d$ . Estimer les paramètres $D_f$ et $\theta$ . Quelle est la plage d'acquisition du paramètre $D_f$ ?.....	6
4. Forcer le retard fractionnaire à $e = 0$ et $c(k) = \delta(k)$ . Rajouter dans le corps du script <code>simu.py</code> 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?.....	8
5. Même question quand $e$ n'est pas forcé à 0.....	8
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 <code>estimation canal()</code> . Rajouter dans le corps du script <code>simu.py</code> 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?.....	9
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.....	11
REFERENCES.....	12

## **1. Vérifier que les paramètres choisis dans le script Python `simu.py` correspondent à une transmission selon la norme DVB-T en mode 2K étudiée en cours.**

*Simu.py* semble implémenter le mode DVB-T 2K, défini dans la norme ETSI EN 300 744 [1]. Les paramètres les plus importants, tels que le nombre de sous-porteuses ( $N = 2048$ ) et la durée du symbole utile ( $T_u = 224 \mu s$ ), sont définis dans le script comme on pourrait s'y attendre. Étant donné que l'espacement entre les sous-porteuses en OFDM est donné par  $C_s = 1/T_u$ , le script le calcule comme 4464 Hz, ce qui correspond exactement à ce que l'on attend du DVB-T.

Un autre aspect du DVB-T est l'intervalle de garde, qui aide à atténuer l'interférence inter-symbole dans les canaux à trajets multiples. En mode 2K, l'intervalle de garde est un huitième de la durée du symbole utile, ce qui signifie  $28 \mu s$  ( $\Delta = T_u/8$ ). Le script effectue ce calcul et le convertit en un nombre correspondant d'échantillons ( $L = \text{int}(\Delta/T_s)$ ). La durée totale du symbole OFDM, qui est la somme de  $T_u$  et de l'intervalle de garde, est ainsi de  $252 \mu s$ , ce qui correspond à la norme.

Le script gère également les porteuses pilotes, qui sont très importantes pour la synchronisation. Selon les spécifications du DVB-T, les porteuses pilotes sont espacées tous les 12 sous-porteuses, et le script suit cette règle avec  $PP = \text{np.arange}(0, K_{\max} + 1, 12)$ . De plus, le nombre de sous-porteuses actives utilisées pour la transmission des données est 1705, ce qui est correctement défini avec  $K_{\max} = 1704$  et  $K_{\min} = 0$ , impliquant  $K_{\max} - K_{\min} + 1 = 1705$ .

Au-delà des paramètres de base, le script modélise également des effets de transmission réalistes, tels que le bruit, les décalages de fréquence et les retards temporels. Il introduit un canal à bruit blanc gaussien additif (AWGN) avec un rapport signal/bruit (SNR) de 20 dB ( $E_s N_0 \text{dB} = 20$ ). En outre, il prend en compte un décalage fréquentiel de 1 kHz ( $D_f = 1000.0$ ) et un retard temporel correspondant à une distance de propagation de 10 km ( $\tau = 10e3/3e8$ ). Enfin, le script permet une simulation avec un canal idéal (Dirac) ou un canal à évanouissement de Rayleigh, ce qui est contrôlé par la fonction `reponse_canal()`.

