

# Spectral remapping of natural signals

Md. Shoaibur Rahman  
Baylor College of Medicine  
Houston, TX 77054  
[shoaibur.rahman@bcm.edu](mailto:shoaibur.rahman@bcm.edu)

## Abstract

*Here we present an algorithm to procedurally remap spectral contents of natural signals. The algorithm takes in a signal and a mapping function as inputs, and generates an output, which is a remapped version of the original signal. In the first step, the algorithm performs a Hilbert transformation of the input signal. This transformation operation separates out the envelope information and the spectral contents. In the second step, the spectral content is modified based on the mapping function. In the third and final step, the modified spectral content is combined with the envelope information to reconstruction the output signal.*

## 1. Introduction

This paper presents an algorithm to procedurally remap spectral contents of natural signals. Procedural remapping of natural signals' spectra is of interest in numerous applications. For example, computer music composers are interested in pitch transposition by modifying spectra of a wide variety of sound signals. Systematic pitch transpositions could produce many of our desired perceptual effects of sound signals. To achieve these perceptual effects, efficient remapping of spectral contents of sound signals has been a great importance in audio applications for a long time (Grey and Moorer, 1977; Grey and Gordon, 1978; Samson et al., 1997; Krumhansl and Iverson, 1992; Alluri and Toiviainen, 2010; Vos and Rasch, 1881; Deutsch, 2013). Analogous applications of spectral remapping have been reported in other fields as well. For instance, clinical applications using EEG (Chemin et al. 2018; Wang et al. 2016) and fMRI (Meszlényi et al. 2017), Acoustic oceanographic measurements (Niu et al. 2014; Bonnel et al. 2017), speech recognition (Chang et al., 2017; Goupell et al., 2008; Noda, 1988; Lee and Rose, 1998).

The process of spectral remapping is equivalent to warping of each frequency in the signal. So, spectral remapping can be achieved by warping each frequency of the signal based on a given remapping function (**Fig. 1**).

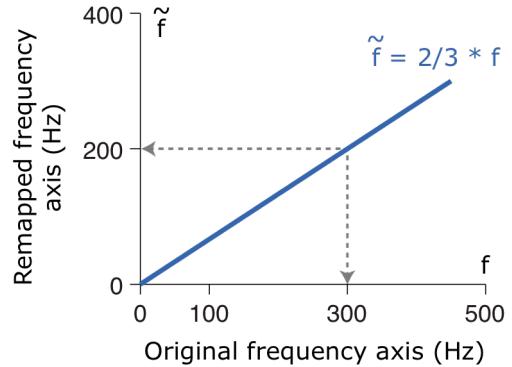


Figure 1. Frequency remapping explained with a remapping function,  $\tilde{f} = 2/3 * f$ .  $f$  and  $\tilde{f}$  are the frequencies of the original and remapped signals, respectively. So, a frequency of 300Hz in original axis will be projected to 200Hz in remapped axis, and vice versa.

The idea of frequency warping was initially proposed in terms of phase vocoders (Flanagan and Golden, 1966; Moorer 1978). Phase vocoder was originally developed for timescale modification of audio, and was implemented using filter-banks and Fourier transforms (Dolson, 1986). Then other versions of phase vocoders were reported (Puckette, 1995; Laroche and Dolson, 1999). Although the implementation techniques of the later versions were different from the initial ones, the goal was same – to change the temporal characteristics of a sound by stretching or compressing the time-base of the spectrogram. These methods introduce errors in the spectral remapping process (Puckette and Brown, 1998). Further efforts had been made to develop more general methods for frequency warping, for instance, using arbitrary allpass maps (von Schroeter, 1999), and more recently, using short time Fourier transforms and Gabor frames (Evangelista and Cavaliere, 2007; Mejstrik and Evangelista, 2016; Wabnik et al., 2005). These methods perform well for remapping spectral contents of simple signals like pure tones or even frequency sweeps. However, these algorithms are susceptible to high spectral leakage during the remapping process for complex natural signals like human speech.

