A CONSTANT-Q SPECTRAL TRANSFORMATION WITH IMPROVED FREQUENCY RESPONSE

Danillo B. Graziosi, Cristiano N. dos Santos, Sergio L. Netto, and Luiz W. P. Biscainho

PEE/COPPE & DEL/POLI, UFRJ
POBox 68504, Rio de Janeiro, RJ, 21945-970, BRAZIL
{danillo, csantos, sergioln, wagner}@lps.ufrj.br

ABSTRACT

This paper introduces a new transform intended for audio processing. The proposed transform exhibits two interesting features in the frequency domain, namely: a constant-Q characteristic and a steep response for each output channel. Constant Q implies that the spectral description of the transformed signal is performed along a log-like scale, as opposed to the linear scale of standard transforms, such as the discrete Fourier transform (DFT). The consequent variable resolution makes the new transform especially suited for the analysis of musical audio signals. The improved frequency response, as compared also to the DFT, is achieved by its implementation as a bank of very selective filters based on the frequency-response masking (FRM) approach. Such selectivity results in higher isolation between adjacent channels of the overall transform. Application of the new transform to the analysis of musical signals is illustrated through a computer experiment.

I. INTRODUCTION

The discrete Fourier transform (DFT) is a powerful tool for signal analysis, constituting a true mathematical bridge between time and frequency domains [1]. The DFT-based analysis, however, can be shown to present significant interference between the outputs of adjacent channels. In addition, considering the way the occidental musical scales were historically built, one can deduce that for musical audio signals, a logarithmic frequency representation would be more natural than the linear-frequency scale inherent to the DFT. This paper then proposes a new transform which attempts to overcome these two drawbacks related to the DFT. The new transform, the so-called constant-Q fast filter bank (CQFFB), is generated by the combination of the constant-Q transform (CQT) introduced in [2], [3] with the improved response characteristic of the fast filter bank (FFB), based on the frequency-response masking (FRM) approach and presented in [4], [5], [6].

The remaining of this paper is organized as follows: In Section II a brief description of the CQT is provided, focusing on the positive aspects of having a log-like frequency scale. In Section III, the sliding fast Fourier transform (sFFT) is interpreted under a filter-bank perspective for implementing the DFT. The FFB is then described in Section IV as a generalization of the sFFT, whose prototype filters can be replaced by more selective ones to achieve an improved frequency response. Section V presents the novel CQFFB, combining the positive issues of the CQT (log-like frequency scale) and the FFB (selective frequency response). Finally, computer experiments are included in Section VI to illustrate the capability of the CQFFB to analyze musical audio signals. Computational complexity issues will be addressed in a future paper.

II. THE CONSTANT-Q SPECTRUM TRANSFORMATION

When using the DFT to analyze musical audio signals, the resulting linear frequency scale yields a badly balanced signal description, since it concentrates too much information in the high-frequency region.

In [2], a constant-Q method, which allows the description of the frequency domain in a logarithmic scale, is presented as a tool for the analysis of musical audio signals. A strong motivation behind a constant-Q transform (CQT) is to allow harmonic frequencies to be represented in equal intervals in the transform domain. In that manner, any fundamental frequency together with its associate harmonics define a linear pattern which can be easily identified.

For the DFT, the frequency resolution $(\delta f)_{\mathrm{DFT}}$ is a constant value, given by the sampling frequency f_s divided by the total number of samples N being transformed:

$$(\delta f)_{\mathrm{DFT}} = \frac{f_s}{N}. \tag{1}$$

The N frequencies directly represented are

$$f_k = \frac{k}{N} f_s, \tag{2}$$

for $k = 0, 1, \ldots, (N-1)$.

