Débruitage de Vuvuzela

V. Choqueuse et C. Ansquer

31 mars 2019

1 Introduction

1.1 Contexte / Problématique

FIGURE 1 – Le Vuvuzela

Lors de la coupe du Monde de Football 2010, les téléspectateurs du monde entier ont pu découvrir un instrument de musique d'Afrique du Sud : le Vuvuzela. Passé la phase d'étonnement, le son continu généré par cet instrument est rapidement devenu gênant lors de la diffusion des matchs (`https://www.youtube.com/watch?v=bKCIFXqhLzo`). Dans ce projet, nous allons mettre en place un filtre "anti-vuvuzela" pour atténuer les sonorités de cet instrument.

1.2 Méthodologie

Le repo github contient un enregistrement sonore nommé `vuvuzela.wav`. Cet enregistrement a été réalisé lors d'un match de la coupe du monde 2010. Pour supprimer les sonorités de Vuvuzela, nous allons procéder de la manière suivante :

1. Analyse du contenu spectral du fichier `vuvuzela.wav`.
2. Analyse de différents "rejecteurs" avec Python :
   - un filtre RLC,
   - un filtre Twin T passif,
   - un filtre Twin T actif.
3. Etude expérimentale d'un des deux filtres passifs :
   - Détermination expérimentale de la résistance interne d'une bobine et réglage de la résistance de compensation à rajouter dans la branche capacitive du RLC.
   - Détermination expérimentale du coefficient d'amortissement du twin T passif par mesure de la bande rejetée à partir du déphasage entre les signaux d'entrée et de sortie.

4. Implémentation des différents Twin T actif nécessaires pour rejeter les différentes harmoniques du son à éliminer. Preuve sur LTspice.
5. Répartition des différentes cellules à câbler entre les différents groupes.
6. Câblage de la cascade et test de l'ensemble sur le signal audio fourni.

Pour chacune de ces étapes, vous devez rédiger un notebook Jupyter expliquant votre démarche et présentant vos différents résultats.

2 Déroulement du Projet

2.1 Analyse Spectrale

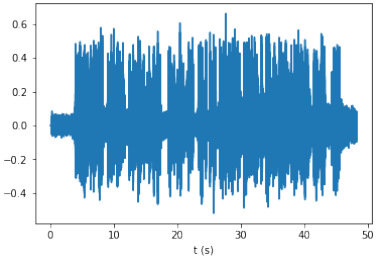

FIGURE 2 – Enregistrement sonore contenant des sonorités de Vuvuzela et des commentaires sportifs.

La figure 2 donne la représentation temporelle de l'enregistrement sonore. Cet enregistrement contient des parties contenant uniquement les sonorités de Vuvuzela (entre 0 et 2s par exemple) et d'autres parties contenant les sonorités de Vuvuzela ainsi que des commentaires de journalistes sportifs.

Mathématiquement, les sonorités de Vuvuzela peuvent être modélisés par une somme de plusieurs sinusoïdes de fréquence f_k c-a-d

$$x(t) = \sum_{k=1}^{L} a_k \sin(2\pi f_k t + \varphi_k) \tag{1}$$

Dans un premier temps, il est nécessaire de déterminer les fréquences $f_1, \cdots, f_L$ en réalisant une analyse spectrale (c-a-d fréquentielle) de l'enregistrement sonore. L'analyse pourra être réalisée en utilisant Python (ou Scilab). En Python, cette partie nécessitera d'utiliser les fonctionnalités suivantes :

- Importation de fichiers sonores (Documentation : `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.io.wavfile.read.html`)
- Calcul du spectre (Documentation : `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.periodogram.html`)

Question 1. *En utilisant `matplotlib`, représentez le contenu temporel puis fréquentiel de l'enregistrement sonore.*

Question 2. *Isolez une portion temporelle de l'enregistrement sonore ne contenant que des sonorités de Vuvuzela (pas de commentaires sportifs). Représentez alors le contenu fréquentiel de cette portion.*

Question 3. *Identifiez les composantes fréquentielles du son de Vuvuzela c-a-d les valeurs de f_k ($k = 1, \cdots, L$). Retrouvez alors l'unique note (!!!) de la gamme tempérée qu'est capable de générer le Vuvuzela.*

