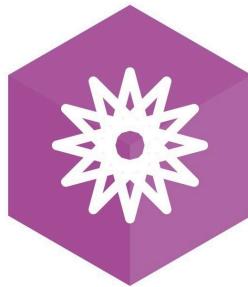


Compte rendu SAE 22 : Mesurer ou caractériser un signal ou un système

Réalisé par : Di GUSTO Gabin, YILMAZ Semi, SAOUD Nohame
Avec l'aide de **Mr GIVRON, Mr VANSTRACEELE et Mr VUILLEMIN**

IUT Nord Franche Comté
BUT Réseaux et Télécommunication
2024-2025



R&T
DUT Réseaux &
Télécommunications
IUT Belfort-Montbéliard



1. Introduction	3
2. Configuration du matériel	3
2.1 Présentation de la SAE	3
3. Mesure et calculs	4
3.1 Mesurer le temps de traitement de la chaîne de conversion CAN-CNA	4
3.2 La plage d'utilisation en tension	5
3.3 La valeur du quantum, pour plusieurs codecs	9
3.4 La fréquence d'échantillonnage	11
3.5 La bande passante occupée par le signal	14
3.6 Mesure du SNR	17
4. Conclusion	20
4.1 Les problèmes rencontrés ainsi que les solutions mise en place	20
4.2 Conclusion	20
5. Annexe	22
5.1 Auto-évaluation	22

1. Introduction

La SAE 22 porte sur la mesure et la caractérisation des signaux et des systèmes. Elle a pour objectif d'analyser les signaux générés par un système de transmission afin d'approfondir la compréhension de leur comportement et de leurs performances.

Dans ce cadre, nous avons mobilisé nos compétences en mesure, en caractérisation et en analyse des signaux. L'étude portera sur différents types de systèmes de transmission, notamment les réseaux de communication, les systèmes audio.

Les résultats obtenus feront l'objet d'une analyse approfondie et seront présentés sous forme d'un rapport détaillé

2. Configuration du matériel

2.1 Présentation de la SAE

La SAÉ 22 a pour objectif de nous faire manipuler et analyser des signaux numériques dans un contexte de téléphonie VoIP. Nous avons mis en œuvre une chaîne complète de transmission grâce à différents outils comme Raspberry Pi, FreePBX, Linphone et mené et nous avons utilisé GNU radio pour caractériser les signaux en fonction des codecs, du débit et de la fréquence d'échantillonnage.

Le système VoIP a été mis en place avec :

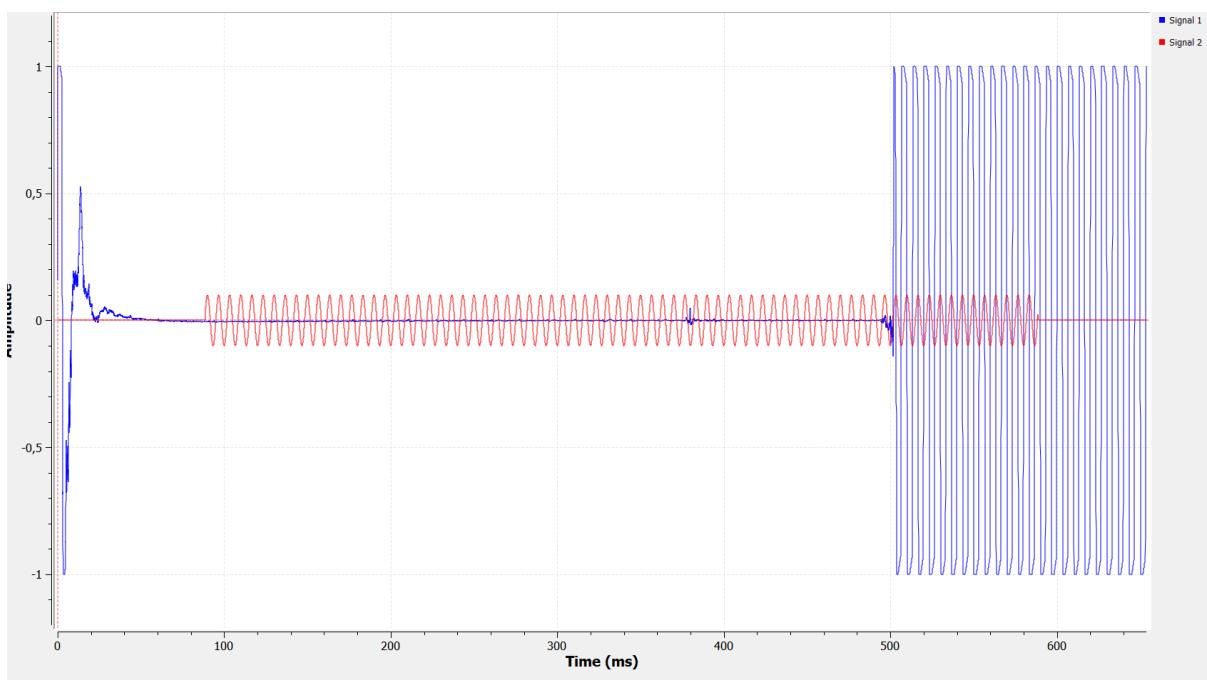
- Un Raspberry Pi comme serveur SIP via FreePBX,
- Une borne Linksys configurée en DHCP (192.168.1.0/24),
- Deux PC clients avec Linphone installé,
- Un pc avec GNU radio pour analyser les signaux
- Connexion des appareils via un réseau local filaire.

Les extensions SIP ont été créées sur FreePBX. Linphone a été configuré pour permettre la sélection de codecs, du débit et de la fréquence d'échantillonnage.

3. Mesure et calculs

3.1 Mesurer le temps de traitement de la chaîne de conversion CAN-CNA

Après avoir configuré le matériel à l'aide du fichier *latence.grc* et exécuté ce dernier sur GNU Radio, nous avons obtenu le graphique suivant :



Dans l'analyse du signal observé, on distingue une émission représentée en rouge, apparaissant quelques instants après une sinusoïde bleue. Cette dernière constitue la réponse initiale du système.

Afin de déterminer le temps de latence, il est nécessaire de mesurer l'écart temporel entre le début de l'émission de la sinusoïde bleue et le début de l'émission de la sinusoïde rouge. La différence de ces deux instants permet d'évaluer précisément le délai de réponse du système. Dans notre cas, l'écart temporel entre le début de l'émission de la sinusoïde bleue et le début de l'émission de la sinusoïde rouge est d'environ 408μs.