To address this gap, here we present an algorithm of

frequency warping to remap spectral contents of various signals, including pure tones, frequency sweeps, and a wide variety of natural signals. We have shown that although the algorithm concedes a nominal spectral leakage, it can faithfully remap the spectra of the signals to the desired ones. The validity of the remapped signals was tested using signals of different complexity levels, i.e. from pure tones to human voice with instruments.

## 2. Algorithm

A high-level description of the proposed algorithm is presented in the block-diagram (**Fig. 2**). The algorithm involves analysis and synthesis steps. The analysis steps start with taking a Hilbert transform of the input signals to separate out the envelope and instantaneous phase information. Instantaneous phases are then converted into instantaneous frequencies, which are warped based on the warping function provided. Finally, in the synthesis step, the warped frequencies or phases are combined with the intact envelope to reconstruct the output signals.

A summary of the algorithm is shown in **Table 1**. The details of the inputs, analysis/synthesis steps and output are explained in the following subsections. The mathematics has been shown for discrete signals to keep it aligned with computer implementations/codes. Codes will be publicly available. Similar mathematics is also valid for continuous signals.

### 2.1. Inputs

The algorithm requires two inputs: first, a real signal  $x[n]$  in time domain with a length of  $M \in \mathbb{Z}_+$  with  $n$  being the time index; second, a frequency warping function,  $\tilde{f}[n] = \text{func}(f[n])$ , where  $f[n]$  and  $\tilde{f}[n]$  are the instantaneous frequencies of the input and output signals, respectively.

### 2.2. Analysis

Step 1 of analysis is to form an analytic signal by taking Hilbert transform (HT) of the input signal. However, before taking HT, the mean signal  $\mu$  is computed and subtracted from the input signal.

$$\mu = \frac{1}{M} \sum_{k=1}^M x[k] \quad (1)$$

$$s[n] = x[n] - \mu \quad (2)$$

$$\hat{s}[n] = \mathcal{H}(s[n]) \quad (3)$$

where,  $\mathcal{H}$  represents an operator for HT.  $s[n]$  and  $\hat{s}[n]$  are used to form the analytical signal  $s_a[n]$ , which is a complex signal in time and is defined as:

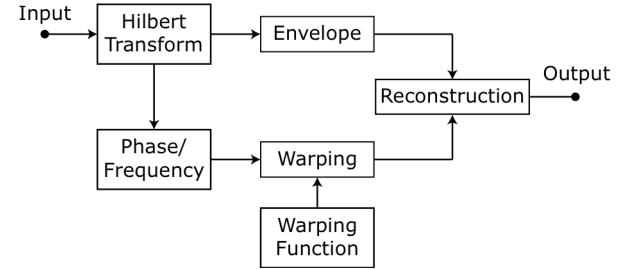


Figure 2. Block diagram representation of the frequency warping algorithm. Input signal passes through Hilbert transform that separates out the envelope and phase/frequency. Frequency is then warped based on the warping function. Warped frequency and the envelope are used to reconstruct warped output signal.

$$s_a[n] = s[n] + j\hat{s}[n] \quad (4)$$

In step 2, this analytic signal is used to extract envelope  $e[n]$  and instantaneous phase  $\phi[n]$  in radians.

$$e[n] = \sqrt{|s[n]|^2 + |\hat{s}[n]|^2} \quad (5)$$

$$\phi[n] = \arctan\left(\frac{\hat{s}[n]}{s[n]}\right) \quad (6)$$

In step 3, instantaneous phases are converted into instantaneous frequencies  $f[n]$  by applying a series of unwrap/wrap and differential/integral operations. However, the envelope remains intact and no operation is applied on it before synthesis. The phases computed in (6) are wrapped between  $-\pi$  and  $\pi$  for every signal. For instance, the computed phases of  $\sin(2\pi + \pi/3)$  and  $\sin(-4\pi + \pi/4)$  are  $\pi/3$  and  $\pi/4$  instead of their actual phases  $2\pi + \pi/3$  and  $-4\pi + \pi/4$ , respectively. So, before any operation being applied, the actual phases are restored by adding appropriate multiples of  $2\pi$  to each phase computed in (6). This process is phase unwrapping. Let's denote the unwrapped phases as  $\theta[n]$  and  $\mathcal{U}$  be the unwrap operator, then the process can be expressed as:

$$\theta[n] = \mathcal{U}(\phi[n]) \quad (7)$$

$\theta[n]$  are then converted into instantaneous frequencies  $f[n]$  by applying a differential operation, i.e.

