INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

A Praat-Based Algorithm to Extract the Amplitude Envelope and Temporal Fine Structure Using the Hilbert Transform

Lei He, Volker Dellwo

Phonetics Laboratory, Department of Comparative Linguistics, University of Zurich, Switzerland
{lei.he|volker.dellwo}@uzh.ch

Abstract

A speech signal can be viewed as a high frequency carrier signal containing the temporal fine structure (TFS) that is modulated by a low frequency envelope (ENV). A widely used method to decompose a speech signal into the TFS and ENV is the Hilbert transform. Although this method has been available for about one century and is widely applied in various kinds of speech processing tasks (e.g. speech chimeras), there are only very few speech processing packages that contain readily available functions for the Hilbert transform, and there is very little textbook type literature tailored for speech scientists to explain the processes behind the transform. With this paper we provide the code for carrying out the Hilbert operation to obtain the TFS and ENV in the widely used speech processing software Praat, and explain the basics of the procedure. To verify our code, we compare the Hilbert transform in Praat with a widely applied function for the same purpose in MATLAB (“hilbert(...)”). We can confirm that both methods arrive at identical outputs.

Index Terms: amplitude envelope, temporal fine structure, Hilbert transform, Praat

1. Introduction

Speech signals are characterized as a composition of both high-frequency carriers containing the temporal fine structures (TFS) which have been modulated at lower frequencies, the amplitude envelopes (ENV) [1]. The roles of TFS and ENV in speech perception have been studied among both normal listeners (e.g. [2, 3, 4, 5, 6]), and hearing impaired listeners, in particular patients with cochlear implants (e.g. [7, 8, 9, 10, 11]). In these studies, the Hilbert transform was applied to segregate the TFS and ENV from the acoustic signal. The mathematical infrastructure was first developed by the mathematician David Hilbert [12], and the application of the transform in communication engineering was first described by Gabor [13]. Signal processing packages containing ready-to-use functions are rare. To the knowledge of the authors, the “hilbert(…)” function in the Signal Processing Toolbox™ of MATLAB® is probably the most common solution to performing the Hilbert transform. However, for many phoneticians, Praat [14] is the first language for signal processing, and the most comfortable platform to perform acoustic analysis, create stimuli, and run perception experiments. The present paper addresses the practical issue of implementing the Hilbert transform to split a speech signal into its ENV and TFS using Praat script lines. Furthermore, the basics of the Hilbert transform and how TFS and ENV are obtained from the transformed signal are explained (Section 2). We compare the Hilbert TFS and ENV in Praat with the “hilbert(...)” function in MATLAB to verify whether they arrive at identical outputs (Section 3).

On way of extracting the Hilbert envelope in Praat is by using the “Sound: To Harmonicity (gne)…” function (which is based on the glottal-to-noise excitation ratio (GNE) algorithm [19]). However, the caveats with this procedure are 1) TFS is not extracted contemporaneously; 2) the result is a Matrix object, requiring further scripting to obtain the ENV; and 3) users are not free to choose filter types. Our script returns both ENV and TFS as Sound objects, and users can opt for different filter types (readily available in Praat are the Hann filter and gammatone filter) before applying our script to different filtered bands.

To perform a Hilbert transform with our Praat script (Figure 1), load the script from the main menu “Praat ▸Open Praat script…”. Then select a sound from the list of objects for which you wish to perform the Hilbert transform to obtain the TFS and ENV. Next, click “Run” in the menu of the script window. The TFS and ENV will be created in the objects list as sound objects.

2. Mathematical basics and Praat code

To extract the TFS and ENV from a time-domain signal $a(t)$, two major steps are required: 1) obtaining the analytic signal of $a(t)$, i.e. $\tilde{a}(t)$, using the Hilbert transform (more precisely, the Hilbert transform computes the imaginary part of $\tilde{a}(t)$, see § 2.1 below); and 2) manipulating $\tilde{a}(t)$ to get the TFS and ENV of $a(t)$. Both steps are explained in sections 2.1 and 2.2 respectively with mathematical formulas and Praat script lines.

2.1. The analytic signal and the Hilbert transform

The analytic signal $\tilde{a}(t)$ is a complex-valued time-domain signal, where the original signal $a(t)$ acts as the real part, and the Hilbert transformed signal $\hat{a}(t)$ acts as the imaginary part:

$$\tilde{a}(t) = a(t) + j\hat{a}(t), \text{where } \hat{a}(t) = \mathbb{H}\{a(t)\} \tag{1}$$

In (1), j is the imaginary number $\sqrt{-1}$, and $\mathbb{H}\{\cdot\}$ denotes the operation of the Hilbert transform.