3.2 La plage d'utilisation en tension

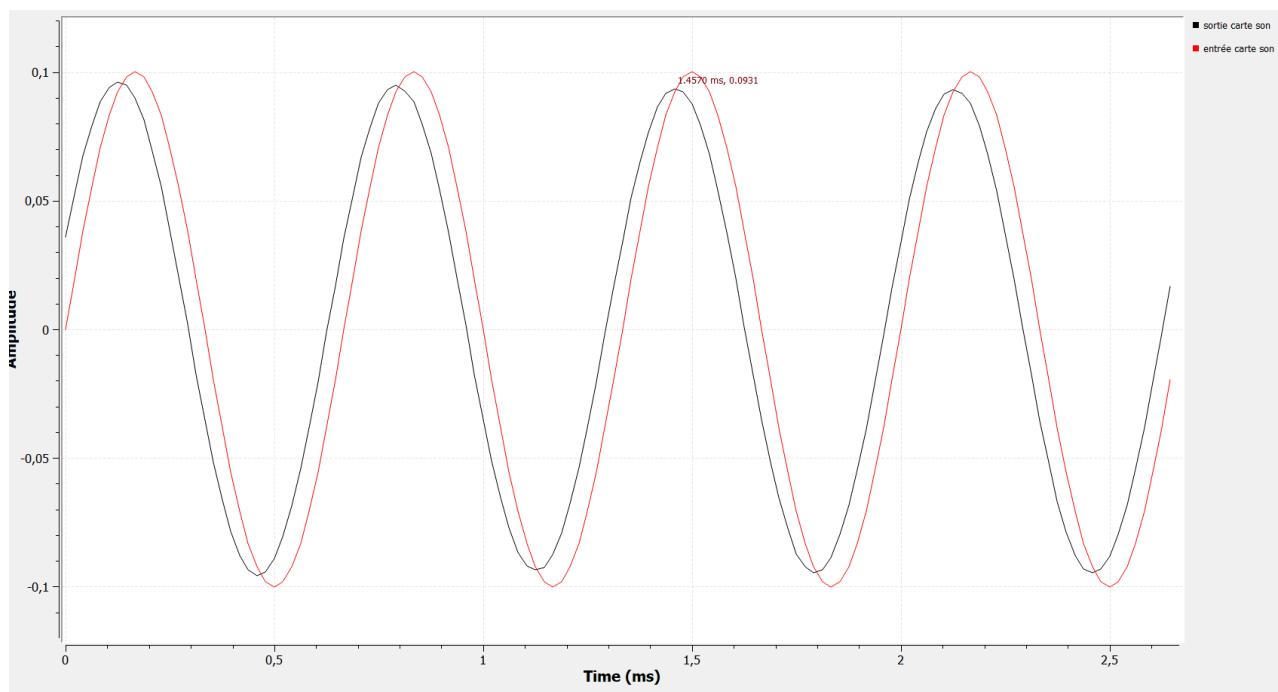
Pour établir la plage d'utilisation de la tension, nous avons utilisé le fichier limite_tension.grc, que nous avons exécuté sous GNU Radio.

L'approche adoptée consiste à comparer les valeurs de Ve et Vs sur plusieurs intervalles d'amplitude. Afin d'assurer une analyse précise, nous avons choisi d'effectuer ces comparaisons avec un pas de 0.3 unités. Cette méthodologie permet d'obtenir une estimation fiable des limites de fonctionnement en tension.

Valeur obtenue pour:

