Proceedings of Frontiers of Research on Speech and Music 2004, Annamalai University, India

APPLICATIONS OF THE BLUESTEIN FILTER IN BOWED-STRING ANALYSIS

Knut Guettler
Norwegian Academy of Music
P.O. Box 5190 Majorstuen, 0302 Oslo, Norway
knut.guettler@samson.nmh.no

Abstract
In this study, the search for the origin of noise produced in the bowed string serves as the main example of advantageous use of the Bluestein filter[1] in a certain type of analyses. The task is here to separate the stochastic part of the signal from the deterministic one, without shifting the time-domain position of the noise signal, the amplitude of which is rapidly (but periodically) varying over each steady-state oscillation period. Or, as a compromise, at least keeping the noise amplitudes unaltered with respect to the phase of this periodic signal. This can be accomplished by using a correctly dimensioned rectangular time-domain window in combination with a perfectly periodic comb filter in the frequency domain. The routine is facilitated by using the Bluestein filter, which permits arbitrarily sized DFT to be computed as FFT, giving a substantial time advantage over conventional DFT. Other examples of practical filter usage related to bowed string analyses are also discussed.
Key words: Signal analysis, Noise filtering, Bowed string.

Introduction

After J. W. Cooley and J. W. Tukey published their algorithm for speedy computation of the Fourier transform ("An algorithm for the machine calculation of complex Fourier series"[2]—now commonly referred to as FFT), the majority of practical discrete Fourier analyses have been based on arrays of *N* data elements, where *N* is 2 raised to the power of any reasonable integer. For many applications, however, the original discrete Fourier transform, DFT (where *N* is chosen arbitrarily), provides cleaner images in the frequency domain, particularly when the time-domain signal is periodic or nearly so, as is the case for many music instruments with phase-locked oscillations. The image could become even more blurred if the signal needs another transformation back to the time domain after having been filtered in the frequency domain. However, already in 1968 L. I. Bluestein published "A linear filter approach to the computation of the fast Fourier transform"[1], which provided fast transformation of arrays containing any number of elements, and with accuracy limited only by the machine's round off error—still using the Cooley/Tukey's FFT as the central vehicle for transformation.

The later technique has proven advantageous in many applications, one of them being separation of stochastic and deterministic components of musical signals. Several methods have already been suggested for this kind of separation[3,4]. One problem with these, however, is that they do not seem to give proper information about the generation of noise in the time domain for cases where the noise amplitude envelope is periodically changing. For most kinds of spectral-modeling (re)synthesis this represents no problem, as the ear is rather insensitive to phase relations above a relatively low fundamental frequency. For analyses of the noise's origin, however, the time-domain information is crucial. This information can be well kept if using DFT window length of the proper size, exactly matching an integer multiple of the fundamental-period length.

Other examples of usage of the Bluestein filter within bowed-string analysis are speedy resampling between arbitrary sampling rates, and highly improved sound-to-noise ratios when utilizing window lengths that match arbitrarily chosen numbers of fundamental periods in (nearly) harmonic signals. Both types of techniques will be discussed in this paper. Compared to conventional FFT the Bluestein filter reduces computation speed by a factor approx. 0.15 for most applications.

The Bluestein filter used for spectral analysis—
a comparison between true DFT and the zero-padded FFT

In order to visualize the difference between true DFT (by application of the Bluestein filter) and zero-padded FFT the following experiment was performed: A bowed string driven in steady state was simulated. The fundamental period was steadily occupying 306 time steps. Two spectral analyses were done using the same time-domain signal multiplied by a 12×306 (=3672) elements' Hann window. Panel ***a*** of Figure 1 shows DFT over 3672 elements with Bluestein filtering, while panel ***b*** shows FFT over the same window zero-padded up to 4096 elements. Then the experiment is repeated after adding a sinuous vibration to the bow speed. The vibrational frequency is 1.5 times the string's fundamental (see arrow in panel ***c***), while the amplitude lies 46 dB below the constant bow speed. The effect of adding this vibration is clearly visible in panel ***c***: A subharmonic—half of the string's fundamental frequency—is founded with a complete overtone series. In panel ***d*** this effect is entirely hidden by the "interpolation" and side lobes created by the zero padding. A comparable situation would occur if stretching the Hann window over all 4096 elements of the FFT in order to avoid zero padding.

