SLIDING WITH A CONSTANT Q

Russell Bradford,
Dept. of Computer Science
University of Bath
England
`rjb@cs.bath.ac.uk`

*John ffitch,**
Codemist Ltd
Combe Down
Bath, UK
`jpff@codemist.co.uk`

Richard Dobson,†
Composer's Desktop Project,
Frome, Somerset
United Kingdom
`richarddobson@blueyonder.co.uk`

ABSTRACT

The linear frequency (constant-bandwidth) scale of the FFT has long been recognised as a disadvantage for audio processing. Long analysis windows are required for adequate low-frequency resolution, while small windows offer lower latency, better handling of transients, and reduced computation cost. A constant-Q form of analysis offers the possibility of increased low-frequency resolution for a given window size, this resolution being essential for many fundamental processing tasks such as pitch shifting.

We consider the application of the Sliding Discrete Fourier Transform to a Constant-Q analysis. The increased flexibility of sliding allows for a variety of data alignments, and we produce the mathematical formulation of these. Windowing in the frequency domain introduces further complications. Finally we consider the implementation of the analysis on both serial and parallel computers.

1. INTRODUCTION

The Discrete Fourier Transform (DFT) is usually calculated by the FFT algorithm, but this forces a number of constraints, like a power of 2 window size[1] and frequency bins of equal width in Hertz. With the increased computational power now available it is possible to consider alternative methods. We have already presented the Sliding Discrete Fourier Transform[1] and the Sliding Phase Vocoder[2, 3] in both theory and practice, following the lead of Moorer[4]. There we used the Sliding Discrete Fourier Transform (SDFT) as a direct replacement for the FFT, incorporated into a full sliding phase vocoder (SPV) and gained musically both from the increased bandwidth in each bin and from the single-sample update. Latency is also reduced. Pitch shifting was shown to be greatly simplified, as the bandwidth of each bin is bounded only by the Nyquist rate. For example we have demonstrated its use in applying classic audio-rate FM to an incoming audio stream.

In this paper we consider the other often quoted advantage of the SDFT, that the frequency bins do not need to be the same size. We will consider that implication of a constant-Q variant of the DFT, performing the stepping via sliding. By Q we mean for each bin the ratio of centre frequency to the bandwidth. We show that the computation is viable, and the results are indeed as expected.

The problem of constant-Q frequency representation is certainly not new: for example Brown[5] considers this problem, and there are many others. It was considered sufficiently important that Izmirli[6] was willing to consider multiple FFTs to get the information. There have been various attempts to approximate a constant-Q transform from a conventional FFT, such as Härmaä *et al.*[7] who used a filter to warp the original signal according to psychoacoustic principles. Other approximations can be found in [8], [9] and [10].

2. OFFSET CONSTANT Q SLIDING DFT

Traditional constant Q DFTs take all the samples in their analysis starting at the beginning of the frame (*e.g.* [11]). Thus the high frequency bins, having shorter frames, ignore the later samples from the frame, as shown in table 1. Some of these samples might be covered in an overlap as the frames are stepped.

Table 1: *Normal alignment of frames.*

	0 summing over these samples $N-1$
low	$\xleftrightarrow{\hspace{14em}}$
med	$\xleftrightarrow{\hspace{7em}}$
high	$\xleftrightarrow{\hspace{3em}}$

In the sliding constant Q DFT, the situation is somewhat different as each sample is analysed at some point as it slides through every frame.

In effect Q measures how many cycles are needed to make an analysis. To resolve a fraction f of a tone we have $Q = 1/(2^{f/6}-1)$, so for a quarter-tone resolution ($2^{1/24}$ ratio between bin centres) Q is approximately 34. The lowest frequency bin is

---

* Also at University of Bath

† Sometime Research Officer, University of Bath

1 Some implementations, such as FFTW, employ a mixed-radix approach to relax the power-of-2 restriction at the cost of some increased computation

the longest: for a frequency of 27.5HZ (the lowest A on a standard piano) at standard CD quality 44100 samples/sec this corresponds to 54728 samples, which translates to a latency of 1.24 seconds. At the Nyquist frequency this drops to 70 samples, or 1.59ms. Indeed, if we slide in a single sample at a time, it will be about 1.24 seconds before a given sample hits the smaller bins.