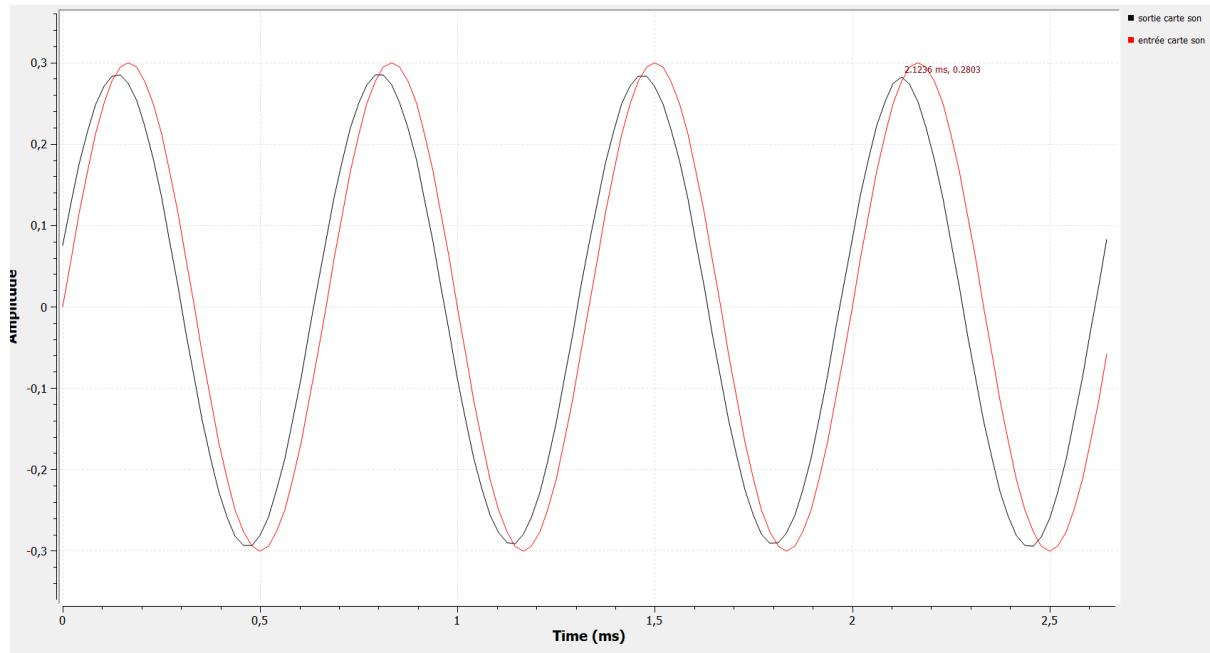
$$vs = 0.1v$$

$$ve = 0.096$$



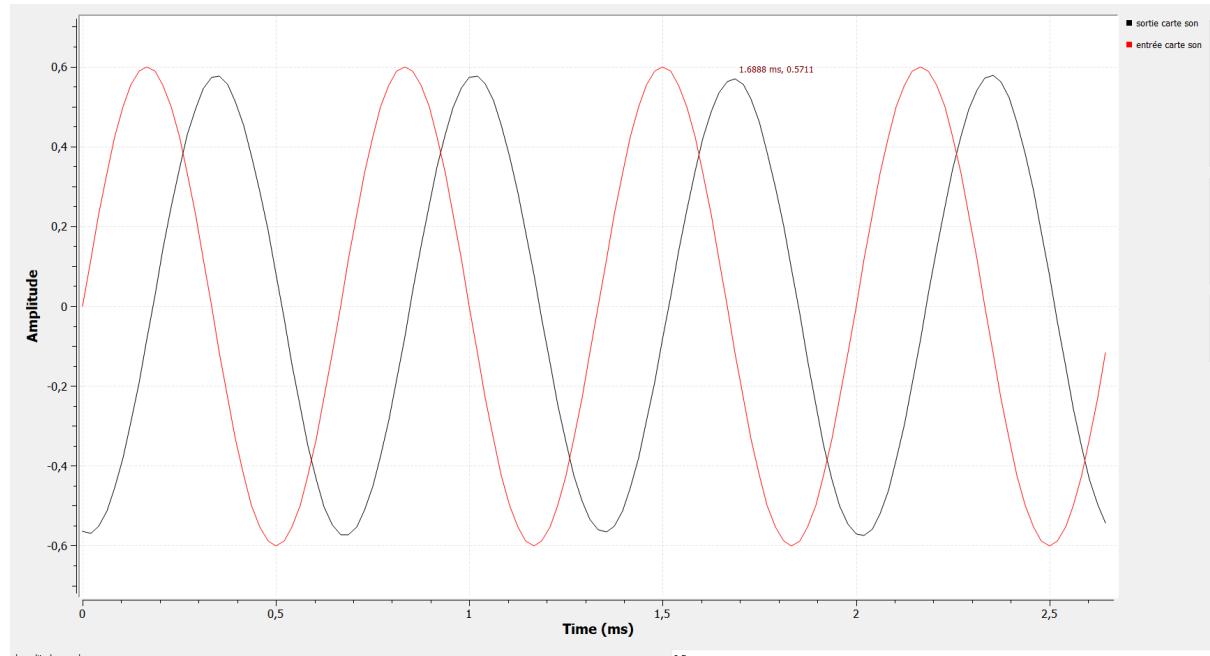
$v_e = 0.3v$

$v_s = 0.28$



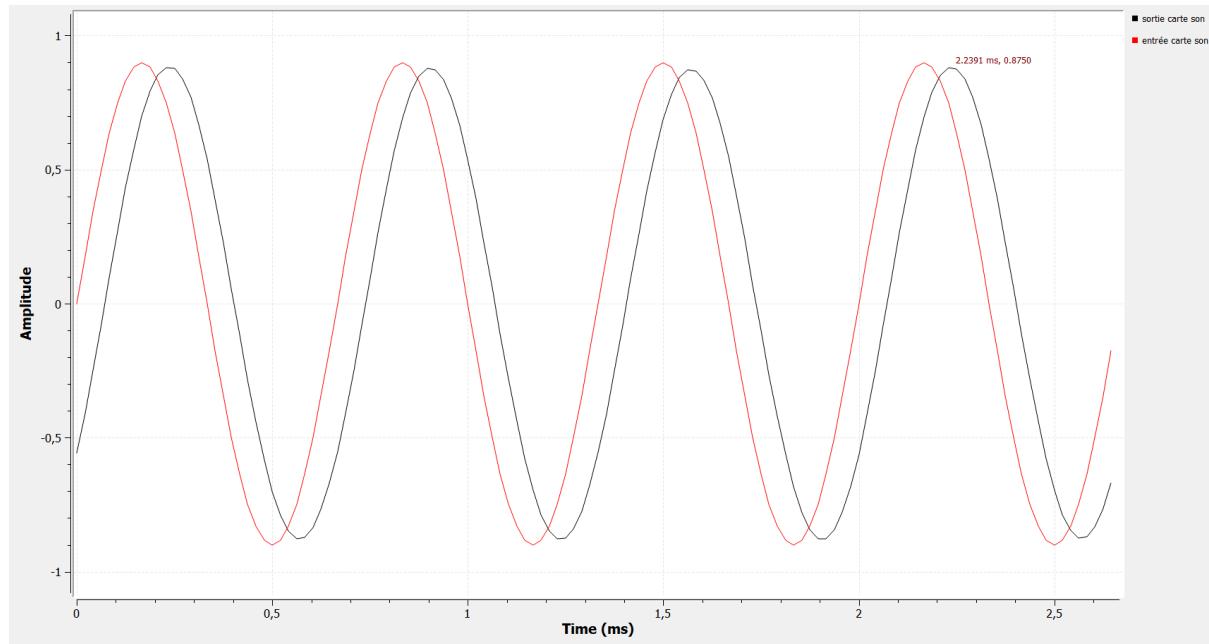
$v_e = 0.6v$

$v_s = 0.57$



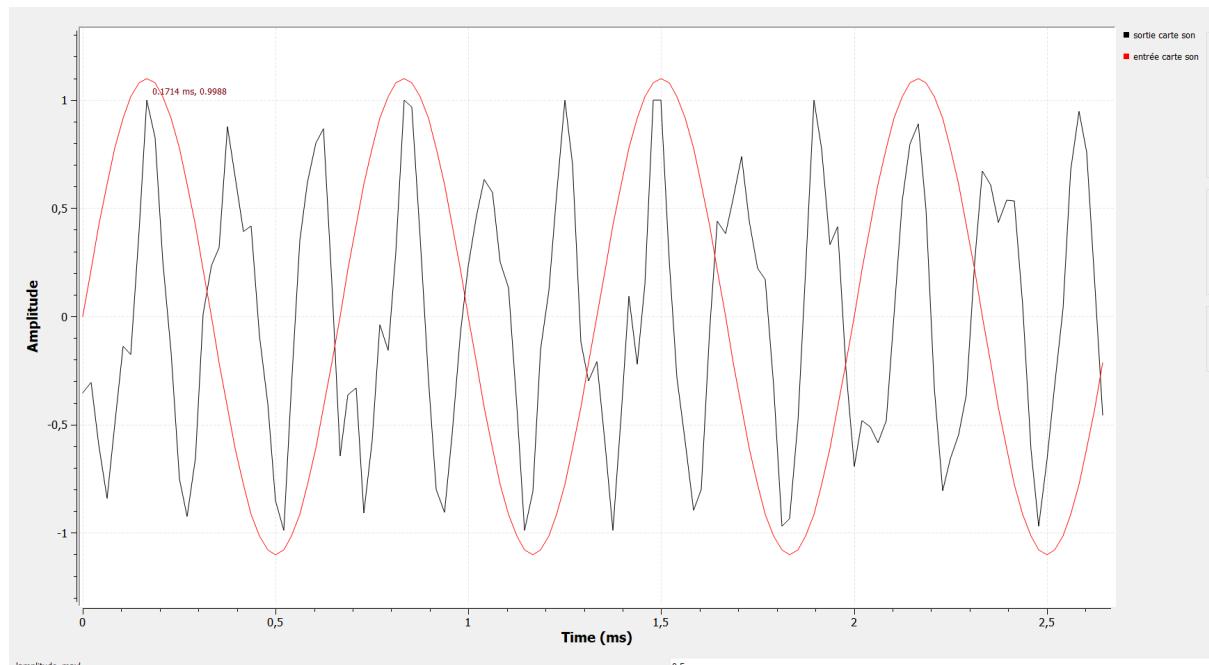
$v_e = 0.9v$

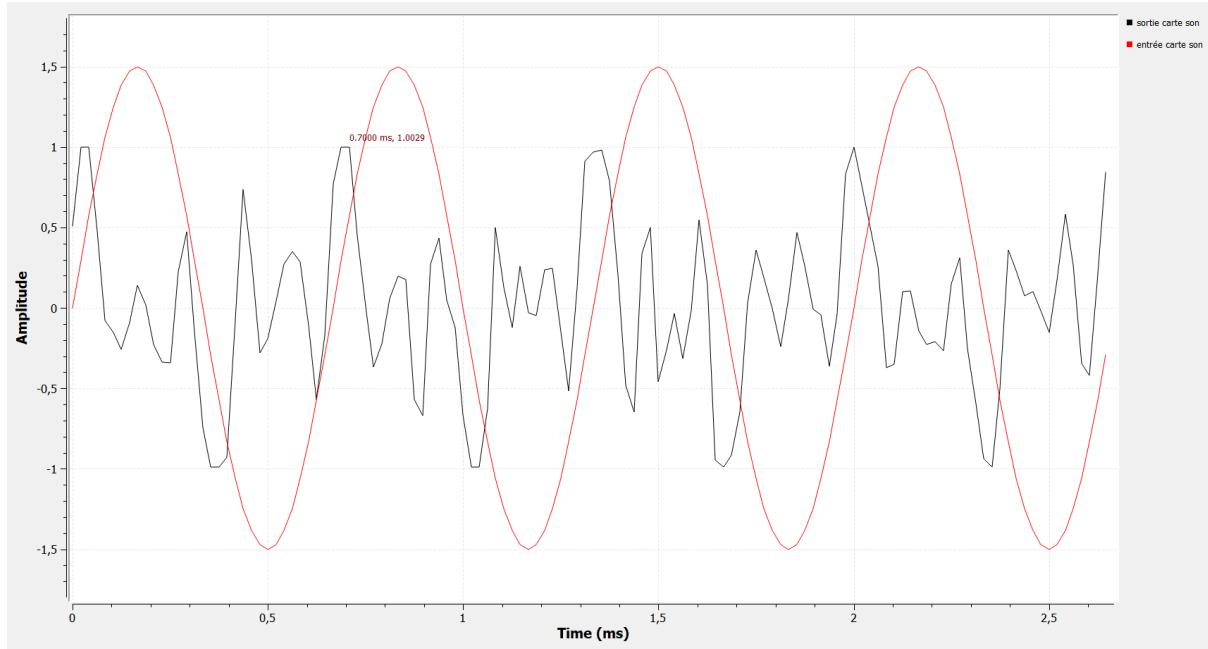
$v_s = 0.88v$



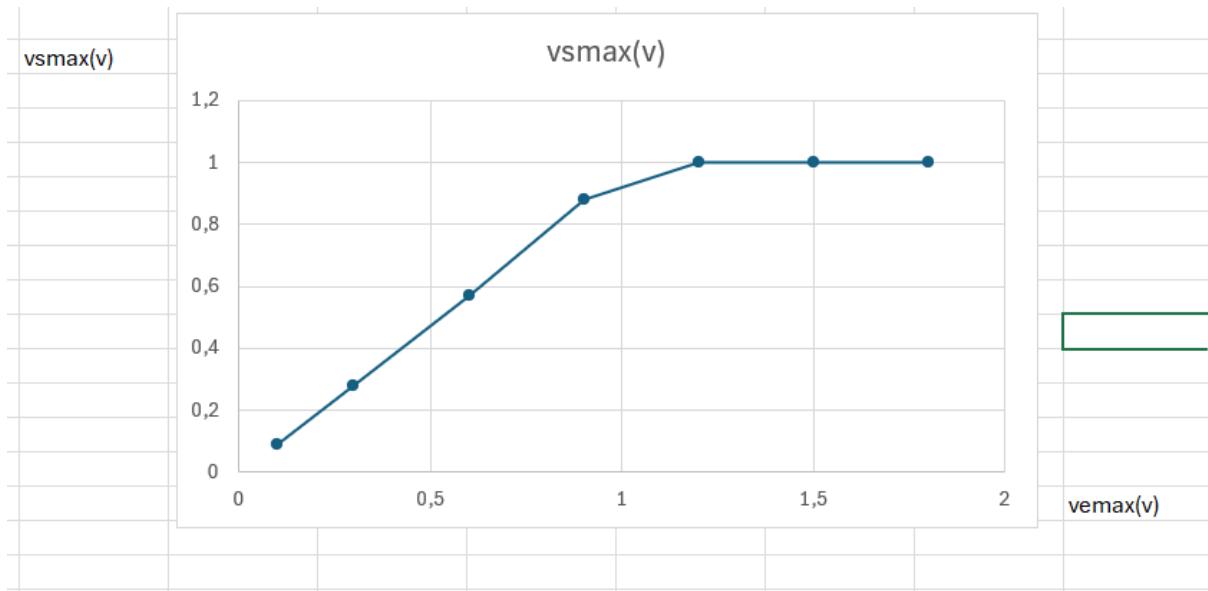
$v_e = 1.2$

$v_s = 1$



$v_e = 1.5$ $v_s = 1$ 

Une fois toutes les valeurs déterminées, il est possible de tracer la caractéristique entrée-sortie $V_{smax} = g(V_{max})$. Ce tracé permet d'analyser la relation entre la tension d'entrée maximale (V_{max}) et la tension de sortie maximale (V_{smax}), mettant ainsi en évidence le comportement du système en fonction des variations de l'entrée.

