ZERO-PHASE SOUND VIA GIANT FFT

Vesa Välimäki, Roope Salmi

Acoustics Lab, Dept. of Information & Communications Engr.
Aalto University
Espoo, Finland
vesa.valimaki@aalto.fi, rpsalmi@gmail.com

Stefan Bilbao

Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

Sebastian J. Schlecht

Multimedia Communications and Signal Processing
Friedrich-Alexander-Universität Erlangen-Nürnberg
Erlangen, Germany
sebastian.schlecht@fau.de

David Zicarelli

Cycling '74
Covina, CA, USA
zicarell@cycling74.com

ABSTRACT

Given the speedy computation of the FFT in current computer hardware, there are new possibilities for examining transformations for very long sounds. A zero-phase version of any audio signal can be obtained by zeroing the phase angle of its complex spectrum and taking the inverse FFT. This paper recommends additional processing steps, including zero-padding, transient suppression at the signal's start and end, and gain compensation, to enhance the resulting sound quality. As a result, a sound with the same spectral characteristics as the original one, but with different temporal events, is obtained. Repeating rhythm patterns are retained, however. Zero-phase sounds are palindromic in the sense that they are symmetric in time. A comparison of the zero-phase conversion to the autocorrelation function helps to understand its properties, such as why the rhythm of the original sound is emphasized. It is also argued that the zero-phase signal has the same autocorrelation function as the original sound. One exciting variation of the method is to apply the method separately to the real and imaginary parts of the spectrum to produce a stereo effect. A frame-based technique enables the use of the zero-phase conversion in real-time audio processing. The zero-phase conversion is another member of the giant FFT toolset, allowing the modification of sampled sounds, such as drum loops or entire songs.

1. INTRODUCTION

Fast Fourier transforms (FFTs) of very long sequences are nowadays very quickly computed, and with the vast memory capacity of modern computers, it is possible to transform whole songs or even albums of music in one pass [1]. This motivates investigations into frequency-domain audio processing techniques for very long signals, including this study. This paper focuses on the FFT-based zero-phase conversion technique to modify audio signals.

Oppenheim and Lim [2] discussed the importance of phase in image and speech signals. They modified the complex spectrum of a speech signal and observed that intelligibility was lost when the phase was zeroed before the inverse Fourier transform. Julius Smith considered the zero-phase signal when discussing the symmetry and delay in the context of the discrete Fourier transform (DFT) (see Chapters 7.4.3-7.4.4 in [3]). The conversion of short signal frames to the zero-phase form has been applied to noise suppression [4, 5, 6]. Zeroing the phase of short windowed signal frames in the short-time Fourier transform is a technique for creating robot-like voices, sometimes called robotization [7].

A concept closely related to zero-phase signals is the zero-phase filter, which is encountered in the design of linear-phase finite impulse response (FIR) filters [8, 9, 10, 11]. When the symmetric impulse response of such a filter is centered at time zero, the frequency response becomes real. It has a remarkable property that it can have both positive and negative values, and that's why it is sometimes called the "amplitude response" to distinguish it from the magnitude response, which is non-negative [9, 10]. The zero-phase FIR filter is usually delayed to ensure causality, resulting in a complex-valued frequency response. Rather surprisingly, a recent paper showed that zero-phase digital filtering can be imitated in real time using a recurrent neural network [12].

Theodore Burt described a sound conversion method in which a whole piece of music was transformed using a single large FFT, and the phase was replaced with a constant before transforming the signal back to the time domain using the inverse FFT (IFFT) [13]. Burt used the method in one of his compositions and explained that neglecting the phase led to unexpected results, as it was impossible to predict what aspects were lost with the phase of the signal, which represented half of the original information. Burt also observed that when the source music material contained a clear pulse, the converted signal maintained the same rhythm [13].

At the second DAFx conference in 1999, Hammer and Sundt presented a system called Mammut, which processes sound files using a long FFT [14]. Mammut offers multiple options for sound manipulation by modifying the FFT spectrum. The closest one to this work is the scaling of the phase by a constant, which may be positive or negative. In 2018, Välimäki et al. [15] showed that a short sound can be extended by computing its zero-padded FFT, randomizing the phase spectrum, and computing the IFFT. The FFT length defined the repetition period of the resulting extended signal, which could be looped seamlessly because of the time periodicity of the inverse DFT (IDFT) [15].

Wright and Stabile have studied the minimum-phase conversion of sounds to make them more percussive or to increase their

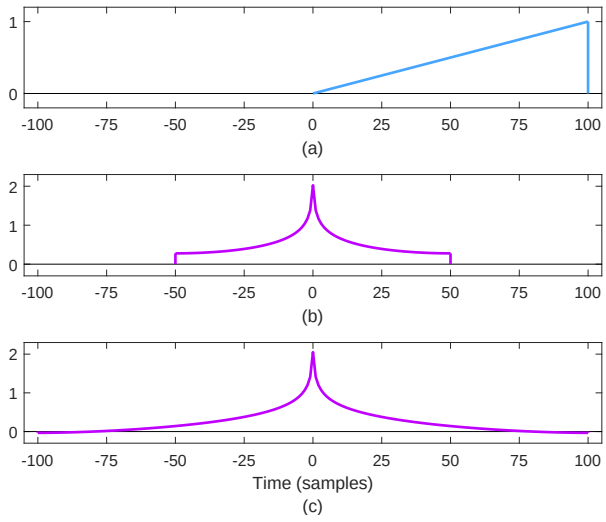

Figure 1: *(a) A ramp of 101 samples and its zero-phase versions (b) without and (c) with zero-padding ($P \approx 2$).*