However there is nothing to stop us moving where the frames start analysing the samples. We can allow each frame to start at an arbitrary offset, and could even allow the frame to move about during the analysis[2].

In the sequel we use the following notation and nomenclature:

- α: bin centre ratio, $\alpha = 1 + 1/Q$
- bin: centred on an analysis frequency (indexed by k, with frequency $f_{\min}\alpha^k$)
- bin width: range of frequencies a bin covers (roughly $f_{\min}\alpha^{k-1/2}$ to $f_{\min}\alpha^{k+1/2}$)
- frame: a set of contiguous samples contributing to a single bin
- frame size: number of samples to compute the bin (N_k)
- window: a cosine-like operation on a frame to reduce smearing

3. THE OFFSET FORMULAE

Table 2: *Frames aligned to end at the same sample.*

	0 summing over these samples $N-1$
low	$\longleftrightarrow$ (samples 0 to $N-1$)
med	$\longleftrightarrow$
high	$\longleftrightarrow$

Table 3: *Frames aligned to their middles.*

	0 summing over these samples $N-1$
low	$\longleftrightarrow$ (samples 0 to $N-1$)
med	$\longleftrightarrow$
high	$\longleftrightarrow$

Suppose frame k has an offset of s_k. Then the constant Q DFT for bin k at time t is

$$F_t(k) = \frac{1}{N_k} \sum_{j=0}^{N_k-1} f(j + s_k + t) e^{-2\pi i j Q/N_k} \tag{1}$$

where N_k is the size of frame k.

[2] though probably this is not a terribly useful thing to do!

Sliding by one sample, at time $t+1$:

$$\begin{aligned} N_k F_{t+1}(k) &= \sum_{j=0}^{N_k-1} f(j + s_k + t + 1) e^{-2\pi i j Q/N_k} \\ &= \sum_{j=1}^{N_k} f(j + s_k + t) e^{-2\pi i j Q/N_k} e^{2\pi i Q/N_k} \\ &= e^{2\pi i Q/N_k} \Bigg(\sum_{j=0}^{N_k-1} f(j + s_k + t) e^{-2\pi i j Q/N_k} + \\ &\qquad f(s_k + N_k + t) e^{-2\pi i Q} - f(s_k + t) \Bigg) \end{aligned} \tag{2}$$

Thus

$$\begin{aligned} F_{t+1}(k) &= e^{2\pi i Q/N_k} \times \\ &\left(F_t(k) + \frac{e^{-2\pi i Q} f(N_k + s_k + t) - f(s_k + t)}{N_k} \right) \end{aligned} \tag{3}$$

In this, $f(s_k + N_k + t)$ is the new sample being slid in, while $f(s_k + t)$ is the old sample moving out.

We now consider three particular cases; aligned left, centrally and right.

3.1. Case $s_k = 0$

The simple left-aligned constant Q sliding DFT corresponds to setting all s_k values to zero. The equation 3 then reduces to

$$F_{t+1}(k) = e^{2\pi i Q/N_k} \left(F_t(k) + \frac{e^{-2\pi i Q} f(N_k + t) - f(t)}{N_k} \right) \tag{4}$$

This has the large latency for analysis of high frequencies, as noted above, based on the window size of the lowest frequency.

3.2. Case $s_k = N - N_k$

The other extreme case is when the different frequency frames are right-aligned, as shown in figure 2. So if N is the length of the longest frame we get that $F_{t+1}(k)$ is

$$e^{2\pi i Q/N_k} \left(F_t(k) + \frac{e^{-2\pi i Q} f(N + t) - f(N - N_k + t)}{N_k} \right) \tag{5}$$

where $f(N+t)$ is the new sample (the same sample for all frames) and $f(N - N_k + t)$ is the oldest sample in each frame.

This has low latency for high frequencies and larger latency for lower frequencies; the latency increases with the wavelength.

3.3. Case $s_k = (N - N_k)/2$

The other potentially interesting case is to align all the windows on the middle of the windows. This is the middle-aligned constant Q sliding DFT, shown in figure 3:

$$F_{t+1}(k) = e^{2\pi i Q/N_k} \times \left(F_t(k) + \frac{e^{-2\pi i Q} f(\frac{N+N_k}{2}+t) - f(\frac{N-N_k}{2}+t)}{N_k} \right) \quad (6)$$

where again $f(\frac{N+N_k}{2}+t)$ is the new sample, $f(\frac{N-N_k}{2}+t)$ the old sample.