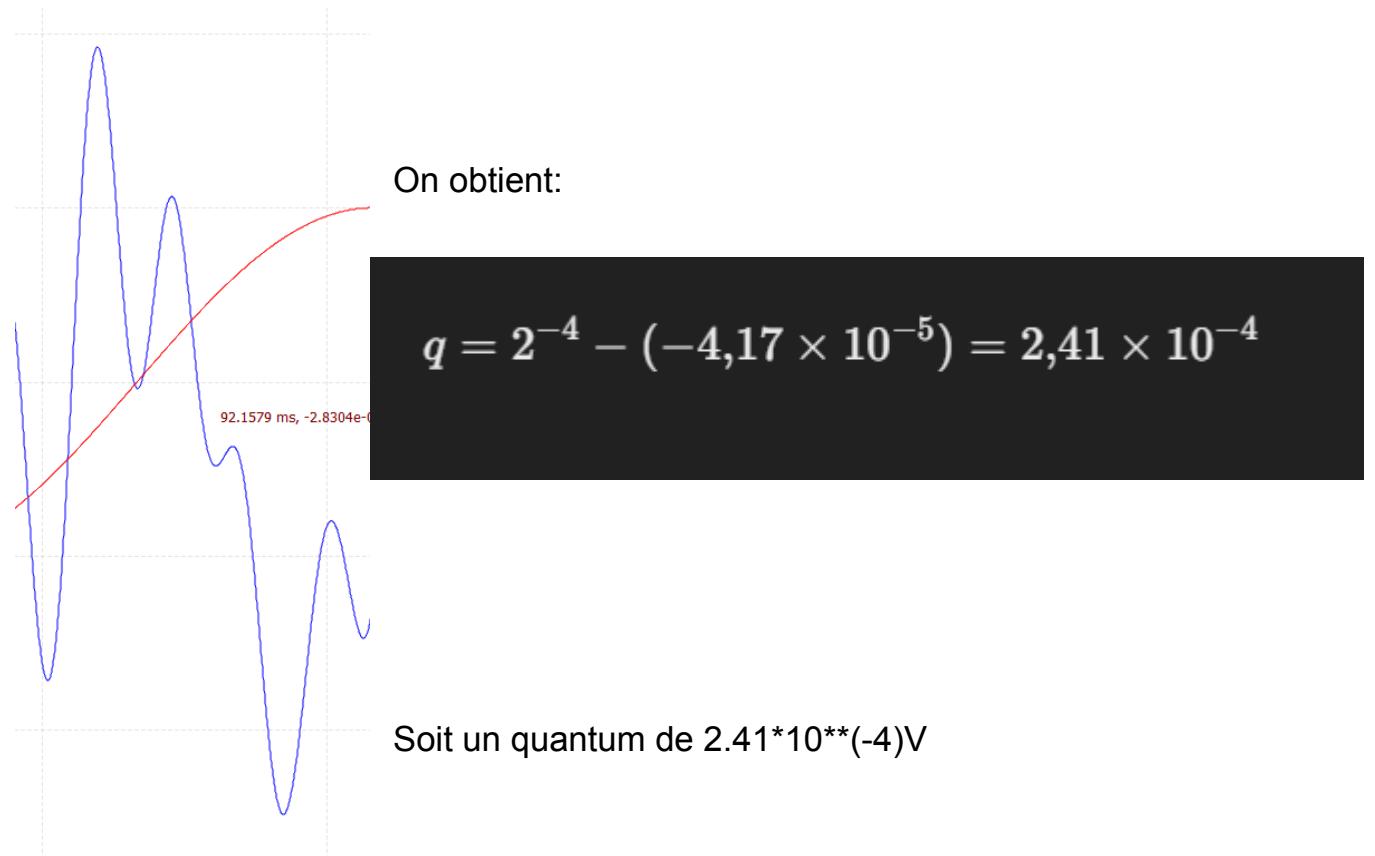


3.3 La valeur du quantum, pour plusieurs codecs

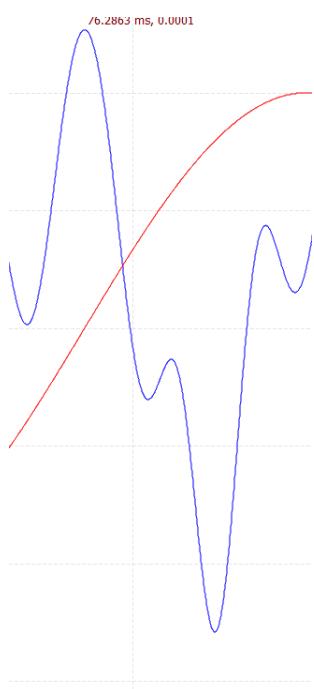
Pour déterminer le quantum, nous avons exécuté le fichier quantum.grc mis à notre disposition.

Dans cette étape, nous avons comparé les valeurs du quantum en utilisant trois codecs différents. L'analyse des courbes obtenues révèle des formes en marches, caractéristiques du processus de quantification. Afin d'estimer la valeur du quantum, nous calculons la différence entre deux niveaux successifs de ces marches. Cette approche permet d'obtenir une mesure précise du pas de quantification propre à chaque codec testé.