In the time domain, the Hilbert transform is defined as the convolution of $a(t)$ with $1/\pi t$, which is equivalent to the integration form shown in (2) [15]:

$$\hat{a}(t) = \mathbb{H}\{a(t)\} = \frac{1}{\pi t} * a(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} a(\tau)\frac{1}{t-\tau}d\tau \tag{2}$$

However, directly manipulating the time-domain signal $a(t)$ using formula (2) is difficult using Praat script lines. To avoid working directly on the time-domain signals, we can apply the

http://dx.doi.org/10.21437/Interspeech.2016-1447

convolution theorem and work with their corresponding frequency-domain signals. The theorem states that the convolution of two time-domain signals is equivalent to the product of their corresponding frequency-domain signals yielded from the Fourier transform [15]. Therefore, the Fourier transform of $\hat{a}(t)$, namely $\hat{\mathrm{A}}(\omega)$ can be expressed as:

$$\hat{\mathrm{A}}(\omega) = \mathrm{FH}(\omega) \cdot \mathrm{A}(\omega) \tag{3}$$

where ω is the angular frequency (the relationship between the angular frequency ω and the ordinary frequency f can be expressed as $\omega[\text{rad/sec}] = 2\pi \cdot f[\text{Hz}]$). FH (ω) is the Hilbert transfer function of $1/\pi t$ which takes the form of a piecewise function: it evaluates to $-j$ for all positive frequencies, and j for all negative frequencies [16, 18]. This enables that all negative frequencies of $a(t)$ are shifted by +90°, and all positive frequencies, –90°. In-depth explanations are available in [18, Ch. 9]. A(ω) is the Fourier transform of $a(t)$, which can be expanded as:

$$\mathrm{A}(\omega) = +\int_0^T a(t)\cos(\omega t)\,dt - j\int_0^T a(t)\sin(\omega t)dt \tag{4}$$

where T is the total duration of the signal (i.e., number of samples × sampling period for digitized signal), and $t \in [0, T]$.

To obtain the A(ω) from a selected sound object $a(t)$ in Praat is straightforward by creating its spectrum object as script lines 6-8 in Figure 1 show. Please note that the default setting to use the Fast Fourier Transform (FFT) is disabled in script line 7, because the FFT algorithm zero-pads the sound signal $a(t)$ so that the total number of samples are equal to the nearest 2^{N} ($\mathrm{N} \in \mathbb{Z}^{+}$), resulting in a different signal duration.

Next we multiply the resulting spectrum object with the Hilbert transfer function FH(ω) to get $\hat{\mathrm{A}}(\omega)$ in formula (3). In Praat, only positive frequencies are stored in the spectrum object, hence are available for the users to manipulate [17]. Spurious negative frequencies can be recovered from the trigonometric properties of $\cos(-\vartheta) = \cos(\vartheta)$, and $\sin(-\vartheta) = -\sin(\vartheta)$. Therefore, we only need to multiply the positive part of FH(ω), namely $-j$, with the spectrum of the original signal. In other words, we multiply $-j$ with formula (4) and obtain the Hilbert spectrum of $\hat{\mathrm{A}}(\omega)$:

$$\hat{\mathrm{A}}(\omega) = -\int_0^T a(t)\sin(\omega t)dt - j\int_0^T a(t)\cos(\omega t)dt \tag{5}$$

Comparing formulas (4) and (5), we find that the real part of $\hat{\mathrm{A}}(\omega)$ is equal to the imaginary part of A(ω), and the imaginary part of $\hat{\mathrm{A}}(\omega)$ is equal to the real part of A(ω) times minus 1. Based on this finding, it is possible to manipulate the spectrum object of A(ω) in Praat and obtain $\hat{\mathrm{A}}(\omega)$. Praat stores the spectrum object as a two-row matrix. The real and imaginary parts of the Fourier transform are stored in the first and second rows respectively [17]. By swapping the two rows of A(ω), and multiplying the second row by –1, we get the Hilbert spectrum of $\hat{\mathrm{A}}(\omega)$ in formula (5). This step is accomplished by script lines 10 and 11 in Figure 1. Next, we convert the spectrum object $\hat{\mathrm{A}}(\omega)$ back to its time-domain signal to get $\hat{a}(t)$ (i.e. the Hilbert transformed $a(t)$), aimed at in formula (2). This step is accomplished by script line 12 in Figure 1.