This has a fairly high latency for high frequencies, but has the advantage that all the frames' midpoints are aligned so in some sense they are all analysing the same part of the signal.

4. COMPUTATIONAL ASPECTS

The formulae look complicated, but the inner loop is simply

```
F[k] = exps[k]*(F[k] + (expm2piIQ*fx[sk + Nk]
                         - fx[sk])/Nk);
```

Where `exps` is an array of pre-computed complex exponentials, `expm2piIQ` is a single pre-computed complex exponential, `fx` the array of real input samples and `F` the output array of the complex DFT. This loop can be trivially parallelised.

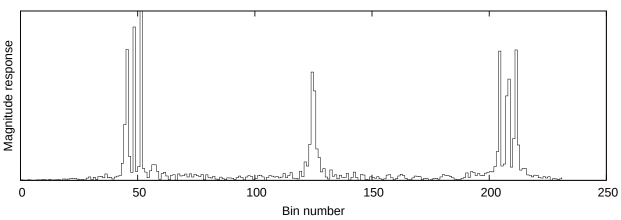

Figure 1: *Constant Q DFT*

To show the advantage of a constant Q sliding DFT we consider a signal constructed from a sum of sine waves of frequencies of 100, 110, 120, 1000, 10000, 11000, 12000Hz. In figure 1 we show the DFT of this signal. No windowing was used. The low frequencies are as clearly separated as the high frequencies.

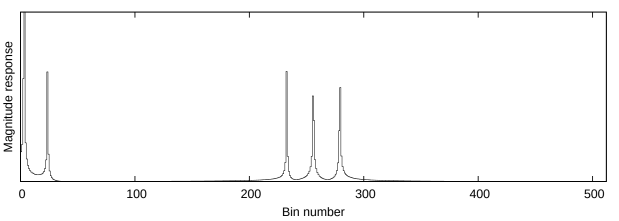

Figure 2: *Standard DFT*

In contrast, Figure 2 shows the lower half of a 1024-sample standard DFT of the same signal; the separation of the low frequencies is totally missing.

It is also worth noting that with constant Q we do not get the symmetry of the bins about the middle due to the complex conjugate and so there are no redundant bins.

The time to add one sample is proportional to the number of bins. For quarter tone separations and a $f_{\text{min}} = 27.5$Hz, at sample rate 44100Hz this is 232 bins. This is a considerable saving on the regular SDFT, and the reduction of bins by a log makes the computational cost similar to a more normal FFT analysis.

5. WINDOWING

In a SDFT any sample windowing to reduce smearing has to be applied in the frequency domain, after the transform, as described in [3]. For the constant Q variant this process is not so easy as in the variable Q sliding variant. So let

$$F_{Qt}(k) = \frac{1}{N_k} \sum_{j=0}^{N_k-1} f(j+s_k+t) e^{-2\pi i j Q/N_k} \quad (7)$$

where we make Q explicit in F, and

$$F^w_{Qt}(k) = \frac{1}{N_k} \sum_{j=0}^{N_k-1} w_k(j) f(j+s_k+t) e^{-2\pi i j Q/N_k} \quad (8)$$

be the windowed transform, where $w_k(j) = a + b\cos(2\pi j/N_k)$ is a standard cosine window. Different values of the constants a and b correspond to a range of common windowing methods. For example, $a = 1/2$ and $b = -1/2$ is the von Hann window. For the following we will restrict the discussion to single cosine windows.

So

$$\begin{aligned}
N_k F^w_{Qt}(k) &= \sum_{j=0}^{N_k-1} w_k(j) f(j+s_k+t) e^{-2\pi i j Q/N_k} \\
&= a \sum_{j=0}^{N_k-1} f(j+s_k+t) e^{-2\pi i Q/N_k} + \\
&\quad b \sum_{j=0}^{N_k-1} \cos(2\pi j/N_k) f(j+s_k+t) e^{-2\pi i Q/N_k} \\
&= a N_k F_{Qt}(k) + \\
&\quad b \sum_{j=0}^{N_k-1} \left(\frac{e^{2\pi i j/N_k} + e^{-2\pi i j/N_k}}{2} \right) f(j+s_k+t) e^{-2\pi i Q/N_k} \\
&= a N_k F_{Qt}(k) + \\
&\quad \frac{b}{2} \sum_{j=0}^{N_k-1} f(j+s_k+t) \left(e^{-2\pi i j(Q-1)/N_k} + e^{-2\pi i j(Q+1)/N_k} \right) \\
&= N_k \left(a F_{Qt}(k) + \frac{b}{2} F_{Q-1,t}(k) + \frac{b}{2} F_{Q+1,t}(k) \right) \quad (9)
\end{aligned}$$

