OPEN ACCESS Freely available online
V. Välimäki and S. Bilbao,
"Giant FFTs for Sample-Rate Conversion"
J. Audio Eng. Soc., vol. 71, no. 3, pp. 88–99, (2023 March).
DOI: https://doi.org/10.17743/jaes.2022.0061

Giant FFTs for Sample-Rate Conversion

VESA VÄLIMÄKI,[1] *AES Fellow,* **AND STEFAN BILBAO,**[2] *AES Associate Member*
(vesa.valimaki@aalto.fi) (s.bilbao@ed.ac.uk)

[1]*Acoustics Laboratory, Department of Information and Communications Engineering, Aalto University, Espoo, Finland*
[2]*Acoustics and Audio Group, University of Edinburgh, Edinburgh, United Kingdom*

The audio industry uses several sample rates interchangeably, and high-quality sample-rate conversion is crucial. This paper describes a frequency-domain sample-rate conversion method that employs a single large ("giant") fast Fourier transform (FFT). Large FFTs, corresponding to the duration of a track or full-length album, are now extremely fast, with execution times on the order of a few seconds on standard commercially available hardware. The method first transforms the signal into the frequency domain, possibly using zero-padding. The key part of the technique modifies the length of the spectral buffer to change the ratio of the audio content to the Nyquist limit. For up-sampling, an appropriate number of zeros is inserted between the positive and negative frequencies. In down-sampling, the spectrum is truncated. Finally, the inverse FFT synthesizes a time-domain signal at the new sample rate. The proposed method does not result in surviving folded spectral images, which occur in some instances with time-domain methods. However, it causes ringing at the Nyquist limit, which can be suppressed by tapering the spectrum and by low-pass filtering. The proposed sample-rate conversion method is targeted to offline audio applications in which sound files need to be converted between sample rates at high quality.

0 INTRODUCTION

Sample-rate conversion (SRC) is probably more heavily used in audio than in any other field of signal processing [1]. The audio industry commonly uses many different sample rates [2, 3], and it is necessary to be able to switch between them without sacrificing sound quality [4]. This paper describes a high-quality SRC method, which uses the fast Fourier transform (FFT) and inverse FFT (IFFT) algorithms to scale the spectrum of the audio signal, thus allowing an efficient and algorithmically simple conversion between any two sample rates.

The most commonly used sample rate is 44.1 kHz, chosen as the standard for the CD system in the late 1970s [5]. The decision was related to the compatibility with video, which had previously been used for storing digital signals [6, 5]. Later, the 44.1-kHz sample rate was adopted for use in several other consumer audio systems, including Digital Audio Tape players [3], and more recently in music streaming services, such as Spotify [7], Amazon Music [8], Apple Music [9], Deezer [10], Tidal [11], and Qobuz [12].

In professional audio and video, however, 48 kHz is the recommended choice [2]. It was available in DAT players and is also used in the European digital radio standard Digital Audio Broadcasting and on DVD movie soundtracks. Historically, the sample rate of 32 kHz was assigned for broadcasting because it was sufficient for the 15-kHz frequency range of radio broadcasts [2]. Since the 1990s, oversampled rates such as 96 and 192 kHz have become popular in music production [3]. Furthermore, the family of multiples of 44.1 kHz, namely 88.2 and 176.4 kHz, are available in some systems, because they are included in the MPEG-2 audio standard. The highest sample rates used currently for multi-bit audio signals are 352.8 [3] and 384 kHz [13, 14].

SRC is also used as a processing stage within certain audio algorithms, where sound quality can be improved by oversampling a signal prior to nonlinear processing [15, 16]. Furthermore, techniques similar to SRC are required to implement wavetable and sampling synthesis [17–19], the varispeed function [20], and pitch shifting [21], which are all commonly used in music production. These applications require a large variety of arbitrary conversion ratios. However, some such applications require real-time low-latency SRC or time-varying conversion ratios and are not discussed here.

Because of the plurality of sample rates and because practically all music production projects require SRC, such as between 48 kHz or its multiples and the consumer sample rate 44.1 kHz, SRC has become an important research problem [22–27]. Most SRC methods are based on a finite-impulse–response (FIR) filter—either a full-band interpolator for up-sampling or a low-pass interpolator for down-sampling [28, 27, 1]. Conventional SRC methods then produce the output signal at the converted rate by computing the necessary output samples consecutively. Signal processing techniques for SRC include polyphase structures allowing both integer and rational conversion ratios [29–31, 27, 32], the Smith–Gossett algorithm based on a tabulated windowed sinc function [22], and the combination of an integer-factor up-sampler and a low-order variable fractional-delay filter, or Farrow filter [33, 34, 24, 25, 35].

It is well known that the discrete Fourier transform (DFT) ideally interpolates the spectrum of a signal [36]. This has led to the popular zero-padding technique used in connection with the FFT: simply adding zeros at the end of a time-domain signal does not modify the spectrum of the signal but allows for interpolation in the spectral domain [36]. This is a useful and common tool in spectral analysis.

The corresponding property of the DFT that it interpolates any bandlimited signal perfectly in the time domain has remained more obscure in audio, however, although several papers and books hint at applications [37, 38, 36, 39]. When the spectrum of a signal is appended with zeros, the inverse DFT (IDFT) effectively applies the aliased sinc function for time-domain interpolation [37, 36]. Bi and Mitra [40, 41] have suggested using the FFT, frequency-domain zero-padding, and IFFT for SRC either for short sequences or for long samples divided into shorter blocks that could be processed with FFT and IFFT at the time. They also showed that there is a time-domain aliasing error at the beginning and end of the resulting signal [40]. More recently, however, it has become possible to quickly compute very long FFTs containing tens or hundreds of millions of points or more, sufficient to represent many minutes of audio [42].

This paper proposes the giant-FFT method, which employs very large FFT and IFFT lengths for SRC of complete audio files lasting for many minutes. The time-domain aliasing in the interpolation becomes negligible at large IFFT lengths, which implies that the giant FFT is virtually an ideal SRC method. The giant-FFT method is suitable to all integer and rational conversion ratios. It is proposed to convert the whole signal using the FFT at once, which has recently become possible. For example, in MATLAB an FFT of a 50.7-min audio sample at 44.1 kHz, which fits an entire CD album and contains over 134 million samples, takes 3.4 s on an Intel Xeon E3 v5 running on a Lenovo P50.

Note that the giant-FFT technique efficiently implements the limiting case of ideal sinc interpolation. The only downside is a ringing artefact at the Nyquist frequency, which is inaudible and can be suppressed if needed. The frequency-domain tapering or a low-pass filter, which attenuates the ringing, however, causes time-domain smearing, and after that, the method no longer implements perfect sinc interpolation. It is expected that the proposed method will be useful in offline SRC tasks such as mastering audio for different media.

The rest of this paper is organized as follows. SEC. 1 describes the giant-FFT SRC method for both up-sampling and down-sampling, demonstrates its time-domain interpolation capabilities, proposes a frequency-domain tapering method to suppress an artefact at the Nyquist limit, and suggests methods for choosing FFT and IFFT lengths. SEC. 2 compares the method with other techniques and provides examples of applying the proposed method to a critical test signal and a very long musical example. Some concluding remarks appear in SEC. 3.

1 SRC USING THE FFT

This section describes the FFT-based method for interpolating and decimating audio signals. An economic strategy to select the zero-padding factors for both time and frequency buffers is introduced.

For reference, the definition of the DFT and IDFT, relating a time-domain sequence $y[n]$, $n = 0, 1, 2 \ldots, L-1$, and its transform $Y[k]$, $k = 0, 1, 2, \ldots, L-1$, both of length L samples, are provided here:

$$Y[k] = \sum_{n=0}^{L-1} y[n] e^{-2\pi jkn/L}, \tag{1}$$

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} Y[k] e^{2\pi jkn/L}.$$

Here, j is the imaginary unit. Notice that the scaling by $1/L$ is included in the IDFT in Eq. (1), which is the same choice as in MATLAB's FFT and IFFT implementations.