On commence par le quantum du codec opus



Puis on fait avec le codec speex 16000

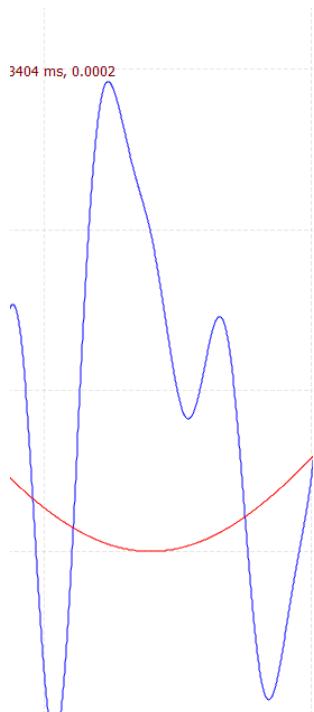


On obtient

$$q = 1.25 \times 10^{-4} - (-4.7 \times 10^{-5}) = 1.72 \times 10^{-4}$$

soit le quantum est 1.72×10^{-4}

Puis on fait avec le codec speex 8000



On obtient

$$q = 2 \times 10^{-4} - 4.55 \times 10^{-5} = 1.545 \times 10^{-4}$$

Soit le quantum est 1.545×10^{-4}

On observe que le quantum le plus élevé correspond au codec Opus, suivi du Speex 16 000, puis du Speex 8 000. Cependant, les valeurs obtenues sont étonnamment proches les unes des autres, ce qui peut sembler incohérent au regard des différences attendues entre ces codecs. Cette anomalie peut s'expliquer par le fait que le quantum mesuré correspond en réalité à celui de la dernière carte son traversée par le signal. Ainsi, cette étape pourrait fausser les résultats en uniformisant les quantum, indépendamment des particularités de chaque codec.

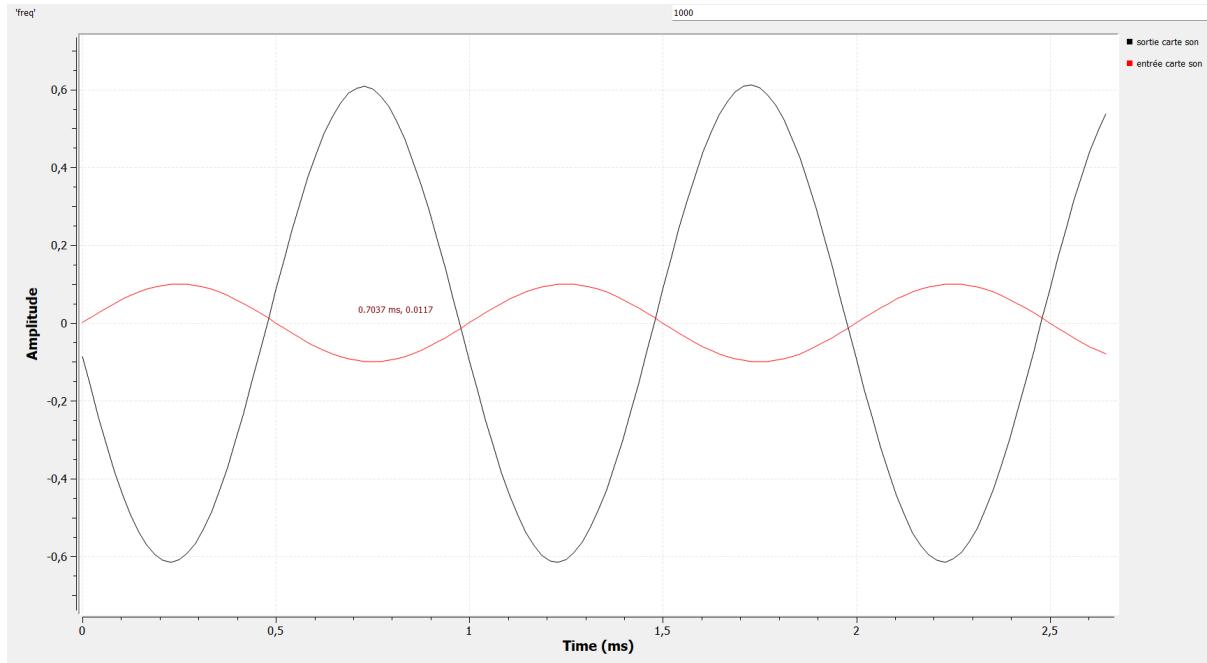
3.4 La fréquence d'échantillonnage

Pour déterminer la fréquence d'échantillonnage effective de chaque codec, nous avons utilisé le fichier Fe.grc sur GNU Radio Companion. L'approche a consisté à exécuter le signal en sortie du décodage et à ajuster progressivement la fréquence d'échantillonnage jusqu'à ce que le spectre du signal de sortie (VS) se stabilise. Cette fréquence stable correspond alors à la fréquence d'échantillonnage réelle utilisée par le codec concerné.

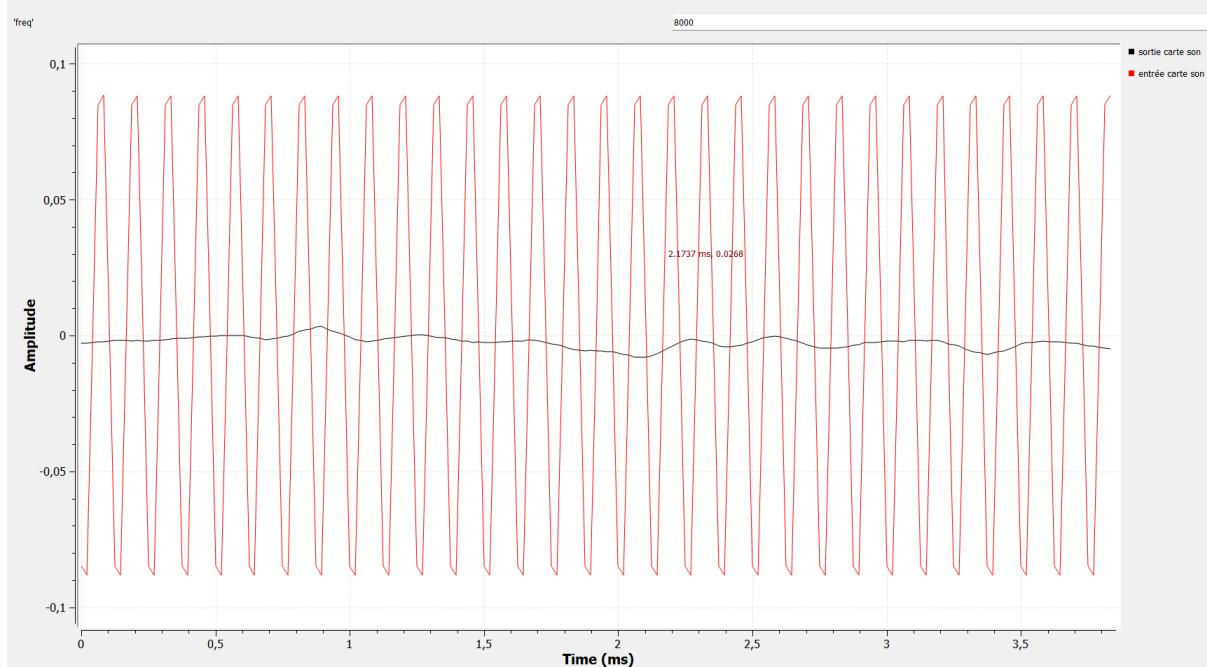
Dans le cas du codec Opus, nous n'avons pas réussi à stabiliser le spectre VS malgré l'augmentation progressive de la fréquence du signal d'entrée. En conséquence, il n'a pas été possible de déterminer expérimentalement la fréquence maximale f_{max} . Cela suggère que la fréquence d'échantillonnage f_e par le codec Opus est significativement plus élevée que celle des autres codecs testés dans les mêmes conditions.