At this point, we have all the components of the time-domain analytic signal $\tilde{a}(t)$. The real part is the original sound signal $a(t)$; the imaginary part is $\hat{a}(t)$. Unlike MATLAB, Praat does not allow complex-valued time-domain signals, so the real and imaginary parts have to be stored as two separate sound objects. Based on this analytic signal $\tilde{a}(t)$, we can calculate the ENV and TFS.

2.2. Getting ENV and TFS from the analytic signal

The ENV of $a(t)$ is defined as the complex modulus (the complex modulus of a complex number $z = a + bj$ ($a, b \in \mathbb{R}$), $||z||$, is defined as $\sqrt{a^2 + b^2}$) of the analytic signal $\tilde{a}(t)$ [18]:

$$\mathrm{ENV}\{a(t)\} \overset{\text{def}}{=} ||\tilde{a}(t)|| = \sqrt{a^2(t) + \hat{a}^2(t)} \tag{6}$$

Geometrically, both real and imaginary parts of the analytic signal $\tilde{a}(t)$ can be represented by the x- and y- axes of a Cartesian plane (i.e. a complex plane). At each sample point of the signal, there exists such a complex plane, from which the ENV value at a particular sample point can be calculated using the Pythagoras theorem (see Figure 2. The ENV values are represented by the dotted arrows). This step can be accomplished using Praat script lines 14 and 15 in Figure 1.

The TFS of $a(t)$ is defined as the cosine of the angle of the analytic signal $\tilde{a}(t)$, namely the angle between the hypotenuse (green arrow, i.e. ENV in Figure 2) and the adjacent (red arrow, i.e. $a(t)$ in Figure 2). The angle can be calculated as the arctangent of $\hat{a}(t)$ over $a(t)$ [18].

$$\mathrm{TFS}\{a(t)\} \overset{\text{def}}{=} \cos\left\{\arctan\left(\frac{\hat{a}(t)}{a(t)}\right)\right\} \tag{7}$$

A trigonometric equivalent of formula (7) is $a(t)$ divided by $\mathrm{ENV}\{a(t)\}$, which is the original signal times its inverse ENV.

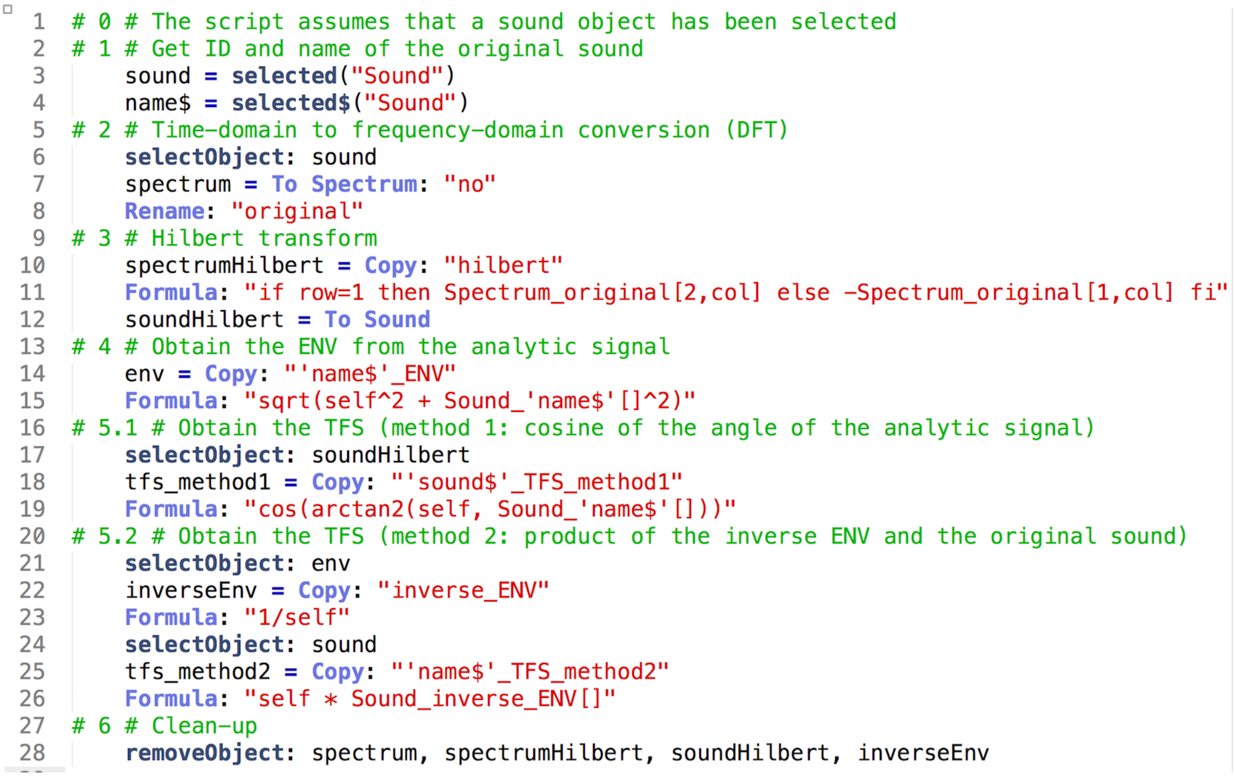
$$\cos\left\{\arctan\left(\frac{\hat{a}(t)}{a(t)}\right)\right\} \equiv \frac{a(t)}{\mathrm{ENV}\{a(t)\}} \tag{8}$$