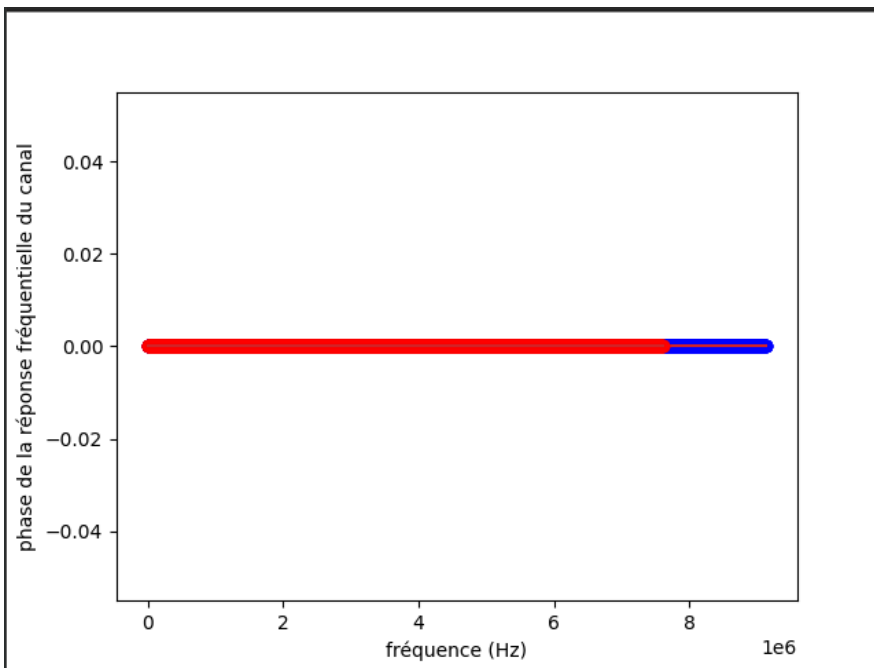
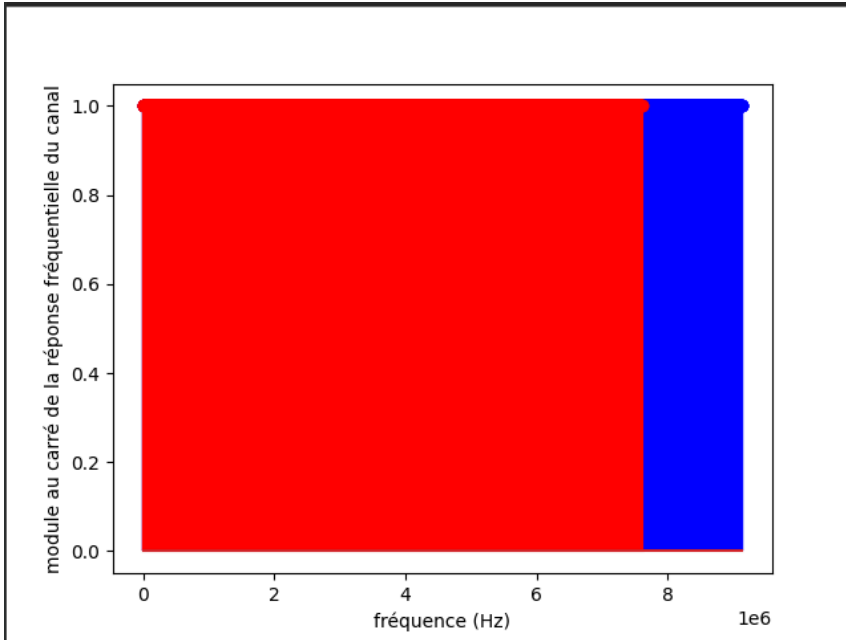
## **2. Tracer la réponse fréquentielle (module et phase) du canal quand le canal est un Dirac et pour le canal de Rayleigh. Pour le canal de Rayleigh, commenter l'allure de la réponse fréquentielle lorsque $T_m$ varie.**