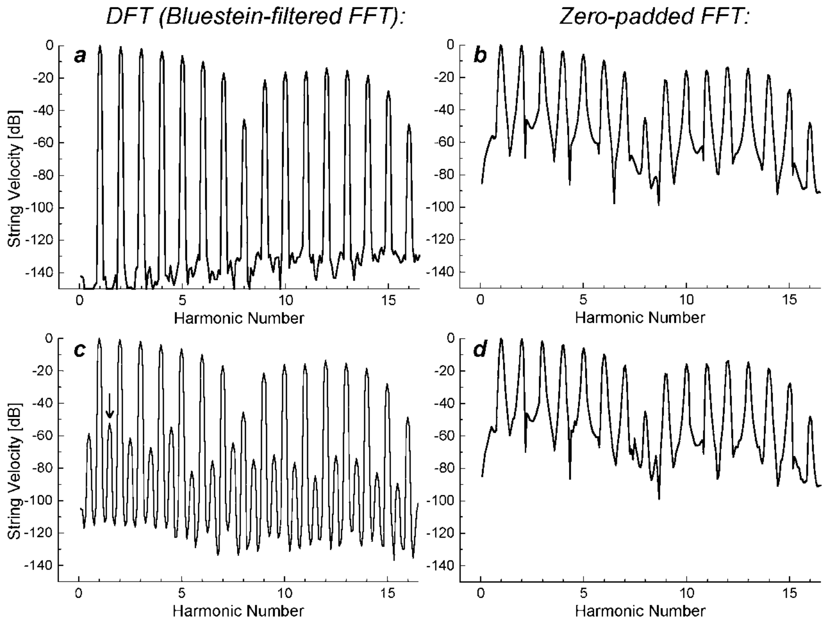

Figure 1: Comparison between Bluestein-filtered and zero-padded FFT. Panels ***a*** and ***b*** show spectra of the same signal, using these two methods respectively. The signal is the string velocity under the bow when oscillating in steady state. In panels ***c*** and ***d***, superimposed on the bow speed of 20 cm/s, a sinuous vibration with amplitude 0.1 cm/s, and frequency 1.5 times the string's fundamental was added. In panel ***c*** a "subharmonic" effect is clearly visible (reflecting frictional non-linearity), while panel ***d*** hardly deviates from panel ***b***.

One might object that had the vibrational amplitude been increased, the effect would have been visible also in panel ***d***. The reason for utilizing the low-amplitude vibration in this experiment, was, however, to avoid affecting the stability of the slip/stick pattern, which might otherwise excurse from the regular 306 time-step periods, and make spectral comparison more difficult.

The calculations of panels ***a*** and ***c*** lasted ca 6.6 times longer than the calculations of panels ***b*** and ***d***. However, compared to conventional DFT methods, the Bluestein filter provides the identical result ca 60 times faster than DFT with the array length chosen here. It is also worth noticing that while zero padding does not in principle affect the half-power bandwidth, it does indeed affect the bandwidths for most lower power ratios.

Resampling through use of the Bluestein filter

For all the examples discussed here, it is important to keep clearly in mind the fact that the discrete Fourier transform either transforms from a continuous signal to a discrete one, or vice versa—and that it has analytical implications which one of the two domains (time or frequency) one defines as continuous. In many practical cases one wants to see the spectrum of an originally continuous signal after having sampled it (and thus *apparently discretized it*) in the time domain. Provided the Nyquist criteria are met so that there is no frequency wrapping, after Fourier transformation, the discrete values of the frequency domain hold all information required for reconstructing the original continuous signal in the time domain, which also per definition is circular with period length equal to the time window. In such cases expanding the series of discrete frequencies (in both positive *and* negative direction) in the frequency domain, while setting each of these magnitudes to zero, provides a quick but correct form of resampling, without any distortion or loss of information when returned to the time domain. With normal FFT the number of elements is even, which means that the discrete frequency-domain representation comprises one more positive than negative frequency (the phase of which is zero when the time-domain signal is real). In the case of DFT and Bluestein-filtered FFT, the number of elements might also be odd, in which case the number of positive frequencies equals that of the negative ones (appearing as conjugate symmetric pairs).

If, however, the signal is truly *discrete* in the time domain (e.g., as the output of a digital z-filter), the application of DFT gives in principle the same transformation as the z-transform, *the frequency domain of which is continuous*. Here zero-padding in the frequency domain would produce a completely different effect, as the zeroes would replace magnitude values already implicitly defined by the transformation.

When wanting to use DFT or the Bluestein filter to *reduce* the sampling rate (e.g., resampling from 4200 to 4096 time steps per window), some loss of energy in the high-frequency end has to be accepted when pairs of positive and negative frequencies are omitted before the inverse transformation is performed.