The standard CQT decomposes the signal into components given by the following frequencies:

$$f_k = \left(2^{1/12}\right)^{\alpha k} f_{\min}, \tag{3}$$

for $k = 0, 1, \ldots, (N-1)$, where α defines the frequency resolution in fractions of a semitone. Then, $\alpha = \frac{1}{2}$ corresponds to a quarter-tone resolution, which suffices for many

0-7803-8251-X/04/$17.00 ©2004 IEEE

applications. Such resolution corresponds to a selectivity factor

$$Q = \frac{f_k}{(\delta f)_{\mathrm{CQT}}} = \frac{f_k}{\left(2^{1/24} - 1\right) f_k} \approx \frac{1}{0.0293} \approx 34. \qquad (4)$$

To achieve a constant Q, one should use a variable number of points

$$N_k = \frac{f_s}{(\delta f)_{\mathrm{CQT}}} = \frac{f_s Q}{f_k} \qquad (5)$$

to determine each transform sample. Noticing that now

$$f_k = \frac{Q}{N_k} f_s, \qquad (6)$$

we get that for the CQT the index k equals the selectivity factor Q. We then define the CQT of a signal $x(n)$ based on the definition of the DFT, but with the number of points given by equation (5), that is,

$$X_{\mathrm{CQT}}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} \mathcal{W}(n,k) x(n) e^{-j \frac{2\pi}{N_k} Qn} \qquad (7)$$

for $k = 0, 1, \ldots, (N-1)$, where $\mathcal{W}(n,k)$ is a windowing function used to reduce blocking effects. A Table listing the characteristics of a 156-channel CQT for a sampling rate of $f_s = 32$ kHz can be found in [2].

III. THE SLIDING FFT

The N-point sliding FFT (sFFT) can be seen as the operation [4]

$$\begin{aligned} X(k) &= \sum_{i=0}^{N-1} x(n+i) W_N^{ki} \\ &= \left(\sum_{i=0}^{N-1} q^{-i} W_N^{ki} \right) \{x(n)\}, \end{aligned} \qquad (8)$$

where q is the delay operator, such that $q^i\{x(n)\} = x(n - i)$ and $W_N = e^{-j\frac{2\pi}{N}}$. Hence, in the z domain, the sFFT operator can be expressed as

$$\mathrm{sFFT}(z) = \sum_{i=0}^{N-1} z^i W_N^{ki} = \prod_{i=0}^{L-1} \left[1 + \left(z W_N^k \right)^{2^i} \right], \qquad (9)$$

where $L = log_2 N$. The FFB, to be reviewed in the next section, is a generalization of the sFFT which results from describing the sFFT(z) as indicated in equation (9).

Example 1: Using equation (9), the transfer function of channel 34 (linked to the quarter-tone CQT) of a 256-point sFFT is given by

$$\begin{aligned} H_{34}(z) &= \prod_{i=0}^{7} \left[1 + \left(z W_{256}^{34} \right)^{2^i} \right] \\ &= G_a^{0,34} G_a^{1,68} G_a^{2,136} G_a^{3,16} G_a^{4,32} G_a^{5,64} G_a^{6,128} G_a^{7,0}, \end{aligned} \qquad (10)$$

where

$$\begin{cases} G_a^{0,34} &= 1 + z^1 W_{256}^{34}; \quad G_a^{1,68} = 1 + z^2 W_{256}^{68} \\ G_a^{2,136} &= 1 + z^4 W_{256}^{136}; \quad G_a^{3,16} = 1 + z^8 W_{256}^{16} \\ G_a^{4,32} &= 1 + z^{16} W_{256}^{32}; \quad G_a^{5,64} = 1 + z^{32} W_{256}^{64} \\ G_a^{6,128} &= 1 + z^{64} W_{256}^{128}; \quad G_a^{7,0} = 1 + z^{128} W_{256}^{0}, \end{cases} \qquad (11)$$

such that $G_a^{i,j} = (1 + z^{2^i} W_N^j)$, with $j = [(34 \times 2^i) \bmod N]$. The corresponding magnitude response of channel 34 is depicted in Figure 1, where one can readily see that the first sidelobes are 13 dB below the channel passband. In order to normalize the sFFT response to the 0 dB level, the channel transfer function should be scaled by a factor of $c = 20 \log_{10} N = 48.2$ dB.

