Comparaison de la Transformée de Fourier et de la Constant Q Transform pour un signal audio

1 Introduction

On se propose dans ce mini-projet d'étudier une représentation fréquentielle du signal autre que la transformée de Fourier conventionnelle : la transformée Q constante ou constant Q transform, CQT. Proche de la manière dont fonctionne le système auditif humain, cette représentation est bien adaptée aux signaux de musique. Après avoir compris les avantages d'une telle représentation, elle sera implantée sous matlab. Elle sera ensuite comparée à la TFD traditionnelle en uilisant divers exemples musicaux.

La constant Q transform est très proche de la transformée de Fourier. Cependant, contrairement à celle-ci, ses composantes fréquentielles sont séparées de manière géométrique. Elle est donc plus adaptée aux signaux de musique. En effet, une note de musique jouée par un instrument produit un son fondamental ainsi que plusieurs harmoniques dont la fréquence est multiple de la fondamentale. Une propriété qui en découle est que la position des harmoniques les unes par rapport aux autres est indépendante de la fréquence fondamentale, si elles sont tracées en log-fréquence. Ceci est illustré sur la Figure 1 où on représente le spectre hypothétique d'un son de fréquence fondamentale f. La distance entre les deux premiéres harmoniques est $log(2)$, entre la seconde et la troisiéme est $log(\frac{3}{2})$ etc. La position absolue des harmoniques sur l'échelle des fréquences dépend de la fréquence fondamentale mais la position relative des différentes harmoniques d'une note est constante et forme un "pattern".

Une autre propriété de la constant Q transform est que la résolution temporelle augmente avec la fréquence. Ainsi, une grande taille de fenêtre d'analyse est utilisée dans les basses fréquences et, plus la fréquence augmente, plus la taille de fenêtre diminue.

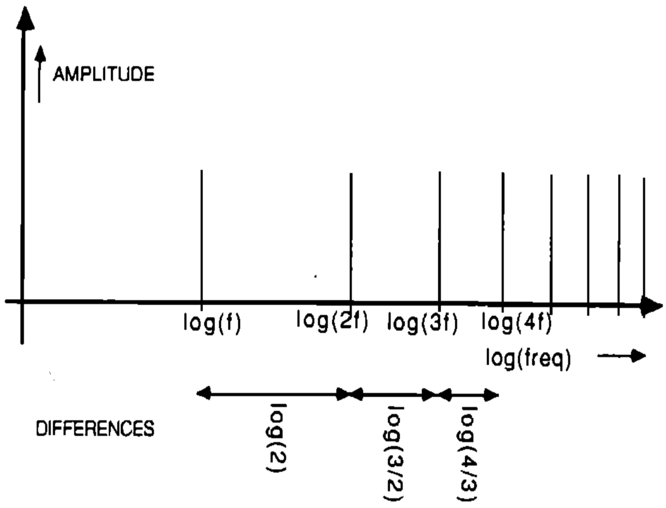

FIGURE 1 – Pattern de la transformée de Fourier d'une note de musique tracé selon une échelle des fréquences logarithmique.

2 Calcul de la CQT

La constant Q transform d'un signal temporel échantillonné x(n) peut être directement calculée par :

$$X^{cq}(k_{cq}) = \sum_{n=0}^{N_{k_{cq}}-1} w(n, k_{cq}) x(n) e^{-j2\pi f_{k_{cq}} n} \tag{1}$$

où $X^{cq}(k_{cq})$ est la k^{eme} composante de la constant Q transform. La fenêtre d'analyse $w(n, k_{cq})$ est fonction de la fréquence ("bin" k_{cq}). Les fréquences correspondant aux bins de la CQT sont géométriquement espacées, en relation avec la gamme tempérée. Ainsi si on note f_{min} la fréquence minimale d'analyse, les autres fréquences peuvent être déduites par :

$$f_{k_{cq}} = (2^{1/12})^{k_{cq}} f_{min} \tag{2}$$

Le nom constant Q transform vient du fait que l'on désire garder un rapport $Q = \frac{f_k}{\Delta f_k}$ constant où $\Delta f_k = f_{k+1} - f_k$. Dans le cas de la gamme tempérée où deux notes adjacentes forment un demi-ton, on a :

$$Q = 1/(2^{1/12} - 1)\#17 \tag{3}$$

Ici, les composantes fréquentielles de la constant-Q transform correspondent aux demi-tons de la gamme tempérée. Elle est ainsi bien adaptée aux sons musicaux.

2.1 Signal à analyser

Dans le travail qui va suivre, la fonction matlab principale sera nommée **construction_CQT**.

Question 1 : Dans **construction_CQT**, convertir le signal audio stéréophonique échantillonné à $11025Hz$ en signal monophonique en prenant la moyenne sur les deux canaux.