Thus

$$F^w_{Qt}(k) = a F_{Qt}(k) + \frac{b}{2} F_{Q-1,t}(k) + \frac{b}{2} F_{Q+1,t}(k) \quad (10)$$

Recall that the range of k, the number of bins, is typically quite small, so it does not take much extra memory to keep all of the $F_{Qt}(k)$, $F_{Q-1,t}(k)$, and $F_{Q+1,t}(k)$ values. We do have extra work in sliding the samples for all three transforms though. But again it should be noted that this is all pathologically parallel.

Figure 3 shows the effect of adding a von Hann window to the analysis of the synthetic signal used above. It greatly reduces the response outside the target bins. Similarly, Figure 4 shows a windowed transform of a simple square wave, showing clearly the $1/n$ drop-off.

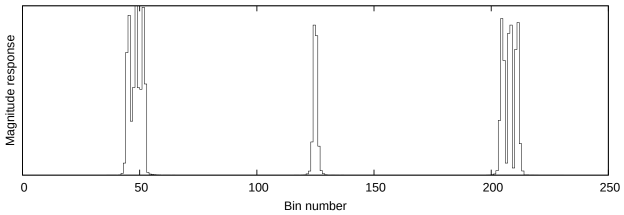

Figure 3: *Windowed Constant Q DFT.*

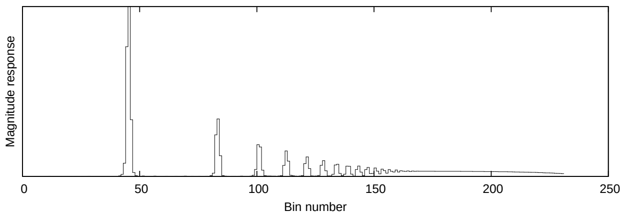

Figure 4: *Windowed Constant Q Sliding DFT of a Square Wave.*

6. RESYNTHESIS

The SDFT can be represented as a matrix that maps samples into frequencies (figure 5). Apart from a few edge cases, this matrix is singular, having determinant zero. Thus we can not expect to be able to reconstruct the exact signal as the SDFT does not retain all the information in the original signal.

There have been attempts at reconstruction, such as Zhuo and Micheli-Tzanakou[12] who uses a single (middle of window) point. Realising the problems of the inverse, FitzGerald, Cranich and Cychowski[13] approximates an inverse by minimisation techniques.

$$\begin{pmatrix}
1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 & \alpha_1^4 & \alpha_1^5 & \alpha_1^6 & \alpha_1^7 & \alpha_1^8 & \alpha_1^9 & \alpha_1^{10} & \alpha_1^{11} \\
0 & 1 & \alpha_2 & \alpha_2^2 & \alpha_2^3 & \alpha_2^4 & \alpha_2^5 & \alpha_2^6 & \alpha_2^7 & \alpha_2^8 & \alpha_2^9 & 0 \\
0 & 0 & 1 & \alpha_3 & \alpha_3^2 & \alpha_3^3 & \alpha_3^4 & \alpha_3^5 & \alpha_3^6 & \alpha_3^7 & \alpha_3^8 & 0 \\
0 & 0 & 1 & \alpha_4 & \alpha_4^2 & \alpha_4^3 & \alpha_4^4 & \alpha_4^5 & \alpha_4^6 & \alpha_4^7 & \alpha_4^8 & 0 \\
0 & 0 & 1 & \alpha_5 & \alpha_5^2 & \alpha_5^3 & \alpha_5^4 & \alpha_5^5 & \alpha_5^6 & \alpha_5^7 & 0 & 0 \\
0 & 0 & 0 & 1 & \alpha_6 & \alpha_6^2 & \alpha_6^3 & \alpha_6^4 & \alpha_6^5 & \alpha_6^6 & 0 & 0 \\
0 & 0 & 0 & 1 & \alpha_7 & \alpha_7^2 & \alpha_7^3 & \alpha_7^4 & \alpha_7^5 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \alpha_8 & \alpha_8^2 & \alpha_8^3 & \alpha_8^4 & \alpha_8^5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \alpha_9 & \alpha_9^2 & \alpha_9^3 & \alpha_9^4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \alpha_{10} & \alpha_{10}^2 & \alpha_{10}^3 & \alpha_{10}^4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \alpha_{11} & \alpha_{11}^2 & \alpha_{11}^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \alpha_{12} & \alpha_{12}^2 & \alpha_{12}^3 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Figure 5: *Example SDFT transformation: αs stand for various roots of unity.*

