Faster than real-time and audio sampling rate extraction of fo candidates using analytic signals with PSWF and cosine series envelope functions

(compiled: 11:12am January 16, 2019)

Hideki Kawahara[1]*, Ken-Ichi Sakakibara*[2]*, Masanori Morise*[3]*, Yuichi Ishimoto*[4]

[1]Wakayama University, Wakayama Japan
[2]Health Science University of Hokkaido, Sapporo Japan
[3]University of Yamanashi, Kofu Japan
[4]National Institute for Japanese Language and Linguistics, Tachikawa Japan

kawahara@sys.wakayama-u.ac.jp, quesokis@gmail.com, mmorise@yamanashi.ac.jp

Abstract

We introduce a set of algorithms and tools for representing local periodicity structure using a set of analytic signals. We introduce several cosine series, PSWF (Prolate Spheroidal Wave Function), and Kaiser windowing functions for their envelope and found that PSWF and a six-term cosine series are the two best choices. .[1] The primary use of them is for designing the excitation source signals of the next generation VOCODERs. The procedure uses efficient instruction sets of vector math libraries, which enabled replacing Flanagan's equation with more direct implementation of the instantaneous frequency and the group delay. The algorithms run faster than real-time and yields fo candidates and event candidates at the audio sampling rate resolution.

One implementation of periodic structure analysis uses an analytic signal with close to isotropic time-frequency resolution for wavelet-based filter arrangement. Integrating the squared group delay and squared temporal derivative of the group delay provides the measure for representing the salience of the periodicity and is proportional to signal to noise ratio and consequently the variance of the frequency.

The other implementation of event structure analysis uses the similar wavelet arrangement of an analytic signal. Combining with an inverse filter for removing responses corresponding resonant frequencies of vocal tract enables clear representation of discontinuities corresponding to excitation events.

We also made tools for real-time visualization of these attributes using MATLAB implementation and off-line tools for detailed analyses. These tools and supporting functions are available as open-source software on GitHub.

Index Terms: speech analysis, speech synthesis, excitation source, periodicity, events, instantaneous frequency, group delay

1. Introduction

We propose an extension of speech signal representation by introducing a generalized tool for periodicity and event analyses. This extension enables analysis and synthesis of less explored aspects of speech signals, excitation source. These excitation attributes enrich speech especially adding para- and non-linguistic contents to speech sounds. These algorithms and tools are designed for fundamental investigation of excitation source attributes. We ask the following fundamental questions. a) Is fo always relevant for representing excitation regularity? b) Is there any perceptual structure for representing irregular vocal fold vibration? c) What is the most relevant description of subharmonic behavior of excitation? d) How to construct excitation signals for reproducing these non-typical excitation sources to stimulate similar perception? e) How to contribute discussion [1] on the terminology of voice attributes?

[1]Release history: 5:33am December 25, 2018, January 5, January 6, January 7, 11:12am January 16, 2019

2. Background

Instantaneous frequency and group delay are useful for analyzing the excitation source of voiced sounds. We introduce a set of simple implementations for calculating these values. Historically their implementations used Flanagan's equation [2] and its variant because those equations do not require phase unwrapping nor inverse trigonometric functions, which have been fragile and inefficient operations.

Recent advances of CPUs and GPUs changed this situation. They provide specialized instruction sets and libraries for those operations (for example, refer to [3]). This situation makes simple, more direct implementations based on the definitions of instantaneous frequency and group delay practical substitutes of the Flanagan's equation.

Phase information is more sensitive to discontinuities (in value, derivative, and higher order derivatives) than power information. We found that a cosine series introduced for deriving closed-form representation of antialiased L-F and F-L models [4, 5] is also relevant for analyzing signal phase and derived attributes [6]. Combining this cosine series and the previously mentioned simple implementation resulted in a procedure that is faster than real-time for calculating those attributes.

3. Instantaneous frequency and group delay

Instantaneous frequency $\omega_i(t)$ of a complex valued signal $x(t) = |x(t)| \exp(j\theta(t))$ is the time derivative of its phase function $\theta(t)$, where $j = \sqrt{-1}$.

$$\omega_i(t) = \frac{d\theta(t)}{dt} \tag{1}$$

Group delay $\tau_g(\omega)$ of a complex valued function $x(\omega) = |x(\omega)| \exp(j\theta(\omega))$ is the (angular) frequency derivative of its phase function $\theta(\omega)$ with the negative sign.

$$\tau_g(\omega) = -\frac{d\theta(\omega)}{d\omega} \tag{2}$$

3.1. Flanagan's equation

By taking logarithm of the $x(t)$ and applying derivative rules yields the following equation to calculate the instantaneous

frequency.

$$\omega_i(t) = \frac{\Re[x(t)]\Im\left[\frac{dx(t)}{dt}\right] - \Re\left[\frac{dx(t)}{dt}\right]\Im[x(t)]}{|x(t)|^2} \quad (3)$$

where $\Re[x]$ and $\Im[x]$ represent the real and the imaginary part of x, respectively. Equation 3 is the Flanagan's equation [2]. Substituting ω with t and adding the negative sign yields the similar equation for the group delay. These equations do not require calculations of phase unwrapping nor inverse trigonometric functions.

3.2. Simple discrete implementation

Argument of the ratio of succeeding samples of a discrete signal $x[n]$ is proportional to the instantaneous frequency $\omega_i[n]$.

$$\omega_i[n] = \angle\left[\frac{x[n+1]}{x[n]} f_s\right], \quad (4)$$

where f_s represents the sampling frequency.

Similarly, argument of the ratio of neighboring samples of a discrete spectrum $X[k]$ is proportional to the group delay $\tau_g[k]$.

$$\tau_g[k] = -\frac{1}{\Delta\omega}\angle\left[\frac{X[k+1]}{X[k]}\right], \quad (5)$$

where $\Delta\omega$ represents the spacing of the discrete (angular) frequencies $\omega[k+1]$ and $\omega[k]$.

3.3. Analytic signal

We used a cosine series envelope of the analytic signal for analyzing the phase of the signal. We found it is necessary to use a relevant envelope when analyzing phase of the signal [6]. The following equation defines the envelope and the analytic signal using the envelope.

$$w_e[n; c_{\mathrm{mag}}, N] = \sum_{m=0}^{M} a_m \cos\left(\frac{2\pi n m}{c_{\mathrm{mag}} N}\right) \quad (6)$$

$$w[n; c_{\mathrm{mag}}, N] = w_e[n; c_{\mathrm{mag}}, N] \exp\left(\frac{2j\pi n}{N}\right) \quad (7)$$

where $M = 5$ represents the highest order of the cosine series, and the support of this function is $[-c_{\mathrm{mag}}N, c_{\mathrm{mag}}N]$. The coefficients of the six-term series are 0.2624710164, 0.4265335164, 0.2250165621, 0.0726831633, 0.0125124215, and 0.0007833203 from a_0 to a_5. The sidelobes have the highest level of -114 dB and the decay rate of -54 dB/oct.

4. Cost functions

We revisit the physical meaning of instantaneous frequency and group delay [7]. Revisiting them provides simple and computationally efficient candidates of cost functions. In this section, we propose to use these cost functions for evaluating candidates of salient frequency and salient event derived from fixed-point analyses [4, 8]. (Actually, the referred articles essentially used these cost functions, in retrospective views.)

5. MATLAB functions

We implemented these functions using MATLAB. We also prepared test scripts for the calibration and evaluation of these implementations. The core functions are as follows:

designCos6Wavelet.m Function to design a set of analytic signals for wavelet analysis.

Table 1: *Duration of envelope functions*

function	duration	length factor	reference & note
six-term	0.4447	3	[10]
Hann	0.2833	1	[12]
Hamming	0.3052	1	[12]
Blackman	0.3565	1.5	[12]
Nuttall-12	0.4021	2	[13] item-12 of Table
Kaiser	0.3619	2	[11] $\beta = 15.03$
DPSS	0.3678	2	[14] $NW = 4.72$

waveletSourceAnalyzer.m Calculates wavelet analysis results using FFT-based efficient convolution.

sourceInformationAnalysis.m Calculates signal attributes using `waveletSourceAnalyzer.m`.

staticTrigBsplinePowerSpec.m Calculates interference-free power spectral representation using fo information [9].

This release provides test scripts for these functions in "test" directory. Appendix A shows excerpts of the test results using scripts. The `sourceInformationAnalysis.m` runs faster than real-time even when calculating fo candidates at the audio sampling rate, for example 44,100 Hz. It runs 12 times faster than real-time with automatic downsampling.

5.1. Updated MATLAB functions: 5, January 2019

We revised core functions to be general and readable. They are as follows:

designAnalyticWavelet.m Function to design a set of analytic signals for wavelet analysis. This a replacement of `designCos6Wavelet.m`.

waveletAttributesAnalyzer.m Calculates wavelet analysis results using FFT-based efficient convolution. This is a replacement of `waveletSourceAnalyzer.m`.

sourceAttributesAnalysis.m Calculates signal attributes using `waveletAttributesAnalyzer.m`. This is a replacement of `sourceInformationAnalysis.m`.