Question 4 (Bonus)**.** *En pratique, les signaux réels sont rarement parfaitement stationnaires car le contenu fréquentiel peut varier dans le temps. C'est le cas notamment des signaux de parole. Pour représentez le contenu d'un signal non-stationnaire, il est possible de recourir à des représentations dites temps-fréquences comme le spectrogramme. En utilisant la fonction* `spectrogram` *de Scipy (Documentation : https ://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html), représentez le contenu temps-fréquence du signal audio.*

2.2 Filtre rejecteur RLC

2.2.1 Analyse du filtre (Python)

Dans un premier temps, nous allons considérer le rejecteur RLC suivant :

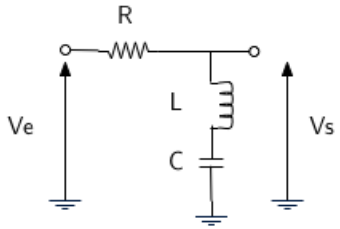

FIGURE 3 – Rejecteur RLC

Question 5. *Déterminez la fonction de transfert $T(p)$ de ce rejecteur.*

Question 6. *Déterminez l'expression du coefficient d'amortissement m et de la pulsation propre ω_0 en fonction des différents composants du filtre.*

Question 7. *A partir d'une analyse théorique, déterminez les pôles et les zeros de ce filtre. Montrez ensuite que ce filtre permet de supprimer complètement une composante fréquentielle f particulière (expression de f à déterminer). Déterminez théoriquement l'expression de la largeur de la bande rejetée à $-3dB$, puis l'expression du gain en basse-et haute fréquence de ce filtre.*

Question 8. *En utilisant Python, réalisez une étude complète de ce filtre (réponse indicielle, réponse fréquentielle).*

2.2.2 Analyse de la mise en cascade (Python/ LT-spice)

Pour supprimer les différentes composantes fréquentielles du Vuvuzela, nous allons cascader plusieurs rejecteurs RLC en série.

Question 9. *Sous l'hypothèse que $R = R_1 = \cdots = R_L$ et que le coefficient d'amortissement m est égal pour chaque étage RLC, réalisez un script python pour déterminer automatiquement les valeurs des composants L_k et C_k. Ces valeurs seront déterminées de manière à filtrer les différentes composantes fréquentielles f_k (voir question 3).*

Question 10. *Câblez le filtre cascadé sous LT-spice.*

Question 11. *Analysez la réponse fréquentielle du filtre cascadé.*

Question 12. *En utilisant les fonctionnalités audio de LT-spice, importez le fichier sonore* `vuvuzela.wav` *puis sauvegardez le signal de sortie dans un fichier sonore au format wav. Modifiez votre valeur de m de manière à obtenir un résultat sonore acceptable.*

Question 13.
- *En renseignant le paramètre série résistance de la bobine, expliquez la problématique liée à ce type de filtre.*
- *Effectuer une recherche pour trouver les schémas permettant de compenser cette résistance de la branche inductive.*

2.3 Filtre rejecteur *Twin-T* passif

2.3.1 Analyse du filtre (Python)

Pour éviter d'utiliser plusieurs bobines, nous allons maintenant considérer un filtre rejecteur Twin-T passif. Ce filtre est décrit dans la figure suivante :

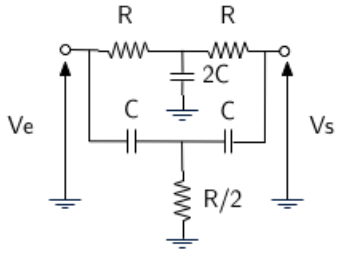

FIGURE 4 – Filtre rejecteur Twin-T passif

Question 14. *Déterminez la fonction de transfert $T(p)$ de ce filtre.*

Question 15. *Similaire aux questions 6-8.*

Question 16. *Justifiez la valeur limite de m et critiquez le choix de ce filtre.*

2.4 Filtre rejecteur Twin-T actif

Le filtre précédent ne permet pas d'obtenir des résultats sonores satisfaisants. Pour améliorer le résultat sonore, il est possible d'utiliser un filtre Twin-T actif. Contrairement au filtre Twin-T passif, le filtre Twin-T actif permet le contrôle du coefficient d'amortissement m. Le tutoriel MT-225 d'Analog Devices (voir Gitlab) décrit la structure de ce filtre et donne plusieurs recommendations pour choisir les composants.

Question 17. *Similaire aux questions 9-12.*

Question 18. *Utiliser cette structure pour implémenter sous LTSpice la cascade nécessaires à filtrer les harmoniques déterminer à la question 3*