$$f[n] = \frac{F_s}{2\pi} (\theta[n+1] - \theta[n]) \quad (8)$$

where,  $F_s$  denotes sampling frequency. Instantaneous frequencies are modified or warped based on the frequency warping function  $\tilde{f}[n] = \text{func}(f[n])$ . The warped frequencies,  $\tilde{f}[n]$ , are converted back to phases  $\tilde{\theta}[n]$  by applying an integral operation, i.e.

$$\tilde{\theta}[n] = \frac{2\pi}{F_s} \sum_{k=1}^n \tilde{f}[k], \quad n = 1, 2 \dots M \quad (9)$$

$\tilde{\theta}[n]$  are the actual warped phases. So, before they can be used to reconstruct the warped signal at the synthesis step, they must be wrapped between  $-\pi$  and  $\pi$  by subtracting appropriate multiples of  $2\pi$  from  $\tilde{\theta}[n]$ . Let's denote the wrapped wrapped phases as  $\tilde{\phi}[n]$ , which is computed as:

$$\tilde{\phi}[n] = \mathfrak{u}(\tilde{\theta}[n]) - 2\pi \left\lfloor \frac{\mathfrak{u}(\tilde{\theta}[n]) + \pi}{2\pi} \right\rfloor \quad (10)$$

where,  $\lfloor \cdot \rfloor$  denotes a floor operation.

### 2.3. Synthesis

Step 4 is the synthesis step, in which the warped output signal is reconstructed. To execute this step, an element-wise dot product between the envelope and the cosine of wrapped phases is computed. The results of their product are added up with the mean signal to reconstruct the frequency-warped output signals,  $\tilde{x}[n]$ , i.e.

$$\tilde{x}[n] = \mu + e[n] \cdot \cos \tilde{\phi}[n] \quad (11)$$

To understand how (11) works at the synthesis step, here we give an example of perfect reconstruction, i.e. we set up an example so the warped output signal will be identical to the original input signal. An identity frequency warping function, i.e.  $\tilde{f}[n] = f[n]$ , is an obvious choice for the perfect reconstruction. With the identity frequency warping function, we find that the wrapped phases for original input signal and warped output signal are identical, i.e.  $\tilde{\phi}[n] = \phi[n]$ , after applying unwrap/unwrap and differential/integral operations in the analysis steps. So, according to (11), the reconstructed signal:

$$\tilde{x}[n] = \mu + e[n] \cdot \cos \phi[n] \quad (12)$$

Using trigonometric identities (**Fig. 3**):

$$\begin{aligned} \tilde{x}[n] &= \mu + e[n] \cdot \cos \arctan \left( \frac{\hat{s}[n]}{s[n]} \right) \\ &= \mu + e[n] \cdot \cos \arccos \left( \frac{s[n]}{e[n]} \right) \\ &= \mu + e[n] \cdot \frac{x[n] - \mu}{e[n]} \\ &= x[n] \end{aligned} \quad (13)$$

Therefore, with an identity frequency warping function, the warped output signal is identical to the original input

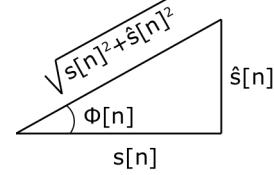


Figure 3. Trigonometric identities. To make this triangular representation analogous to (6), base and height must be  $s[n]$  and  $\hat{s}[n]$ , respectively. Accordingly, the hypotenuse is  $\sqrt{s[n]^2 + \hat{s}[n]^2}$ , which is equal to  $e[n]$  (5). Thus,  $\arctan \left( \frac{\hat{s}[n]}{s[n]} \right) = \Phi[n] = \arccos \left( \frac{s[n]}{e[n]} \right)$ .

signal. This verifies that (11) can be used to estimate the warped output signal.