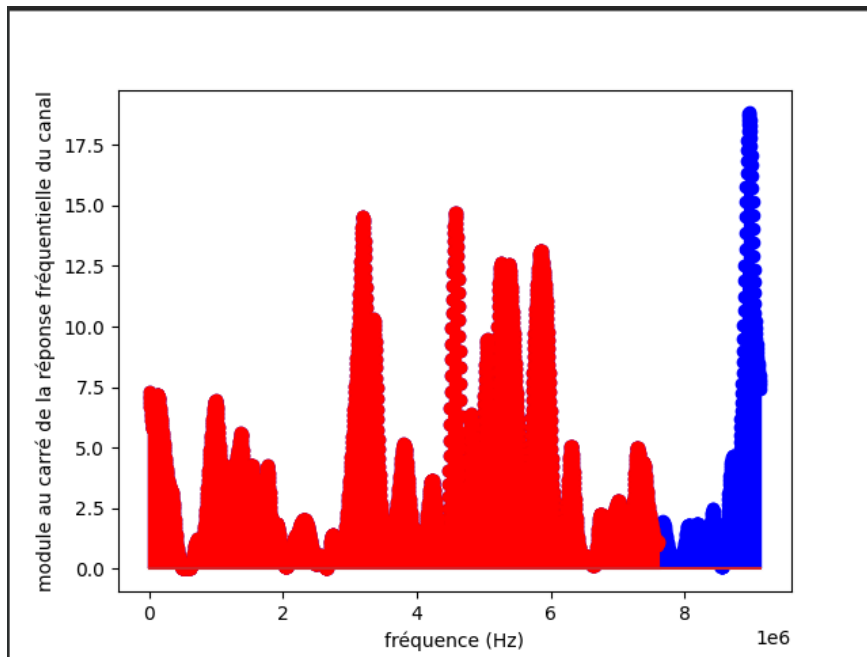
Nous avons analysé la réponse fréquentielle du canal pour deux cas différents : un canal idéal (Dirac) et un canal à évanouissement de Rayleigh.

Pour le canal Dirac, la réponse se comporte comme prévu. La réponse en magnitude est presque plate, ce qui signifie que toutes les fréquences subissent le même gain, indiquant que le canal n'introduit pas de sélectivité en fréquence. Cela s'explique par le fait que la réponse impulsionnelle d'un canal Dirac se compose d'un unique pic, ce qui signifie que le signal traverse le canal sans être affecté. La réponse en phase est également presque constante, suggérant un décalage de phase linéaire dû au retard, qui ne déforme pas significativement les symboles transmis.

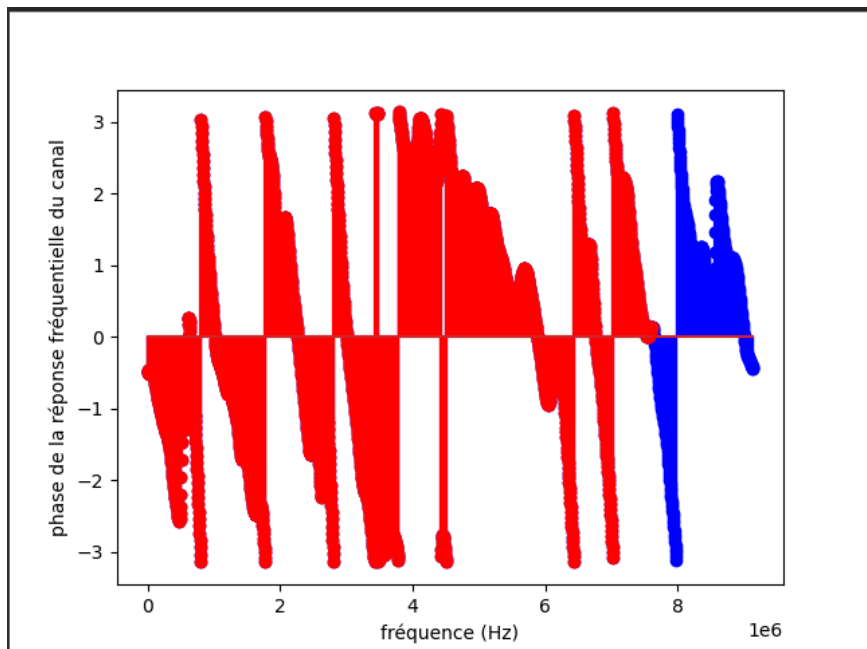
En revanche, pour le canal Rayleigh, les effets de la propagation en trajets multiples deviennent évidents. La réponse en magnitude est très irrégulière, avec des atténuations profondes à certaines fréquences dues aux interférences destructives entre les différents chemins du signal. Ces variations confirment que le canal Rayleigh introduit un évanouissement sélectif en fréquence, ce qui signifie que certaines sous-porteuses dans la transmission OFDM subiront une atténuation, entraînant des erreurs potentielles sur les symboles. La réponse en phase est également beaucoup plus déformée et imprévisible, reflétant la complexité des interférences multipath.