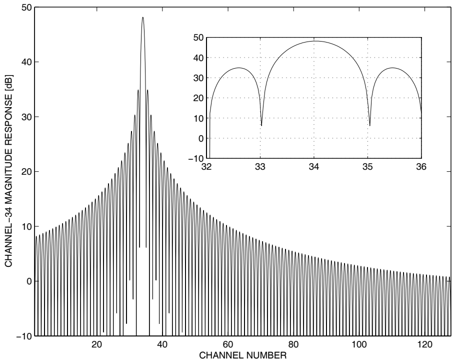


Fig. 1. Example 1: Magnitude response of channel 34 of a 256-point sFFT.

■

IV. THE FAST FILTER BANK

In equation (9), following the description in [4], the factor

$$G_a^{0,k}(z) = 1 + z W_N^k \qquad (12)$$

can be seen as the kernel filter of the sFFT(z), since all other factors can be derived from it following a frequency scaling operation

$$z W_N^k \rightarrow (z W_N^k)^i. \qquad (13)$$

A general fast filter bank (FFB) can be then generated from the sFFT by substituting the first-order kernel filter $G_a^{0,k}(z)$ by any other filter. Of course, higher order filters can yield improved frequency responses [4].

Example 2: Figure 2 depicts the magnitude response of channel 34 of a 256-point FFB using the filters given in [4].

Another interpretation for the FFB, based on the frequency-response masking (FRM) approach [7], is given in [6], where alternative FFB subfilters are also provided. Applications of the FFB include a programmable filter [6] and automatic music transcription [8].

■

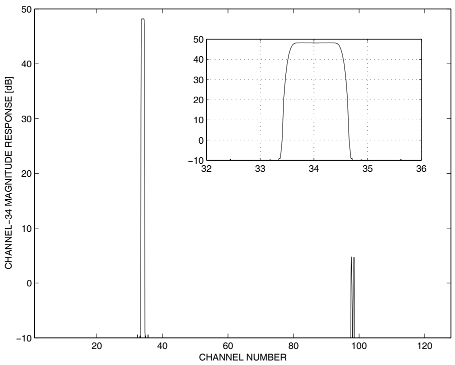


Fig. 2. Example 2: Magnitude response of channel 34 of a 256-point FFB.

V. AN IMPROVED CONSTANT-Q TRANSFORMATION

The so-called constant-Q fast filter bank (CQFFB) was devised as a blend of the CQT and FFB techniques. In that manner, we combine the constant-Q behavior of the CQT with the improved frequency response of the FFB. To achieve such symbiotic combination, however, we must first overcome a discrepancy between these two techniques. In fact, while the CQT calls for a variable number of time-domain samples to compute each frequency-domain sample, as given in equation (5), the FFB requires the underlying DFT to be computed from a power-of-two number of time-domain samples. The adopted strategy was to modify the CQT in the following way: instead of varying the number of input samples employed to compute each output sample in order to guarantee constant Q, we keep the number of input samples fixed as a power-of-two integer and change accordingly the sampling frequency, which becomes

$$f_s(k) = f_k \frac{N}{Q}, \tag{14}$$

with f_k defined by equation (3), as before. After such modification, considering the quarter-tone resolution (Q=34), the Nyquist criterion requires

$$2f_k < f_s(k) \;\Rightarrow\; N > 68 \;\Rightarrow\; N = 128, 256, \ldots. \tag{15}$$

Naturally, resampling of the input signal may need to include some kind of anti-aliasing filtering. In a preliminary version of the CQFFB we used the MATLAB®command `resample`.

Table I lists the characteristics of the 156-channel CQT, which uses a fixed sampling rate f_s = 44.1 kHz, and CQFFB, which uses a fixed number of N = 256 samples in each channel.