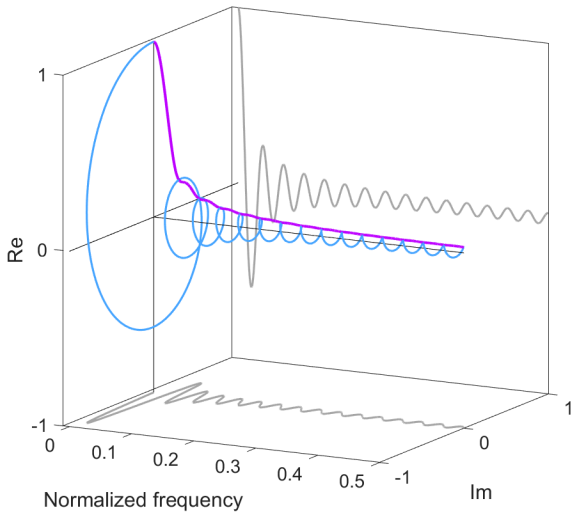

Figure 2: *Complex spectrum of a short ramp signal (blue) and the corresponding magnitude spectrum (purple). The shadows (gray) show the real and imaginary parts of the complex spectrum.*

attack sharpness [16]. The minimum-phase conversion, however, is a distinctly different method than the zero-phase conversion discussed in this paper. Donahue has also studied the use of the long FFT for sound processing in his studies on extended cross-synthesis of two or more sounds [17]. Peer et al. have studied the linearization of the phase of speech signals to enhance the consonants, but this was implemented in short frames [18].

The background of the present study is in experiments that the third author made in 1992 at IRCAM, Paris. He observed the effects on a long audio signal when converted to a corresponding zero-phase signal by first computing its complex spectrum using the FFT, then zeroing the phase angle, and finally, computing the IFFT. The rest of this paper is structured as follows. Section 2 discusses the zero-phase conversion technique and its properties. Section 3 applies the method to long audio signals and introduces a stereo version of the method. Section 4 suggests frame-based zero-phase conversion and discusses its real-time implementation in Max/MSP. Some concluding remarks appear in Section 5.

2. FFT-BASED ZERO-PHASE CONVERSION

Any audio signal can be converted to a corresponding zero-phase signal by first computing its complex spectrum, zeroing its complex phase, and finally, resynthesizing the signal with the inverse Fourier transform. The DFT of a real-valued time-domain signal $s(n)$, $n = 0, 1, 2, \ldots, N-1$, when zero-padded to length $L \geq N$ samples, is defined as

$$S(k) = \sum_{n=0}^{L-1} s(n) e^{-j2\pi kn/L}, \qquad k = 0, \ldots, L-1, \quad (1)$$

where k is the frequency bin index, j is the imaginary unit, and L is the transform length in samples, which may be very large in this work, such as $L = 2{,}646{,}000$ for a 1-min sound clip sampled at $f_s = 44.1$ kHz. The index k refers to the frequency $f_k = kf_s/L$ in Hz. We assume that all forward and inverse Fourier transforms are implemented with the FFT.

The zeroing of the phase of the complex spectrum can be implemented by applying the IDFT to the magnitude spectrum [5]:

$$s_0(n) = \frac{1}{L} \sum_{k=0}^{L-1} |S(k)| e^{j2\pi kn/L}, \quad n = 0, \ldots, L-1, \quad (2)$$

where the subscript "0" refers to "zero-phase" and the absolute value of the complex spectrum for all k is computed as

$$|S(k)| = \sqrt{\mathrm{Re}[S(k)]^2 + \mathrm{Im}[S(k)]^2}. \quad (3)$$

We use a rising ramp, shown in Fig. 1(a), to demonstrate the properties of the zero-phase conversion. Fig. 2 presents a 3-D view of the complex spectrum of a short ramp signal of 33 samples. The magnitude of the spectrum is also shown in purple in Fig. 2. The complex spectrum has been normalized to have a maximum magnitude of 1.0, and only the positive frequencies up to the Nyquist limit, or normalized frequency 0.5, are displayed.

Fig. 1(b) presents the zero-phase version of the ramp obtained with Eqs. (1) to (3) using the same FFT length as the test signal itself, $L = N = 101$. We place the center of the zero-phase signal at the time origin ($n = 0$) to highlight the fact that zero phase means no delay. Fig. 1(b) shows typical features of the zero-phase signal: its maximum is at the center, and it spreads symmetrically in both directions in time, i.e., it is even. These characteristics are common to zero-phase signals. The total energy, or the squared sum of sample values, of both the original and the zero-phase signal is 33.2, which verifies that removing the phase does not affect the signal energy. This is an expected property due to the Rayleigh energy theorem [3].

Fig. 3(a) compares the magnitude spectra of the original and zero-phase test signals obtained using the IFFT of the same length as the original signal. We have computed the spectra using considerably more FFT points than in sampled signals ($L = 8192$), which means we are approximating the discrete-time Fourier transform (DTFT), which is continuous in frequency. The two magnitude spectra deviate visibly from each other, although they are

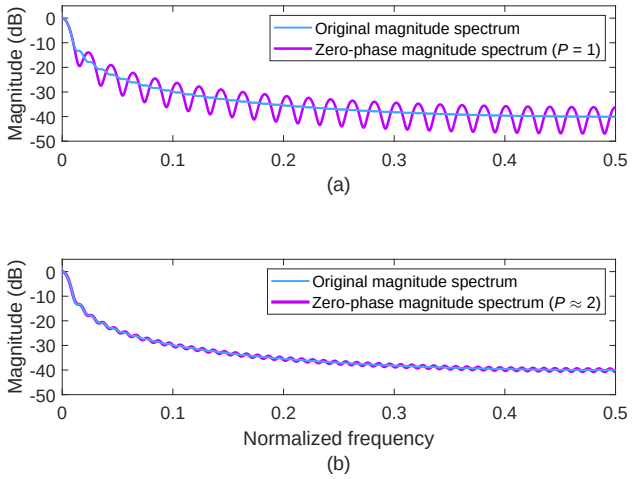

Figure 3: *Comparison of the normalized magnitude spectra of the original ramp signal and its zero-phase versions from Fig. 1 (a) without zero-padding and (b) with zero-padding ($P \approx 2$). In (a) the two spectra match at 52 points whereas in (b) they match at 101 points.*

exactly the same at 52 points. Between those points, there is much ripple in the spectrum of the zero-phase signal Fig. 3(a). This teaches us that simply zeroing the phase does not retain the spectral content. The time-domain shape of Fig. 1(b) appears truncated at both ends, suggesting that the signal has not decayed in the 101 samples, but has aliased in time according to the circular properties of the DFT. This implies that the zero-phase signal could be allowed to be longer than the original signal.