Pour le codec speex 16000

On a obtenu ce graphique à une fréquence de 1000 hz



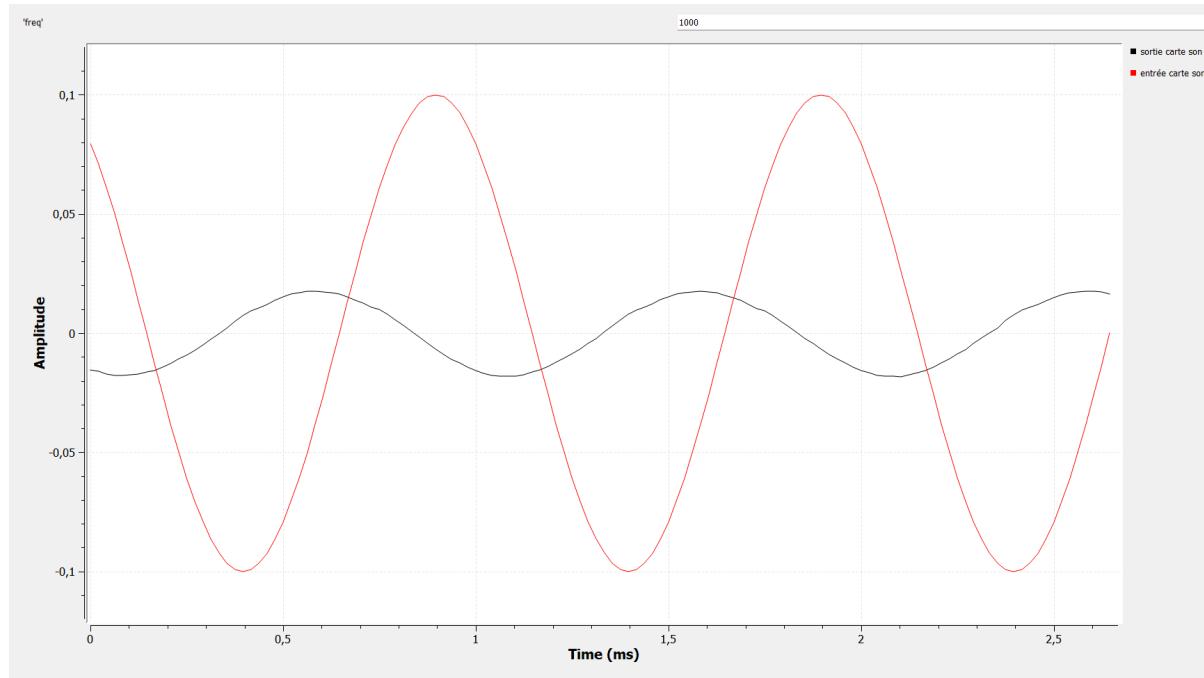
A une fréquence de 8000 hz



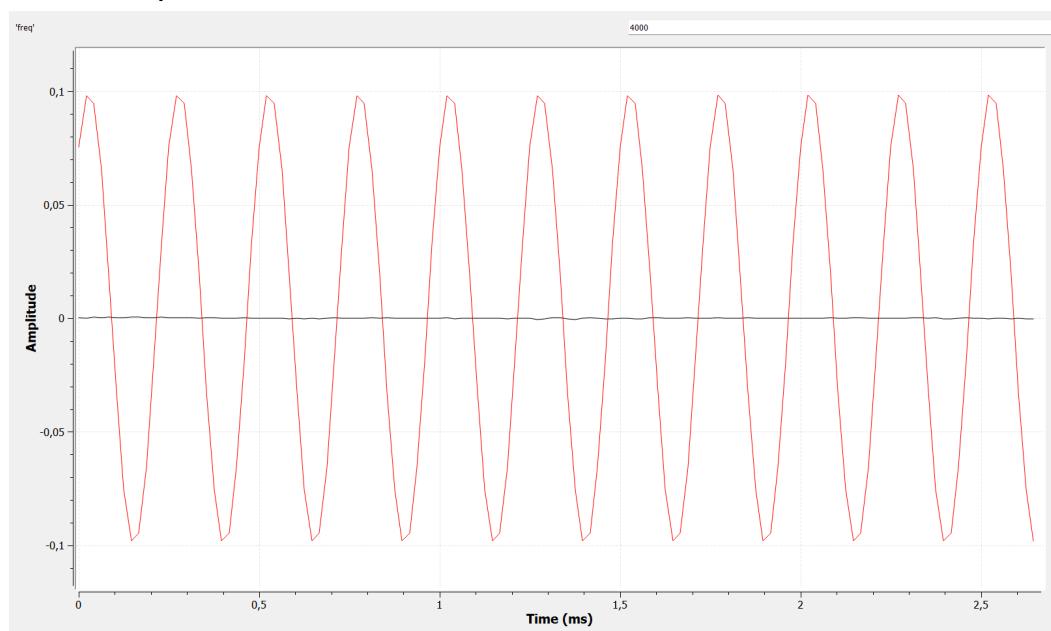
Ce qui veut dire qu'à 8000 hz on a atteint fmax or on sait que $f_e = 2f_{max}$
donc dans ce cas si on obtient $f_e = 16000$ hz

Codec speex 8000

On a obtenu ce graphique à une fréquence de 1000 hz



A une fréquence de 4000 hz



Soit f_{max} a été atteint à une fréquence de 4000 hz soit $f_e = 2 * 4000$ hz = 8000 hz

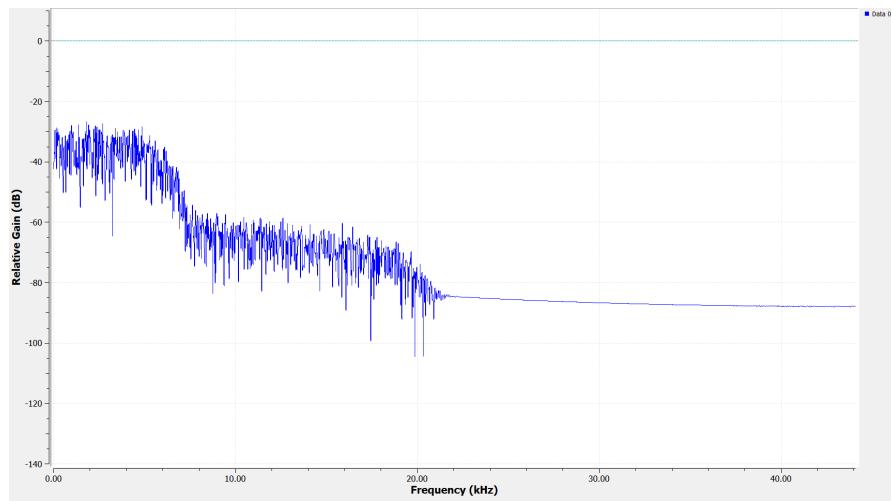
D'après nos observations, le codec présentant la fréquence d'échantillonnage la plus élevée est Opus, suivi de Speex 16000, puis de Speex 8000. Cette hiérarchie a été déduite à partir de l'analyse des signaux de sortie (VS) pour chaque codec, en identifiant la fréquence maximale transmise sans dégradation, ce qui permet d'estimer indirectement la fréquence d'échantillonnage selon le théorème de Nyquist.

3.5 La bande passante occupée par le signal

Dans le cadre de cette étude, nous avons cherché à analyser la bande passante effective des signaux traités par différents codecs audio, en lien avec leur fréquence d'échantillonnage. À l'aide du fichier Fe_spectre.grc sur GNU Radio Companion, et en utilisant comme source le fichier sae.wav, nous avons observé le spectre fréquentiel du signal en sortie du décodeur pour chaque codec testé.