5.1.1. designAnalyticWavelet

This is a generalized version of the previous function `designCos6Wavelet.m`. This function supports the six-term cosine series [10], Hann, Hamming, Blackman, Nuttall, Kaiser [11–13] windowing functions, and the prolate spheroidal wave function [14].

Figure 1 shows the shape of envelope functions using the nominal time axis which is normalized by the carrier frequency $t_c = 1/f_c$ and the frequency gain functions with adjusting to have the same duration $\sigma_{t:X}$ defined below.

$$\sigma_{t:X}(c_{\mathrm{mag}}, f_c) = \sqrt{\frac{\int_{-t_w/2}^{t_w} \tau^2 w_{e:X}^2(\tau; c_{\mathrm{mag}}, f_c) d\tau}{\int_{-t_w/2}^{t_w} w_{e:X}^2(\tau; c_{\mathrm{mag}}, f_c) d\tau}} \quad (8)$$

$$\text{where } t_w = \frac{c_{\mathrm{mag}} K(X)}{f_c} \quad (9)$$

where X represents the name of the envelope function $w_{e:X}^2(\tau; c_{\mathrm{mag}}, f_c)$ and $K(X)$ represents the nominal length factor for determining the domain of definition.

Table 1 shows the duration of envelope functions. The parameters of Kaiser and DPSS (prolate spheroidal wave

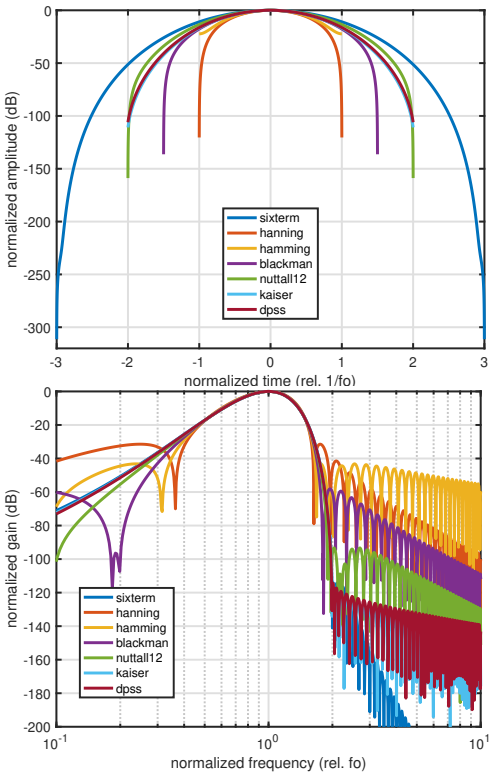

Figure 1: *Envelope function shapes (upper plot) and the gain with duration normalization (lower plot). The stretching factor is* $c_{\mathrm{mag}} = 1.05$

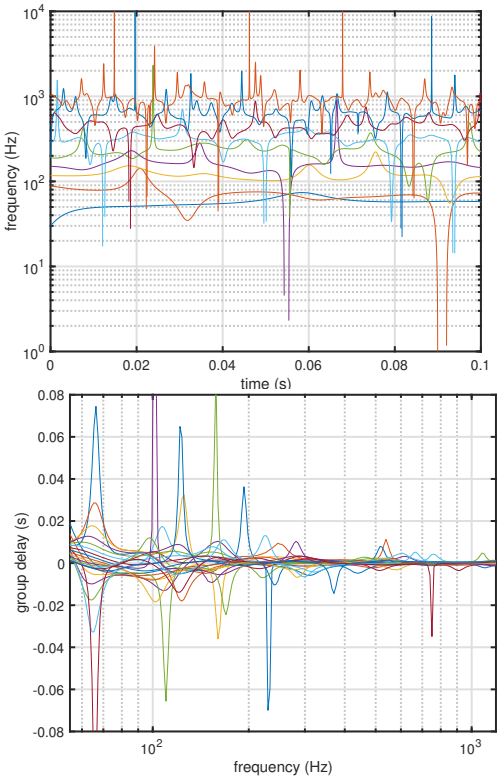

Figure 2: *Spikes in instantaneous frequency and group delay.*

Table 2: *Window length in samples at 44,100 Hz sampling rate*

function	27.5 Hz	40 Hz	55 Hz
six-term	9623	6617	4811
Hann	5035	3462	2517
Hamming	4674	3214	2337
Blackman	5996	4123	2998
Nuttall-12	7092	4877	3546
Kaiser	7878	5417	3938
DPSS	7756	5333	3877

function) make the maximum side-lobe level closest to that of the six-term cosine series. The following modified stretching factor $c_{\mathrm{mag}:X}$ makes analysis using envelope function X obey the same duration condition.

$$c_{\mathrm{mag}:X} = \frac{\sigma_{t:six-term}(c_{\mathrm{mag}}, f_c)}{\sigma_{t:X}(c_{\mathrm{mag}}, f_c)} c_{\mathrm{mag}}. \tag{10}$$

Note that the coefficient to c_{mag} does not depend on c_{mag} and f_c. (Their effects are cancelled out.) It is a constant defined for each envelope function. Table 2 shows impulse response length of the lowest channel for each envelope function. The lengths represented in samples assume 44,100 Hz sampling and $c_{\mathrm{mag}} = 1.0$. Note that for 27.5 Hz and 55 Hz case, the PSWF and the six-term series fit in the different (efficient) FFT buffer lengths, such as 16,384, 8,192, and 4096.

5.1.2. waveletAttributesAnalyzer

This function calculates the convolution using the designed filters. The function also calculates the sample-wise instantaneous frequency and the sample-wise group delay. The previous function calculates using the asymmetric implementation in Eqs. 4, and 5. This updated version also implements the symmetric versions by using squared absolute value weighted averaging. However, in conclusion, they are not very different. The major difference is $0.5/f_s$ delay.

Figure 21 in AppendixB.1 shows the elapsed time with and without symmetric implementation. Because the results are virtually the same and the symmetric implementation takes significant processing time, I commented out the symmetric implementation from the release candidate.

The raw instantaneous frequency and the raw group delay usually have close to singular spikes. Figure 2 shows spikes found in the group delay and the instantaneous frequency. The input signal is a Gaussian white noise. The upper plot shows the instantaneous frequency trajectories. The separation of each trajectory is a half octave. The lower plot shows the group delay. This case, the channel density is 48 channels per octave. These spikes occur when the complex trajectory of filter output passes close to the origin in the complex plane. The Flanagan's equation (Eq. 3) represents how it behaves. The equation also suggests what is the relevant weighting for calculating weighted averages [15, 16]. In short, weighting the square of the absolute value removes these singular behaviors.

5.1.3. sourceAttributesAnalysis

This function calculates weighted average of attributes and cost functions and conducts fixed-point analysis to extract

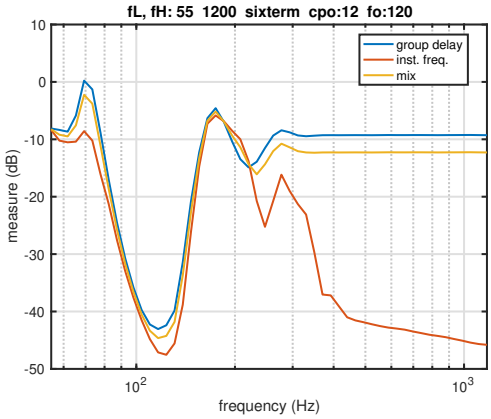

Figure 3: *Average of constituent measures and their average. The input is a sum of a regular pulse train and Gaussian white noise at 30 dB SNR.*

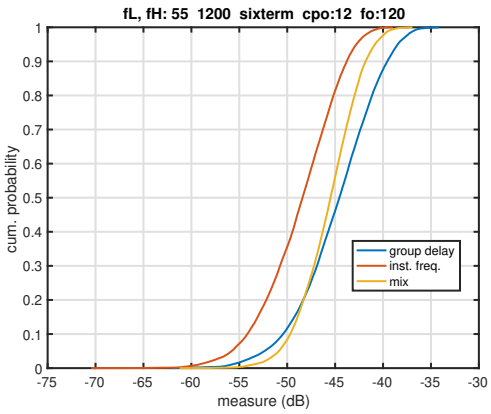

Figure 4: *Cumulative distribution of the constituent measures and the mixed measure of the closest filter to the input fo, 120 Hz.*

periodicity candidates which represents the fundamental component, if it is relevant. Based on a series of preliminary tests, we selected two constituent measure. One is the relative deviation of group delay τ_g and the other is the relative deviation of the speed of the instantaneous frequency change $\frac{df_i(t)}{dt}$. For group delay, $1/f_c$ normalizes the deviation and for the speed of change in instantaneous frequency, f_c^2 normalizes the speed of change.