The FFT and IFFT are simply fast algorithms to compute these transformations and were originally proposed for power-of-two values for L [36]. The terms FFT and IFFT are used here to indicate the wider family of methods that yield a fast computation for any integer L (even a prime [43]), with some variation in the resulting efficiency. Currently, even the worst-case choice of prime L is not out of range of audio applications. For the example mentioned above of 3.4-s computation time for 50.7 min of audio at 44.1 kHz, L was chosen to be a power of two: $L = 2^{27} = 134{,}217{,}728$. For a very slightly larger $L = 134{,}217{,}757$ (prime) length transform, computation time is 62.2 s—still not excessive for an album-length SRC task. Such prime-length transforms are easily avoided, as described below.

1.1 Up-Sampling

Up-sampling is the process of increasing the sampling density in an audio signal. Ideally, the spectral content of the signal should be kept unchanged. In practice, the resampled signal is oversampled, since at least a small fraction of its spectrum is empty. In the context of SRC, up-sampling is often called interpolation.

In Fig. 1(b), the basic principle of interpolation using the giant FFT is illustrated, as applied to a real signal of length N samples. (Here and henceforth in this article, for simplicity, N is assumed to be even.) First the FFT is applied to compute the complex spectrum of the original signal, the

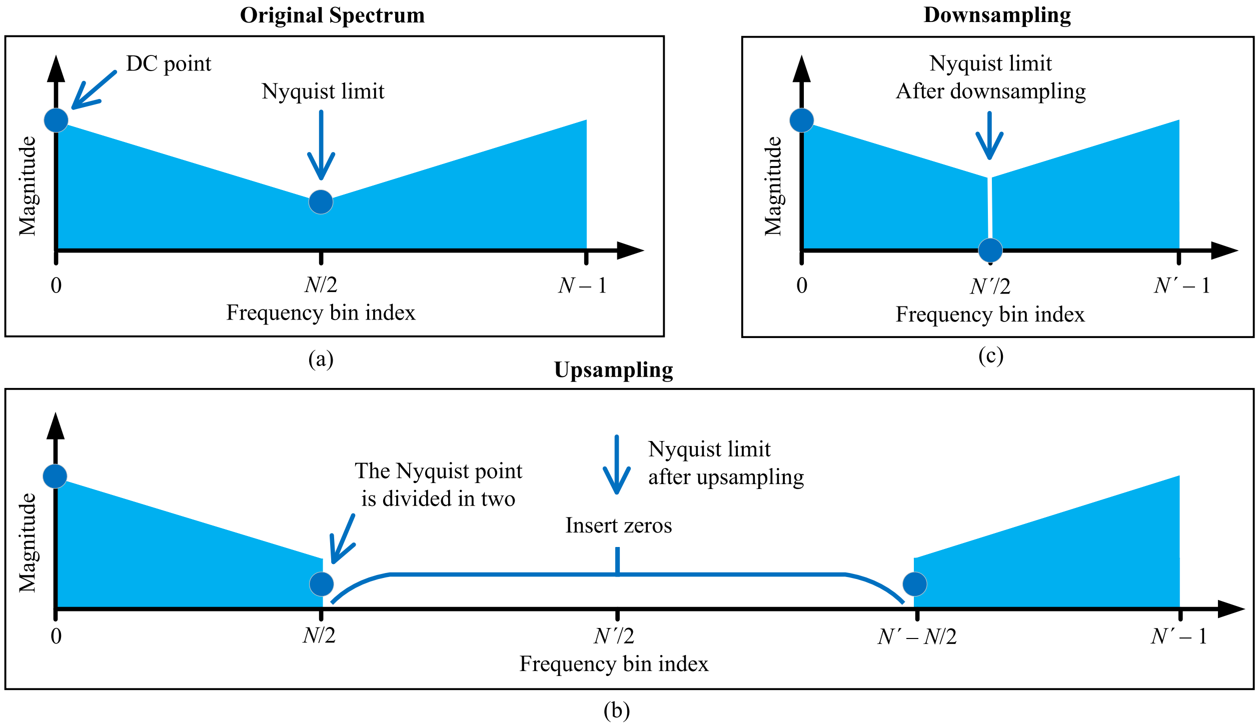


Fig. 1. (a) Original magnitude spectrum of the signal, (b) the principle of up-sampling using the giant-FFT SRC algorithm by zero-padding in the frequency domain, and (c) down-sampling by truncation of the middle part of the spectrum. The handling of the special point at the Nyquist limit is explained in (b) and (c).

magnitude of which is shown in Fig. 1(a). For real signals, all spectral information is included in bins 0 to $N/2$, where bin 0 corresponds to 0 Hz and bin $N/2$ to the Nyquist limit, or half of the sampling frequency. Values in the remaining bins $(N/2) + 1 \ldots N - 1$, which are sometimes called "negative frequencies" or an "image spectrum," are redundant and satisfy Hermitian symmetry.

Frequency-domain zero-padding is illustrated in Fig. 1(b). The modified spectrum, here called X', has length $N' > N$ and retains all information in the original spectrum X, except for at the Nyquist bin. The image spectrum is reproduced at the other end of X'. The data contained at the Nyquist limit, which is real, is divided so that half is inserted at its original locations (bin number $N/2$) and half at the bin location just before the image spectrum, as suggested by Adams [44]. The rest of the buffer is filled with zeros, so that there will be exactly $N' - N - 1$ zeros filling the center part of the buffer.

The construction of the modified spectral buffer $X'(k)$ of length $N' > N$ can be formally expressed as follows:

$$X'(k) = \begin{cases} X(k), & 0 \le k < \frac{N}{2} \\ \frac{1}{2}X(\frac{N}{2}), & k = \frac{N}{2} \\ 0, & \frac{N}{2} < k < N' - \frac{N}{2} \\ \frac{1}{2}X(\frac{N}{2}), & k = N' - \frac{N}{2} \\ X(k - N' + N), & N' - \frac{N}{2} < k < N', \end{cases} \tag{2}$$

where $X(k)$ is the original complex spectrum and k is the spectral bin index. The second and fourth cases above indicate how the spectral value at the Nyquist bin is divided in two. When constructed this way, the modified spectrum $X'(k)$ retains Hermitian symmetry, and thus, after the IFFT is applied, the result is a real-valued signal. This signal should then be scaled by the factor F_s'/F_s, where F_s' and F_s are the new and original sample rates, respectively. The conversion process is zero-phase because the output signal is time-synchronous with the input signal.

In theory, this procedure should be exact in the case of a time-limited and bandlimited input sequence. Although it is not possible to achieve this in infinite precision, this property can effectively hold in finite precision (here, double-precision floating point) for some signal types, such as a Gaussian-windowed sinusoid. See Fig. 2, where Fig. 2(a) shows samples of such a 10-kHz windowed sinusoid. The length of the test signal is 2,205,000 samples, corresponding to 50 s at 44.1 kHz. Consider now up-sampling to 96 kHz. The FFT is applied, and the spectrum is zero-padded to the length of 4,800,000 points, where the ratio $4{,}800{,}000/2{,}205{,}000 = 2.177$ corresponds to that of the new and old sample rates 96.0 and 44.1 kHz. Then, the 4.8-million–point IFFT is executed, which takes about 0.05 s in MATLAB. Fig. 2(b) shows the converted signal samples (dots) and the exact modulated sinusoid (solid line). Fig. 2(c) shows that the approximation error does not exceed 2.8×10^{-14} and that the resampled signal is exact to near machine precision.