2.1. Benefits of Zero Padding

Zero padding with factor $P = 2$ is tested next. To keep the signal length odd, we use $L = 2N - 1 = 201$ FFT points (which corresponds to $P = 1.99$ to be exact). The resulting zero-phase signal, again with a total energy of 33.2, is presented in Fig. 1(c). Its magnitude spectrum is compared with the original ramp spectrum in Fig. 3(b), showing a better match than before. Now the two magnitude spectra match at 101 bins, and the zero-signal spectrum has less freedom to oscillate between the points. The remaining maximum error in the spectral magnitude is 0.55 dB, which is small enough for audio. Nevertheless, the error can be further reduced by increasing the zero-padding factor and allowing the zero-phase signal to oscillate further in both directions in time. It appears that zero-phase signals obtained with the IFFT are generally not limited in time, and this property is briefly discussed next.

2.2. Decay of Zero-Phase Signals

Extending the zero-padding to infinity and taking the DTFT allows us to analyze the behavior of the zero-phase conversion without time aliasing. The spectrum $S(\omega)$, with $\omega = 2\pi f/f_s$, is then considered as a continuous, periodic function. The inverse DTFT is equivalent to taking the Fourier series of this function, and then reversing the result in time. After taking the absolute value, it can be shown that the resulting spectrum $|S(\omega)|$, a continuous version

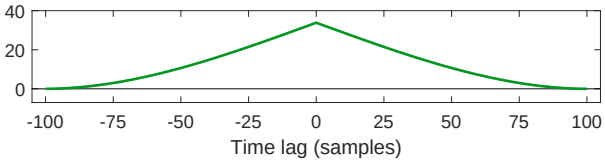

Figure 4: *Autocorrelation function of the ramp in Fig. 1(a) is inherently 201 samples long.*

of the one appearing in Eq. (2), is real analytic unless the spectrum has zeros. Going through zero, which is a singularity of the absolute value function, introduces a discontinuity in the derivative, making the resulting spectrum non-analytic. The Fourier series coefficients of a real analytic function decay exponentially [19]. Therefore, the signal $|s_0(n)|$ decays at the rate of $e^{-\alpha|n|}$ as $|n| \to \infty$ for some $\alpha > 0$. The tightest value of α is affected by the dynamic range of the spectrum $|S(\omega)|$.

If the spectrum $|S(\omega)|$ has zeros, the resulting signal may only decay at the rate of $1/|n|^2$ [19]. A basic example of this is the zero-phase conversion of a rectangular pulse. Practical time-limited signals typically do not have exact zeros in their spectra since they are complex-valued. This suggests that not very much zero padding is needed. Symmetric, real signals of the form $s(n) = \pm s(N-n)$ for $n = 1, \ldots, N-1$ are a notable exception, as their spectrum is purely real or imaginary (up to a linear phase shift) and can have multiple zeros. This property is familiar from linear-phase FIR filters [10].

2.3. Comparison with the Autocorrelation Function

Next, we consider the autocorrelation, which has similarities to the zero-phase signal. The autocorrelation function is defined as [8]:

$$c(m) = \sum_{n=0}^{N-1} s(n)s(n+m), \tag{4}$$

where the index m is generally called the time lag ($m = -N+1, ..., 0, ..., N-1$), and where the sequence is assumed extended such that $s(n) = 0$ when $n < 0$ and $n \geq N$. We use the unnormalized autocorrelation function here, although often a normalization factor is included [20, 3].

The autocorrelation function $c(m)$ is known to be a zero-phase signal. This makes sense as the two sequences involved in Eq. (4) cause the same phase shift in opposite directions in time, thus canceling each other out. However, unlike zero-phase signals, which are not limited in the time domain, the autocorrelation function of a signal of length N is exactly $2N-1$ samples long because it is the result of the convolution of two sequences of length N.

Fig. 4 shows the autocorrelation function of the ramp signal of Fig. 1(a), which we also like to center at $m = 0$. The peak value of the autocorrelation function, $c(0)$, is the same as the signal's variance or energy, which is 33.2 in this case. Comparison of Figs. 4 and 1(c) teaches us that the zero-phase signal and the autocorrelation function have a notably different shape, even when the signal lengths are the same. This comparison reassures us that the zero-phase conversion does not lead to the same result as autocorrelation. Still, the two share common features: the maximum at the center, even symmetry, and the decay toward zero far away from the zero lag.