### 2.4. Output

The algorithm returns a single output, which is the frequency-warped signal of the original input signal. Since the envelope of the input signal remained intact in analysis steps and is just used in synthesis step, the output signal maintains almost the same envelope properties that were in the original input signal.

### 3. Test and validation of the algorithm

To test and validate the performance of the algorithm we considered examples with three types of signals: pure tone sinusoid, frequency sweep sinusoid, and natural sound signals. For simplicity and to keep the analyses consistent across the examples, same warping function was used in all examples. Specifically, we used the following simple linear function for frequency warping:

$$\tilde{f}[n] = 2/3 * f[n] \quad (14)$$

This implies the following inverse frequency warping or de-warping function:

$$f[n] = 3/2 * \tilde{f}[n] \quad (15)$$

#### 3.1. Validation with pure tone sinusoid

A pure tone sinusoidal input signal was generated with a sampling frequency of 16kHz. The frequency of the signal was 300Hz (**Fig. 4A**). The power spectrum (PS) of the signal confirmed that its frequency was 300Hz, i.e. the power of the signal was concentrated around 300Hz (**Fig. 4B**). The signal and the warping function (14) were used as the inputs to the frequency-warping algorithm, which outputted a sinusoidal signal of a different frequency (**Fig. 4C**). Theoretically, according to (14), the frequency of the warped output signal should be:  $\tilde{f} = 2/3 * 300 = 200$ Hz. PS analysis confirmed that the frequency of the warped signal was 200Hz (**Fig. 4D**). Now, for a lossless

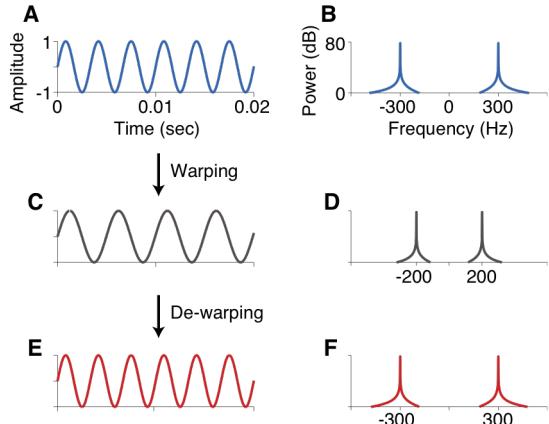


Figure 4. Validation of frequency warping algorithm with pure tone sinusoid. **A)** Pure tone sinusoidal input signal with a frequency of 300 Hz. Original signal was 1 second long; a shorter version is shown for better visualization. **B)** Power spectrum density (PSD) of the input signal. **C)** Warped signal obtained after applying frequency warping function on the input signal. **D)** PSD of warped signal. **E)** De-warped signal obtained after applying inverse warping function on warped signal. **F)** PSD of de-warped signal. For a lossless algorithm, de-warped signal presented in E must be same to the original input presented in A, therefore, PSD presented in F must be same to that in B.

algorithm, if the warped signal and the de-warping function are used as the inputs to the algorithm, then we should obtain an output that is identical to the original input signal. Hence, according to (15), the frequency of the de-warped signal must be:  $f = 3/2 * 200 = 300\text{Hz}$ . The de-warped signal was estimated (**Fig. 4E**). The cross-correlation between this estimated de-warped and the original signals was unity, i.e.  $r = 1$ . Additionally, PS analysis confirmed that the frequency of the de-warped signal was 300Hz, i.e. the power spectra of the de-warped and original signal were highly correlated ( $r = 0.978$ ) (**Fig. 4F**). A correlation value of less than 1 implies that the algorithm concedes some spectral leakage during the warping. However, because of high correlation between the original and de-warped signals (and between their power spectra), the leakage can be negligible.