L'objectif était de mesurer expérimentalement la bande passante réellement transmise et de vérifier s'il existe une relation cohérente entre cette bande passante et la fréquence d'échantillonnage définie par le codec.

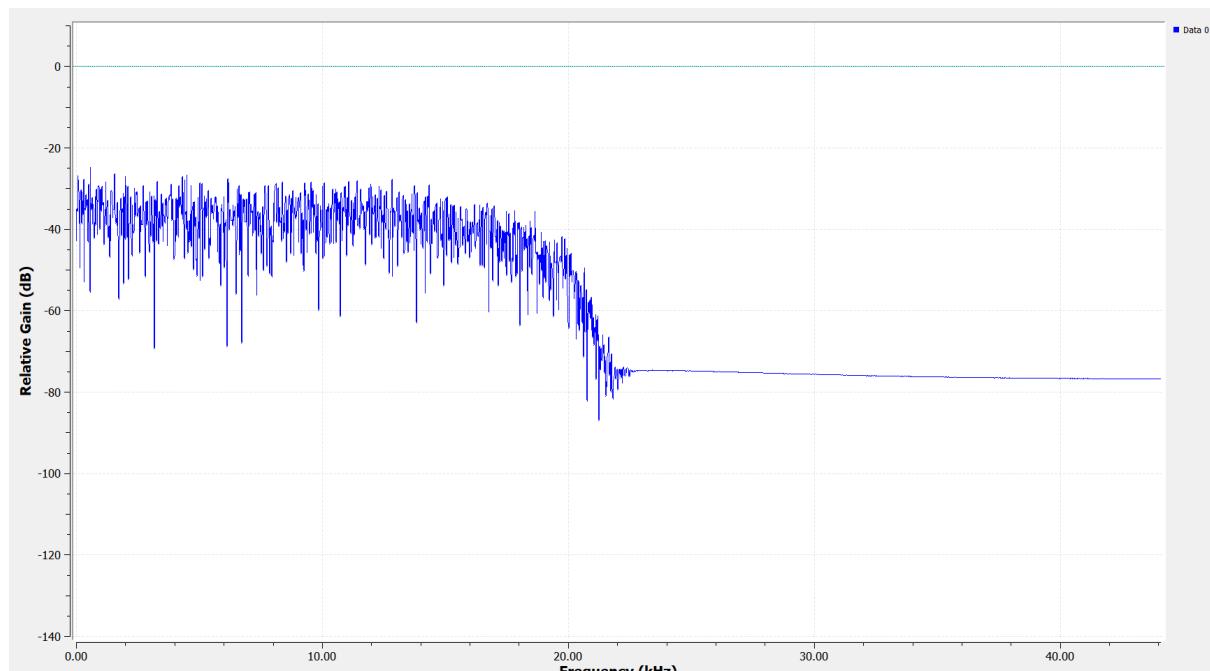
Codec speex 16000



Dans le cas du codec Speex 16000, le spectre montre une atténuation significative à partir d'environ 8 kHz, puis une quasi disparition autour de 16 kHz, ce qui indique que la bande passante utile est d'environ 8 kHz. Ce résultat est cohérent avec la fréquence d'échantillonnage associée au codec, soit 16 kHz, conformément au théorème de Nyquist :

$$f_{\max} = f_e / 2$$

Codec opus



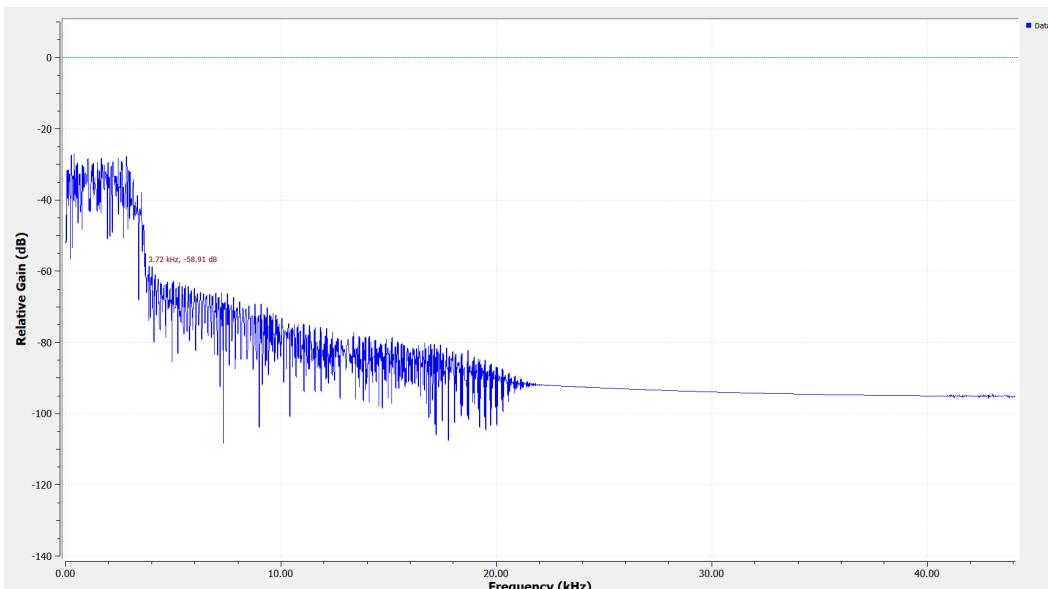
Sur le spectre ci-dessus obtenu à l'aide du fichier Fe_spectre.grc, on observe que le signal reste significatif jusqu'à environ 22 kHz, avec une atténuation marquée à partir de cette fréquence. Cela indique que la bande passante utile du signal est d'environ 22 kHz.

En appliquant le théorème de Nyquist, on en déduit que la fréquence d'échantillonnage associée au codec Opus est approximativement :

$$f_e = 2 * f_{max} = 2 * 22\ 000 \text{ hz} = 44\ 000 \text{ hz}$$

Ce résultat confirme que le codec Opus utilise une fréquence d'échantillonnage nettement plus élevée que les codecs Speex 8000 et Speex 16000.

Codec speex 8000



Le spectre ci-dessus, obtenu avec le fichier Fe_spectre.grc, montre que le signal décodé présente une atténuation marquée à partir d'environ 4 kHz. Le contenu fréquentiel devient ensuite très faible et fortement atténué, indiquant que la bande passante utile du signal est d'environ 4 kHz.

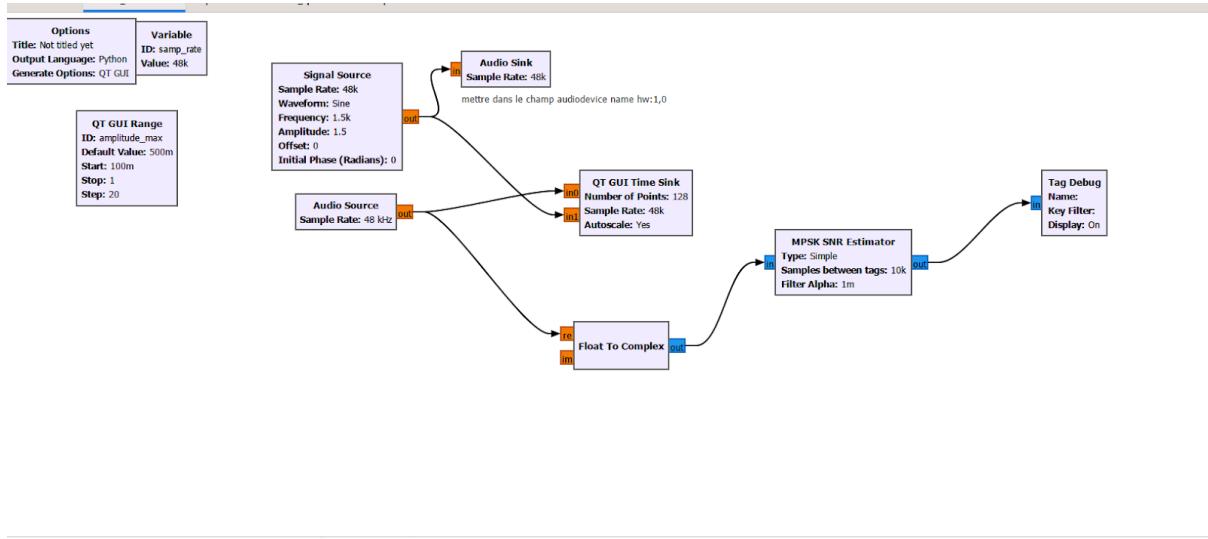
Selon le théorème de Nyquist, cela signifie que la fréquence d'échantillonnage utilisée par le codec est :

$$f_e = 2 * f_{max} = 2 * 4000 = 8000 \text{ hz}$$

Ce résultat est cohérent avec la fréquence d'échantillonnage annoncée pour ce codec.