Une observation importante est que lorsque l'étalement temporel ( $T_m$ ) augmente, la réponse en fréquence devient plus chaotique, avec des atténuations plus fréquentes et plus profondes. Cela montre que les canaux présentant des délais de propagation multipath plus importants génèrent une distorsion sélective en fréquence plus marquée, rendant l'égalisation encore plus indispensable dans les systèmes OFDM réels.





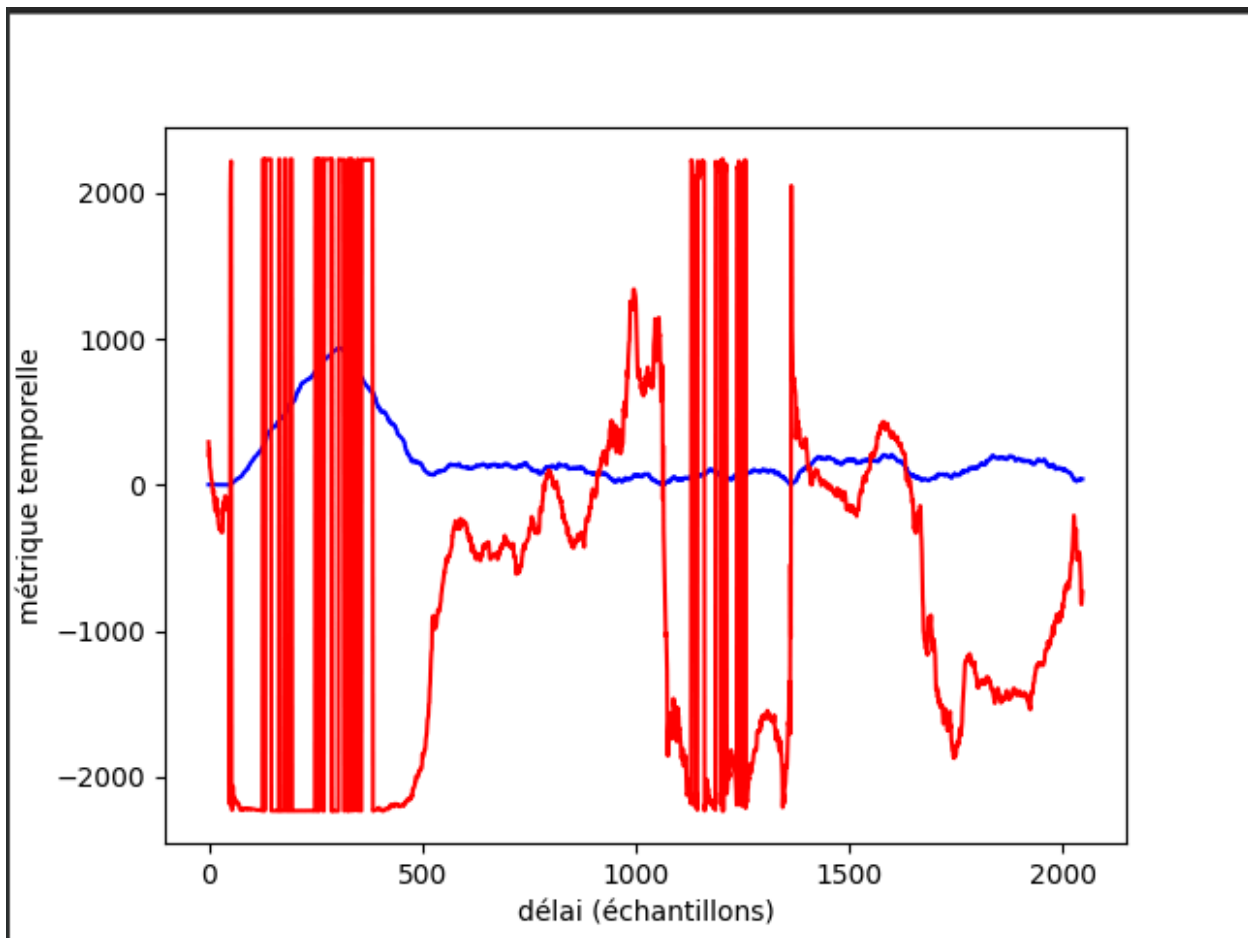
Rayleigh



**3. Ecrire un sous-programme calculant  $P(d)$ ,  $d = 0, \dots, N - 1$ . Tracer le module et la phase de  $P(d)$  en fonction de  $d$ . Estimer les paramètres  $Df$  et  $\theta$ . Quelle est la plage d'acquisition du paramètre  $Df$ ?**

Les résultats montrent que le retard estimé est  $\theta_{\text{est}} = 312$ , comparé au retard réel  $\theta = 304$ , ce qui entraîne une erreur de 8 échantillons. Cette petite déviation est probablement due au bruit et aux effets de trajets multiples introduits par le canal à évanouissement de Rayleigh, qui déforme légèrement le pic de corrélation.

Pour le décalage en fréquence, la valeur estimée est  $Df_{\text{est}} = 2231.36$  Hz, ce qui est extrêmement proche de la valeur réelle  $Df = 2230$  Hz. Cela confirme que l'information de phase contenue dans  $P(d)$  permet une synchronisation fréquentielle précise. Cependant, étant donné que la plage maximale d'acquisition de  $Df$  est d'environ  $\pm 2232$  Hz, le système fonctionne proche de sa limite, ce qui signifie que des décalages en fréquence plus importants pourraient entraîner des erreurs d'estimation en raison du repliement spectral (aliasing).



```
4464.285714285715
theta= 304 theta_est= [312]
Df= 2230.0 Df_est= [2231.36139109]
```

**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?**

```