### 3.2. Validation with frequency sweep sinusoid

A frequency sweep sinusoidal input signal was generated with a sampling frequency of 16kHz. The frequency of the signal changed logarithmically from 21Hz to 480Hz in 1 second (**Fig. 5A**). PS analysis confirmed that the start and the end frequencies of the signal were 21Hz and 480Hz, respectively (**Fig. 5B**). The signal and the warping function were used as the inputs to the algorithm, which outputted a warped frequency sweep sinusoidal signal, whose start and end frequencies were different from that in original signal (**Fig. 5C**). Theoretically, according to (14), the start and end

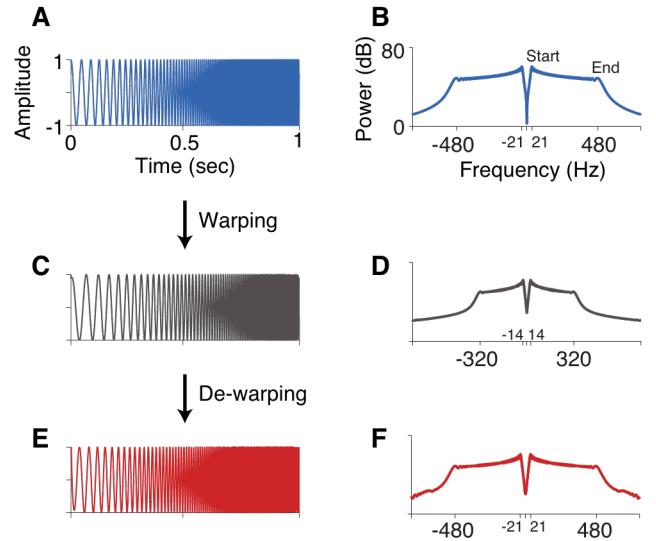


Figure 5. Validation of frequency warping algorithm with frequency sweep sinusoid. **A)** Frequency sweep sinusoidal input signal, whose frequency increases logarithmically from 21Hz to 480Hz in 1 second. **B)** PSD of the input signal showing the start and end frequencies. **C)** Warped signal obtained after applying frequency warping function on the input signal. **D)** PSD of warped signal. **E)** De-warped signal obtained after applying inverse warping function on warped signal. **F)** PSD of de-warped signal. For a lossless algorithm, de-warped signal presented in E must be same to the original input signal presented in A, therefore, PSD presented in F must be same to that in B.

frequencies should be 14Hz and 320Hz. PS analysis confirmed that the start and end frequencies of the warped signal were 14Hz and 320Hz (**Fig. 5D**). Now, to check for the spectral leakage, the warped signal and the de-warping function were used as the inputs to the algorithm, which outputted the de-warped signal (**Fig. 5E**), which was similar to the original signal ( $r = 0.998$ ). Also, their power spectra were alike ( $r = 0.996$ ); hence, the start and the end frequencies of this de-warped signal were 21Hz and 480Hz, respectively (**Fig. 5F**).

### 3.3. Validation with natural signals

Seven natural signals were used to validate the algorithm. The signals were available in Matlab2018b either in .mat or .wav formats. A description of the signals is listed in **Table 2**.

A train whistles signal was used as the input to the algorithm (**Fig. 6A**). The signal comprised several major frequency bands as displayed in power distribution over time and frequency (**Fig. 6B**). Alternatively, the power of the signal was concentrated at certain frequencies as indicated by the larger peaks on the power spectrum of the signal (**Fig. 6C**). The signal and the warping function were used as the inputs to the algorithm, which outputted a warped signal. The overall envelope of the warped signal remained intact but the frequencies were changed as