Both methods to get the TFS can be realized by Praat. Script lines 17-19 calculated the TFS using the method of formula (7); script lines 21-26 extracted the TFS using the method of formula (8). Both methods yield identical results.

At this point, the mathematical basics of the Hilbert transform and the extractions of ENV and TFS have been explained with Praat script lines (Figure 1). In the next section, we verify our script in Figure 1 by comparing the signal processing results with the ones produced by the MATLAB "hilbert(…)" function.

3. Verification of the Praat code

We used three test signals to verify our Praat code (Figure 1) with the MATLAB "hilbert(…)" function.



```
# 0 # The script assumes that a sound object has been selected
# 1 # Get ID and name of the original sound
    sound = selected("Sound")
    name$ = selected$("Sound")
# 2 # Time-domain to frequency-domain conversion (DFT)
    selectObject: sound
    spectrum = To Spectrum: "no"
    Rename: "original"
# 3 # Hilbert transform
    spectrumHilbert = Copy: "hilbert"
    Formula: "if row=1 then Spectrum_original[2,col] else -Spectrum_original[1,col] fi"
    soundHilbert = To Sound
# 4 # Obtain the ENV from the analytic signal
    env = Copy: "'name$'_ENV"
    Formula: "sqrt(self^2 + Sound_'name$'[]^2)"
# 5.1 # Obtain the TFS (method 1: cosine of the angle of the analytic signal)
    selectObject: soundHilbert
    tfs_method1 = Copy: "'sound$'_TFS_method1"
    Formula: "cos(arctan2(self, Sound_'name$'[]))"
# 5.2 # Obtain the TFS (method 2: product of the inverse ENV and the original sound)
    selectObject: env
    inverseEnv = Copy: "inverse_ENV"
    Formula: "1/self"
    selectObject: sound
    tfs_method2 = Copy: "'name$'_TFS_method2"
    Formula: "self * Sound_inverse_ENV[]"
# 6 # Clean-up
    removeObject: spectrum, spectrumHilbert, soundHilbert, inverseEnv
```

Figure 1. *A Praat script to extract the ENV and TFS from a signal using the Hilbert transform. Syntax highlighting was enabled by* [27].

(1) A 440 Hz sine wave modulated by an exponentially decaying envelope (formula: $0.5 \cdot \sin(2\pi \cdot 440t) \cdot e^{-5t}$; duration: 1 second; sf = 44 100 samples/second).

(2) A sinusoidally modulated sinusoid (so called beat signal; formula: $0.3(\sin(2\pi \cdot 440\text{t}) + \sin(2\pi \cdot 443\text{t}))$; duration: 1 second; sf = 44 100 samples/second).

(3) A speech signal in the 85-180 Hz frequency band. Male fundamental frequencies are most likely to fall within this band. But the purpose of choosing this band is purely illustrative in this paper. We did not choose the broad-band either, because the Hilbert transform works well only for signals with narrow frequency bands [26]. The ENV from the broad-band speech signal contains a substantial amount of temporal fine structure information.