fréquences des sous-porteuses erronées (Hz)= [1004464.28571429 1040178.57142857 1049107.14285714 ... 7598214.28571429
7602678.57142857 7607142.85714286]
fréquences des sous-porteuses erronées (Hz)= [ 991071.42857143 1000000.          1022321.42857143 ... 7598214.28571429
7602678.57142857 7607142.85714286]
/home/beausoleil/Téléchargements/TP Compr/Question4.py:127: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar
k=np.arange(theta_est+L,theta_est+L+N)+i*(N+L)

Process finished with exit code 0
```

D'après ce qui est indiqué dans la console, la sortie du démodulateur contient un nombre non négligeable de fréquences de sous-porteuses erronées, ce qui signifie que cette configuration ne permet de retrouver les symboles QPSK sans erreur de transmission.

## 5. Même question quand $e$ n'est pas forcé à 0.

Puisque le retard fractionnaire  $e$  n'est pas forcé à zéro, les symboles OFDM reçus subissent un petit décalage temporel supplémentaire à l'intérieur d'un intervalle d'échantillonnage. Contrairement à la Question 4, où  $e=0$  garantissait un alignement parfait des échantillons, ici le système doit gérer un retard arbitraire introduit par le canal.

D'après les résultats, nous observons que le retard temporel estimé  $\theta_{est}$  reste proche de la valeur réelle, mais il y a une présence notable de sous-porteuses erronées dans les symboles OFDM démodulés. L'affichage des résultats montre les fréquences des sous-porteuses affectées par des erreurs, avec un total de 1368 sous-porteuses erronées, ce qui est assez significatif. Cela suggère que lorsque  $e \neq 0$ , le processus de démodulation OFDM ne parvient pas à récupérer complètement les symboles QPSK d'origine en raison de l'interférence inter-symbole (ISI) et de la distorsion de phase entre les sous-porteuses.