The following equations provide the definition of constituent measure for each carrier frequency f_c:

$$\eta_{GD}(t, f_c) = \frac{\int_{\Omega_u} w_I(u) f_c^2 \tau_g^2(u, f_c) |y(u, f_c)|^4 du}{\int_{\Omega_u} w_I(u) |y(u, f_c)|^4 du} \tag{11}$$

$$\eta_{IF}(t, f_c) = \frac{\int_{\Omega_u} w_I(u) \frac{1}{f_c^4} \left(\frac{df_i(u, f_c)}{du} \right)^2 |y(u, f_c)|^4 du}{\int_{\Omega_u} w_I(u) |y(u, f_c)|^4 du} \tag{12}$$

$$\text{where } \Omega_u = \left\{ u \,\middle|\, t - \frac{t_I}{2} \le u \le t + \frac{t_I}{2} \right\}, \tag{13}$$

where $y(u, f_c)$ represents the filter output and $w_I(u)$ represents a weight for integration. In this implementation we used Hann windowing function for the weight $w_I(u)$. The mixed measure $\eta_{\text{MIX}}(t, f_c)$ is the average of $\eta_{GD}(t, f_c)$ and $\eta_{IF}(t, f_c)$. Appendix shows their behavior to random inputs and a set of pulse train plus random noise test signals.

$$\eta_{\text{MIX}}(t, f_c) = \eta_{GD}(t, f_c) + \eta_{IF}(t, f_c) \tag{14}$$

Simple mixing of these measures provides a useful measure for finding periodicity in outputs. A periodic pulse train plus noise produces the following profile of outputs. Figure 3 shows an example. The fundamental frequency of the test signal is 120 Hz. The figure illustrates that the outputs of filters which consists of the fundamental component are dominated by the component and behave stable and yield low measure values.

Figure 4 shows the distribution of measure values of the filter output having the closest carrier frequency to the fundamental frequency 120 Hz. The figure indicates that

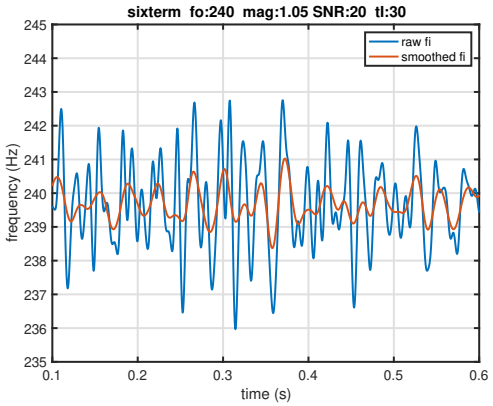

Figure 5: *Raw instantaneous frequency and the power weighted average of the instantaneous frequency of the matching filter output to the input fo, 240 Hz.*

averaging two measures make distribution more concentrated. This is a relevant behavior for evaluating salience of periodicity of the filter output.

This function also calculates a power weighted average of the instantaneous frequency. As also shown in Flanagan's equation, power weighted averaging removes singular behavior of instantaneous frequency.

$$f_{iS}(t, f_c) = \frac{\int_{\Omega_u} w_I(u) f_i(u, f_c) |y(u, f_c)|^2 du}{\int_{\Omega_u} w_I(u) |y(u, f_c)|^2 du}, \tag{15}$$

where $f_i(t, f_c)$ represents the raw output of the instantaneous frequency and $f_{iS}(t, f_c)$ represents the power weighted version of the instantaneous frequency.

Figure 5 shows an example of the raw instantaneous frequency and the power weighted average of the instantaneous frequency. The input signal is a sum of a pulse train of 240 Hz fo and 20 dB SNR. The smoother is Hann window of 30 ms in length.

Figure 6 shows gain transfer functions for several integration time t_I. The legend uses ms for unit. This

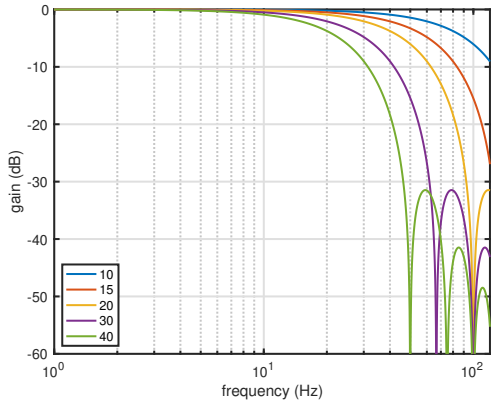

Figure 6: *Gain of smoother for several t_I (in ms).*

smoother assumes that the underlying system which governs the dynamics of fo trajectory is generally stable and does not change at the speed of fundamental frequency. This assumption is not true for chaotic conditions which usually occur around voice onset and offset. We will introduce event based approach for these conditions. At least for fo extraction, we use this post-processing smoother. Behavior of this weighted average for several envelope functions are given in Appendix B.2.6

6. Tools

We introduced a set of tools for real-time interaction with wavelet analyses and post-processing tools. This section provides their roles and a brief introduction of each tool.

6.1. Real-time visualization of wavelet analysis

These tools are for observing the behavior of signal attributes using wavelet analysis. It does not provide any decision results other than the fundamental component. The fundamental component extraction uses an ad hoc stability measure and provides stabilization of waveform display and fine-tuning feedback.

Figure 7 shows the GUI of the real-time visualizer of wavelet analysis. It visualizes the output of a filter bank consisting of the analytic signal impulse response. The GUI has four panels. From top left with counterclockwise, the panels show a) input waveform, b) phase map of the filter output, c) power of filter outputs, and d) fo fine tuner with closest musical note name. It also has several UI controllers. They are "START," "STOP," "SAVE," and "QUIT" buttons, and popup menus for selecting display information (phase, amplitude, instantaneous-frequency, and group delay) and operation modes (Normal or Experimental). The bottom image of Fig. 7 shows the "Phase" information in the "Experimental" mode.

6.1.1. Waveform

The top left panel shows the waveform which corresponds to the center location of the map display below. The waveform center corresponds to the assigned phase of the fundamental component of the signal. The edit box UI controller provides a way to define the assigned phase. The default assigned value is -120 degree, which roughly corresponds to the GCI (Glottal Closure Instant).

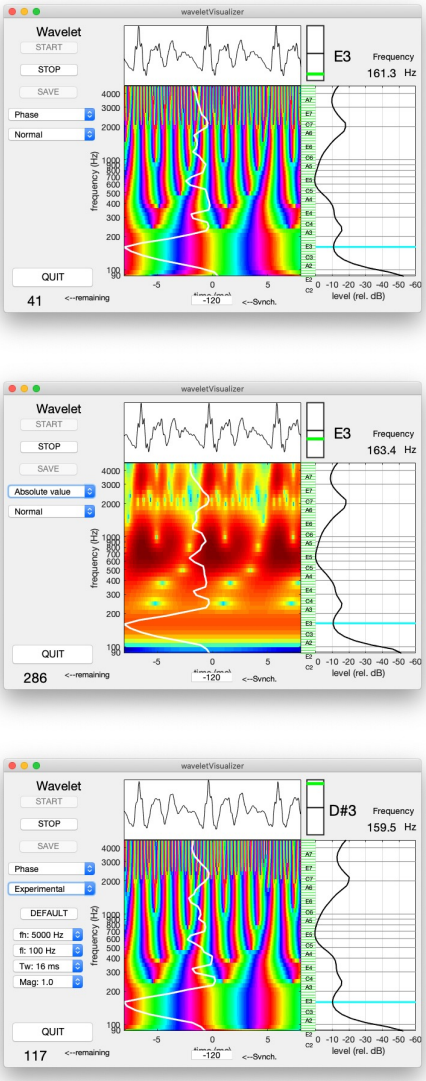

Figure 7: *Realtime wavelet visualizer. From top to bottom, phase view, dB absolute value view and phase view with extra control parameters.*

6.1.2. Attribute map

The lower left panel of each image shows the time-scale map of attributes. Currently, four types of map are available; phase map, amplitude map, instantaneous frequency (deviation) map, and group delay map. This panel has a chromatic scale axis at the right side having green lines indicating the position with a semitone step. It also has musical note names.

The top image of Fig. 7 shows the phase map. The phase map uses "hsv" colormap of MATLAB because the phase and the hue of color share the same cyclic topology.

The middle image of Fig. 7 shows the amplitude map. Since our auditory system has a wide dynamic range in sound level, the pseudo color mapping uses dB representation of the amplitude. This value has a monotonic topology and uses "jet" colormap of MATLAB.

The top image of Fig. 8 shows the normalized instantaneous frequency map. The corresponding center frequency

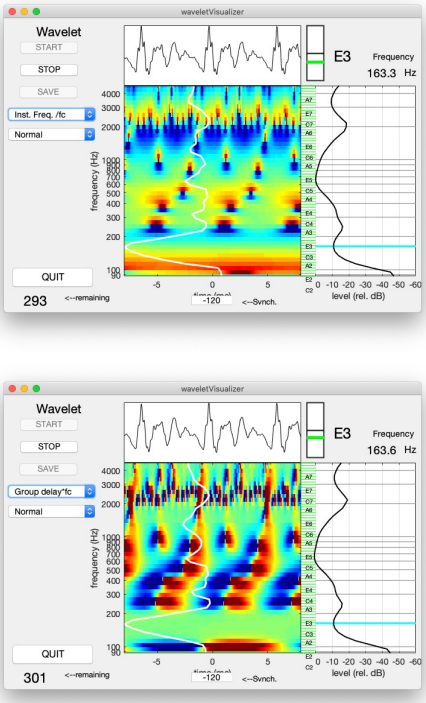

Figure 8: *Realtime wavelet visualiser. Upper image shows the normalized instantaneous frequency map and the lower image shows the normalized group delay map.*

normalized each instantaneous frequency. The topology of this attribute is linear, but the value has a chance to be singular. This map uses "jet" colormap of MATLAB. This map uses logarithmic conversion for mapping to pseudocolor and used truncation inside 1/2 to 2. The color green corresponds to zero.