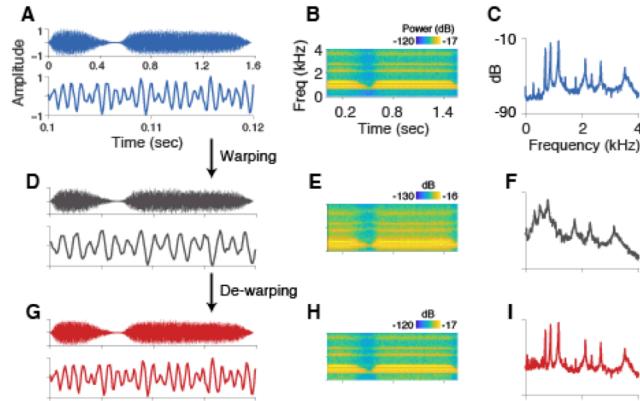


Figure 6. Validation the algorithm with more common natural signal. **A)** Top: Original input signal of train whistles (train.mat,  $F_s = 8192\text{Hz}$ , 1.57 seconds). Bottom: Input signal zoomed-in between 0.1 and 0.12 seconds. **B)** Time-frequency plot of power distribution of input signal. **C)** PSD of the input signal. **D)** Warped signal (top: entire duration, bottom: zoomed-in version). **E)** Time-frequency plot of power distribution of warped signal. **F)** PSD of the warped signal. **G)** De-warped signal (top: entire duration, bottom: zoomed-in version). **H)** Time-frequency plot of power distribution of de-warped signal. **I)** PSD of the de-warped signal.

expected (Fig. 6D). The power of the warped signal essentially shifted to lower frequency bands as shown in its power distribution over time and frequency (Fig. 6E). Indeed, the frequencies of the original signal were reduced by a factor of  $3/2$ , which is more obvious on the frequency spectrum (Fig. 6F). Now, to check for the spectral leakage, the warped signal and the de-warping function were used as the inputs to the algorithm, which outputted the de-warped signal (Fig. 6G). We found that the de-warped signal was nearly identical to the original input signal ( $r = 0.99$ ). Moreover, the distribution of power over time and frequency was shifted back to the higher frequency bands (Fig. 6H). More precisely, the frequencies of the warped signal were increased by a factor of  $3/2$ , which was more obvious on the power spectrum of the de-warped signal (Fig. 6I). This shift in frequency made the spectrum of the de-warped signal similar to that of the original signal ( $r = 0.851$ ).

Similar analyses were performed with the six remaining example signals. We found that frequency spectra of the original input signal and the de-warped signals were highly correlated (Fig. 7A-F). High correlation values in all these examples demonstrate that although the algorithm concedes some spectral leakage, it performs well for a frequency warping for a wide variety of natural signals, including vocals with instruments.

### 3.4. Comparison with existing methods

Similar analyses with pure tones, frequency sweeps, and seven natural signals were performed using sampling

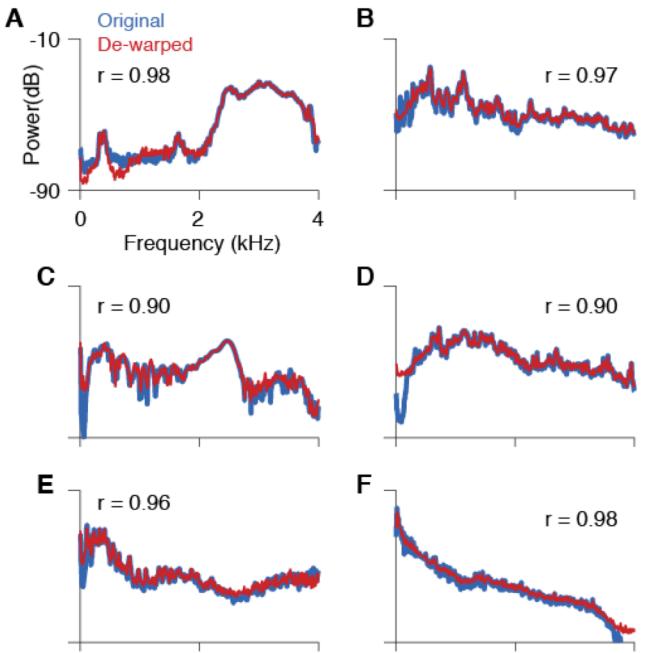


Figure 7. Validation with different natural signals. Power spectrum of: **A)** Chirping of a bird. **B)** Hallelujah chorus from Handel's Messiah. **C)** Fall and splat sound. **D)** Sound of laughing of humans. **E)** Male voice speaking. **F)** Rock with vocals, drums and guitar (used  $F_s = 8000, 53.78$  seconds to keep consistent with sampling frequencies of the other signals in this figure. Conclusions are not changed if using 48000Hz though).

expansion () and phase vocoder () methods. We found that the cross-correlations between the spectra of the original and de-warped signals were higher for the proposed algorithm in this paper. This comparison implies that the proposed algorithm concedes smaller amount of spectral leakage during the remapping process.