Fig. 3(a) shows the magnitude of the FFT spectrum of the test signal, including the image spectrum above the Nyquist limit at 22.05 kHz. Fig. 3(b) shows the magnitude spectrum of the interpolated signal up to the new Nyquist limit of 48 kHz. It can be seen that the spectrum has remained unchanged up to 22.05 kHz. Notably, the spectrum is empty

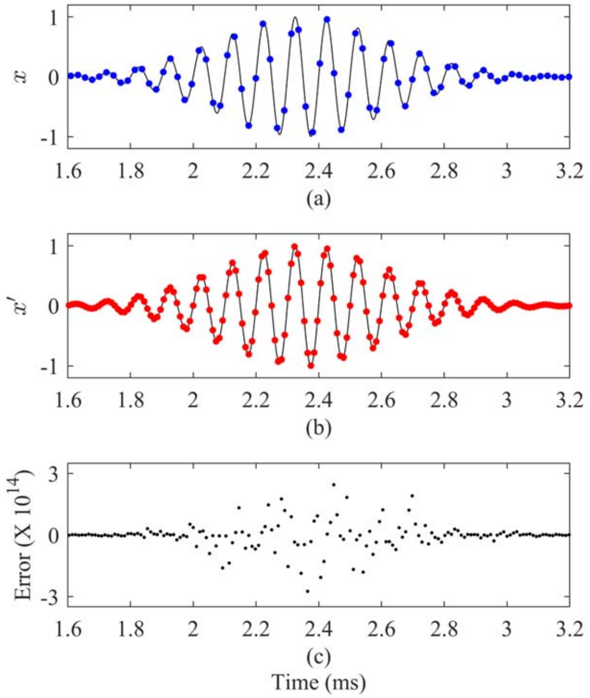


Fig. 2. (a) A Gaussian windowed 10-kHz sinusoid (solid line) sampled at 44.1 kHz (points), (b) the same signal interpolated by factor 2.177 to 96 kHz using the giant FFT (points) and the exact signal (solid line), and (c) the difference of the interpolated and exact signals.

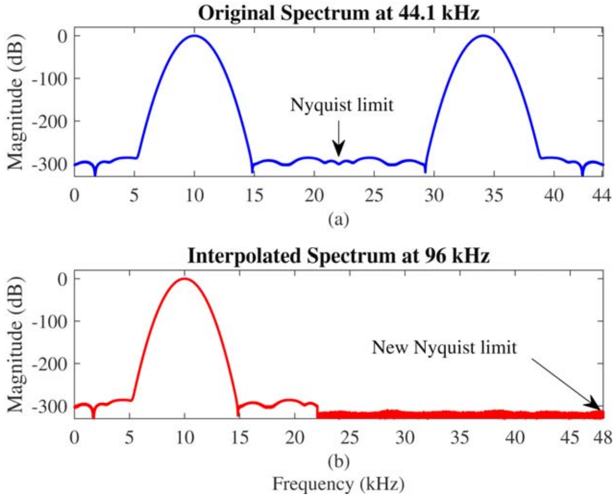


Fig. 3. Magnitude spectra of (a) the windowed 10-kHz sinusoid sampled at 44.1 kHz, showing both positive and negative (above the Nyquist limit) frequencies, and (b) the same signal converted to the sample rate of 96.0 kHz using the 4.8-million–point FFT. Notice the absence of spectral images in (b).

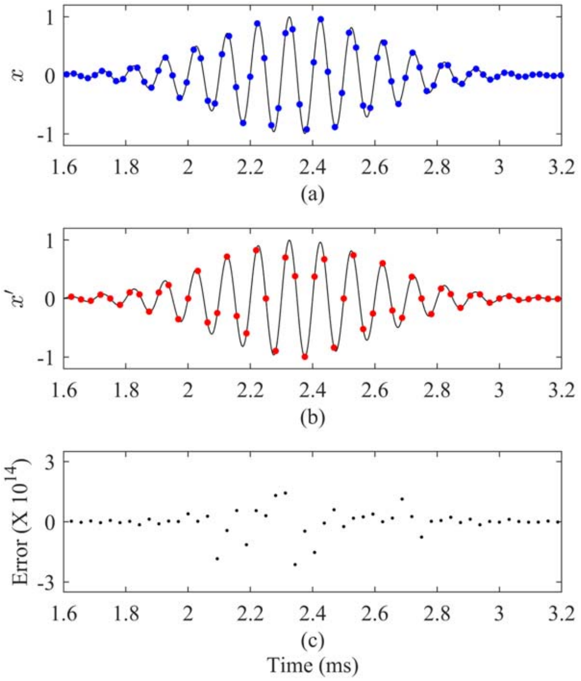


Fig. 4. (a) Gaussian windowed 10-kHz sinusoid (solid line) sampled at 44.1 kHz (points), (b) decimated by factor 0.7256 to 32 kHz using the giant FFT (points) with the exact signal (solid line), and (c) the approximation error.

in the stop-band between 22.05 and 48 kHz, staying 300 dB lower than the peak. The noise floor visible in Fig. 3(b) between 22.05 and 48 kHz is numerical noise typical to the FFT algorithm in MATLAB. It is the convention in hifi audio to require disturbances not to exceed −100 dB, whereas in professional audio, during studio production, the dynamic range should be at least 120 dB for highest-quality work. The giant-FFT method meets these requirements, with operation in double precision.

1.2 Down-Sampling

Down-sampling, or decimation, refers to the lowering of the sample rate, which requires ensuring that the spectrum of the signal does not exceed the new Nyquist limit. In conventional SRC techniques, down-sampling is always paired with an appropriate low-pass filter.

In the giant-FFT method, down-sampling is accomplished by truncating the FFT buffer of the input signal according to the ratio of the output and input sample rates: The lower part of the spectrum up to the new Nyquist limit, together with its image spectrum, are retained, but values in the rest of the bins are discarded, as illustrated in Fig. 1(c). This can be expressed formally as

$$X'(k) = \begin{cases} X(k), & \text{for } 0 \le k < \frac{N'}{2} \\ 0, & \text{for } k = \frac{N'}{2} \\ X(N - N' + k), & \text{for } \frac{N'}{2} < k \le N' - 1, \end{cases} \qquad (3)$$

where it is assumed that $N > N'$. Note that in down-sampling, according to Eq. (3), a zero is inserted at the Nyquist bin $k = N'/2$ in the new spectrum buffer X'.

Fig. 4 shows an example of down-sampling the Gaussian windowed sinusoid from 44.1 kHz to 32 kHz, which is one of the common large rational conversion ratios ($320/441 = 0.7256$). The input signal has been first zero-padded to the length of $N = 2{,}205{,}000$ samples prior to the FFT, which is also the length of the FFT buffer. The FFT spectrum was truncated to the length $N' = 1{,}600{,}000$ points, as described in Eq. (3) and illustrated in Fig. 1(c).

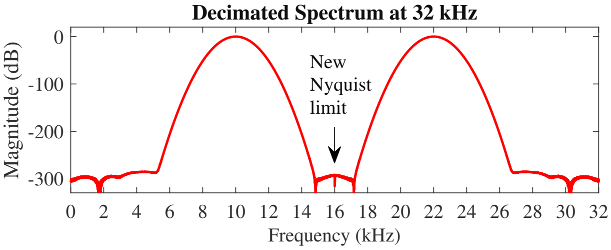


Fig. 5. Magnitude spectrum of the Gaussian windowed 10-kHz sinusoid down-sampled from 44.1 kHz to 32 kHz using the giant FFT, showing both positive and negative frequencies (above 16 kHz).

The ratio of the output and input buffer lengths, 1.6 million and 2.205 million, is equal to the desired conversion ratio, so there is no frequency error. There is no need for any low-pass filter design or execution in the FFT-based decimation, apart from the truncation of the spectral buffer.

The time-domain approximation error shown in Fig. 4(c) is again near machine precision. Fig. 5 shows the spectrum of the converted signal, which may be compared with that of the original presented in Fig. 3(a). It is seen that the spectrum has been cleanly cut at 16 kHz, which is the new Nyquist limit at the sample rate of 32 kHz. There is no visible aliasing or numerical noise in Fig. 5.