The bottom image of Fig. 8 shows the normalized group delay. The group delay of each filter uses its center frequency to normalize the value. The topology of this attribute is linear, but the value has a chance to be singular. This map uses "jet" colormap of MATLAB. Since the time axis is linear, we used no logarithmic conversion but used truncation from -0.75 to 0.75. The color green also corresponds to zero.

6.1.3. Level and fo indicator

The right bottom plot of each image shows the averaged power of each filter output. The duration of averaging uses the segment shown in the waveform display and the attribute map display. The cyan line indicates the frequency of the most salient periodic component. Usually, in a voiced segment, it corresponds to conventional fo.

6.1.4. Tuning indicator

The top right corner shows the frequency of the most salient periodic component numerically and visually. The left indicator shows the corresponding musical note name and the deviation of the salient frequency from the reference frequency of the musical note (in equal temperament with the 440 Hz reference). The vertical span of the box is 100 musical cent, and the horizontal black line indicates the frequency of the musical note. The green horizontal line shows the average frequency

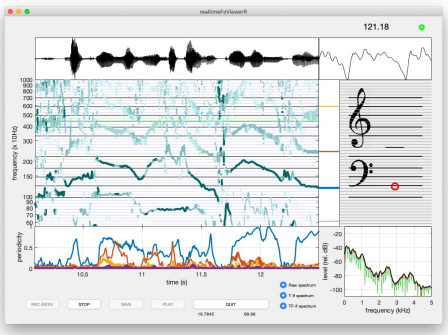

Figure 9: *Realtime visualization of fo candidates.*

of the most salient periodic component.

6.1.5. Experimental mode

The left popup menu provides to change "Normal" mode to "Experimental" mode. In experimental mode, the user can change the following analysis settings: a) High-end frequency of periodicity check, b) Low-end frequency of periodicity check, c) Signal and map viewing width, and d) Temporal stretching factor of the envelope (1 indicate isotropic resolution both in the time and the frequency domain.)

6.2. Real-time fo candidate analysis

Figure 9 shows the GUI of the real-time extractor of fo candidates. The left three panels are horizontally scrolling viewers of (from top to bottom) waveform, fo candidates, and periodicity indicator. They are updated every 50 ms. The top right plot shows the stabilized view of the waveform. The center of the view locks to the zero-phase of the most salient periodic component. Usually, it is the fundamental component. The middle right panel shows the frequency of the most salient component on the musical score-like representation using a red circle. The G and F clefs represent the reference points. The panel has gray horizontal lines representing the chromatic scale (semitone step). The bottom right panel shows the interference-free spectral representations [9] and usual power spectrum using the Blackman window.

The tool keeps running endlessly. The scrolling buffer length is 30 s. When reaching to the end, it initializes and starts again. The user can stop running by clicking the "STOP" button. Then, by using the "PLAY" button, the user can replay the contents of the buffer. By clicking the "SAVE" button, the user can save the contents in the current buffer. The tool generates a unique name for recording the contents. The "START" button restarts the tool.

We designed this tool also for voice therapy and training. The periodicity panel in the bottom provides qualitative feedback about the salience of the voice periodicity. It uses a heuristic conversion to map periodicity into 0 to 1 range (0: completely random, and 1: purely periodic). We are trying to find the better heuristic conversion for this panel.

6.3. Post-processing tool for detailed analysis

We introduced an interactive tool for analyzing speech signals using the analysis functions of signal attributes. The faster than real-time analysis procedures enabled flexible interaction.

Figure 10 shows the scatter plot of the extracted periodicity candidates on the frequency vs. periodicity measure (converted

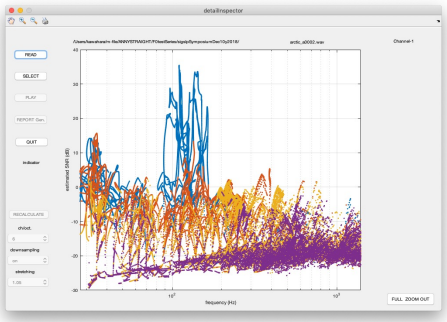

Figure 10: *Scatter distribution of periodicity candidates on the frequency vs. periodicity measure plane.*

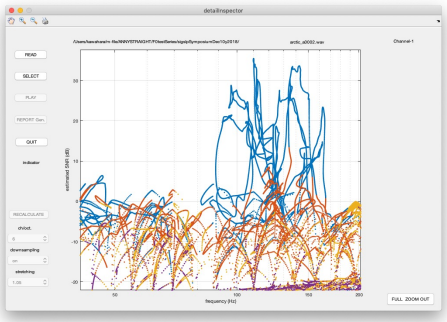

Figure 11: *Scatter distribution of periodicity candidates on the frequency vs. periodicity measure plane. This shows just after region selection.*

to SNR) plane. The tool displays this view just after reading a speech file. The user can select the region where the relevant fo likely exists by using a MATLAB UI tool on the top left corner of the GUI. The tested signal is an excerpt from the CMU ARCTIC database [17] (arctic_a0002.wav by talker bdl.). The signal consists of the speech signal and the EGG signal which are simultaneously recorded.

Figure 11 shows the scatter plot of the extracted periodicity candidates just after selecting the relevant region. If it is actually relevant the user can click the "SELECT" button to proceed to the next step. The user can change the selection using the UI tools on the top left corner of the window.

Figure 12 shows the whole view of the analysis results. The top panel shows the waveform, and the middle panel shows the frequency of the candidates. The bottom panel shows the estimated SNR of the candidates. Then the user can select a region to inspect in detail. After selecting the focused region, click of the "APPLY" botton broadcasts the selected region to the other panels.

Figure 13 shows the magnified view of the selected region. The default setting uses automatic downsampling for analysis. By selecting "on" of the downsampling popup menu and clicking the "RECALCULATE" bottun starts the audio sampling rate analysis.

Figure 14 shows the magnified view of the selected region without downsampling. In the magnified mode, by clicking the "PLAY" button, the user can playback the sound of the displayed portion.

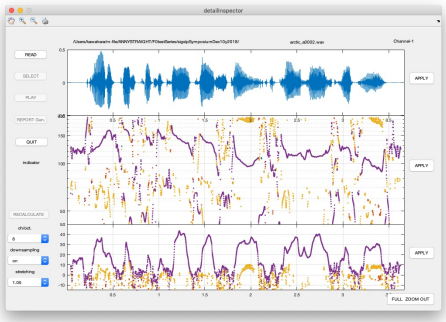

Figure 12: *Whole view display of the analysis results. The top panel shows the waveform, and the middle panel shows the frequency of the candidates. The bottom panel shows the estimated SNR of the candidates.*

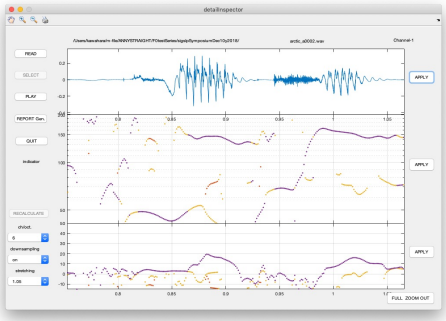

Figure 13: *Selected view display of the analysis results. The top panel shows the waveform, and the middle panel shows the frequency of the candidates. The bottom panel shows the estimated SNR of the candidates.*

Figure 15 shows the EGG signal analysis resluts using the similar setting. In the voiced part, both the speech and the EGG analysis results behave similarly.

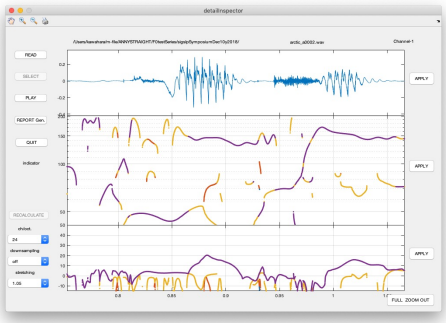

Figure 14: *Selected view display of the analysis results without downsampling. The top panel shows the waveform, and the middle panel shows the frequency of the candidates. The bottom panel shows the estimated SNR of the candidates.*

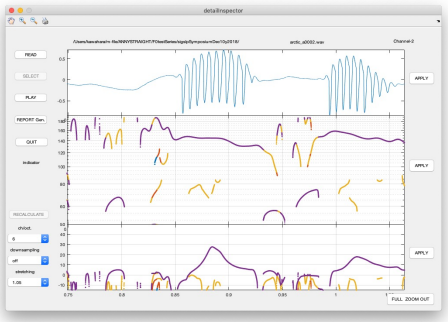

Figure 15: *Selected view display of the analysis results without downsampling. The analyzed signal is the simultaneously recorded EGG. The top panel shows the waveform, and the middle panel shows the frequency of the candidates. The bottom panel shows the estimated SNR of the candidates.*

7. Discussion

The core functions are general enough to be used in a wide range of applications. This implementation uses a log-linear filter setting. However, it is possible to design filter bank elements using more general frequency axes and frequency resolution [18, 19] The current release does not have event related functions, which will be available soon.