The results are shown in Figures 3-5, with the original signals (top signals), ENVs (superimposed on the original signals) and TFSs (bottom signals). The same signals were processed using MATLAB, and the results were visually the same to what we obtained from our Praat code. We further saved the output signals from both platforms as time-series data, and performed correlations on them. The signals from both platforms showed a correlation equal to 1 (see exemplary confirmation for one of these signals in Figure 6).

4. Conclusion

This paper explained the the Hilbert transform and how to extract the ENV and TFS from the transformed signal. We also provide a Praat script for this purpose, which researchers interested in cochlear implant design, speech chimera, or amplitude envelope-based approaches to speech rhythm [20, 21, 22, 23, 24, 25] may find useful. Depending on the need, this script can be easily modified into a procedure to be implemented in a bigger program, or saved as a button in the Praat object window.

5. Acknowledgements

The study is supported by the Gebert Rüf Stiftung (Grant No. GRS-027/13). The authors would like to thank Stuart Rosen and Christian Lorenzi for providing us with a MATLAB script for extracting the Hilbert ENV and TFS, and the reviewers for offering many helpful comments and suggestions.

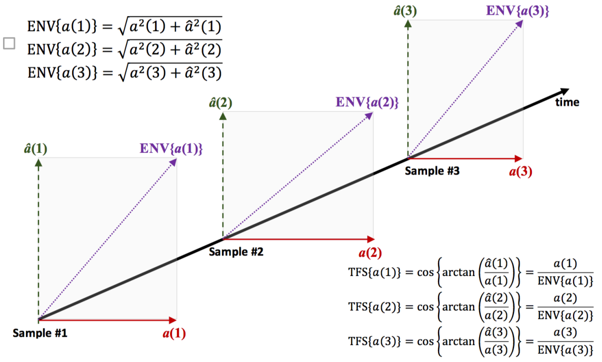

$\text{ENV}\{a(1)\} = \sqrt{a^2(1) + \hat{a}^2(1)}$
$\text{ENV}\{a(2)\} = \sqrt{a^2(2) + \hat{a}^2(2)}$
$\text{ENV}\{a(3)\} = \sqrt{a^2(3) + \hat{a}^2(3)}$
â(3)
ENV{a(3)}
time
â(2)
ENV{a(2)}
Sample #3
a(3)
â(1)
ENV{a(1)}
Sample #2
a(2)
$\text{TFS}\{a(1)\} = \cos\left\{\arctan\left(\frac{\hat{a}(1)}{a(1)}\right)\right\} = \frac{a(1)}{\text{ENV}\{a(1)\}}$
$\text{TFS}\{a(2)\} = \cos\left\{\arctan\left(\frac{\hat{a}(2)}{a(2)}\right)\right\} = \frac{a(2)}{\text{ENV}\{a(2)\}}$
$\text{TFS}\{a(3)\} = \cos\left\{\arctan\left(\frac{\hat{a}(3)}{a(3)}\right)\right\} = \frac{a(3)}{\text{ENV}\{a(3)\}}$
Sample #1
a(1)

Figure 2. *Geometric illustration of calculating the ENV and TFS from an idealized analytic signal. The real parts (i.e. the original signal $a(t)$) are represented by the solid arrows, and imaginary parts (i.e. the Hilbert transformed signal $\hat{a}(t)$) are represented by the dashed arrows.*

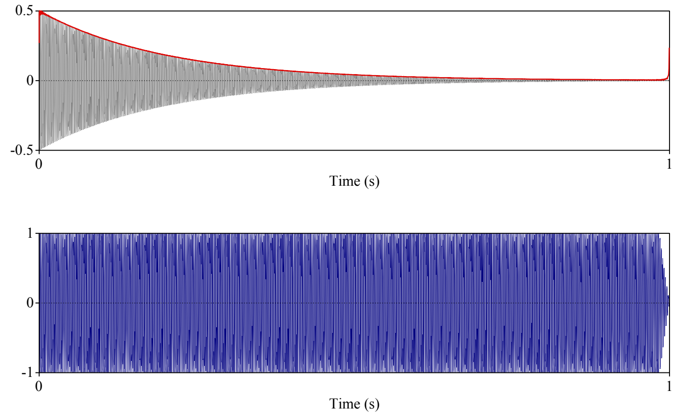


Figure 3. *The ENV (superimposed in red on top signal) and TFS (bottom signal) extracted from an exponentially decaying sine wave (top signal).*