1.3 Frequency-Domain Tapering

Although the previous examples show excellent performance, the basic giant-FFT SRC method does not produce acceptable results for full-band audio signals. The reason is that both Eqs. (2) and (3) implement an abrupt truncation of the frequency response at the Nyquist limit, which causes an artefact at that frequency range, whenever the input signal contains energy at the highest frequencies. This section illustrates this problem and offers as a solution a frequency-domain tapering technique, which suppresses the ringing.

A suitable tapering can be obtained with half of the cosine function. A real-valued non-negative weighting sequence $W(k)$ is defined here for a spectrum containing M bins:

$$W(k) = \begin{cases} 1, & 0 \le k, k_\mathrm{c} \\ \dfrac{1 + \cos\left(\frac{\pi}{2}\frac{k - k_\mathrm{c} + 1}{M/2}\right)}{2}, & k_\mathrm{c} \le k \le M/2 \\ \dfrac{1 + \cos\left(\frac{\pi}{2}\frac{k_\mathrm{c} - k + 1}{M/2}\right)}{2}, & M/2, k \le M/2 + k_\mathrm{c} \\ 1, & M/2 + k_\mathrm{c}, k, M, \end{cases} \tag{4}$$

where k_c is the bin index at which the tapering begins (at positive frequencies). The sequence is symmetric with respect to its center value $W(M/2)$, which corresponds to the Nyquist bin. Fig. 6 shows an example of a million-point weighting function $W(k)$, which starts the tapering at 90% of the Nyquist limit, or at bin 450,000. Above the Nyquist bin 500,000 the same behavior is reflected to the image spectrum frequencies. When the weighting function is used, the IFFT is applied to the spectrum $W(k)X'(k)$, which requires point-wise complex multiplication. A clever implementation modifies only the spectral bins for which the weighting function is not 1.

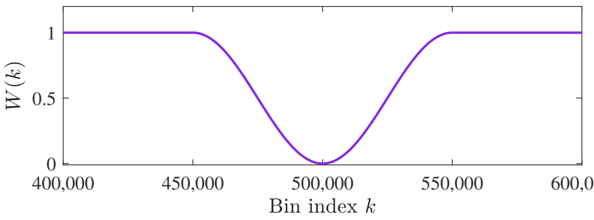


Fig. 6. Frequency-domain weighting function tapering the highest frequencies to zero at the Nyquist bin to suppress the artefact at the Nyquist frequency. Only the center part of the million-point sequence is presented.

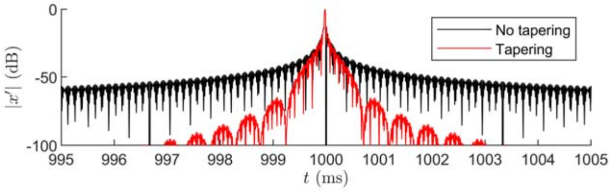


Fig. 7. Impulse, after sample-rate conversion from 44.1 to 96 kHz, both with and without frequency-domain tapering.

As an example, consider the worst case of SRC for an impulse; ideal interpolation naturally produces oscillations that decay gradually in the new sample rate. The proposed tapering in the frequency domain greatly decreases the spread of such oscillations away from the target location of the impulse. In Fig. 7, impulses are plotted on a log scale, after conversion from 44.1 to 96 kHz, both with and without tapering. If it is desired to suppress the ringing more, the converted signal may be low-pass filtered.

1.4 Selecting Input and Output FFT Lengths

Suppose the input signal to be converted consists of N_in samples, at sample rate F_s. In order to sample-rate convert this sequence to sample rate F'_s using a single pair of FFT and IFFT operations, one must first zero pad the original sequence to N samples, with $N \ge N_\mathrm{in}$. Furthermore, the length of the sample-rate–converted sequence will be N'. It is important to find good choices of integers N and N' that minimize zero-padding and thus FFT length.

The sequence lengths N and N' must be in the ratio of the sample rates F'_s and F_s. Suppose that this ratio can be written as

$$\frac{F'_\mathrm{s}}{F_\mathrm{s}} = \frac{P}{Q} \tag{5}$$

for integers P and Q with no common factors. See Table 1 for (P, Q) for commonly encountered audio sample rates.

Now, let

$$N = QM \tag{6}$$

and

$$N' = PM, \tag{7}$$

Table 1. Pairs (P, Q) for sample rate ratios F_s'/F_s as indicated, for input and output sample rates F_s and F_s' in kilohertz.

	Input sample rate F_s				
Output sample rate F_s'	32 kHz	44.1 kHz	48 kHz	96 kHz	192 kHz
32 kHz	···	(320, 441)	(2, 3)	(1, 3)	(1, 6)
44.1 kHz	(441, 320)	···	(147, 160)	(147, 320)	(147, 640)
48 kHz	(3, 2)	(160, 147)	···	(1, 2)	(1, 4)
96 kHz	(3, 1)	(320, 147)	(2, 1)	···	(1, 2)
192 kHz	(6, 1)	(640, 147)	(4, 1)	(2, 1)	···

for some integer M yet to be determined. Note that, given that P/Q is in lowest terms, at least one of P and Q must be odd. This implies that, in order to obtain frame sizes N and N' that are even, M must be chosen to be even. Also, one would like to select M such that N is as close to N_{in} as possible. This implies that

$$M \geq M_{\min} = 2\left\lceil \frac{N_{\text{in}}}{2Q} \right\rceil, \tag{8}$$

where $\lceil \cdot \rceil$ indicates a ceiling operation. Given that M appears as a factor in both N and N', for the sake of performance, it may be desirable to choose M even such that $M > M_{\min}$ if a suitable M can be found that is composite and ideally factorizable into small prime numbers. The precise question of how to choose M subject to such transform efficiency issues will not be discussed further here.

As an example, consider an initial set of $N_{\text{in}} = 600{,}000$ samples at $F_s = 44.1$ kHz, which corresponds to 13.6 s and where the new sample rate is $F_s' = 48$ kHz. In this case, $Q = 147$ according to Table 1, and $M_{\min} = 4{,}082$, yielding according to Eq. (6) a frame size $N = QM = 600{,}054$ that is only slightly larger than N_{in}. Unfortunately, 4,082 possesses the relatively large factor of 157. A better choice for M might then be $M = 4{,}096$, which is a power of two, and yields a slightly larger frame size of $N = QM = 602{,}112 = 2^{12} \times 3 \times 7^2$. In practice, this means that the input sequence must be padded with 2,112 zeros before the FFT. For N' then, using Eq. (7) and $P = 160$, which can be read from Table 1, the value $N' = PM = 160 \times 4{,}096 = 655{,}360$ is obtained. This means that $N' - N = 53{,}248$ zeros must be inserted in the FFT buffer prior to the IFFT to achieve the desired SRC from 44.1 to 48 kHz.

Finally, the extra zeros can be truncated from the end of the output sequence. The output signal length N_{out} in samples, at the new sample rate, is

$$N_{\text{out}} = \left\lceil \frac{F_s'}{F_s} N_{\text{in}} \right\rceil. \tag{9}$$

This implies that there are $N' - N_{\text{out}}$ excessive zeros in the IFFT result x' that must be truncated. In the example case above, 655,360 – 653,062 = 2,298 zero samples are discarded from the end of the signal to reach the correct length. Notice that in the case above (and in most practical cases, including those common choices of sample rates in Table 1), the amount of zero-padding required in the time domain to obtain N samples from N_{in} samples, is very small—here approximately 0.35%.

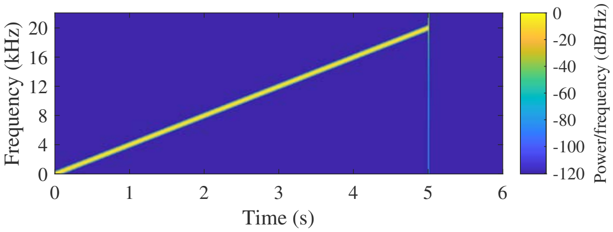


Fig. 8. Spectrogram of a 5-s linear chirp at 44.1 kHz, where 1,024-point Chebyshev windows with a 130-dB stop-band rejection and 512-point overlap were used.