TABLE I
CHARACTERISTICS OF THE 156-CHANNEL CQT (WITH $f_s = 44.1$ KHZ) AND CQFFB (WITH $N = 256$ SAMPLES), BOTH WITH A SELECTIVITY FACTOR $Q = 34$.

Channel	Midinote	f_k	N_k	$f_s(k)$	Time
			[CQT]	[CQFFB]	
		(Hz)	(samples)	(Hz)	(ms)
0	53	175	8568	1318	194.3
6	56	208	7209	1566	163.5
12	59	247	6070	1860	137.6
18	62	294	5100	2214	115.6
24	65	349	4296	2628	97.4
30	68	415	3613	3125	81.9
36	71	494	3035	3720	68.8
42	74	587	2554	4420	57.9
48	77	699	2145	5263	48.6
54	80	831	1804	6257	40.9
60	83	988	1518	7439	34.4
66	86	1175	1276	8847	28.9
72	89	1398	1073	10526	24.3
78	92	1664	901	12529	20.4
84	95	1978	758	14893	17.2
90	98	2350	638	17649	14.5
96	101	2797	536	21060	12.2
102	104	3327	451	25050	10.2
108	107	3956	379	28536	8.6
114	110	4710	318	35464	7.2
120	113	5608	267	42225	6.1
126	116	6675	225	50259	5.1
132	119	7942	189	59799	4.3
138	122	9461	158	71236	3.6
144	125	11216	134	84450	3.0
150	128	13432	112	101135	2.5

VI. COMPUTER EXPERIMENT

Example 3: We formed a test signal $x(n)$ composed by six sinusoids of different frequencies 185.0, 196.0, 587.3, 622.3, 1046.5, and 1108.7 Hz (corresponding to F#, G, D, D#, C, and C#, respectively), sampled at a rate of $f_s = 44.1$ kHz. The entire signal has a total of 44100 samples, or 1 s. This signal was processed by the sFFT, FFB, CQT, and CQFFB tools, all with 100 frequency bands between 130.8 and 2282.4 Hz (corresponding to C and halfway between C# and D, respectively), for the sake of uniformity. Hence, the frequency resolution for the sFFT and FFB, which perform a linear frequency sampling, was

$$(\delta f)_{\text{sFFT}} = (\delta f)_{\text{FFB}} = \frac{2282.4 - 130.8}{100} = 21.5 \text{ Hz}. \tag{16}$$

Accordingly, for the CQT and CQFFB we had $Q = 34$, corresponding to a quarter-tone frequency resolution, as suggested before.

The output magnitudes obtained by all four analysis tools for the input signal $x(n)$ are depicted in Figure 3. The sFFT response can be seen in Figure 3(a), from which one may notice that the sFFT yielded a nonzero spectrum for all frequency components and it was unable to resolve properly the low-frequency signal components. The FFB response is shown in Figure 3(b), where one may easily see that although the FFB was able to point out the purely sinusoidal characteristic of the input signal, it was not able, however, to separate properly the two low-frequency components. In Figure 3(c), the response of the CQT indicates that it succeeds in analyzing all sinusoidal components clearly, while presenting some sort of noisy behavior throughout the spectrum. Finally, Figure 3(d) depicts the magnitude response of the proposed CQFFB, which is able to show clearly the sinusoidal nature of the input signal, due to its selective response, with all six components clearly visualized, due to its logarithmic frequency scale.

■

VII. CONCLUSION

In this paper, two techniques previously known in the literature were reviewed, namely: the constant-Q transform (CQT), presented in [2], and the fast filter bank (FFB), introduced in [4]. A novel transform was then proposed, exhibiting a constant-Q characteristic, as opposed to the linear frequency resolution of the FFB (including the traditional sFFT), and improved frequency response, if compared to the standard sFFT-like response of the CQT. These two features combined together make the so-called constant-Q fast filter bank (CQFFB) especially suitable for audio applications, including analysis, coding, and transcription, as indicated by a computer experiment.