8. Conclusions

We introduced efficient core functions using simpler implementation of phase related attributes than Flanagan's equation. We also introduced interactive (and some are in real-time) tools by making use of this efficient implementation. The tools and constituent functions implemented using MATLAB are accessible online in GitHub. We are hoping users to acquire deeper understanding and grasp of phase related signal attributes.

9. Acknowledgements

This research was supported by KAKENHI (Grant in Aid for Scientific Research by JSPS) 16H01734 and 15H03207 and 18K00147.

10. References

[1] I. R. Titze, R. J. Baken, K. W. Bozeman, S. Granqvist, N. Henrich, C. T. Herbst, D. M. Howard, E. J. Hunter, D. Kaelin, R. D. Kent, J. Kreiman, M. Kob, A. Löfqvist, S. McCoy, D. G. Miller, H. Noé, R. C. Scherer, J. R. Smith, B. H. Story, J. G. Švec, S. Ternström, and J. Wolfe, "Toward a consensus on symbolic notation of harmonics, resonances, and formants in vocalization," *The Journal of the Acoustical Society of America*, vol. 137, no. 5, pp. 3005–3007, 2015.

[2] J. L. Flanagan and R. M. Golden, "Phase Vocoder," *Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, Nov 1966.

[3] Intel, "Vector Mathematics (VM): Performance and Accuracy Data," 2018, (Access date: 2018-10-12). [Online]. Available: https://software.intel.com/sites/products/documentation/

[4] H. Kawahara, H. Katayose, A. d. Cheveigné, and R. D. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity," in *Sixth European Conference on Speech Communication and Technology*, 1999, pp. 2781–2784.

[5] H. Kawahara, K.-I. Sakakibara, M. Morise, H. Banno, and T. Toda, "A modulation property of time-frequency derivatives of filtered phase and its application to aperiodicity and fo estimation," in *Proc. Interspeech 2017*, 2017, pp. 424–428.

[6] H. Kawahara, "Pitfalls in digital signal processing," *The Journal of the Acoustical Society of Japan*, vol. 73, no. 9, pp. 592–599, 2017, [in Japanese].

[7] L. Cohen, *Time-frequency analysis: Theory and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[8] H. Kawahara, Y. Atake, and P. Zolfaghari, "Accurate vocal event detection method based on a fixed-point analysis of mapping from time to weighted average group delay." in *Icslp-2000*, Beijing, China, 2000, pp. 664—-667.

[9] H. Kawahara, M. M., and K. Hua, "Revisiting spectral envelope recovery from speech sounds generated by periodic excitation," in *APSIPA ASC 2018, Hawaii*. APSIPA,, 2018, pp. 1674–1683.

[10] H. Kawahara, K.-I. Sakakibara, M. Morise, H. Banno, T. Toda, and T. Irino, "A new cosine series antialiasing function and its application to aliasing-free glottal source models for speech and singing synthesis," in *Proc. Interspeech 2017*, Stocholm, August 2017, pp. 1358–1362.

[11] J. Kaiser and R. W. Schafer, "On the use of the I_0-sinh window for spectrum analysis," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 1, pp. 105–107, 1980.

[12] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[13] A. H. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. Audio Speech and Signal Processing*, vol. 29, no. 1, pp. 84–91, 1981.

[14] D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty-I," *Bell System Technical Journal*, vol. 40, no. 1, pp. 43–63, 1961.

[15] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, 1992.

[16] ——, "Estimating and Interpreting the Instantaneous Frequency of a Signal - Part 2: Algorithms and Applications," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 540–568, 1992.

[17] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[18] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing research*, vol. 47, no. 1-2, pp. 103–138, 1990.

[19] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Jan 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17112

[20] Y. Atake, T. Irino, H. Kawahara, J. Lu, S. Nakamura, and K. Shikano, "Robust estimation of fundamental frequency using instantaneous frequencies of harmonic components," *IEICE Trans. D. Information*, vol. J83-D2, no. 11, pp. 2077–2086, 2000, [in Japanese].

A. Test resuls: outdated: 05/Jan./2019

This appendix shows excerpts of the test results. Figure 16 shows the speed test results. The sampling frequency is 44,100 Hz, and the periodicity search range is from 55 Hz to 1000 Hz with 12 channels per octave arrangement. The temporal stretching factor is 1. The results indicate that it runs three times faster than real-time without downsampling. It also shows that it runs 12 times faster than real-time with automatic downsampling.

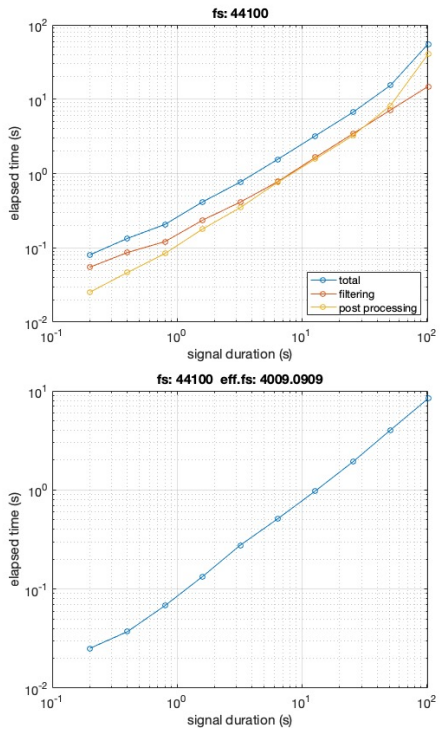

Figure 16: *Speed test results. The pper plot shows the speed without using downsampling. The lower plot shows the speed with automatic downsampling. The horizontal axis shows the duration of the test signal and the vertical axis shows the elapsed time.*

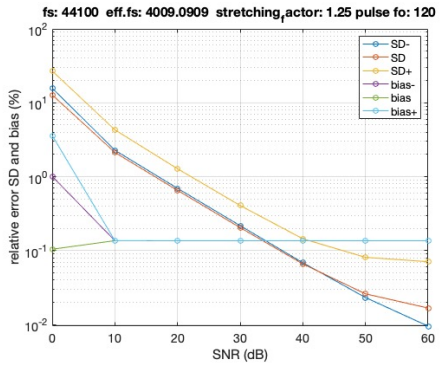

Figure 17: *Frequency estimation error and signal SNR.*

Figure 17 shows the standard deviation and the bias of the calculated frequency. The test signal is a periodic pulse train plus Gaussian white noise. The horizontal axis represents the local SNR and the vertical axis shows the standard deviation and the bias. The plots shows the best filter output and the surrounding filter outputs.

Figure 18 shows the periodicity measure and the input SNR. he test signal is a periodic pulse train plus Gaussian white noise. The average of measure and the SNR have close to linear relation from 5 to 45 dB SNR. The standard deviation of the measure is about 3 dB indicating that it is a reliable measure.

Figure 19 shows the standard deviation and the bias in the

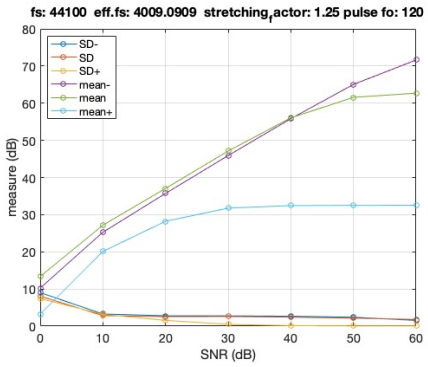

Figure 18: *Average of the periodicity measure and the standard deviation.*

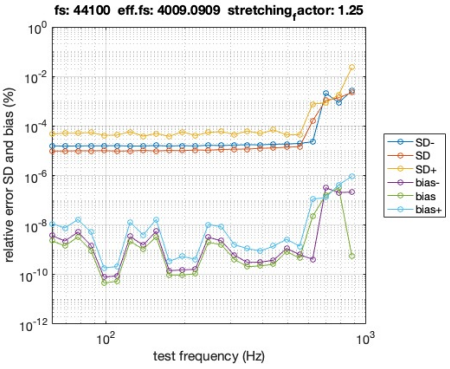

Figure 19: *The standard deviation and the bias in frequency calculation without any additive noise.*

calculated frequency when no additive noise exists. The results indicate that they are negligible for usual situations.

Figure 20 shows the response to FM. This indicates that this procedure can track the fo trajectory of vibrato accurately.