1.5 General Algorithm Summary

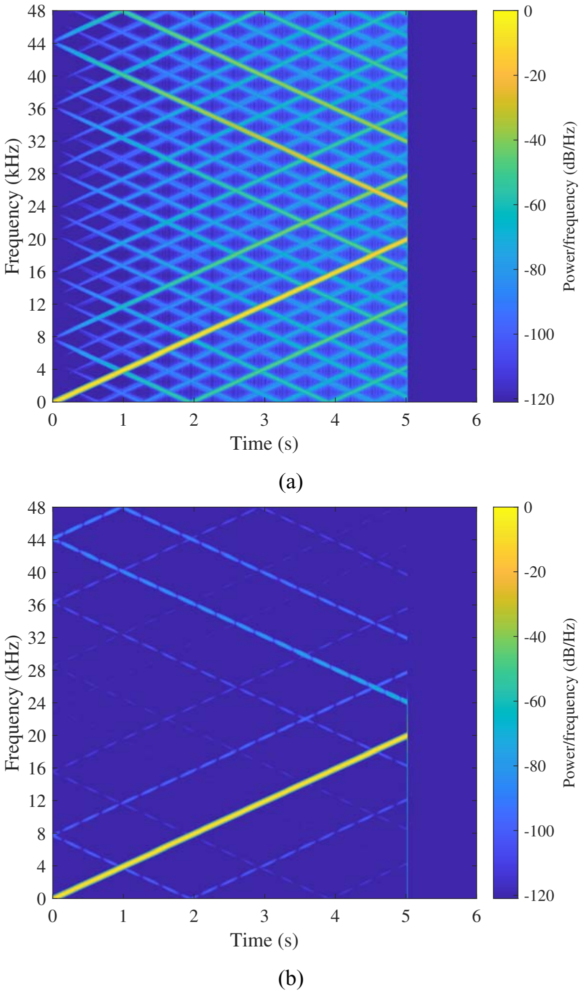
A general form of the FFT-based SRC algorithm, which applies to both up-sampling and down-sampling cases, is described next. Given input sequence x_{in} (length N_{in} samples) at original sample rate F_s, which is to be converted to the new sample rate F_s', do the following steps:

1. Choose sequence lengths N and N', see Sec. 1.4;
2. Zero-pad x_{in} to obtain x of length $N \geq N_{\text{in}}$ samples;
3. Perform FFT to obtain X with N bins;
4. Apply frequency-domain tapering with W, if desired;
5. Zero-pad [Eq. (2)] or truncate [Eq. (3)] spectrum X symmetrically to obtain X' with $N' = N F_s'/F_s$ bins;
6. Perform IFFT to obtain x' of length N' samples;
7. Scale x' by a factor P/Q;
8. Remove additional zeros to obtain output signal x_{out}, see Sec. 1.4; and
9. Apply a low-pass filter at the Nyquist limit, if desired.

2 EVALUATION AND COMPARISON

This section shows how the proposed method compares with traditional SRC techniques, which process the input signal in the time domain using an FIR filter. The effectiveness of the frequency-domain tapering technique is also demonstrated in both up-sampling and down-sampling. Finally, a very long music signal is first up-sampled and then down-sampled back to the original sample rate, to analyze the conversion error.

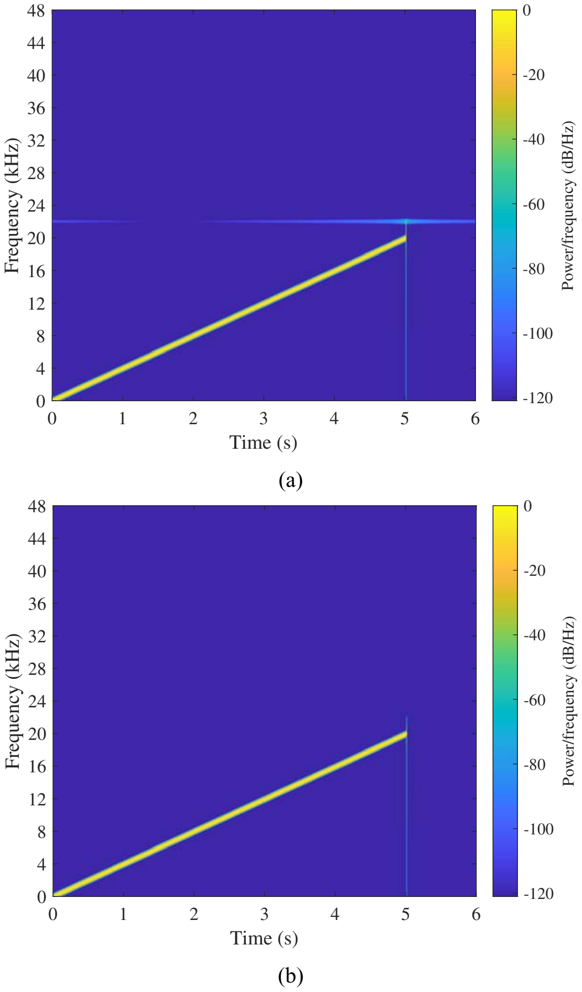
A linear chirp running from 0 Hz to 20 kHz in 5 s is used as the test signal. It is generated at the sample rate of 44.1 kHz. One second of silence (zero samples) is inserted at the end of the test sound file. Fig. 8 shows its spectrogram



(a)

(b)

Fig. 9. For comparison, SRC from 44.1 to 96 kHz using (a) linear interpolation, showing heavy imaging, and (b) a polyphase FIR filter having 12,801 coefficients, which performs better.



(a)

(b)

Fig. 10. Spectrograms of the chirp after converting from 44.1 to 96 kHz using the proposed method (a) without and (b) with the proposed frequency-domain tapering above 20 kHz, which suppresses the ringing at the original Nyquist frequency associated with up-sampling.

containing a single spectral component. Notice the faint vertical line at 5.0 s, which is the transient caused by the abrupt ending of the chirp.

2.1 Up-Sampling Example and Comparison

The test signal is converted to the sample rate of 96 kHz using three different methods. First, Fig. 9(a) shows the result of the conversion using the simplest method, linear interpolation, which resamples the signal using a two-tap FIR filter [45]. Fig. 9(a) shows many extra spectral components, which are images. Some of the images are attenuated by only about 40 dB, which means that the result is badly corrupted and useless for high-quality audio work.

Fig. 9(b) shows the spectrogram after the signal has been converted using a high-order polyphase FIR filter, which is a better method than linear interpolation (the resample function of MATLAB was used). The polyphase filter requires only 20 FIR filter coefficients to be applied per output sample, while giving identical results as if a polyphase low-pass FIR filter of length 12,801 was used in a direct implementation. The cutoff frequency of the filter is 21,920 Hz, and it attenuates the images much better than linear interpolation. The loudest image component at about 24 kHz remains below −65 dB. At lower (audio) frequencies, the images remain at −98 dB or lower, which suggests that the result is close to being acceptable for hifi use.

Fig. 10(a) shows the result of using the proposed method. The original chirp contains 264,600 samples and is first zero-padded to $336 \times 882 = 296{,}352$ samples, and the FFT is applied. In the frequency domain, the spectrum is zero-padded to the length of $731 \times 882 = 645{,}120$ points, after which the IFFT is used to obtain the output signal. The extra samples are discarded. The spectrogram is generally very clean, but a horizontal artefact, or ringing, at the original Nyquist limit 22 kHz is observed, which originates from the abrupt truncation of the spectrum through zero-padding.