Examples where resampling is advantageous are numerous (although use of the Bluestein filter would significantly reduce the number). E.g., “Reconstruction of bowing point friction force in a bowed string” as suggested by J. Woodhouse et al.[5] requires resampling of a steady state signal in their application. A simpler case occurs when wanting to compare how the string velocity during an averaged slip cycle changes with bow speed (as shown by Ref.[6]). Here the string’s velocity under the bow had been recorded with the sampling rate of 44.1 kHz. Since this proved to be too small for alignment and visual comparison of transition slopes during release and capture, the sampling rate was subsequently increased to 148 kHz through use of the Bluestein filter combined with zero padding in the frequency domain. The result is shown in Figure 2.

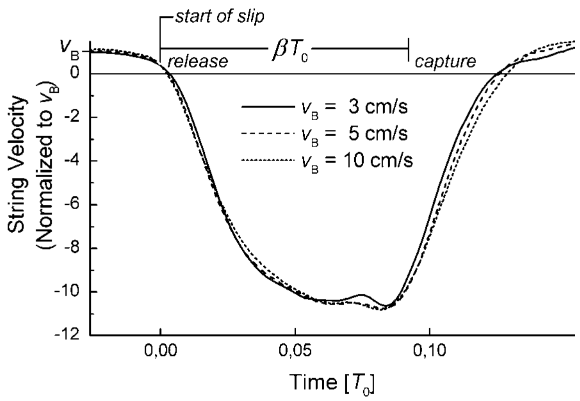

Figure 2: When reducing the bow speed while maintaining the bow force, the string’s velocity slopes at release and capture become steeper, which implies increased relative energy in the spectrum’s high end. In order to facilitate comparison, the signal was resampled from 44.1 to 148.0 kHz (by use of the Bluestein-filter method) so that the three patterns could be properly aligned. v_B = bowing velocity; β = relative bowing position; T_0 = nominal fundamental period.

Separation of stochastic and deterministic parts of a quasi-stationary signal

The present experiment was designed to determine how much noise was produced during the different phases (stick/slip) of a bowed-string fundamental period. It was important that the position of the noise signal with respect to slip/stick was not in any way altered during the process of separation. The following procedure was used:

(1) Precise determination of an (open/rectangular) time window comprising an integer number of fundamental periods.
(2) Transformation to the frequency domain without zero padding or other forms of windowing.
(3) Determination of, and multiplication with a comb filter, circularly continuous over the entire frequency array, and symmetrical around the zeroth element (the DC component) *and* the Nyquist frequency.
(4) Transformation back to the time domain, where preferably only the middle part of the signal will be used.

(1) In order to determine the length of this time window optimally, an algorithm where the fundamental period is extracted on base of a weighed best-fit of spectral peaks, is preferable, (e.g., R. C. Maher and J. W. Beauchamp: “Fundamental frequency estimation of musical signals using a two-way mismatch procedure”[7]—in which case *zero padding in the time domain* will improve accuracy). The window length could of course alternatively be determined manually by evaluating how well the periodic comb filter covers the harmonically placed peaks of the spectrum. [Notice: Since the comb filter used under point (3) of this procedure will (wrongly) determine the frequencies in the near vicinity of the zeroth element (i.e., the DC component) as *deterministic*, and harmonically related to the signal’s fundamental frequency (see Figure 3), it might be advantageous to have utilized an appropriate high-pass filter upon recording the signal, thus removing all energy in this region.]

(2) The chosen window length should contain an integer number of fundamental periods, each comprising an integer number of sample elements. If not, resample as described above, so that these two criteria are met. If the window length does not fit to the requirements of FFT (or another prime-factor transformation routine available), use the Bluestein filter (described in the Appendix) for DFT without padding.

(3) By fitting an exact integer number of signal periods into the (open/rectangular) time window one has ensured that the fitting comb filter becomes periodic in the (quasi circular) frequency domain (see Figure 3), as well as preventing interpolation or scattering between frequencies to take place before filtering (compare the columns of Figure 1). The teeth of the comb filter should preferably be as narrow as possible for the following reason: When returning to time domain after multiplication, depending on the comb’s “tooth width”, a certain percentage of the stochastic signal will, however, be misplaced like echoes in intervals of $n2\pi$ with respect to the phase of the fundamental period (*n* being an integer). For analyses of the generation of noise in bowed strings during steady state, this represents no problem as long as the variation in the noise energy is periodic with period T_0. In cases where it is important to minimize the scattering effect, one might round off the corners of the comb. This will make the echoes fade out more quickly—at the expense of reduced separation. The function shown in the right panel of Figure 3 is the DFT of the comb filter.