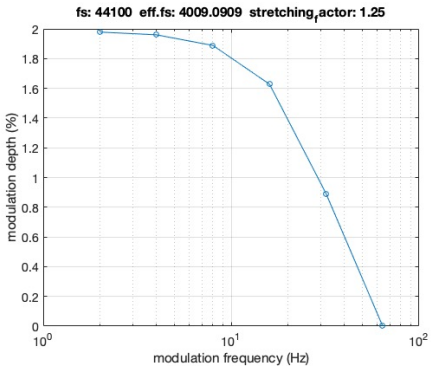

Figure 20: *Response to FM. The horizontal axis represents the modulation frequency. The modulation depth of the test signal is 2.07%.*

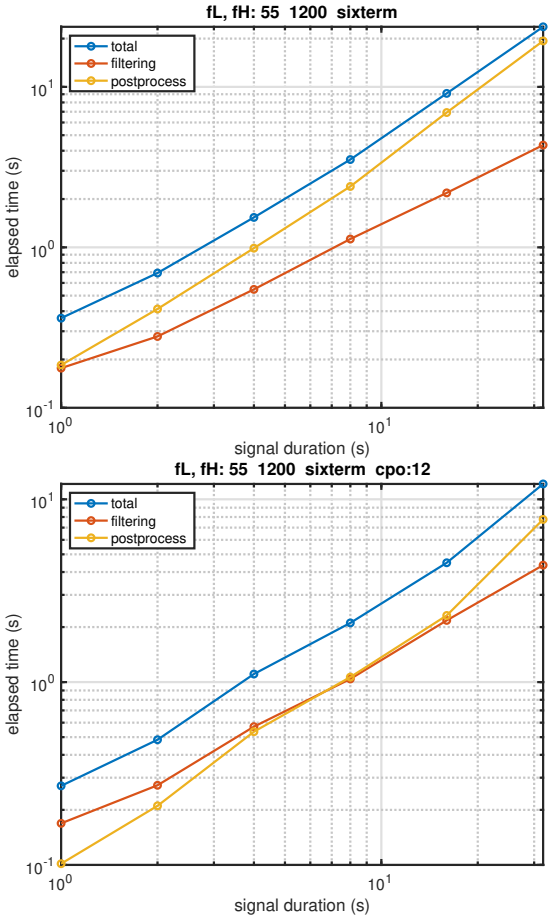

Figure 21: *Processing speed at 44,100 Hz sampling. The upper plot shows results with symmetric implementation of instantaneous frequency and group delay. The lower plot shows results without them.*

B. Test results: 5, January 2019

B.1. Speed test: waveletAttributesAnalyzer.m

We used to test the processing speed using a Gaussian noise input. The tested sampling frequency is 44,100 Hz and the channel density is 12 channels per octave.

Figure 21 compares with and without calculating the symmetric implementations of the instantaneous frequency and the group delay in the discrete signal domain. These figures show that the elapsed time for filtering is close to linear. However, the elapsed time for post processing is growing faster than the linear relation.

B.2. Measure test: sourceAttributesAnalysis.m

The signal attributes extraction uses these updated core functions. We conducted tests to calibrate measures and check for distribution of estimates.

B.2.1. SNR and measure

A series of tests provide information to estimate SNR based on calculated periodicity measure $\eta_{\mathrm{MIX}}(t, f_c)$.

Figure 22 shows the results using mixtures of periodic pulse trains and Gaussian white noise. Their fundamental frequencies

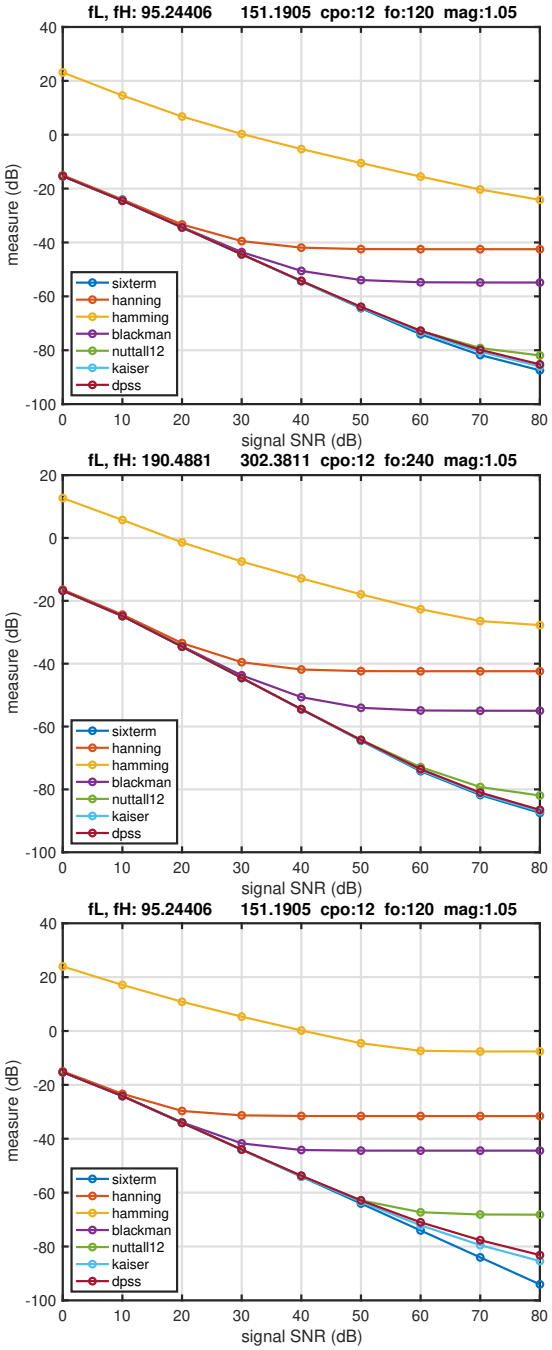

Figure 22: *Signal SNR and the average value of the measure. The top and middle panel show the output of matching channel. The bottom panel shows the output of a neighboring channel. Plots show results using seven types of envelope functions.*

are 120 Hz and 240 Hz representing average male and female voices. We tested seven envelope functions. Their durations are set equal by using the effective stretching factor $c_{\mathrm{mag}:X}$ given by Eq. 10. The vertical axis of the plots show the mean of

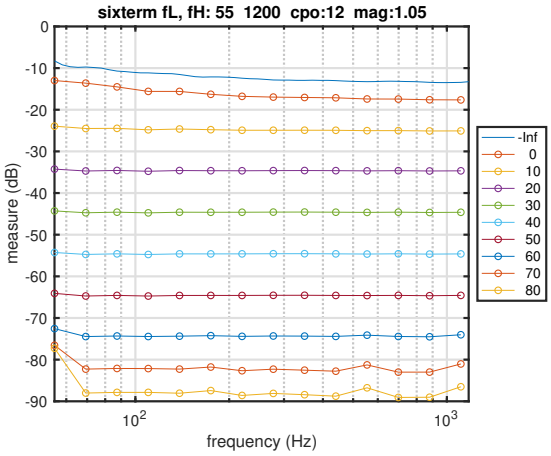

Figure 23: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is the six-term series.*

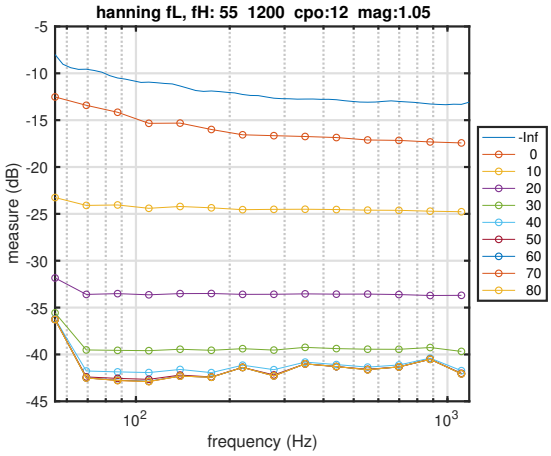

Figure 24: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is Hann windowing function.*

periodicity measure $\tilde{\eta}_{\mathrm{dB}}(f_c)$ defined below.

$$\tilde{\eta}_{\mathrm{dB}}(f_c) = 10\log_{10}\left(\int_{t_{mg}}^{T-t_{mg}} \eta_{\mathrm{MIX}}(t, f_c)dt\right), \qquad (16)$$

where T represents the duration of the test signal and t_{mg} represents the margin for eliminating the transitional part. In these tests, we set the value $t_{mg} = 0.2$ s.

The results indicates that the average measure is independent of the signal fundamental frequency. These figures also show that commonly used windows such as Hann, Hamming and Blackman are not relevant for evaluating periodicity for winder SNR range.

For envelope functions having highly attenuated side-lobes (Nuttall-12, Kaiser, PSWF, and the six-term cosine series) the derived measures show relevant linear relation to the given SNR. As shown in the bottom panel of Fig. 22 functions other than the six-term one deviate from linear relation. However, their deviations are not problematic for speech applications.

B.2.2. Dependency on fo

We tested the measure dependency on the fundamental frequency of the input test signal with SNR as the test parameter. The SNR test range is from 0 dB to 80 dB. The sampling frequency of the test signal is 44,100 Hz. A test signal is a sum of a regular pulse train and Gaussian white noise. It also shows noise only condition. It is indicated as "-Inf" condition. The fundamental frequency range is from 55 Hz to 1200 Hz in a 1/3 octave step. The actual frequencies are slightly shifted to make pulse intervals integer.

Figure 23 shows the results for the six-term series. The vertical axis represents the averaged measure defined by Eq. 16. It indicates that the measure is independent on the fundamental frequency for SNR higher than 10 dB. This is relevant behavior for periodicity detector. The plot also shows the average of measure for random signal input. The line indicated by "-Inf" in the legend shows it.