Fig. 10(b) demonstrates how the frequency-domain tapering proposed in SEC. 1.3 helps reduce the artefact at the original Nyquist frequency. A cosine weighting function is used to taper the spectrum above 19,845 Hz, which is outside the audio range. As a result, the artefact at 22 kHz has been suppressed in Fig. 10(b). A click, which can be seen

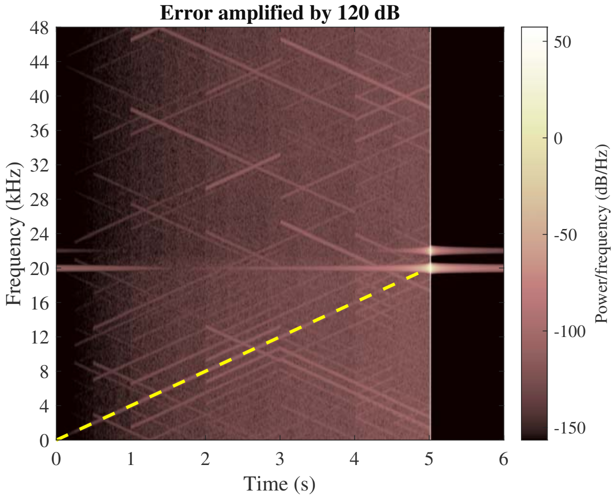
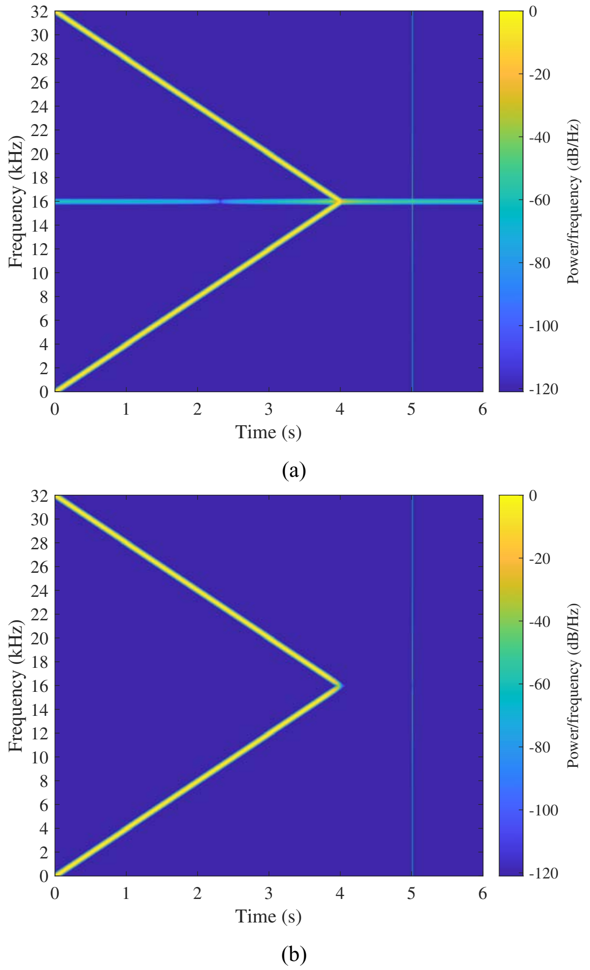


Fig. 11. Spectrogram of the error in the converted chirp at 96 kHz, for which 1,024-point Chebyshev windows with a 200-dB stopband rejection and 512-point overlap were applied. The dashed line indicates the location of the chirp that has been canceled, cf. Fig. 10(b). Note the scaling is different than that in the other spectrograms.



(a)

(b)

Fig. 12. Spectrograms of the chirp at 32 kHz decimated with the giant-FFT technique (a) without and (b) with the proposed frequency-domain tapering. Here both the baseband and image spectrogram are presented to show the ringing at the new Nyquist frequency 16 kHz and how it is suppressed using tapering prior to down-sampling.

as a faint vertical line at 5.0 s, is still visible but is part of the ideal test signal (see Fig. 8).

Because the frequency-domain SRC method is a zero-phase technique, in the case of the synthetic test signal, it is possible to extract the conversion error. A copy of the linear chirp is generated at the 96-kHz sample rate and is subtracted from the converted signal, which corresponds to Fig. 10(b). Because the error signal is very faint, it has been amplified by factor 1,000,000 or by 120 dB prior to computing its spectrogram shown in Fig. 11. It can be seen that the spectrogram contains some similar diagonal patterns to those seen in Figs. 9(a) and 9(b), but they are fragmented and very faint, appearing over 200 dB below the peak signal level. Furthermore, a soft ringing appears at about 20 kHz, which is where the chirp ends. This test verifies that the imaging and rounding errors remain sufficiently small for hifi audio applications.

2.2 Down-Sampling Example

Fig. 12 gives an example of down-sampling the 5-s chirp from 44.1 to 32 kHz. This time the chirp is cut at 4.0 s, as it exits the new frequency range, see Fig. 12(a). Again, without the frequency-domain weighting function, an artefact is born at the Nyquist frequency, which is now 16 kHz. The image spectrum is included to better show the ringing. Fig. 12(b) reveals that the weighting function tapering the spectrum above 15 kHz, which is 93.75% of 16 kHz, deletes the artefact. Notice a faint click at 5.0 s, which comes from the discontinuity at the end of the original chirp and is also visible in Fig. 8.

2.3 Testing With a Long Music Signal

Finally, the proposed FFT-based SRC method is tested with a musical signal. As the test signal, "Tom's Diner" by Suzanne Vega was chosen. The duration of the original stereo recording at the sample rate of 44.1 kHz is 2 min 11 s and contains 5,776,512 samples per channel. To study the performance of the proposed method in the case of a lengthy recording, such as a full music album, the same song is concatenated 24 times, yielding a long signal of 52 min and 24 s, or 138,636,288 samples per stereo channel.

The long test signal is first converted from 44.1 to 48 kHz using $P = 160$, $Q = 147$, and $M = 943{,}104$, which produces an interpolated signal of 150,896,640 samples per channel. To allow a direct comparison with the original, this signal is converted back to 44.1 kHz using $P = 147$, $Q = 160$, and $M = 943{,}104$ (same M as above). The resulting signal has 138,636,288 samples, the same as the long test signal before any conversion. The same processing is applied to both stereo channels separately. The conversion to a higher sample rate and back gives access to an error signal by simply subtracting the twice-converted signal from the original one.

Fig. 13 shows the envelope of the long test signal (maximum value, −11.8 dB) and the conversion error (maximum

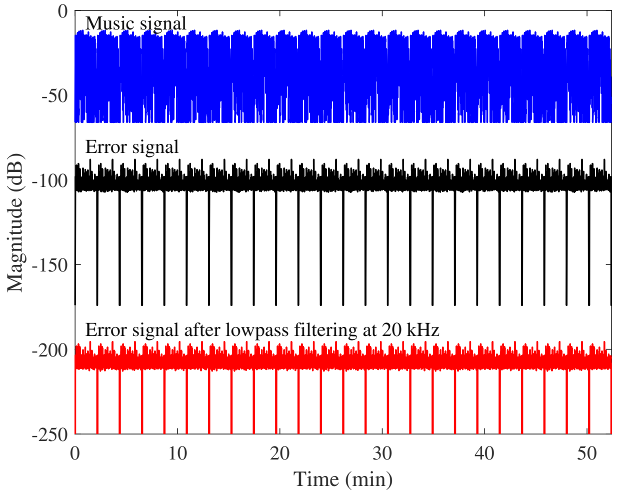


Fig. 13. Temporal envelopes of the 52-min—long music signal and the error after converting it from 44.1 to 48 kHz and back. The bottom curve is the error signal low-pass filtered at 20 kHz to reduce the ringing artefact near the Nyquist limit.

value, −88.0). Unfortunately, the error exceeds the −120-dB and −100-dB limits. Further analysis shows that the peak error occurs near the original Nyquist limit of 22.05 kHz, where it does not affect the sound quality. For this reason, the original and converted signals are low-pass filtered with a linear-phase FIR of length 215 coefficients to remove the inaudible error. The filter's cutoff frequency (−6-dB point) is set at 20.2 kHz and the stop-band ripple is −100 dB at frequencies above 20.7 kHz. The conversion error after the filtering operation, which is shown as the bottom curve in Fig. 13, now remains below −195 dB in the audio band, which is sufficient for hifi use.