(4) Do inverse DFT to obtain the time domain signal of the noise alone (remove imaginary residues and adjust for DC offset). To obtain the deterministic part of the signal, just flip the comb filter.

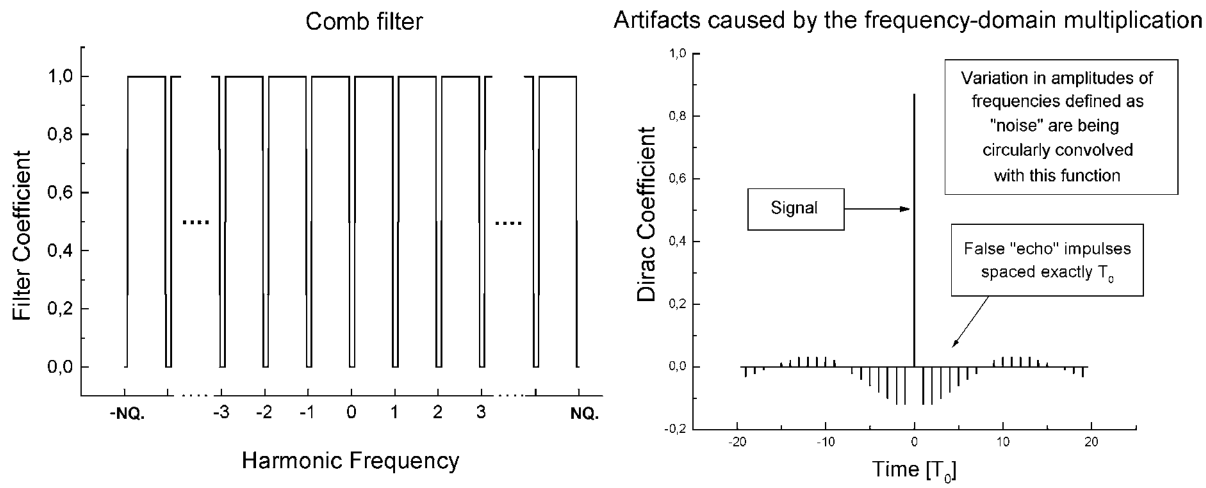

Figure 3: Comb filter and artifacts in the time domain for the stochastic signal. Notice that the Nyquist frequency (NQ) is a multiple of the 1st harmonic frequency. All energy between the comb’s narrow teeth is defined as noise. When returning to the time domain, the stochastic signal will to some extend be “echoed” in intervals of nT_0 (see right panel).

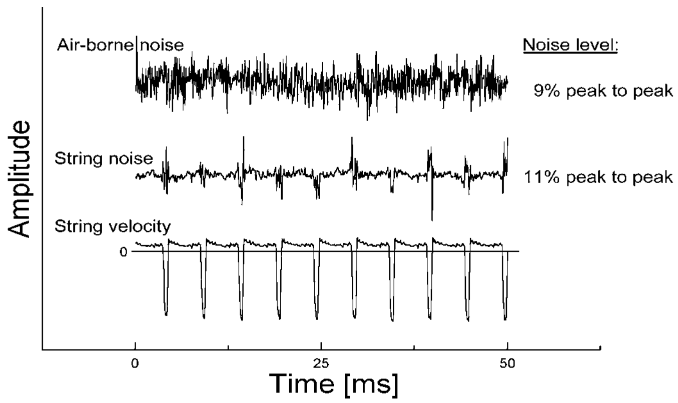

Figure 4: Noise isolated from a violin tone and the string signal in steady-state Helmholtz motion. The plots show that noise—although apparently continuous in the sound pressure—is mainly generated in pulses during the slipping intervals. Some noise, however, is caused by partial slipping across the bow-hair ribbon during stick[8]. Amplitudes up to about 10% of the signal amplitudes are typically seen.

Although this method is designed for analyses of signals in steady state—and only for short windows at a time—relatively good separation of complete, articulated tones can for audio playback purposes be achieved by: (a) dividing the entire original signal by its smoothed RMS curve; (b) performing the procedure described above separately for short pieces with individual determination of pitch, and thus length; (c) splicing the filtered pieces with a fade/overlap technique; (d) multiplying the result by the same RMS curve.