### 4. Discussion

The algorithm can faithfully warp the frequencies of diverse signals with various levels of complexities, i.e. from pure tones to human voice with instruments. However, it also subjects to a number of limitations. First, the algorithm relies on the estimation of instantaneous frequencies. So, without a good estimation of the instantaneous frequency, the algorithm will perform poorly. The method of estimating frequencies used in this paper has widely been studied, but a better method will increase the performance of the algorithm. Second, spectral leakage is higher for complex signals as compared to simple signals like pure tones. Again, the estimation of instantaneous frequencies of complex is less accurate. This error in estimation may be accounted as the spectral leakage. However, the spectral leakage was nominal. Third, the computational complexity of the algorithm has not been assessed in this paper. The algorithm performs

pretty fast in general as it uses FFT to compute Hilbert transform in analysis step. For instance, each example in this paper took a fraction of seconds to perform warping and de-warping in a MacOSX. Larger signals may be partitioned in time and used each partition as separate inputs to make the computation faster.

Despite these limitations, the proposed algorithm can be a powerful tool.

Table 1. Summary of the algorithm

**Algorithm:** Frequency warping

**Input:** Real signal  $\mathbf{x}[n]$  with a signal length of  $M \in \mathbb{Z}_+$   
**Output:** Estimates of frequency warped signals

**Analysis:**

- 1: Construct analytic signal
  - 1a: Compute mean signal,  $\mu$
  - 1b: Compute operation signal,  $s[n]$  (subtract mean from the original signal)
  - 1c: Compute Hilbert transform,  $\hat{s}[n]$
  - 1d: Form analytic signal,  $s_a[n] = s[n] + j\hat{s}[n]$
- 2: Extract properties of the analytic signal
  - 2a: Extract envelope,  $e[n] = |s_a[n]|$
  - 2b: Extract instantaneous phase,  $\phi[n] = \arg(s_a[n])$
- 3: Warp frequency and phase
  - 3a: Unwrap phase,  $\theta[n] = \arg(\phi[n])$
  - 3b: Obtain instantaneous frequency,  $f[n]$
  - 3c: Apply frequency warping function and obtain warped frequency,  $\tilde{f}[n]$
  - 3d: Convert  $\tilde{f}[n]$  into warped phase,  $\tilde{\phi}[n]$

**Synthesis:**

- 4: Reconstruct warped output signal
  - 4a: Perform dot product between the envelope and the cosine of wrapped phases
  - 4b: Add the product to the mean signal and obtain the warped output signal,  $\tilde{\mathbf{x}}[n]$

Table 2. Natural signals used for validation

Signals (name and format available in Matlab2018b)	Fs (Hz)	Duration (sec)
1. Whistles of a train (train.mat)	8192	1.57
2. Chirping of a bird (chirp.mat)	8192	1.5
3. Hallelujah chorus (handel.mat)	8192	8.92
4. Fall and splat sound (splat.mat)	8192	1.22
5. Sound of laughing (laughter.mat)	8192	6.42
6. Male voice speaking (speech_dft_8kHz.wav)	8000	4.99
7. Rock with vocals, drums and guitar (audio48kHz.wav)	48000	8.96

References

- [1] J. L. Flanagan, and R. M. Golden. Phase vocoder. Bell System Technical Journal, 45:1493-1509, 1966.