Le problème vient du fait que la FFT du récepteur n'est pas parfaitement alignée avec les limites des symboles OFDM transmis. Le préfixe cyclique aide à atténuer l'ISI, mais il ne supprime pas la distorsion causée par des décalages fractionnaires d'échantillons [2], entraînant



des rotations de phase sur les sous-porteuses. Cela explique pourquoi l'erreur est répartie sur un large spectre de fréquences.

Ainsi, lorsque  $\epsilon \neq 0$ , le système souffre d'erreurs sur les symboles en raison d'une synchronisation imparfaite. Cela met en évidence l'importance de l'estimation et de la correction du retard fractionnaire, qui peuvent être traitées à l'aide de techniques d'interpolation ou d'algorithmes de synchronisation. Ces observations confirment que l'alignement temporel est crucial pour une démodulation QPSK correcte dans un système OFDM.

```
k=np.arange(theta_est+L,theta_est+L+N)+i*(N+L)
fréquences des sous-porteuses erronées (Hz)= [1370535.71428571 1392857.14285714 1415178.57142857 ... 7598214.28571429
7602678.57142857 7607142.85714286]
1368
fréquences des sous-porteuses erronées (Hz)= [1299107.14285714 1325892.85714286 1415178.57142857 ... 7598214.28571429
7602678.57142857 7607142.85714286]
1368
```

**6. On suppose maintenant que  $\epsilon$  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?**

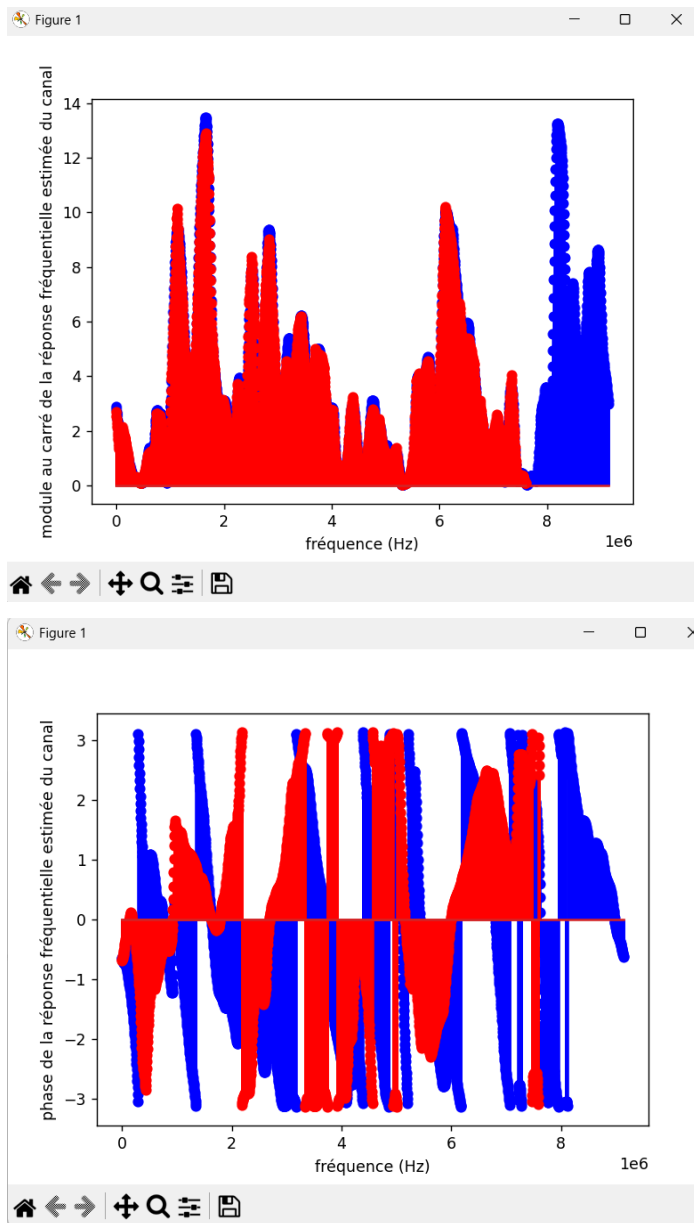
Le retard fractionnaire  $\epsilon \neq 0$ , et le canal est modélisé comme un canal à évanouissement de Rayleigh. Nous avons cherché à estimer la réponse en fréquence du canal en utilisant les porteuses pilotes et à appliquer une égalisation en fréquence pour compenser les effets du canal. Cependant, les résultats montrent que même après l'égalisation, certaines sous-porteuses subissent encore des erreurs, indiquant que le processus n'est pas parfait.