This test demonstrates that the proposed giant-FFT SRC method does not suffer from accumulating numerical errors even when extremely long signals are processed—not at least when double-floating-point computing is in use. However, the test also suggests that the frequency-domain tapering by itself may be insufficient in suppressing the high-frequency ringing for the most critical cases, and additional postprocessing using a low-pass filter (or a Nyquist notch filter) in the time domain may be worthwhile.

3 CONCLUSION

An FFT-based technique is proposed for converting audio signals between arbitrary sample rates without notable quality degradation. The method performs the rate conversion without explicitly having to design or apply any interpolation or decimation filters, which have been traditionally used for implementing SRC. Furthermore, the method is conceptually simpler than SRC systems based on interpolating filters, because there is no need to keep track of the exact time locations of output samples.

The key idea in the described method is to employ the theoretically perfect time-domain interpolation capability of the Fourier transform for bandlimited signals. This is achieved by zero-padding or truncating the FFT spectrum of the signal, which essentially changes the ratio of the spectral content and Nyquist limit. The IFFT yields all converted signal samples at once. The method is called the giant FFT, because the whole signal is processed in one go using a single large FFT. This requires the use of very long FFTs, typically millions of points for audio signals that last for minutes, which are still quick to compute.

The giant-FFT method has been compared with conventional SRC techniques. It has been shown that the resulting spectrum of the converted signal is notably free of spectral imaging, which is a nuisance in time-domain SRC methods. However, in the FFT-based method, a ringing artefact appears near the original or new Nyquist limit and may have to be suppressed. This paper suggests a frequency-domain tapering toward the Nyquist limit to significantly attenuate the artefact. Nevertheless, a test case shows that for best quality, it may be necessary to use a low-pass or Nyquist notch filter to further reduce the high-frequency ringing.

It remains to be investigated how the sequence of the very long FFT and IFFT operations together keep the numerical precision high. The numerical error grows with FFT size—a feature that has seen substantial investigation [46, 47]. As double-precision floating-point precision is not always available, it would be of interest to check whether the proposed method provides good results using a single-precision floating-point number system. Future work also includes a formal listening test with experienced listeners, such as mastering engineers, to verify that the sound quality is indeed preserved using the proposed SRC method. Various critical test signals, such as nonstationary and noisy sounds and speech should be included in such a test. It is expected that the giant-FFT method described in this paper will be useful in many audio and music applications, where sound files need to be converted between sample rates at high quality.

4 ACKNOWLEDGMENT

The main part of this work was conducted during a research visit of the second author to the Aalto Acoustics Lab between May 30 and June 13, 2022. This research belongs to the activities of the Nordic Sound and Music Computing Network—NordicSMC (NordForsk project no. 86892). For the purpose of open access, the second author has applied a creative commons attribution (CC BY) license to any author-accepted manuscript version arising. The authors would like to thank the Associate Technical Editor for noticing the surprising mismatch between the numerical precisions of the FFT spectrum and the time-domain signal reconstructed using the IFFT. Special thanks go to the anonymous reviewer who suggested testing the method by converting a signal to another sample rate and back.

5 REFERENCES

[1] A. Zeineddine, A. Nafkha, S. Paquelet, C. Moy, and P. Y. Jezequel, "Comprehensive Survey of FIR-Based Sample Rate Conversion," *J. Signal*

Process. Syst., vol. 93, pp. 113–125 (2021 Jul.). https://doi.org/10.1007/s11265-020-01575-6.

[2] AES, "AES Recommended Practice for Professional Digital Audio – Preferred Sampling Frequencies for Applications Employing Pulse-Code Modulation," *AES Standard AES5-2018* (2018 Dec.).

[3] V. R. Melchior, "High-Resolution Audio: A History and Perspective," *J. Audio Eng. Soc.*, vol. 67, no. 5, pp. 246–257 (2019 May). https://doi.org/10.17743/jaes.2018.0056.

[4] U. Zölzer, *Digital Audio Signal Processing* (Wiley, Chichester, UK, 2008) 2nd ed.

[5] K. A. Immink, "The Compact Disc Story," *J. Audio Eng. Soc.*, vol. 46, no. 5, pp. 458–460, 462, 464, 465 (1998 May).

[6] T. T. Doi, Y. Tsuchiya, and A. Iga, "On Several Standards for Converting PCM Signals Into Video Signals," *J. Audio Eng. Soc.*, vol. 26, no. 9, pp. 641–649 (1978 Sep.).

[7] Spotify AB, "FAQs: Are There Tracks That Have an SpSampleFormat Different From 44kHz Stereo?" https://developer.spotify.com/documentation/commercial-hardware/implementation/faqs/ (accessed Jun. 6, 2022).

[8] Amazon, "Amazon Music Unlimited in HD FAQ," https://www.amazon.co.uk/b?node=3022219031 (accessed Sep. 11, 2022).

[9] Apple, "About Lossless Audio in Apple Music," https://support.apple.com/en-us/HT212183 (accessed Sep. 11, 2022).

[10] Deezer, "High Fidelity (HiFi)," https://support.deezer.com/hc/en-gb/articles/115004588345-Deezer-HiFi (accessed Sep. 11, 2022).

[11] Tidal, "Clearly the Best Sound," https://tidal.com/sound-quality (accessed Nov. 8, 2022).

[12] Qobuz, "What Is in the Streaming Catalogue?" https://help.qobuz.com/hc/en-us/articles/360026640751 (accessed Sep. 11, 2022).

[13] A. P. Hill, P. Prince, E. P. Covarrubias, et al., "AudioMoth: Evaluation of a Smart Open Acoustic Device for Monitoring Biodiversity and the Environment," *Methods Ecol. Evol.*, vol. 9, no. 5, pp. 1199–1211 (2017 Dec.). https://doi.org/10.1111/2041-210X.12955.

[14] K. Kaleris, B. Stelzner, P. Hatziantoniou, D. Trimis, and J. Mourjopoulos, "Laser-Sound Transduction From Digital $\Sigma\Delta$ Streams," *J. Audio Eng. Soc.*, vol. 70, no. 1/2, pp. 50–61 (2022 Jan.). https://doi.org/10.17743/jaes.2021.0053.

[15] J. Kahles, F. Esqueda, and V. Välimäki, "Oversampling for Nonlinear Waveshaping: Choosing the Right Filters," *J. Audio Eng. Soc.*, vol. 67, no. 6, pp. 440–449 (2019 Jun.). https://doi.org/10.17743/jaes.2019.0012.

[16] D. Albertini, A. Bernardini, and A. Sarti, "Antiderivative Antialiasing Techniques in Nonlinear Wave Digital Structures," *J. Audio Eng. Soc.*, vol. 69, no. 7/8, pp. 448–464 (2021 Jul.). https://doi.org/10.17743/jaes.2021.0017.

[17] D. Rossum, "Constraint Based Audio Interpolators," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 161–164 (New Paltz, NY) (1993 Oct.). https://doi.org/10.1109/ASPAA.1993.379972.

[18] R. C. Maher, "Wavetable Synthesis Strategies for Mobile Devices," *J. Audio Eng. Soc.*, vol. 53, no. 3, pp. 205–212 (2005 Mar.).

[19] A. Franck and V. Välimäki, "Higher-Order Integrated Wavetable and Sampling Synthesis," *J. Audio Eng. Soc.*, vol. 61, no. 9, pp. 624–636 (2013 Sep.).

[20] P. S. Gaskell, "A Hybrid Approach to the Variable-Speed Replay of Digital Audio," *J. Audio Eng. Soc.*, vol. 35, no. 4, pp. 230–238 (1987 Apr.).

[21] J. Driedger and M. Müller, "A Review of Time-Scale Modification of Music Signals," *Appl. Sci.*, vol. 6, no. 2, paper 57 (2016 Feb.). https://doi.org/10.3390/app6020057.