- [2] J. M. Grey, and J. A. Moorer. Perceptual evaluations of synthesized musical instrument tones. Journal of the Acoustic Society of America, 62:454-462, 1977.
- [3] J. M. Grey, and J. W. Gordon. Perceptual effects of spectral modifications on musical timbres. Journal of the Acoustic Society of America, 63:1493-1500, 1978.
- [4] S. Samson, R. J. Zatorre, and J. O. Ramsay. Multidimensional scaling of synthetic musical timbre: Perception of spectral and temporal characteristics. Canadian Journal of Experimental Psychology, 51(4):307-315, 1997.
- [5] C. L. Krumhansl, and P. Iverson. Perceptual interactions between musical pitch and timbre. Journal of Experimental Psychology: Human Perception and Performance, 18(3):739-751, 1992.
- [6] V. Alluri, and P. Toiviainen. Exploring perceptual and acoustic correlates of polyphonic timbre. Music Perception: An Interdisciplinary Journal, 27(3):223-242, 2010.
- [7] D. Deutsch. The psychology of music. 3<sup>rd</sup> Edition, 2013.
- [8] J. Vos, and R. Rasch. The perceptual onset of musical tones. Perception & Psychophysics, 29(4):323-335, 1981.
- [9] J. A. Moorer. The use of the phase vocoder in computer music applications. Journal of the Audio Engineering Society, 24(9):717-727, 1978.
- [10] M. Dolson. The phase vocoder: A tutorial. Computer Music Journal, 10(4):14-27, 1986.
- [11] J. Laroche, and M. Dolson. New phase vocoder technique for pitch-shifting, harmonizing and other exotic effects. Proc. IEEE Workshop on Applied Signal Processing and Audio and Acoustics, New Paltz, NY, 1999.
- [12] M. Puckette. Phase-locked vocoder. Proc. IEEE ASSP Conference on Applied Signal Processing, Audio and Acoustics, New Paltz, NY, 1995.
- [13] M. S. Puckette, and J. C. Brown. Accuracy of frequency estimates using the phase vocoder. IEEE Transactions on Speech and Audio Processing, 6(2):166-176, 1998.
- [14] T. von Schroeter. Frequency warping with arbitrary allpass map. IEEE Signal Processing Letters, 6(5):116-118, 1999.
- [15] H. Noda. Frequency-warped spectral distance measures for speaker verification in noise. ICASSP, 1:576-579, 1988.
- [16] L. Lee, and R. Rose. A frequency warping approach to speaker normalization. IEEE Transactions on Speech and Audio Processing, 6(1):49-60, 1998.
- [17] S. A. Chang, J. H. Won, H. Kim, S. H. Oh, R. S. Tyler, and C. H. Cho. Frequency-limiting effects on speech and environmental sound identification for cochlear implant and normal hearing listeners. Journal of Audiology and Otology, 22(1):28-38, 2017.
- [18] M. J. Goupell, B. Laback, P. Majdak, and W. D. B. Baumgartner. Effects of upper-frequency boundary and spectral warping on speech intelligibility in electrical simulation. Journal of the Acoustic Society of America, 123(4):2295-2309, 2008.
- [19] J. Bonnel, S. Caporale, and A. Thode. Waveguide mode amplitude estimation using warping and phase compensation. Journal of the Acoustic Society of America, 141(3):2243, 2017.
- [20] H. Niu, R. Zhang, and Z. Li. Theoretical analysis of warping operators for non-ideal shallow water waveguides. Journal of the Acoustic Society of America, 136(1):53-65, 2014.

- [21] B. Chemin, G. Huang, D. Mulders, and A. Mouraux. EEG time-warping to study non-strictly-periodic EEG signals related to the production of rhythmic movements. *J Neurosci Methods*, 308:106-115, 2018.
- [22] L. Wang, J. B. Arends, X. Long, Y. Wu, and P. J. Cluitmans. Seizure detection using dynamic warping for patients with intellectual disability. *IEEE EMBS*, 1010-1013, 2016.
- [23] R. J. Meszlenyi, P. Hermann, K. Buza, V. Gál, and Z. Vidnyánszky. Resting state fMRI functional connectivity analysis using dynamic time warping. *Frontiers of Neuroscience*, 11:75, 2017.
- [24] S. Wabnik, G. Schuller, U. Kramer, and J. Hirschfeld. Frequency warping in low delay audio coding. *ICASSP*, 2005.