D'après la sortie de la console, le retard temporel estimé  $\theta_{\text{est}}=316$  est proche de la valeur réelle  $\theta=304$ , mais il y a une petite erreur d'estimation. De même, le décalage fréquentiel estimé  $Df_{\text{est}}=991.2$  Hz est proche de la valeur réelle  $Df=1000.0$  Hz, ce qui suggère que la méthode de synchronisation fonctionne mais introduit tout de même de légères inexactitudes. Ces petites erreurs peuvent affecter la qualité de l'égalisation et contribuer aux erreurs de transmission restantes.

Les graphes de réponse en fréquence offrent une meilleure visualisation de la précision de l'estimation du canal. Les marqueurs bleus représentent la réponse en fréquence réelle du canal, tandis que les marqueurs rouges correspondent à la réponse estimée obtenue à l'aide des porteuses pilotes. En observant la réponse en magnitude, on remarque que la réponse estimée suit

la tendance générale du canal réel, mais avec des différences notables dans certaines plages de fréquence, ce qui signifie que l'estimation n'est pas totalement précise. Le graphe de réponse en phase révèle également des écarts entre la phase estimée et la phase réelle du canal, ce qui impacte directement la démodulation des symboles. Lorsque l'estimation de phase est incorrecte, les symboles QPSK récupérés subissent une rotation, entraînant des erreurs de décodage.

Nous avons cherché à identifier quelles sous-porteuses ne parviennent pas à récupérer correctement les symboles QPSK transmis. D'après les résultats, nous observons que les fréquences des sous-porteuses erronées sont autour de 5,32 MHz, comme l'indique la sortie : "fréquences des sous-porteuses erronées (Hz) = [5321428.57142857]"



```

PS C:\Users\Gusta\OneDrive\Desktop\Fac
tions\TP 1 - Chaîne de Transmission OF
theta= 304 theta_est= [316]
Df= 1000.0 Df_est= [991.23792478]