We have a kind of reconstruction:

$$\hat{f}(t) = \sum_{k=0}^{B} F_t(k) e^{2\pi i Q/N_k} \tag{11}$$

where B is the number of bins. This is fast, as B is relatively small, and all the $e^{2\pi i Q/N_k}$ are already precomputed as part of the forward transform.

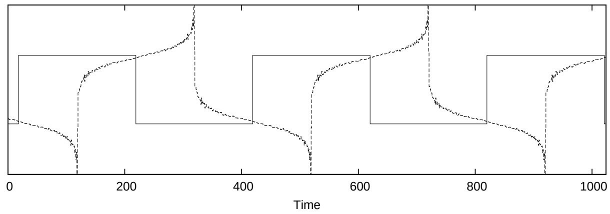

Figure 6: *Square wave and a reconstruction.*

In practice, this reconstructed signal *sounds* remarkably similar to the original. Figure 6 show a square wave and its reconstruction using this formula. They are clearly quite different in their phase content. However they sound almost the same: after all, they have the same representation in the constant Q.

The utility of the transform can be seen from the original and resynthesis of a slap-low bass guitar sound. In figure 7 we show part of the wave-forms, which while not bit-accurate is very similar. The latency of the process is also clear in this figure. The spectral view in figure 8 emphasises the similarities.

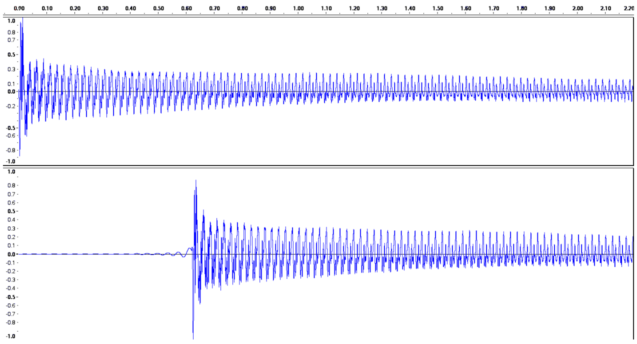

Figure 7: *Wave form of slap low bass.*

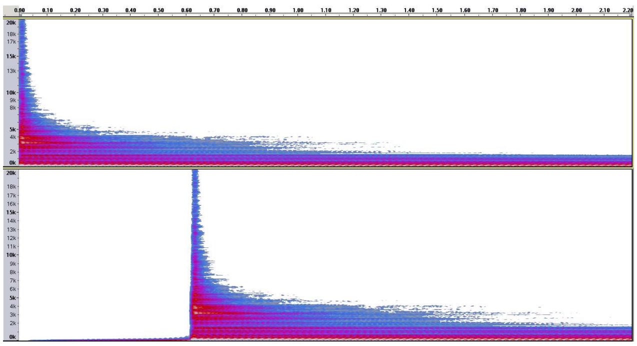

Figure 8: *Spectrum of slap low bass.*

7. TRANSFORMATIONS

In implementing the sliding constant-Q transform we are at the beginning of a journey through largely unexplored territory. The