[22] J. Smith and P. Gossett, "A Flexible Sampling-Rate Conversion Method," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 112–115 (San Diego, CA) (1984 Mar.). https://doi.org/10.1109/ICASSP.1984.1172555.

[23] R. Adams and T. Kwan, "A Stereo Asynchronous Digital Sample-Rate Converter for Digital Audio," *IEEE J. Solid-State Circ.*, vol. 29, no. 4, pp. 481–488 (1994 Apr.). https://doi.org/10.1109/4.280698.

[24] K. Rajamani, Y.-S. Lai, and C. W. Farrow, "An Efficient Algorithm for Sample Rate Conversion From CD to DAT," *IEEE Signal Process. Lett.*, vol. 7, no. 10, pp. 288–290 (2000 Oct.). https://doi.org/10.1109/97.870683.

[25] K.-J. Cho, J.-S. Park, B.-K. Kim, J.-G. Chung, and K. K. Parhi, "Design of a Sample-Rate Converter From CD to DAT Using Fractional Delay Allpass Filter," *IEEE Trans. Circ. Syst. II: Express Briefs*, vol. 54, no. 1, pp. 19–23 (2007 Jan.). https://doi.org/10.1109/TCSII.2006.885067.

[26] A. Franck, "Arbitrary Sample Rate Conversion With Resampling Filters Optimized for Combination With Oversampling," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 149–152 (New Paltz, NY) (2011 Oct.). https://doi.org/10.1109/ASPAA.2011.6082271.

[27] A. Kumar, S. Yadav, and N. Purohit, "Generalized Polyphase Multistep FIR Structures: Modular Realization of Polyphase Filters," in *Proceedings of the 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 537–541 (Noida, India) (2020 Feb.). https://doi.org/10.1109/SPIN48934.2020.9071059.

[28] A. Kumar, S. Yadav, and N. Purohit, "Exploiting Coefficient Symmetry in Conventional Polyphase FIR Filters," *IEEE Access*, vol. 7, pp. 162883–162897 (2019 Nov.). https://doi.org/10.1109/ACCESS.2019.2951706.

[29] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter Banks," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 24, no. 2, pp. 109–114 (1976 Apr.). https://doi.org/10.1109/TASSP.1976.1162788.

[30] V. Melchior, "Multiphase Filters for Sample Rate Conversion of High Resolution Audio," presented at the *105th Convention of the Audio Engineering Society* (1998 Sep.), paper 4854.

[31] S. Tassart, "Time-Invariant Context for Sample Rate Conversion Systems," *IEEE Trans. Signal*

Process., vol. 60, no. 3, pp. 1098–1107 (2012 Mar.). https://doi.org/10.1109/TSP.2011.2176336.

[32] F. Harris, "Polyphase Interpolators With Reversed Order of Up-Sampling and Down-Sampling," in *Proceedings of the 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 918–924 (Pacific Grove, CA) (2021 Oct.). https://doi.org/10.1109/IEEECONF53345.2021.9723220.

[33] T. Ramstad, "Digital Methods for Conversion Between Arbitrary Sampling Frequencies," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 3, pp. 577–591 (1984 Jun.). https://doi.org/10.1109/TASSP.1984.1164362.

[34] T. Saramäki and T. Ritoniemi, "An Efficient Approach for Conversion Between Arbitrary Sampling Frequencies," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2, pp. 285–288 (Atlanta, GA) (1996 May). https://doi.org/10.1109/ISCAS.1996.541702.

[35] A. Chinaev, P. Thüne, and G. Enzner, "Low-Rate Farrow Structure With Discrete-Lowpass and Polynomial Support for Audio Resampling," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, pp. 475–479 (Rome, Italy) (2018 Sep.). https://doi.org/10.23919/EUSIPCO.2018.8553469.

[36] J.O. Smith III, *Mathematics of the Discrete Fourier Transform (DFT) With Audio Applications* (Palo Alto, CA) (W3K Publishing, 2007), 2nd ed.

[37] T. J. Cavicchi, "DFT Time-Domain Interpolation," in *Digital Signal Processing*, pp. 441–454 (Wiley, New York, NY, 2000).

[38] D. Fraser, "Interpolation by the FFT Revisited—An Experimental Investigation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 5, pp. 665–675 (1989 May). https://doi.org/10.1109/29.17559.

[39] R. G. Lyons, *Understanding Digital Signal Processing* (Pearson Education, Boston, MA, 2011), 3rd ed.

[40] G. Bi and S. K. Mitra, "Sampling Rate Conversion in the Frequency Domain [DSP Tips and Tricks]," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 140–144 (2011 May). https://doi.org/10.1109/MSP.2011.940413.

[41] G. Bi and S. K. Mitra, "FFT-Based Sampling Rate Conversion," in *Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 428–431 (2012 Jul.). https://doi.org/10.1109/ICIEA.2012.6360765.

[42] V. Välimäki, J. Rämö, and F. Esqueda, "Creating Endless Sounds," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx)*, pp. 32–39 (Aveiro, Portugal) (2018 Sep.).

[43] C. M. Rader, "Discrete Fourier Transforms When the Number of Data Samples is Prime," *Proc. IEEE*, vol. 56, no. 6, pp. 1107–1108 (1968 Jun.). https://doi.org/10.1109/PROC.1968.6477.

[44] J. Adams, "A Subsequence Approach to Interpolation Using the FFT," *IEEE Trans. Circ. Syst.*, vol. 34, no. 5, pp. 568–570 (1987 May). https://doi.org/10.1109/TCS.1987.1086169.

[45] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay [FIR/All Pass Filters Design]," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60 (1996 Jan.). https://doi.org/10.1109/79.482137.

[46] G. U. Ramos, "Roundoff Error Analysis of the Fast Fourier Transform," *Math. Comp.*, vol. 25, no. 116, pp. 757–768 (1971 Oct.). https://doi.org/10.2307/2004342.

[47] T. Thong and B. Liu, "Accumulation of Roundoff Errors in Floating Point FFT," *IEEE Trans. Circ. Syst.*, vol. 24, no. 3, pp. 132–143 (1977 Mar.). https://doi.org/10.1109/TCS.1977.1084316.

THE AUTHORS

Vesa Välimäki

Stefan Bilbao

Vesa Välimäki is Full Professor of Audio Signal Processing and Vice Dean for Research at Aalto University, Espoo, Finland. He received his M.Sc. and D.Sc. degrees from the Helsinki University of Technology in 1992 and 1995, respectively. In 1996, he was a Postdoctoral Research Fellow at the University of Westminster, London, UK. In 2001–2002, he was Professor of Signal Processing at the Pori unit of Tampere University of Technology. In 2008–2009, he was a visiting scholar at the Stanford University Center for Computer Research in Music and Acoustics (CCRMA). He is a Fellow of the AES and a Fellow of the IEEE. He was the General Chair of the 11th International Conference on Digital Audio Effects (DAFx) in 2008 and of the 14th International Sound and Music Computing Conference (SMC) in 2017. Prof. Välimäki is the Editor-in-Chief of the *Journal of the Audio Engineering Society*.

●

Stefan Bilbao (B.A. Physics, Harvard, 1992; M.Sc. and Ph.D. Electrical Engineering, Stanford, 1996 and 2001, respectively) is currently Professor of Acoustics and Audio Signal Processing in the Acoustics and Audio Group at the University of Edinburgh, and he previously held positions at the Sonic Arts Research Centre, at the Queen's University Belfast, and the Stanford Space Telecommunications and Radioscience Laboratory. He has been the Principal Investigator of two ERC-funded projects concerned with audio synthesis and virtual acoustics: the NESS project (ERC-2011-StG-279068-NESS) and the WRAM project (ERC-2016-PoC-737574-WRAM). He is an Associate Editor of the *IEEE/ACM Transactions on Audio, Speech, and Language Processing* and was the General Chair of the 20th International Conference on Digital Audio Effects (DAFx) held in Edinburgh in 2017. He was born in Montreal, Quebec, Canada.