Figure 24 shows the results for Hann windowing function. The measure saturates at 30 dB SNR. This measure is useless handling signals which have 30 dB or higher SNR.

Figure 25 shows the results for Hamming windowing function. It introduces strong fo dependency. Hamming

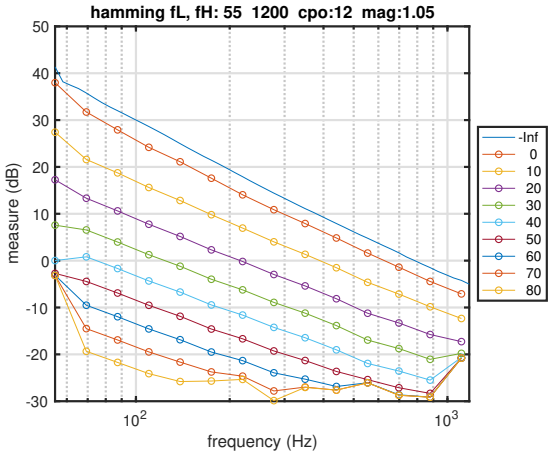

Figure 25: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is Hamming windowing function.*

windowing function is not relevant for deriving SNR related measure.

Figure 26 shows the results for Blackman windowing function. It saturate around 40 dB SNR. This envelope can be useful, if usual speech does not have higher than 40 dB SNR. It is necessary to test high-quality recording data to judge relevance of this envelope function. This function is attractive, because the window length is short.

Figure 27 shows the results for Nuttall windowing function. This Nuttall windowing function is not the MATLAB built-in function. It is 12-th item of Table in the reference [13]. The saturation level of this function is about 65 dB. In terms of the saturation level, this function is relevant.

Figure 28 shows the results for Kaiser windowing function. This function shows linear behavior to SNR up to 70 dB. Kaiser widowing function is an approximation of the following PSWF. The window length is slightly longer than corresponding PSWF.

Figure 29 shows the results for the discrete version of PSWF (Prolate Spheroidal Wave Function) named DPSS (Discrete Prolate Spheroidal Sequences). This function also

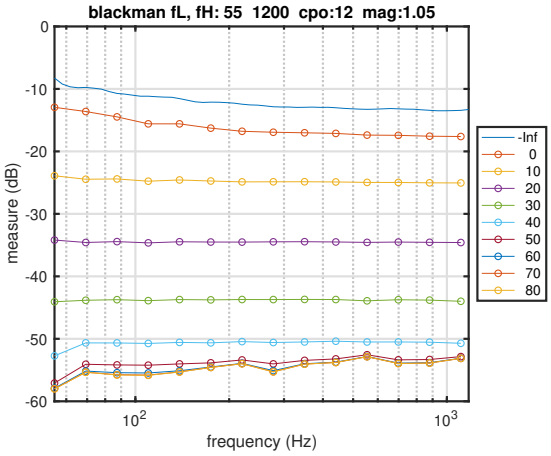

Figure 26: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is Blackman windowing function.*

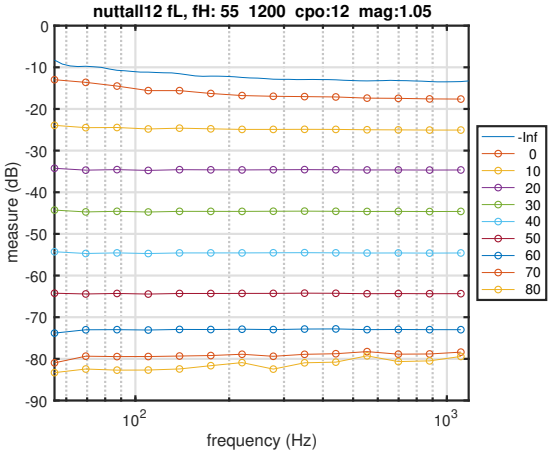

Figure 27: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is Nuttall's four term window (12-th item of Table).*

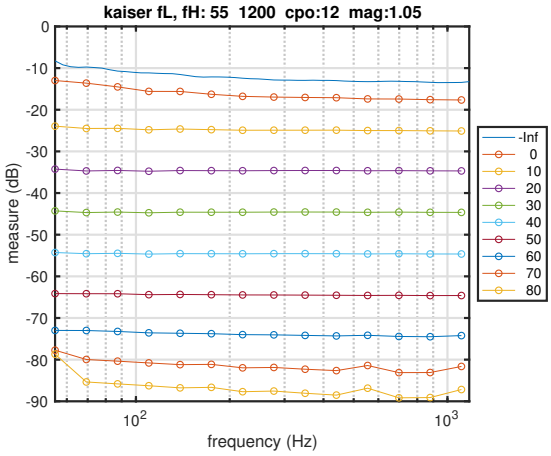

Figure 28: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is Kaiser windowing function.*

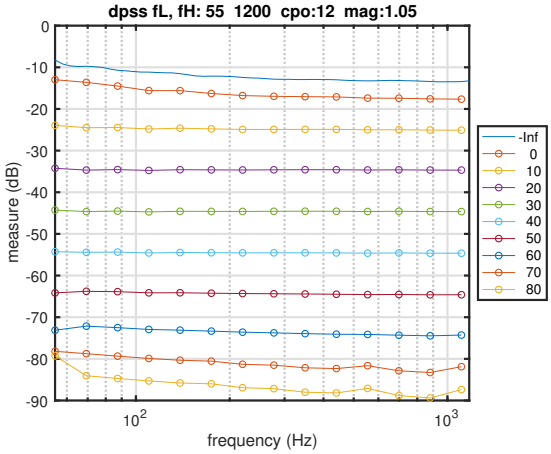

Figure 29: *Measure value dependency on the fundamental frequency of the input with SNR for parameter. The envelope is the first DPSS.*

provides linear relation to SNR up to 70 dB. The window length is the shortest of relevant functions. In periodicity analysis application, the only disadvantage of this function, heavy computational demand, is not a problem.

B.2.3. Distribution of the measure

The previous section showed that relevant envelope functions do not introduce fo dependency in the periodicity measure. This section checks detail of the distribution. We tested at 120 Hz and 240 Hz fundamental frequencies for test signals, because they represent usual male and female voices. Then, we found the results using 120 Hz and 240 Hz behave similarly. Therefore, this section only shows the results using the 120 Hz test signal.

Figure 30 shows the measure distribution for the six-term series. The vertical axis shows the cumulative probability of the measure is smaller than the value on the horizontal axis. The plot shows that this windowing function does not saturate.

Figure 31 shows the measure distribution for Hann windowing function. The plot shows clear saturation.

Figure 32 shows the measure distribution for Hamming windowing function. The plot seems not show saturation up to 60 dB. However, the measure values are substantially higher than other envelope functions and the relation to SNR is less than linear.

Figure 37 summarizes the distribution results. The results shown in this plot is for 240 Hz test signals. The plot for 120 Hz is essentially the same. This plot shows the spread of the measure distribution for each SNR level. The spread is represented as the distance between 0.5% and 99.5% on the cumulative distribution. The spread shown in this plot is about half of our Interspeech 2017 paper [5].

Important difference between the paper [5] and this implementation is the power weighted averaging of instantaneous frequency and group delay. This weighting completely removed glitches in other windowing functions than the six-term series, which I pointed out in 2017 [6].

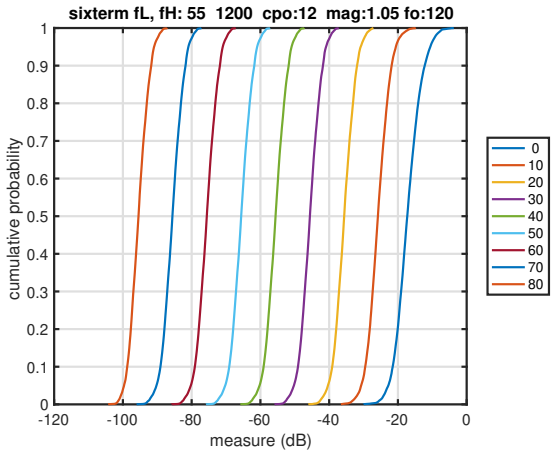

Figure 30: *Cumulative distribution of the measure for different input SNR. The envelope is the six-term series.*

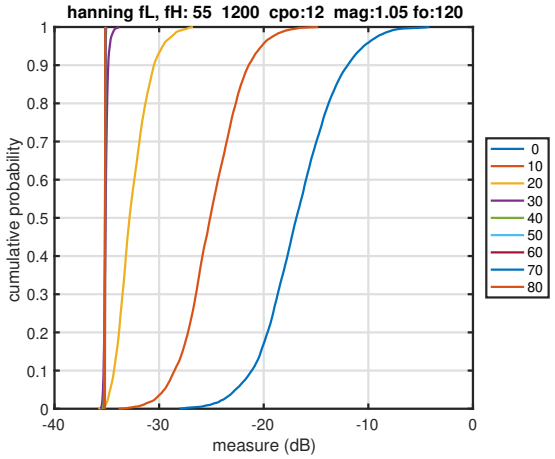

Figure 31: *Cumulative distribution of the measure for different input SNR. The envelope is Hann windowing function.*