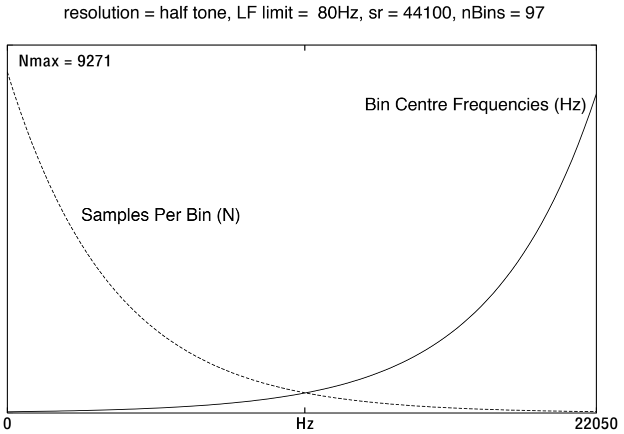

Figure 9: *The Structure of Constant-Q: N per bin v bin frequencies.*

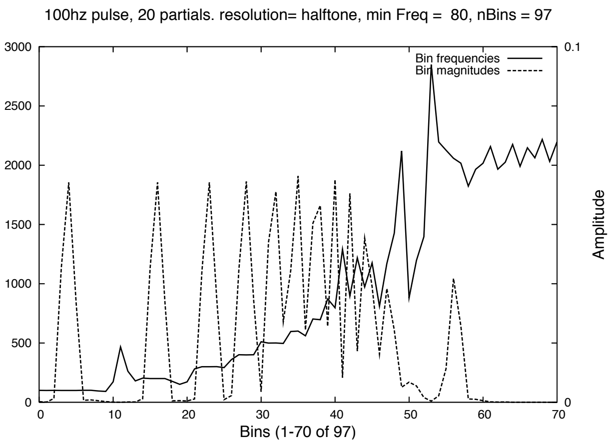

Figure 10: *Pulse-wave: Close-up of Bin frequencies v magnitudes.*

contrast with the standard linear transform is stark. Figure 9 relates the centre frequency of each bin to the number of samples to which they correspond. The complementary exponential nature of both parameters is obvious. Just as interesting is the shape of the input magnitude response. Figure 10 illustrates a typical response, together with the nominal centre frequencies for each bin. In the conventional DFT (sliding or otherwise, we are used to seeing a somewhat indistinct low frequency (LF) response, with increasing resolution at high frequency. As expected, the sliding constant-Q transform reverses this behaviour.

Figure 11 shows the result of an upwards pitch shift by 3 semitones. Clearly the phases especially of the higher components are disturbed (in some orderly way: there is a clear pattern to the variations; we expect further analysis will provide a simple formula to describe this), but nevertheless it sounds very reasonable. Note that while the latency is extreme because of the large Nmax, the number of bins passed through the transformation is (in this case) only 97, comparable more to a peak-tracked stream than to a standard DFT-based stream. Initial tests with music inputs demonstrate a surprising degree of fidelity given such a small number of bins, though much more work is needed to establish the determining

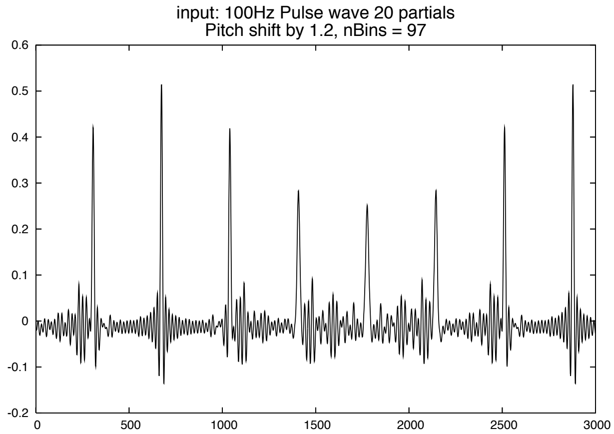

Figure 11: *Waveform of pitch-shifted pulse wave.*

factors for high-quality transformations.