Appendix—The Bluestein filter routine

In order to perform a Fast Fourier Transform of $\mathbf{y}(n)$, where $\mathbf{y}(n)$ contains an arbitrary number of N elements, the following procedure may be followed:

$$\overline{x}_1 = \exp\left(j\frac{i^2\pi}{N}\right) \qquad (\,i = 0,1,2,3,...(2N-1)),\ (j = \sqrt{-1}).$$

$\overline{x}_1$ is then padded with $(M - 2N)$ zeros at the end, to reach a total of M elements, where M is a power of two, and $M > 2N$.

$$\mathbf{X}_1 = \text{FFT}\{\,\overline{x}_1\} \qquad \text{(meaning: } \mathbf{X}_1 = \text{ FFT of } \overline{x}_1).$$

$$\overline{x}_2 = \exp\left(j\frac{k^2\pi}{-N}\right) \qquad (\,k = 0,1,2,3,...(N-1)).$$

$\overline{x}_2$ is then padded with $(M - N)$ zeros at the end, to reach a total of M elements.

$$\overline{x}_3 = \exp\left(j\frac{(k+N)^2\pi}{-N}\right) \qquad (k = (N-1),(N-2),(N-3),...2,1,0).$$

For any given N, the filter values of $\mathbf{X}_1$, $\overline{x}_2$ and $\overline{x}_3$ (or better $\overline{x}_3/M$) may be stored for later use.

Now, $\mathbf{y}(n)$ has to be padded with $(M - N)$ zeros at the end, to reach a total of M elements.

$$\overline{x}_4 = \text{FFT}\left[\mathbf{X}_1 \cdot \text{FFT}\left\{\overline{x}_2 \cdot \mathbf{y}(n)\right\}\right]$$

Then, $\overline{x}_4$ should be cropped to hold only the N elements:

$index = [(M-2N+2),(m-2N+3),...(m-2N+N+1)]$,

after which

$$\mathbf{Y} = \overline{x}_4 \cdot \overline{x}_3/M$$

is calculated.

For a forward Fourier transform, i.e., $\mathbf{Y}(\omega) = \text{FFT}\{\,\mathbf{y}(n)\}$, the elements in $\mathbf{Y}$ has to be permuted, so that $index = [N,1,2,3...,(N-1)]$ to yield the final result.

For a backwards transform, i.e., $\mathbf{y}(n) = \text{FFT}^{-1}\{\,\mathbf{Y}(\omega\,)\}$, the order of elements in $\mathbf{y}$ should become $index = [N,(N-1),(N-2)...1]$, and the result divided by N:

$$\mathbf{y}(n) = \frac{\overline{x}_4 \cdot \overline{x}_3}{NM}.$$

References

[1] Bluestein, L. I., (1968) "A linear filter approach to the computation of the fast Fourier transform" *Nerem Records 1968*, pp. 218-219.

[2] Cooley, J. W. and Tukey, J. W., (1965) "An algorithm for the machine calculation of complex Fourier series"*Mathematics of Computation*, Vol. 19 No. 90, pp 297-301

[3] Serra, X., and Smith, J., (1990) "Spectral modeling synthesis: a sound analysis/synthesis method based on a deterministic plus stochastic decomposition". *Computer Music Journal,* Vol. 14, No. 4.

[4] d'Alessandro, C., Yegnanarayana, B., and Darsinos, V., (1995) "Decomposition of speech signals into deterministic and stochastic components", *Int. Conf. on Acoust., Speech and Sig. Proc., ICASSP'95*, pp 760 - 763.

[5] Woodhouse, J., Schumacher, R. T. and Garoff, S.(2000): "Reconstruction of bowing point friction force in a bowed string", *J. Acoust. Soc. Am.* 108(1), pp. 357-368.

[6] Guettler, K., Schoonderwaldt, E. and Askenfelt, A., (2003): "Bow speed or bowing position—which one influences the spectrum the most?" *Proceedings of the Stockholm Music Acoustics Conference (SMAC'03)* Vol. I, 67-70.

[7] Maher, R. C. and Beauchamp, J. W., (1994): "Fundamental frequency estimation of musical signals using a two-way mismatch procedure", *J. Acoust. Soc. Am.* Vol. 95(4), pp. 2254-2263

[8] Pitteroff, R., and Woodhouse, J., (1998): "Mechanics of the contact area between a violin bow and a string. Part III: Parameter dependence", Acta Acustica, Vol. 84 No. 5, pp.929-946.