VIII. REFERENCES

[1] P. S. R. Diniz, E. A. da Silva, and S. L. Netto, *Digital Signal Processing; System Analysis and Design*, Cambridge, Cambridge, UK, 2002.

[2] J. C. Brown, “Calculation of a constant Q spectral transform,” *J. Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, Jan. 1991.

[3] J. C. Brown and M. S. Puckette, “An efficient algorithm for the calculation of a constant Q transform,” *J. Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, Nov. 1992.

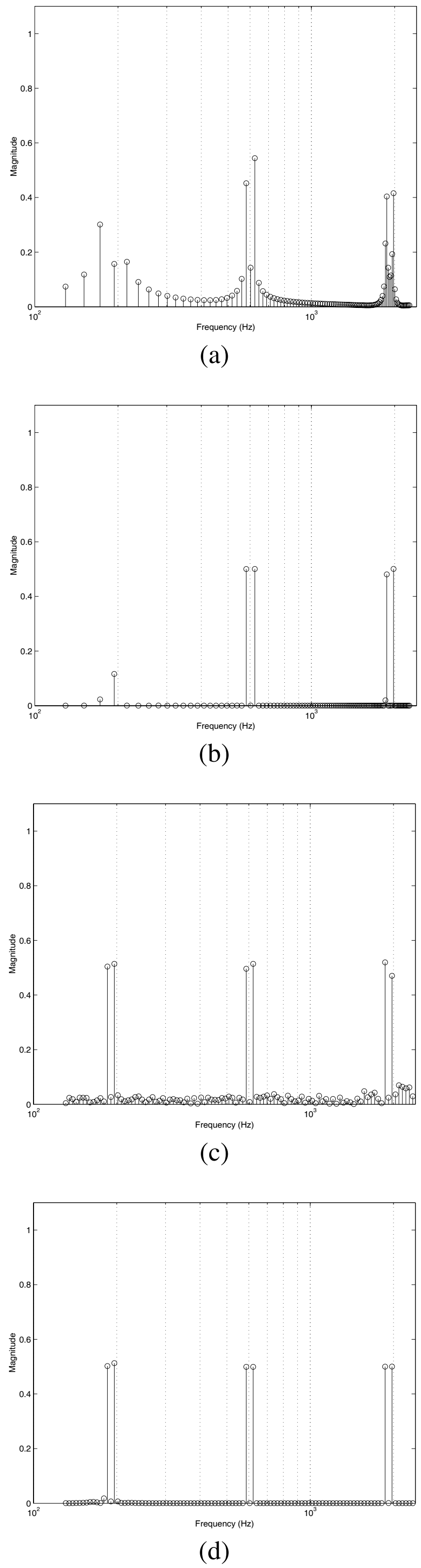
[4] Y. C. Lim and B. Farhang-Boroujeny, “Fast filter bank (FFB),” *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 39, no. 5, pp. 316–318, May 1992.

[5] B. Farhang-Boroujeny and Y. C. Lim, “A comment on the computational complexity of sliding FFT,” *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 39, no. 12, pp. 875–876, Dec. 1992.

[6] Y. C. Lim and B. Farhang-Boroujeny, “Analysis and optimum design of the FFB,” *Proc. IEEE Int. Symp. Circuits and Systems*, pp. II.509–II.512, London, UK, 1994.

[7] Y. C. Lim, “Frequency-response masking approach for the synthesis of sharp linear phase digital filters,” *IEEE Trans. Circuits and Systems*, vol. CAS-33, pp. 357–364, Apr. 1986.

[8] S. W. Foo and W. T. Lee, “Transcription of polyphonic signals using fast filter bank,” *Proc. IEEE Int. Symp. Circuits and Systems*, pp. III.241–III.244, Scottsdale, USA, May 2002.



(a)

(b)

(c)

(d)

Fig. 3. Example 3: Magnitude responses of 100-point transformations of an input signal composed by six sinusoidal components: (a) sFFT; (b) FFB; (c) CQT; (d) CQFFB.