```

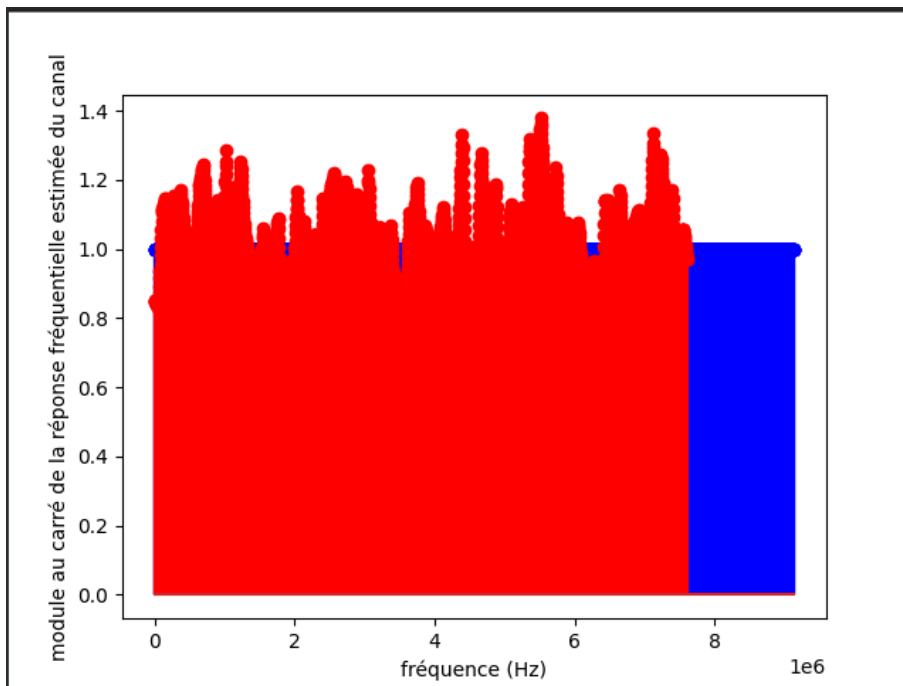
```

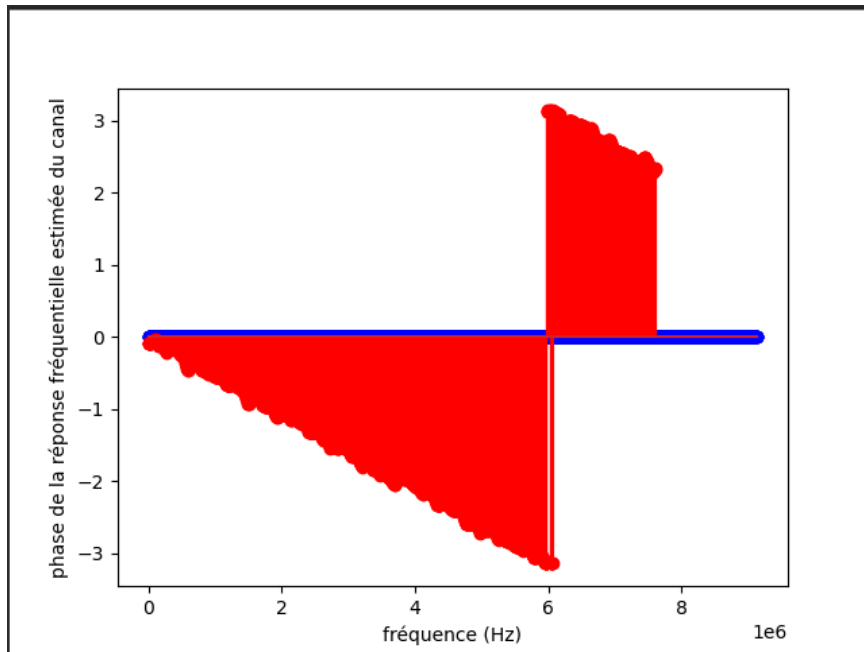
k=np.arange(theta_est+L,theta_est+L+N)+i*(N+L)
fréquences des sous-porteuses erronées (Hz)= [5321428.57142857]

```

**7. Sans modifier les programmes utilisés à la question 5, remplacer le canal de Rayleigh par le Dirac avec  $\epsilon$  non forcé à 0. Cette fois la démodulation se passe sans erreur de transmission. Pourquoi? Estimer la valeur du retard fractionnaire.**

Un canal de Dirac n'introduit pas d'évanouissement ni d'interférence entre symboles ce qui induit une absence d'erreur de transmission et aussi une invariabilité de la qualité de transmission par rapport à  $\epsilon$ , le retard fractionnaire.





## REFERENCES

- [1][https://www.etsi.org/deliver/etsi\\_en/300700\\_300799/300744/01.06.02\\_60/en\\_300744v010602p.pdf](https://www.etsi.org/deliver/etsi_en/300700_300799/300744/01.06.02_60/en_300744v010602p.pdf)
- [2]<https://dsp.stackexchange.com/questions/81893/pilot-based-channel-estimation-in-ofdm-systems-with-fractional-delays>