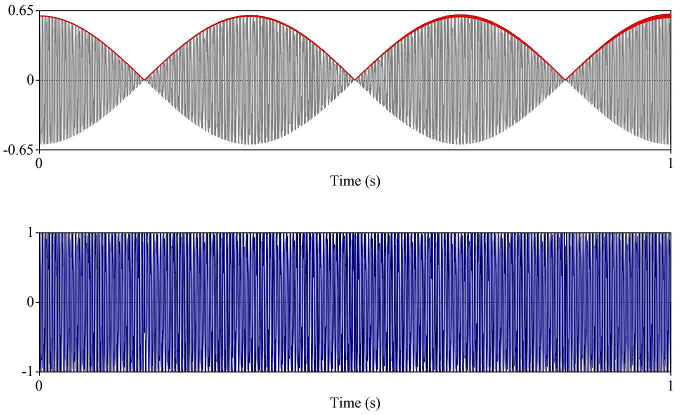


Figure 4. *The ENV (superimposed in red on top signal) and TFS (bottom signal) extracted from a beat signal (top signal).*

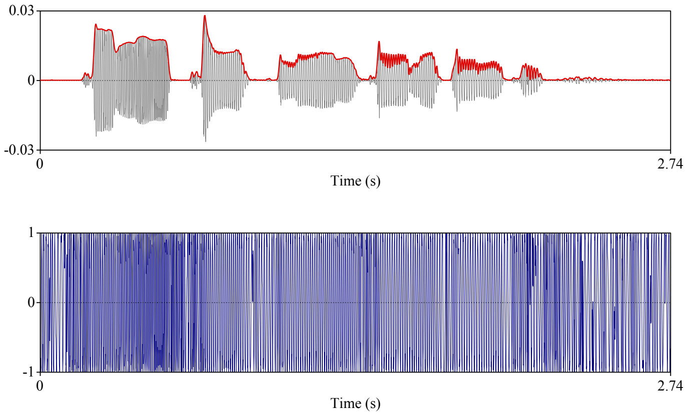


Figure 5. *The ENV (superimposed in red on top signal) and TFS (bottom signal) extracted from a speech signal in the 85–180 Hz frequency band (top signal).*

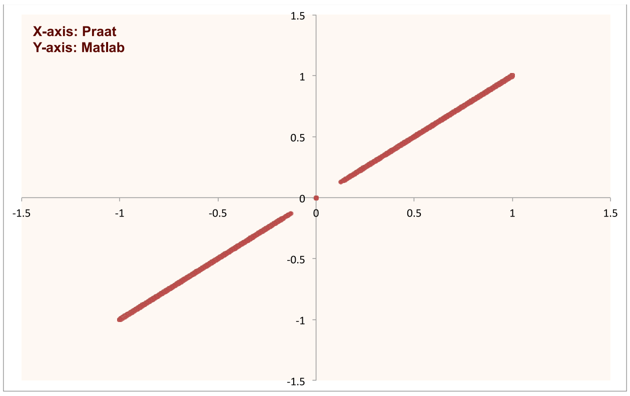


Figure 6. *Scatterplot of the TFS time-series of the test signal #3 (speech) produced using our Praat script in Figure 1 and* MATLAB*. It is an exemplary demonstration that the two platforms arrive at the same outputs. Other test signals also showed the same results, but were not plotted.*

6. References

[1] S. Rosen, "Temporal information in speech: acoustic, auditory, and linguistic aspects," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 336, no. 1278, pp. 367-373, 1992.

[2] Z. M. Smith, B. Delgutte, and A. J. Oxenham, "Chimæric sounds reveal dichotomies in auditory perception," *Nature*, vol. 416, no. 6876, pp. 87-90, 2002.

[3] D. Fogerty, "Perceptual weighting of the envelope and the fine structure across frequency band for sentence intelligibility: effect of interruption at the syllable-rate and periodic-rate of speech," *Journal of the Acoustical Society of America*, vol. 130, no. 1, pp. 489-500, 2011.

[4] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, "Speech recognition with primary temporal cues," *Science*, vol. 270, no. 5234, pp. 303-304, 1995.

[5] L. Xu and B. E. Pfingst, "Relative importance of temporal envelope and fine structure in lexical tone perception," *Journal of the Acoustical Society of America*, vol. 114, no. 6, pp. 3024-3027, 2003.