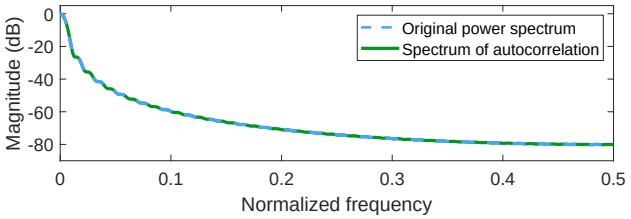

Figure 5: *Comparison of the normalized power spectrum of the ramp signal with the magnitude spectrum of its autocorrelation function shows a perfect match.*

Fig. 5 shows the power spectrum of the ramp (i.e., the squared magnitude of its FFT) and the magnitude spectrum of the autocorrelation function, which are both normalized to have their maximum at 0 dB. They are identical, which was confirmed by checking that their difference is close to machine accuracy. This is expected according to the well-known theorem, which states that the Fourier transform of the autocorrelation function of a signal is the power spectrum of that signal [3]. This also implies that the autocorrelation function of a signal can be obtained as the IFFT of its power spectrum:

$$c(m) = \frac{1}{L} \sum_{k=0}^{L-1} |S(k)|^2 e^{j2\pi km/L}, \tag{5}$$

where we select $L = 2N - 1$. The only difference from Eq. (2) is that here, the magnitude spectrum is squared, $|S(k)|^2$. Furthermore, we see that the autocorrelation function of the zero-phase signal $s_0(n)$ is identical to that of the original signal $s(n)$, since they have the same discrete power spectrum: $|S_0(k)|^2 = |S(k)|^2$.

Besides the general shape of the spectrum, the zero-phase signal and the autocorrelation function appear to mostly share their temporal properties, such as transients and other events, and the decaying temporal envelope, throughout the duration ($2N-1$ samples) of the autocorrelation signal. This may be justified by observing that the majority of the energy in the zero-phase signal $s_0(n)$ is contained near the time origin ($n = 0$), making it close to a time-limited convolution kernel. Since the zero-phase signal $s_0(n)$ is even, convolving it with itself results in the autocorrelation function and retains many temporal properties. The general shape of the spectrum is adjusted by the part of $s_0(n)$ near the time origin, which can be interpreted as a short FIR filter. The rest of the signal contributes a faint noise-like response.

3. ZERO-PHASE SOUND GENERATION

This section discusses the processing chain to produce zero-phase sounds and presents examples of how two different sounds are converted to have zero phase. Additionally, a stereo version of the method is introduced.

3.1. Zero-Phase Conversion of a Long Music Signal

As part of this study, we tested how long FFTs and IFFTs can be executed on a few computers. It was found that on a Windows laptop with 16 GB of RAM, it was possible to run in Matlab an FFT and IFFT of $L = 200{,}000{,}000$ points, when the original signal was kept in memory. At $f_s = 44.1$ kHz, this corresponds to

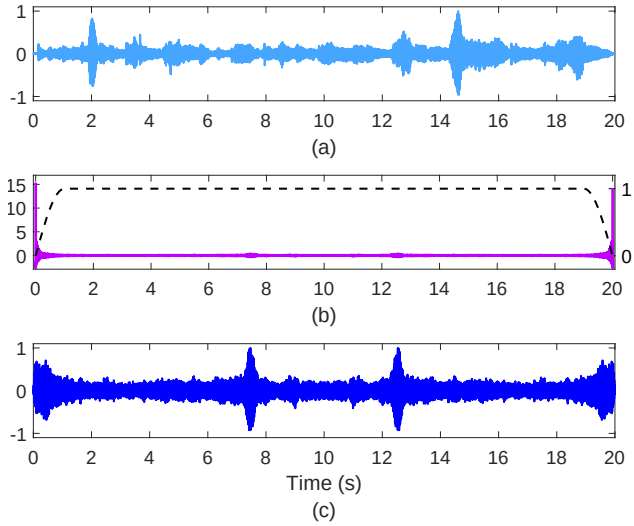

Figure 6: *(a) A 20-s excerpt of a string quartet, (b) its raw zero-phase IFFT ($P = 1$) and the suggested 1-s fade-in and fade-out gains (dashed line), and (c) the corresponding envelope-processed and normalized zero-phase sound, which is a sonic palindrome.*

4535.15 s or about 1 h and 15 min of audio. However, on a computer with 64 GB of RAM, we could run in Python an FFT and IFFT for 10 h of audio, which means 1.6 billion samples. The FFT and IFFT operations took 49 s and 56 s, respectively.

All computations in this paper, as well as the sound examples, were made in Matlab or Python using double-precision (64-bit) floating-point numbers. We also tested the methods using single-precision (32-bit) floating-point numbers in Python. There was no sign of numerical problems even with the longest FFT or IFFT operations in either case, as the difference between the double- and single-precision results was close numerical accuracy. It was found that numerical errors do not accumulate dramatically as the length of the FFT increases, but the bottleneck is the memory capacity. These tests confirm that it is now possible to process whole recordings of long compositions with the giant FFT.

3.2. Temporal Envelope Processing

To test and explain the zero-phase conversion process, we use much shorter audio examples here. Fig. 6(a) shows an example signal, which is an excerpt from a string quartet recording taken from a CD ($f_s = 44.1$ kHz). We apply an 882,000-point FFT to compute the spectrum of the 20-s signal without zero padding. The raw IFFT of its magnitude spectrum, cf. Eq. (2), which is presented in Fig. 6(b), is real and also 20 s in duration. The IFFT used here (Matlab's ifft.m) places the time origin at the beginning of the buffer, at $t = 0.0$ s, where a tall peak (max value 15.4) in the waveform is seen in Fig. 6(b). A second peak appears at the end of the buffer at $t = 20.0$ s, which corresponds to a negative time just before zero, as shown in Fig. 1.