3.6 Mesure du SNR

Dans cette partie, nous avons analyser les performances des modulations MPSK en mesurant le rapport signal/bruit (SNR) pour différents codecs. Pour ce faire, nous avons utilisé les fonctions MPSK, permettant la modulation du signal, SNR, qui quantifie la qualité du signal par rapport au bruit, et Tag Debug, un outil facilitant le suivi et l'analyse des performances du système. Nous avons implémenté ces fonction directement sur le fichier limite_tension.grc



Pour le codec opus trois configurations de débit ont été analysées : 40 kbps, 50 kbps et 55 kbps. Les résultats obtenus sont les suivants :

Opus (40 000)

```

Tag Debug:
Input Stream: 00
Offset: 280000 Source: mpsk_snr_est_cc2 Key: snr Value:
8.29706

```

Opus (50 000)

Tag Debug:

Input Stream: 00

Offset: 853000 Source: mpsk_snr_est_cc2 Key: snr Value:
8.90385Offset: 854000 Source: mpsk_snr_est_cc2 Key: snr Value:
8.90369

Opus (55 000)

Tag Debug:

Input Stream: 00

Offset: 1810000 Source: mpsk_snr_est_cc2 Key: snr Value:
8.98239

On observe une augmentation du SNR avec le débit, ce qui traduit une amélioration de la qualité du signal lorsque la compression est moins agressive. L'écart est particulièrement notable entre 40 kbps et 50 kbps, avec une hausse d'environ 0,6 dB, tandis que le gain entre 50 kbps et 55 kbps est plus faible (+0,08 dB).

Ces résultats confirment que l'augmentation du débit binaire améliore la fidélité du signal.

Pour le speex 16000

Tag Debug:

Input Stream: 00

Offset: 2080000 Source: mpsk_snr_est_cc2 Key: snr Value:
4.60094

À l'aide du bloc MPSK SNR Estimator, le SNR mesuré pour le codec Speex 16000 est de SNR = 4,60094 dB

Ce résultat est nettement plus faible que celui observé pour Speex 8000 (7,70 dB) et bien en dessous des performances du codec Opus (jusqu'à 8,98 dB).

Pour le speex 8000

Tag Debug:

Input Stream: 00

Offset: 8990000 Source: mpsk_snr_est_cc2 Key: snr Value:

0.761023

La mesure effectuée avec le bloc MPSK SNR Estimator pour le codec Speex 8000 donne un SNR de : SNR = 0,761023 dB

Le SNR obtenu pour le codec Speex 8000 est très faible : seulement 0,76 dB. Cela signifie que le signal de sortie est fortement dégradé, avec beaucoup de bruit par rapport au signal original.

Cela peut s'expliquer par le fait que Speex 8000 est un codec très compressé, conçu principalement pour la voix, et non pour des signaux complexes ou de haute qualité. Il est donc normal qu'il donne de mauvais résultats lorsqu'on lui demande de traiter des signaux plus riches que de la parole simple.

4. Conclusion

4.1 Les problèmes rencontrés ainsi que les solutions mise en place

Au cours de la réalisation de cette SAE, plusieurs difficultés techniques ont été rencontrées, ce qui a parfois ralenti l'avancement du projet. Voici les principaux problèmes identifiés :

- Bien que les appels aient été fonctionnels et que les microphones soient correctement détectés sur les deux machines, aucun son n'était transmis d'un ordinateur à l'autre. Après plusieurs vérifications, nous avons identifié que les cartes son n'étaient pas sélectionnées correctement dans les paramètres audio. Une fois ce point corrigé, la communication audio a pu être rétablie.
- Un autre problème majeur concernait la configuration de Linphone. En effet, nous avons constaté que les paramètres de l'application se réinitialisaient de manière aléatoire à des intervalles de temps irréguliers. Cela nous a obligé à reconfigurer régulièrement les comptes SIP et les préférences, ce qui a fortement ralenti notre progression et perturbé certaines phases de test.
- Lors de la partie consacrée à la mesure de la latence, nous avons observé que la fréquence d'émission était trop faible pour que Linphone puisse fonctionner correctement. Cela provoquait une instabilité du signal. Pour résoudre ce problème, nous avons augmenté la fréquence d'émission, ce qui a permis d'obtenir des résultats plus fiables et de poursuivre les mesures dans de meilleures conditions.

4.2 Conclusion

À l'issue des différentes mesures réalisées dans le cadre de cette étude, il apparaît clairement que le codec Opus se démarque des codecs Speex 8000 et Speex 16000 sur l'ensemble des critères techniques évalués : rapport signal/bruit (SNR), bande passante, fréquence d'échantillonnage et quantum.

Tout d'abord, en ce qui concerne le SNR, Opus affiche les meilleures performances avec une valeur maximale de 8,98 dB à un débit de 55 kbps. En comparaison, Speex 16000 atteint seulement 4,60 dB, tandis que Speex 8000 présente un SNR très faible de 0,76 dB. Cela indique qu'Opus restitue le signal avec beaucoup moins de bruit, et donc une qualité bien supérieure.

Sur le plan de la bande passante, Opus permet de transmettre efficacement des fréquences jusqu'à environ 20 kHz, contre 8 kHz pour Speex 16000 et 4 kHz pour Speex 8000. Cette large bande passante est directement liée à une fréquence d'échantillonnage plus élevée (estimée à 40 kHz ou plus pour Opus), qui permet de conserver une plus grande richesse fréquentielle du signal.

Concernant le quantum, qui reflète la finesse de la quantification, les résultats montrent qu'il est plus faible avec Opus, ce qui signifie une meilleure précision dans la représentation numérique du signal. À l'inverse, les codecs Speex présentent un quantum plus élevé, traduisant une quantification plus grossière et donc une perte de fidélité.

De plus, grâce à cette SAÉ, nous avons acquis des compétences dans l'analyse et la manipulation de signaux numériques. Nous avons également appris à utiliser de nouveaux outils tels que FreePBX et Linphone, ainsi que l'importance des différents codecs. Nous avons identifié les facteurs susceptibles d'altérer leur qualité et appris à les analyser.

5. Annexe

5.1 Auto-évaluation

Etape	Non-fait	En cours (date/heure début)	Terminé (date/heure fin)
Installation de FreePBX		31/03/2025 8h	31/03/2025 9h
Configuration FreePBX		31/03/2025 9h	31/03/2025 10h
Configuration de Linphone		31/03/2025 10h	31/03/2025 15h
Test de fonctionnement		31/03/2025 15h	01/04/2025 15h
Mesure de BP(plusieurs config) et spectre et Fe		01/04/2025 15h	01/04/2025 18h
Mesure de SNR (plusieurs config, n, débit)		02/04/2025 8h	02/04/2025 10h
Mesure de Latence/plage/quantum		31/04/2025 16h	01/04/2025 13h
Comparaison audio		02/04/2025 10h	02/04/2025 12h
Autre ?			