[6] S. Liu and F.-G. Zeng, "Temporal properties in clear speech perception," *Journal of the Acoustical Society of America*, vol. 120, no. 1, pp. 424-432, 2006.

[7] S. Wang, L. Xu, and R. Mannell, "Relative contributions of temporal envelope and fine structure cues to lexical tone recognition in hearing-impaired listeners," *Journal of the Association for Research in Otolaryngology*, vol. 12, no. 6, pp. 783-794, 2011.

[8] F. Chen and Y.-T. Zhang, "A novel temporal fine structure-based speech synthesis model for cochlear implant," *Signal Processing*, vol. 88, no. 11, pp. 2693-2699, 2008.

[9] B. S. Wilson and M. F. Dorman, "Cochlear implants: current designs and future possibilities," *Journal of Rehabilitation Research and Development*, vol. 45, no. 5, pp. 695-730, 2008.

[10] K. Nie, A. Barco, and F.-G. Zeng, "Spectral and temporal cues in cochlear implant speech perception," *Ear and Hearing*, vol. 27, no. 2, pp. 208-217, 2006.

[11] B. S. Wilson, C. C. Finley, D. T. Lawson, R. D. Wolford, D. K. Eddington, and W. M. Rabinowitz, "Better speech recognition with cochlear implants," *Nature*, vol. 352, no. 6332, pp. 236-238, 1991.

[12] D. Hilbert, *Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen*. Leipzig and Berlin: B. G. Teubner, 1912.

[13] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers–Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429-457, 1946.

[14] P. Boersma and D. Weenink, "Praat: doing phonetics by computer," www.praat.org, 1992-2016.

[15] N. Thrane, "The Hilbert transform," *Technical Review* (Brüel & Kjær), no. 3, pp. 3-15, 1984.

[16] K. Padmanabhan, S. Anandhi, and R. Vijayarajeswaran, *A Practical Approach to Digital Signal Processing*. New Delhi, New Age International, 2001.

[17] Praat online manual, "Sound: To Spectrum…" URL: http://www.fon.hum.uva.nl/praat/manual/Sound__To_Spectrum___.html (Accessed March 23, 2016).

[18] R. G Lyons, *Understanding Digital Signal Processing* 3rd edn. Upper Saddle River: Pearson, 2011.

[19] D. Michaelis, T. Gramss, and H. W, Strube, "Glottal-to-noise excitation ratio – a new measure for describing pathological voices," *Acustica*, vol. 83, no. 4, pp. 700-706, 1997.

[20] V. Leong, M. A. Stone, R. E. Turner, and U. Goswami, "A role for amplitude modulation phase relationships in speech rhythm perception," *Journal of the Acoustical Society of America*, vol. 136, pp. 366-381, 2014.

[21] V. Leong, and U. Goswami, "Impaired extraction of speech rhythm from temporal modulation patterns in speech in developmental dyslexia," *Frontiers in Human Neuroscience*, vol. 8, no. 96, 2014.

[22] S. Tilsen, and A. Arvaniti, "Speech rhythm analysis with decomposition of the amplitude envelope: characterizing rhythmic patterns within and across languages," *Journal of the Acoustical Society of America*, vol. 134, pp. 628-639, 2013.

[23] S. Tilsen, and K. Johnson, K, "Low-frequency Fourier analysis of speech rhythm," *Journal of the Acoustical Society of America*, vol. 124, pp. EL34-EL39, 2008.

[24] V. Dellwo, P. Mok, and M. Jenny, "Rhythmic variability between some Asian languages: results from an automatic analysis of temporal characteristics," In *Proceedings of INTERSPEECH 2014*, Singapore, 2014.

[25] L. He, and V. Dellwo, "The role of syllable intensity in between-speaker rhythmic variability," submitted.

[26] P. L. Søndergaard, R. Decorsière, and T. Dau, "On the relationship between multi-channel envelope and temporal fine structures," In T. Dau, M. L. Jepsen, T. Poulsen, & J. C. Dalsgaard [Eds], Speech Perception and Auditory Disorders (pp. 363-370). Ballerup, Denmark: Danavox Jubilee Foundation.

[27] M. Figueroa, Praat sublime syntax. Accessed on 28 March 2016. https://github.com/mauriciofigueroa/praatSublimeSyntax