The signal of Fig. 6(b) can be listened to but is dominated by the transients at the beginning and the end. We suggest attenuating them using fade-in and fade-out gains, which may be, for example, a quarter of a sinusoidal function. The duration of the fade-in and fade-out can be the same, but the appropriate length depends on

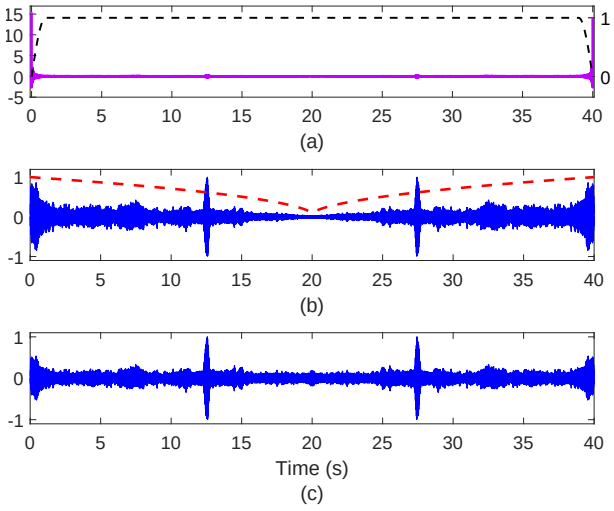

Figure 7: *(a) A 40-s long raw zero-phase IFFT ($P = 2$) of the string quartet signal of Fig. 6(a), the fade-in and fade-out gain (dashed line), (b) the normalized zero-phase signal with fade-in and fade-out, and (c) the gain-compensated zero-phase sound. The dashed red line in (b) is the gain estimate, with $\varepsilon = 0.01$, used to produce (c).*

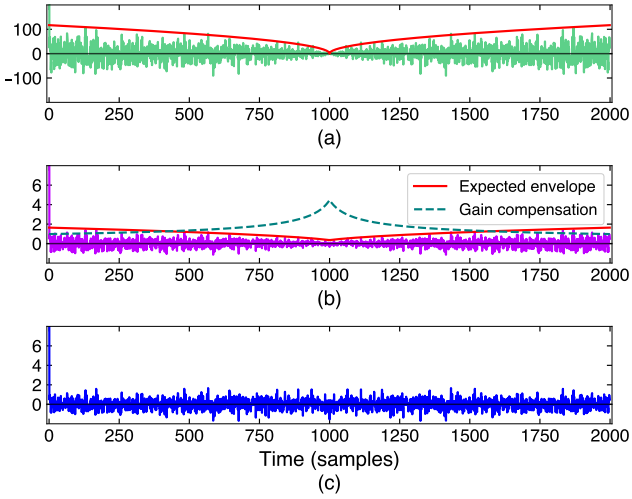

Figure 8: *(a) Autocorrelation function of white noise with $N = 1000, P = 2$, (b) the zero-phase signal formed from the same white noise signal, and (c) the zero-phase signal after gain compensation. The envelope of the autocorrelation function in (a) follows a linear power ramp. The zero-phase signal in (b) has a similar envelope but with an added regularization term $\varepsilon = 0.05$.*

the signal's spectrum. Fig. 6(c) shows the result in which a quarter of a sine and a cosine function of 1 s is used for fading in and for fading out, respectively. After fading both ends, the peak absolute sample value is normalized to 1.0. Now it is apparent that the resulting zero-phase sound is symmetrical w.r.t. the midpoint $t = 10.0$ s, see Fig. 6(c). In other words, the signal is even and forms a palindromic sound—it sounds the same when played in reverse.

Fig. 7(a) shows the result of a 1.764-million-point IFFT taken from a zero-padded string quartet signal (zero-padding factor $P = 2$). The signal is now 40 s long, but otherwise similar to before. By fading the beginning and end and normalizing, we obtain the zero-phase signal of Fig. 7(b), which should be compared with Fig. 6(c), which is its time-aliased version. Notice the different time scales in Figs. 6 and 7.

Nonetheless, when zero-padding is applied ($L = PN$ with $P \geq 2$), the zero-phase signal is quieter in the middle, as seen in Fig. 7(b). This is understandable from the corresponding property of the autocorrelation function discussed in Section 2.3. Here, we propose a gain compensation method to approximately equalize the loudness over time. Each sample of $s_0(n)$ is multiplied by the gain correction $g(n)$ given by

$$g(n) = \frac{1}{\sqrt{p(n) + \varepsilon}}, \tag{6}$$

where $p(n)$ is a piecewise linear ramp such that $p(0) = 1, p(N) = p(L - N) = 0$ and $p(L) = 1$, and ε is a free regularization constant, for which we use values from 0.01 to 0.05. The ramp $p(n)$ models the power envelope of the autocorrelation function. Specifically, $p(m)$ corresponds to the expected power of $c(m)$ when the original signal $s(n)$ is white noise. For time lags $m \neq 0$, the signal is uncorrelated with the delayed signal, so the expected power is determined by the amount of overlap between them. Fig. 8(a) shows the autocorrelation function of $N = 1000$ samples of white noise. The amplitude envelope of the autocorrelation is modeled well by $\sqrt{p(m)}$ (after appropriate scaling).

As discussed in Section 2.3, the temporal structure of the zero-phase signal matches the autocorrelation closely, but not exactly. The regularization constant ε accounts for the residual part and ensures the middle is not amplified too much. Fig. 8(b) shows the modeled amplitude envelope $\sqrt{p(n) + \varepsilon}$ for the zero-phase conversion of the same sample of white noise. The resulting gain compensated signal shown in Fig. 8(c) has an approximately uniform envelope. However, the peaks at the beginning and end of the signal in Fig. 8 are not affected by the gain compensation and should be suppressed using, for example, the quarter-sinewave discussed above. Fig. 7(c) shows the gain-compensated zero-phase version of the string quartet signal in which the middle part does not vanish, as it has been compensated using $g(n)$ with $\varepsilon = 0.01$.