2.2 Conversion des fréquences f_k du spectre en notes midi

Les théoriciens de la musique se réfèrent en général aux différentes classes de hauteur en utilisant des nombres réels (on appelle cela échelle midi).

Question 2 : Ecrire une fonction matlab **freq2midi** qui permet de transformer la fréquence fondamentale f d'un son en un nombre réel p selon l'équation suivante (conversion en échelle midi) :

$$p = 69 + 12log_2 \frac{f}{440} \tag{4}$$

Le le DO du milieu d'un piano correspond à la note midi 60.

Question 3 : Ecrire une fonction matlab **midi2freq** qui permet d'effectuer la transformation inverse.

Ces fonctions seront utilisées dans la suite, afin de s'assurer que l'on utilise des fréquences qui correspondent à des notes de la gamme chromatique de la musique occidentale.

2.3 Fréquences minimales et maximales d'analyse

Question 3 : On considère les fréquences d'analyse entre $f_1 = 50Hz$ et $f_2 = 4000Hz$.À l'aide des fonctions précédentes, déterminer les plus proches fréquences f_{min} et f_{max} correspondant à des notes de musique. Calculer le nombre de notes nb_freq comprises dans l'intervalle $[f_{min} - f_{max}]$.

2.4 Calcul de la CQT

On écrit :

$$X^{cq}(k_{cq}) = \sum_{n=0}^{N_{k_{cq}}-1} x(n)K^*(n, k_{cq}) \tag{5}$$

avec

$$K^*(n, k_{cq}) = w(n, k_{cq})e^{-j2\pi f_{k_{cq}} n} \tag{6}$$

où les $K(n, k_{cq})$ sont appelés noyaux temporels de la transformation. Ici, on utilise pour $w(n, k_{cq})$ une fenêtre de Hamming. Notez que la longueur de cette fenêtre dépend du "bin" de fréquence considéré.

2.4.1 Génération des noyaux

Comme on l'a vu précedement, la fenêtre d'analyse $w(n, k_{cq})$ est fonction de la fréquence k_{cq}. La taille des fenêtres est donnée par :

$$N_{k_{cq}} = Q * \frac{fs}{f_{k_{cq}}} \tag{7}$$

Question 4 : Caculer la taille des fenêtre $w(n, k_{cq})$ en fonction de la fréquence. On stockera ces informations dans un vecteur noté *windsize_v*. Calculer la taille maximale des fenêtres d'analyse (elle correspond à la plus basse fréquence considérée). On la note *windsize_max*.

Question 5 : Écrire une fonction **generer_noyaux** qui permet d'obtenir les noyaux qui sont utilisés pour le calcul de la CQT. On calculera deux matrices *kerncos_m* et *kernsin_m* qui correspondent respectivement aux parties réelles et imaginaires de $K^*(n, k_{cq})$. Elles sont de taille nb_freq x *windsize_max*. Pour chaque "bin" k de fréquence considéré, on a $kerncos_m(k, n) = 0 \; pour \; n < length(windsize\_v(k))$.

2.4.2 Calcul de la CQT

Question 6 : Afin de pouvoir analyser le signal au cours du temps, le découper en "trames" se recouvrant partiellement. On prendra des fenêtres de taille *windsize_max* et un pas d'avancement de $\frac{windsize\_max}{8}$.

Pour chaque trame (vecteur colonne $x$de taille *windsize_max*), calculer sa transormée Q constante (vecteur ligne X^{cq} de taille nb_freq) de la faqn suivante décrite ci-dessous (On calculera d'abord séparément les parties réelles et imaginaires).

$$1) \qquad reCQT_v = kerncos_m * x \tag{8}$$
$$2) \qquad imCQT_v = kernsin_m * x \tag{9}$$
$$3) \qquad X^{cq} = \sqrt{(reCQT_v.^2) + (imCQT_v.^2)} \tag{10}$$

3 Partie Audio

Question 8a : Écrire une fonction qui permet d'afficher la CQT d'un signal en fonction du logarithme des fréquences et en fonction des notes MIDI (utiliser la fonction waterfall de MATLAB).

Question 8b : Écrire une fonction qui permet d'afficher la CQT à la manière d'un spectrogram.

Question 9 : Représenter la CQT pour différentes notes jouées par un violoncelle. Reconnaît-on les notes jouées ? On voit que des pics correspondant à d'autres notes sont présente. En regardant quelles sont les harmoniques des notes émises, essayer d'expliquer ce qui est observé.

Question 10 : Comparer avec la transformée de Fourier usuelle en utilisant diverses tailles de fenêtres. Commenter.

Question 11 : Mêmes questions avec une clarinette. Comparer avec le violoncelle. Que peut-on dire au sujet des harmoniques paires ?

Question 12 : Utiliser des sons plus complexes composés de plusieurs notes à des octaves différentes. Comparer les deux représentations et commenter les résultats.

Question 13 : Même pour des sons percussifs, des sons composés (par exemple violoncelle + persussions).