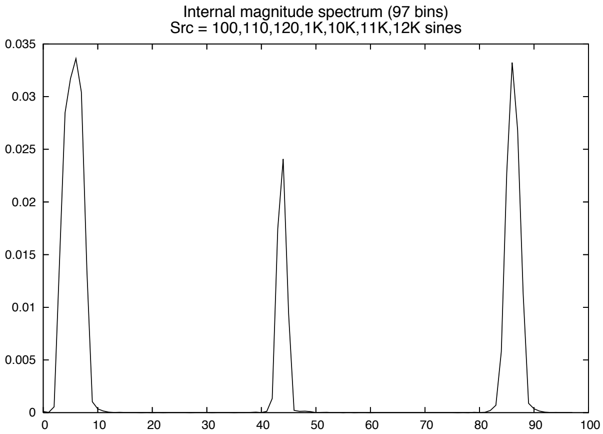

Figure 12: *Internal bin magnitudes representation of Figure 1.*

Central to the use of the sliding constant-Q approach is the simple pitch shifting algorithm enabled by the single-sample update, as described in the SPV[2]. No bin reassignment is required; the frequency value in each bin is simply scaled by the shift factor. With reference to the test signal shown in figure 1, figure 12 shows the internal magnitude values using the same settings used in figures 9, 10 and 11. Unsurprisingly, for a total of 97 analysis bins, the three LF components appear as a single wide peak. However, the spectrum of the pitch-shifted output (when calculated using a suitably large FFT of at least 8192 points, not shown here) shows that all components have retained their clear separation and frequency relationship.

8. DISCUSSIONS

At this early stage of exploration, the sliding constant-Q transform appears capricious, sometimes performing much as intended, and at other times (as we have previously found with the linear sliding phase vocoder) presenting plenty of intriguing and surprising emergent features. We therefore hesitate to make any definitive claims at this stage about its usability for demanding audio work. Instead, we confine ourselves to a number of general observations which indicate likely routes of further investigation. The current theoretical model presents the problem that the low frequencies are in practice over-emphasised. This is reflected both in the very large sample window (Nmax) required for those low frequencies, and in the high ratio of Nmax to the number of analysis bins arising from the constant-Q conversion. Figure 9 offers a clue here — the result of reducing that ratio would be equivalent to reducing the gradient of the curves. In principle (and recalling the widely-used Zoom transform in which just a few arbitrarily selected frequency bins are calculated), we speculate that not only may "less exponential" gradients be available, but perhaps even a combination of curves not necessarily monotonic; this is similar to the proposals by Brown and Puckette[11] and Härmaä *et al.*[7]. Our goal of course is to be able to employ a much smaller overall transform window of (say) 1024 samples, for low latency, while gaining the LF resolution associated with larger FFT windows. This may lead not to a single "ideal" constant-Q transform but to a family of *warped-constant-Q* transforms (loosely analogous to families of wavelets), which may be tuned by the user according to need.

In this respect we note that the current model achieves surprisingly good results (though not yet "pro quality") even with a very small number of analysis bins and with no attempt at phase re-alignment for transients. This suggests (though it is contrary to our primary interest in live performance) that the constant-Q transform may find use as a pre-analysis process (which will then eliminate all issues of latency) for use in samplers. The single sample update offers musical advantages, while the data storage and bandwidth requirements, though substantial, are much less than those for the linear sliding phase vocoder and are by no means unsupportable on modern platforms.

In our initial tests with music sources, we find that the emphasis on the low frequencies is not only analytically significant, it can also manifest as a general sonic LF bias (bass boost). We suspect therefore that the transform carries implications of concomitant EQ correction, if high-quality transformations are to be achieved. It may also be that the suggested warped-constant-Q approach will resolve this issue. The lossy nature of the process suggests that the constant-Q transform is best applied to sources that do not include significant broadband noise components such as reverberation. However, a primary motivation for this work was to find transformations effective on drums and general LF percussion, sounds which demand both low latency and fine LF resolution. We are encouraged by our initial results in this direction.

In its present form, we use only the von Hann window in the analysis stage (that window having proved to be the most effective in the linear Sliding DFT). However, the presence of significant sidelobe-like artifacts in many of the transformations we have tried suggest that other higher-order windows may be more appropriate to the constant-Q transform. It is too soon to discount the transform as a starting-point for a peak-tracking analysis, but initial performance (as suggested by figure 12) currently indicates against this application.

9. CONCLUSIONS.

The concept of the constant-Q Discrete Fourier Transform has been known and talked about for many years, but efficient and effective methods for calculation have been rare. In this paper we remind the community that the Sliding Discrete Fourier Transform is ideal for this. While it might have a daunting complexity of $O(n^2)$ compared with the attractive $O(n \log n)$ of the FFT, by reducing the number of computed bins by the logarithm it becomes competitive. Added to this the pathologically parallel nature of the algorithm suggests that a suitable application of vector processing or multi-cell computers can remove the last concerns.