3.3. Spectral Analysis

Fig. 9 compares the magnitude spectra of the whole 20-s string quartet signal, its autocorrelation function, and its two zero-phase versions: the raw IFFT with $P = 2$ and the final zero-phase after fading the two ends and compensating the envelope with $p(n)$. The four spectra were produced with a 2-million-point FFT ($L = 2^{21}$) without windowing. In the case of the raw IFFT signal, the two halves of the signal were interchanged before taking the FFT (using the Matlab function fftshift.m). This comparison verifies that the overall spectral characteristics of the original and zero-phase sound are practically the same, although the temporal behavior has changed. Additionally, Fig. 9(b) shows that the magnitude spectrum of the autocorrelation function decays much faster with frequency than the other spectra. This implies that the autocorrelation function, whose spectrum is equal to the power spectrum of the original, sounds much darker. It lacks the brightness of the original and zero-phase sounds.

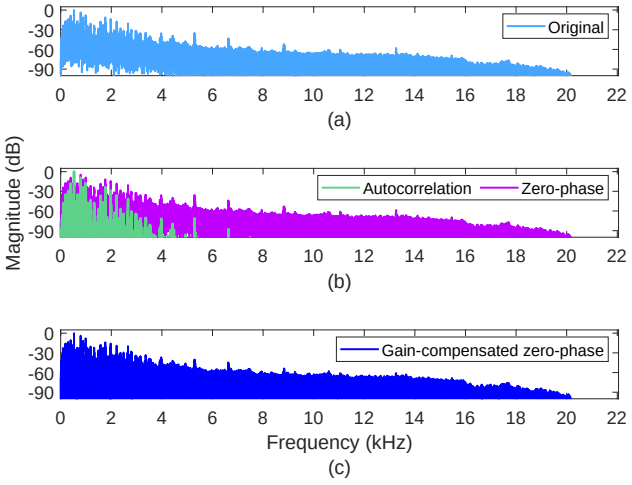

Figure 9: *Magnitude spectra of (a) the original string quartet signal of Fig. 6(a), (b) the raw zero-phase signal (with zero padding, $P = 2$) and the autocorrelation function of Fig. 7(a), and (c) the gain-compensated zero-phase sound of Fig. 7(c). Both zero-phase spectra are remarkably similar to the original.*

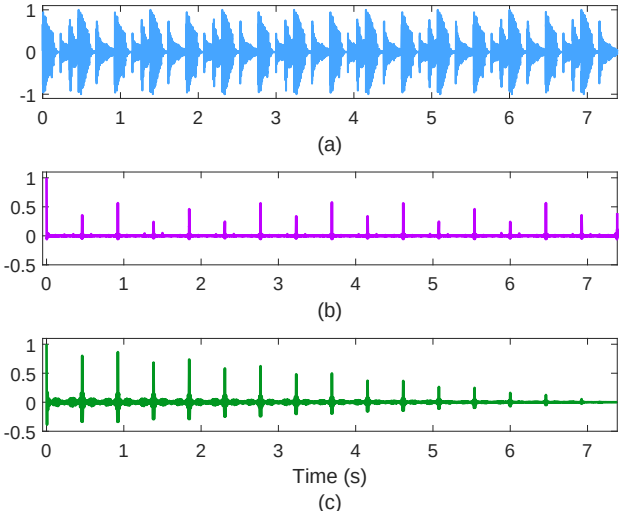

Figure 10: *(a) Electronic drum loop (130 BPM), (b) its raw zero-phase IFFT ($P = 1$), and (c) the autocorrelation function (for positive lags only). Both the zero-phase signal and the autocorrelation show the same periodic pattern at 0.46-s intervals.*

3.4. Zero-Phase Conversion of a Drum Loop

Next, an example is presented of zero-phase processing of an electronic drum loop of Fig. 10(a), which has a very strict rhythm. The raw IFFT of its magnitude spectrum is shown Fig. 10(b) whereas Fig. 10(c) has the autocorrelation of the same signal. It is seen that both the zero-phase and the autocorrelation signal show a clear periodic structure related to the beat of the drum loop and that the repetitive rhythm is indeed emphasized, leaving not much else of the original signal. The zero-phase processed drum loop, among

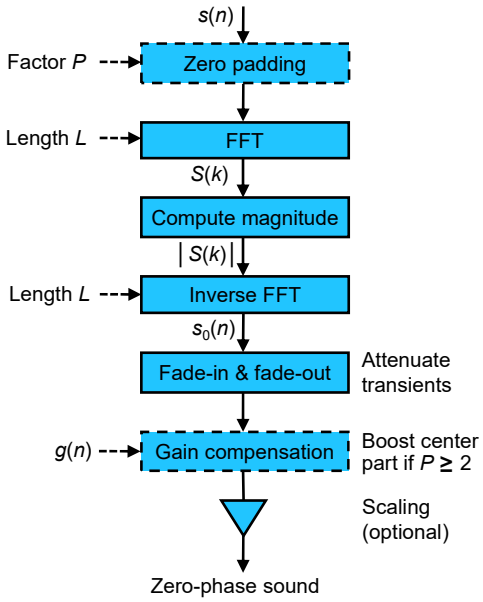

Figure 11: *Proposed zero-phase sound generation chain. The dashed blocks and scaling are optional. The main parameter is the zero-padding factor P. The FFT/IFFT length L depends on the zero-padding factor P and signal length N.*

other examples, can be listened to on a companion web page [1] that uses the audio player from [21].

Fig. 11 summarizes the processing suggested to resynthesize a zero-phase sound. First, the zero-padding factor P is chosen. The signal length N and zero-padding factor P determine the FFT length: $L = PN$. After the FFT, the absolute value of the complex spectral data is computed to form the magnitude spectrum. It is processed with the IFFT of the same length as before, which outputs the raw zero-phase signal. At this point, it may also be necessary to edit out the extra samples if a large zero padding factor P has been used. Since the zero-phase signal is dominated by tall transients at the beginning and end, it is recommended to usually suppress them, for example, using the quarter sinewave function discussed earlier. Finally, if zero-padding has been used, the temporal envelope of the resulting signal can be adjusted using the gain-compensation function $g(n)$, which amplifies the soft middle part of the zero-phase sound. The gain compensation is unnecessary when zero-padding is not used.

3.5. Quadrature Stereo

A stereo version of the method follows when the IFFT is applied separately to the absolute values of the real and imaginary components of the spectrum. This leads to a pair of zero-phase signals,

[1] `http://research.spa.aalto.fi/publications/papers/dafx25-zero-phase/`

$$s_{0,1}(n) = \frac{1}{L}\sum_{k=0}^{L-1} \big|\mathrm{Re}[S(k)]\big| e^{j2\pi kn/L} \tag{7}$$

and

$$s_{0,2}(n) = \frac{1}{L}\sum_{k=0}^{L-1} \big|\mathrm{Im}[S(k)]\big| e^{j2\pi kn/L}, \tag{8}$$

which may be used as the left and right channels of a stereo signal. Fig. 2 shows in gray the real and imaginary parts of the spectrum of an example ramp signal. It is seen that they are in quadrature.

Fig. 12 presents an example of a zero-phase stereo sound produced using the electronic drum loop signal of Fig. 10(a). Fig. 12(a) and (b) show the signals $s_{0,1}$ and $s_{0,2}$ obtained from the real and imaginary parts of the spectrum. They look similar but are slightly different, and form a pleasing stereo image when played together as the left and the right stereo channel. The impression is also slightly noisier than when listening to the single zero-phase sound of Fig. 10(b). The correlation of the two signals is given in Fig. 12(c) to demonstrate the fact that they are highly correlated, except exactly at the drum beats, where the correlation is close to 1.0.

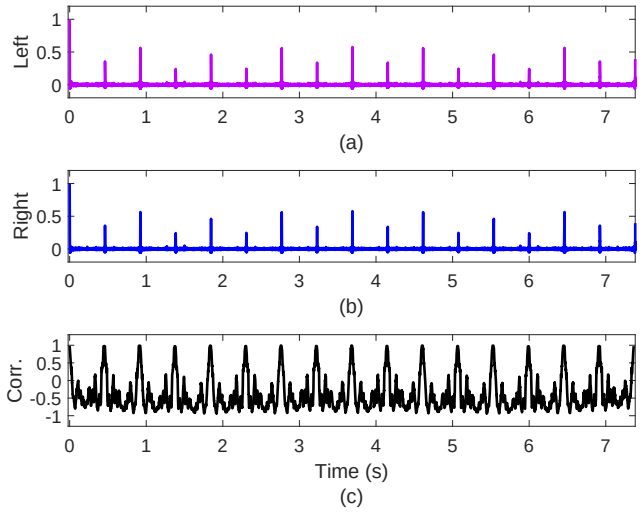

Figure 12: *(a) Left and (b) right stereo signals obtained from the zero-phase real and imaginary spectra of the electronic drum signal of Fig. 10(a), and (c) the correlation between the two signals, computed in 20-ms segments, showing that they form a decorrelated stereo signal.*

4. FRAME-BASED ZERO-PHASE CONVERSION AND MAX/MSP IMPLEMENTATION

It is straightforward to extend this approach to handle a continuous audio stream through the use of the short-time Fourier transform [3], where the frame length is large, such as over 1 s. The general structure is similar to the case of the phase vocoder [7], with a specified hop size, except that instead of performing phase adjustments, the phase is zeroed out, and the frame sizes are as large as desired. In addition to the fade in, fade out, and gain compensation mentioned earlier, a Hann window is applied to each frame.

A real-time version of the frame-based technique was implemented using Max/MSP. As the technique involves large FFT sizes, the input-output latency is quite large, because a number of samples equal to the FFT size must be captured before any spectral transformation can occur. This latency will be on the order of several seconds. Such a delay could be acceptable given the nature of the effect that creates a "wash" of the input. A curious property of the technique is the way it seems to compose different melodies when looping melodic source material.

The technique exposed certain limitations of Max/MSP when working with large FFT sizes. First, the FFT size was limited to $2^{17} = 131{,}072$, which was increased to a million points. Second, implementing the technique using the original fft~ object effectively doubled the latency, since this implementation outputs data in the frequency domain (from lowest to highest frequency bin) over the same amount of time as was needed to capture the input in the time-domain. As a result, the IFFT cannot happen until the entire frequency spectrum is "played" into it.

To eliminate this additional latency, we tried a different strategy using the pfft~ object, which can run both the spectral domain processing and the IFFT in one large block without this additional delay, at the potential risk of introducing drop-outs if the computation takes too much time. This initially proved unworkable due to the post-IFFT scaling and windowing needed in the frame-based approach. Once modifications to the pfft~ output processing system were implemented, a frame-based algorithm could run in real-time with a delay equal to the number of samples needed to perform the FFT.

5. CONCLUSION

A giant-FFT technique to generate zero-phase sounds is proposed. The basic idea is to compute a long FFT of the original sound and apply the IFFT to its magnitude spectrum. The resulting zero-phase signals are real and even in the time domain, which means they form a sonic palindrome.

The zero-phase signal's magnitude spectrum is the same as that of the original. This distinguishes the zero-phase signal from the autocorrelation function, the inverse transform of the power spectrum, which otherwise has similar behavior. The autocorrelation function generally sounds lowpass filtered and often dull, whereas the zero-phase sound entails the original sound's brightness.

When zero padding is not used, a time-aliased zero-phase signal is obtained, which is palindromic. Zero-padding with a factor of 2 can be used to extend the palindromic sound to be twice as long as the original. In this case, the central part of the signal is faint but can be amplified with a gain compensation function proposed in this paper. Regardless of the zero-padding factor, it is recommended to fade in the very beginning and fade out the end of the zero-phase sound to suppress the loud clicks.

Extensions of the method include a stereo version and a continuous audio-stream processing technique suitable for a real-time implementation. Both variations were briefly described, and the latter method has been developed and tested using the Max/MSP visual programming language.