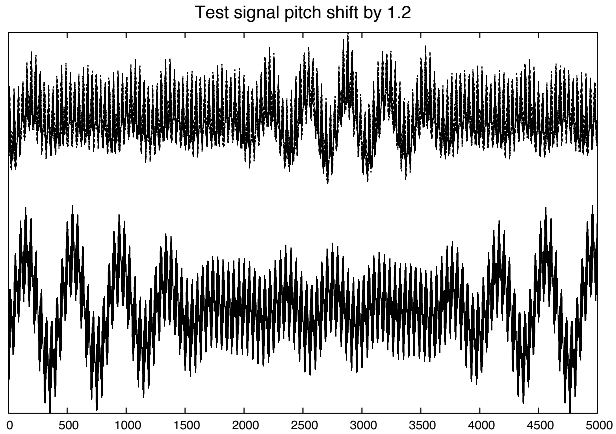

Figure 13: *Test signal of Figure 1. Lower = input, upper = pitch-shifted output.*

Our initial theoretical approach demands very large transform windows, which contradict our goal of low latency. However the results achieved with the very much smaller internal analysis windows are most encouraging. A key goal for future work is reduction of this ratio, so that the expected advantages of low latency and enhanced LF resolution can be realised. While computationally demanding, the algorithm is, like the Sliding Phase Vocoder, highly parallelisable, and (if the latency question can be resolved) available to real-time implementation given suitable hardware. In this respect we believe that the sliding constant-Q transform will permit as wide a range of musical transformation as already identified for the SPV. The question of reconstruction remains moot — while many uses for the analysis stage alone suggest themselves, the inherently lossy nature of the transform means that while it is suited to a wide class of sounds and artistic activities, it will not apply to all. The full extent and seriousness of this limitation remains to be determined.

This work was initially funded by the Arts and Humanities Research Council, and continued with partial support from Clearspeed plc; we wish to express our thanks to both of these for their confidence.

10. REFERENCES

[1] Russell Bradford, Richard Dobson, and John ffitch, "Sliding is Smoother than Jumping," in *ICMC 2005 free sound*. Escola Superior de Música de Catalunya, 2005, pp. 287–290.

[2] Russell Bradford, Richard Dobson, and John ffitch, "The Sliding Phase Vocoder," in *Proceedings of the 2007 International Computer Music Conference*. August 2007, vol. II, pp. 449–452, ICMA and Re:New, ISBN 0-9713192-5-1.

[3] John ffitch, Richard Dobson, and Russell Bradford, "Sliding DFT for Fun and Musical Profit," in *6th International*

Linux Audio Conference, Frank Barknecht and Martin Rumori, Eds., Kunsthochschule für Medien Köln, March 2008, LAC2008, pp. 118–124.

[4] James A. Moorer, "Audio in the New Millennium," *J. Audio Eng. Soc.*, vol. 48, no. 5, pp. 490–498, May 2000.

[5] Judith C Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, January 1991.

[6] Ozgur Izmirli, "A Hierarchical Constant Q Transform for Partial Tracking in Musical Signals," in *Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99)*, NTNU, Trondheim, Norway, 1999.

[7] Aki Härmaä, Matti Karjalainen, Lauri Savioja, Vesa Välimäki, Unto K Lane, and Jyri Huopaniemi, "Frequency-Warped Signal Processing for Audio Applications," *Journal of[] the Audio Engineering Society (AES)*, vol. 48, no. 11, November 2000.

[8] Aki Härmä and Tuomas Paatero, "Discrete Representation of Signals on a Logarithmic Frequency Scale," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2001, pp. 39–42.

[9] Carlo Braccini and Alan V Oppenheim, "Unequal Bandwidth Spectral Analysis using Digital Frequency Warping," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 4, pp. 236–244, August 1974.

[10] Thomas von Schroeter, "Frequency Warping with Arbitrary Allpass Maps," *IEEE Signal Processing Letters*, vol. 6, no. 5, pp. 116–118, May 1999.

[11] Judith C. Brown and Miller S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[12] Wen Zhuo and Evangelia Micheli-Tzanakou, "A High Performance Continuous Data Flow Filter using Sliding Discrete Fourier Transform (DFT) and One Point Inverse DFT," in *Proceedings 1998 IEEE International Conference on Information Technology Applications in Biomedicine*, May 1998, pp. 51–56.

[13] Derry FitzGerald, Matt Cranich, and Marcin T Cychowski, "Towards an Inverse Constant Q Transform," in *Proceedings 120th AES Convention*, 2006, Available from `http://www.audioresearchgroup.com/main.php?page=Publications`.