It is known that the real-valued amplitude response of a linear-phase filter can contain both positive and negative values. Thus, as a continuation of this work, it would be interesting to try estimating the real-valued amplitude spectrum of an audio signal—as an alternative to the non-negative magnitude spectrum. Future work may also consider decomposing the input signal to percussive and non-percussive components, as the zero-phase conversion handles

these in a different manner. Median-filter based signal separation techniques would be a suitable starting point [22, 23].

The exciting characteristic of zero-phase sound is that its temporal events are dramatically different from those taking place in the original audio, although the timbre is similar. Nevertheless, if the original signal has a clear rhythm, it is retained in the zero-phase version, which can be understood from the similarity to the autocorrelation function. The associated audio examples also demonstrate that with speech input, the method produces babble noise, or unintelligible voice-like sound that has the timbral characteristics of the original speech. Combining the voices of multiple speakers yields a cocktail-party-like chatter. Zero-phase processing of recordings of singing leads to effects reminiscent of infinite reverberation. The FFT-based zero-phase sound generation method is a novel, straightforward way to modify music and other audio material.

6. ACKNOWLEDGMENTS

This work was initiated in May 2022 when Stefan Bilbao visited the Aalto Acoustics Lab. Another step forward was made in October 2024 when Vesa Välimäki made a short visit to the University of Edinburgh. Further progress took place when Sebastian Schlecht visited the Aalto Acoustics Lab in December 2024 and in February 2025. The authors are grateful to Tom Mudd for helpful discussions on the use of the zero-phase conversion technique in computer music.

7. REFERENCES

[1] V. Välimäki and S. Bilbao, "Giant FFTs for sample rate conversion," *J. Audio Eng. Soc.*, vol. 71, no. 3, pp. 88–99, Mar. 2023.

[2] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proc. IEEE*, vol. 69, no. 5, pp. 529–541, May 1981.

[3] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT)*, W3K Publishing, 2007, Online book available at https://ccrma.stanford.edu/∼jos/dft/.

[4] W. Thanhikam, A. Kawamura, and Y. Iiguni, "Noise suppression based on replacement of zero phase signal," in *Proc. Int. Symp. Intelligent Signal Processing and Communications Systems (ISPACS)*, Dec. 2011, pp. 1–4.

[5] S. Kohmura, A. Kawamura, and Y. Iiguni, "An efficient zero phase noise reduction method for impact noise with damped oscillation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 892–895.

[6] R. Takehara, A. Kawamura, and Y. Iiguni, "Impulsive noise suppression using interpolated zero phase signal," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2017, pp. 1382–1389.

[7] D. Arfib, F. Keiler, U. Zölzer, V. Verfaille, and J. Bonada, *Musical Signal Processing*, chapter 7 Time-frequency processing, pp. 220–278, U. Zölzer, Ed., Wiley & Sons, Chichester, UK, 2011.

[8] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice-Hall, 2nd edition, 1999.

[9] T. W. Parks and C. S. Burrus, *Digital Filter Design*, Wiley-Interscience, 1987.

[10] T. Saramäki, *Handbook for Digital Signal Processing*, vol. 4, chapter Finite impulse response filter design, pp. 155–277, Mitra, S. K. and Kaiser, J. F. (Eds.), Wiley, New York, NY, 1993.

[11] J. O. Smith, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, 2007, Online book available at https://ccrma.stanford.edu/∼jos/filters/.

[12] T. Sinjanakhom and S. Chivapreecha, "Real-time zero-phase digital filter using recurrent neural network," in *Proc. IEEE Asia Pacific Conf. Circ. Syst. (APCCAS)*, Nov. 2023, pp. 348–352.

[13] T. Burt, "Composition folio," PhD thesis, Univ. of York, Music Research Centre, York, UK, Sep. 2012. Available at https://etheses.whiterose.ac.uk/id/eprint/3817/1/TheoBurt_PhD_Commentary.pdf.

[14] Ø. Hammer and H. Sundt, "Musical applications of decomposition with global support," in *Proc. 2nd COST-G6 Workshop Digital Audio Effects (DAFx)*, Trondheim, Norway, Dec. 1999.

[15] V. Välimäki, J. Rämö, and F. Esqueda, "Creating endless sounds," in *Proc. Int. Conf. Digital Audio Effects (DAFx18)*, Aveiro, Portugal, Sep. 2018, pp. 32–39.

[16] M. Wright and M. Stabile, "Spectrally matched click synthesis," in *Proc. 12th Int. Conf. Digital Audio Effects (DAFx09)*, Como, Italy, Sep. 2009.

[17] C. Donahue, "Extensions to convolution for generalized cross-synthesis," M.S. thesis, University of California, San Diego, CA, 2016.

[18] T. Peer, K.-J. Ziegert, and T. Gerkmann, "Plosive enhancement using phase linearization and smoothing," in *Proc. 14th ITG Conf. Speech Communication*, Sep. 2021, pp. 1–5.

[19] M. A. Pinsky, *Introduction to Fourier Analysis and Wavelets*, American Mathematical Society, 2009, See Chapter 1.2.3.

[20] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 25, no. 1, pp. 24–33, Feb. 1977.

[21] N. Werner, S. Balke, F.-R. Stöter, M. Müller, and B. Edler, "trackswitch.js: A versatile web-based audio player for presenting scientific results," in *Proc. 3rd Web Audio Conf.*, London, UK, Aug. 2017.

[22] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Graz, Austria, Sep. 2010, p. 217–220.

[23] L. Fierro and V. Välimäki, "Enhanced fuzzy decomposition of sound into sines, transients, and noise," *J. Audio Eng. Soc.*, vol. 71, no. 7/8, pp. 468–480, Jul. 2023.