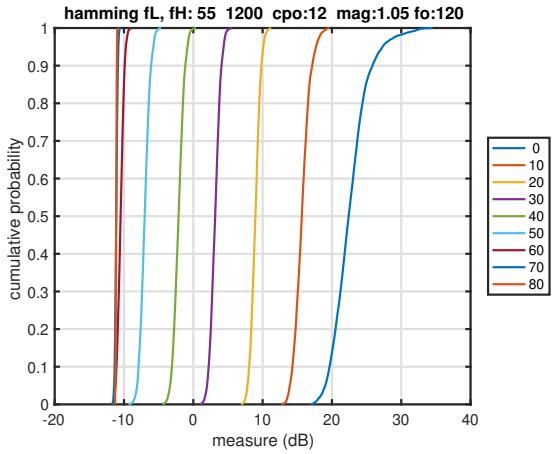

Figure 32: *Cumulative distribution of the measure for different input SNR. The envelope is Hamming windowing function.*

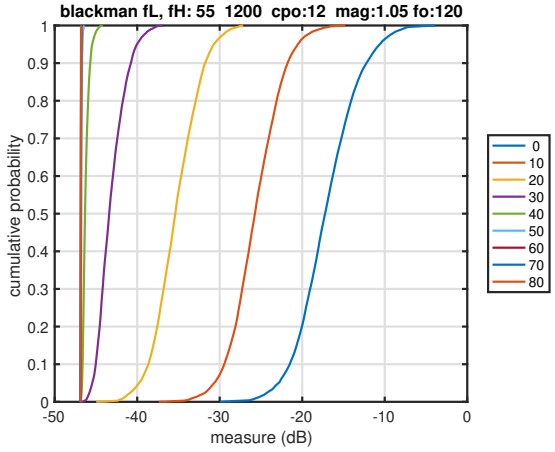

Figure 33: *Cumulative distribution of the measure for different input SNR. The envelope is Blackman windowing function.*

B.2.4. Downsampling rate tuning

For realtime applications, we introduce downsampling option. The initial implementation set the downsampling rate using the following statement:

```
downSamplinbgRate = ...
    max(1, floor (fs / (high_freuency * 4)));
```

This decision was not based on careful tests. Using six-term series envelope with this downsampling yielded the following.

Figure 38 shows the effects of Hann's side-lobes. It introduced saturation.

By changing the target upper frequency using the following statement:

```
downSamplinbgRate = ...
    max(1, floor (fs / (high_freuency * 6)));
```

It yielded slightly better behavior. However, effects of side-lobes are not negligible. Figure 39 shows the results using smaller downsampling rate. The antialiasing function is the same; Hann windowing function.

Finally, we replaced the antialiasing function. The following results are using Blackman windowing function for antialiasing. Figure 40 shows the results using Blackman windowing function. The effects of side-lobes are negligible up to 60 dB SNR, which is practically acceptable.

The following figures show results using different envelope functions using Blackman windowing function for antialiasing. Figure 41 shows the results using Hann windowing function. Figure 42 shows the results using Hamming windowing function. Figure 43 shows the results using Blackman windowing function. Figure 44 shows the results using Nuttall's four term windowing function. Figure 45 shows the results using Kaiser windowing function. Figure 46 shows the results using the first DPSS windowing function.

B.2.5. SNR and frequency deviation

A series of tests provide relation between signal SNR and the standard deviation of the yielded instantaneous frequency.

Figure 47 shows the results for 120 Hz test signal. The results are with and without smoothing. The duration of all envelope functions are set the same. It was surprising that there are no difference between envelope functions with

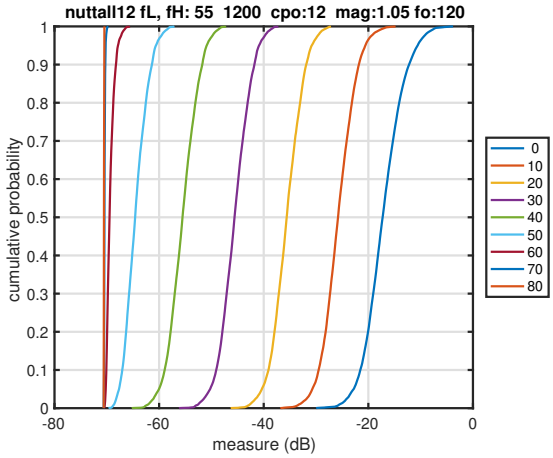

Figure 34: *Cumulative distribution of the measure for different input SNR. The envelope is Nuttall's four term windowing function.*

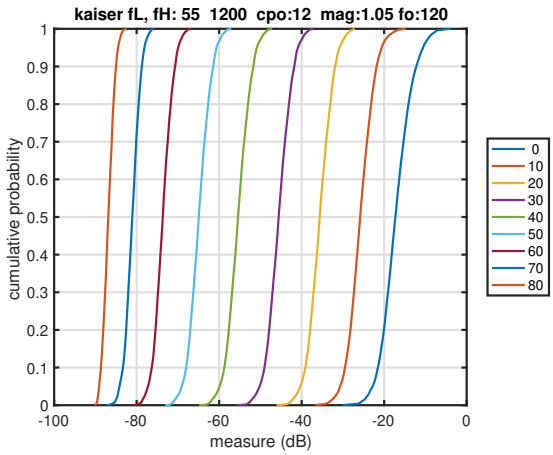

Figure 35: *Cumulative distribution of the measure for different input SNR. The envelope is Kaiser windowing function.*

smoothing. The results without smoothing illustrates that there exists difference between envelope functions. These indicates that deviation components which make differences between envelope functions are from leakage of side-lobes. Those deviation components are essentially rapidly changing and filtered out by the smoother.

The vertical axis of the plots uses RMS (musical) cent for representing the standard deviation. It is 1200 times of $\log_2(\ )$ of frequency.

Figure 48 shows the results for the test signal with 240 Hz fundamental frequency. The standard deviation of the smoothed signal is smaller than that of 120 Hz test signal. This is because the higher fo introduces more fast varying components. This is a merit of this procedure (possibly) over DIO.

B.2.6. Fo trajectory shape

Figure 49 shows power spectra of the output instantaneous frequencies.

B.2.7. Onset and offset behavior

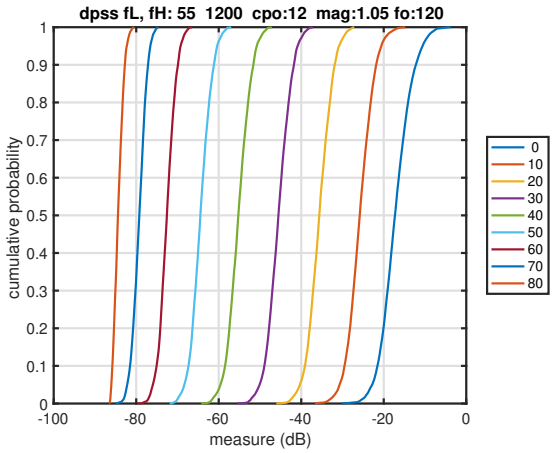

Figure 36: *Cumulative distribution of the measure for different input SNR. The envelope is the first of DPSS.*

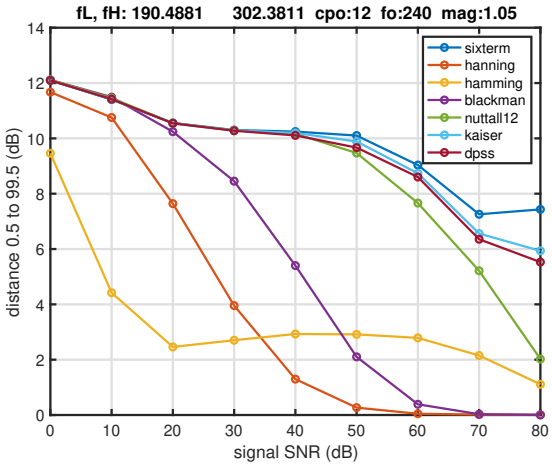

Figure 37: *Summary of the distribution test results. The vertical axis shows the distance of measure values at 0.5% and 99.5% on the cumulative distribution.*

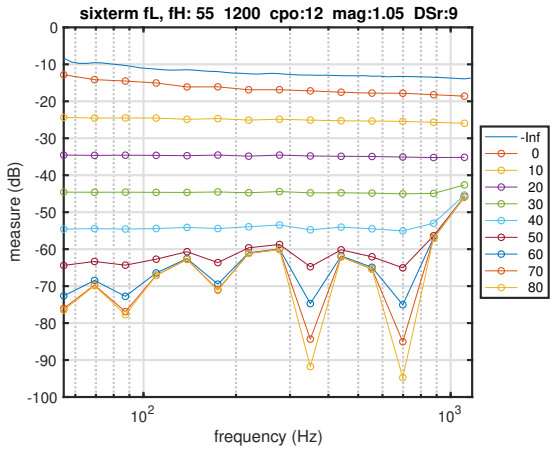

Figure 38: *Fo dependency test using the six-term series. The smoothing function is Hann windowing function.*

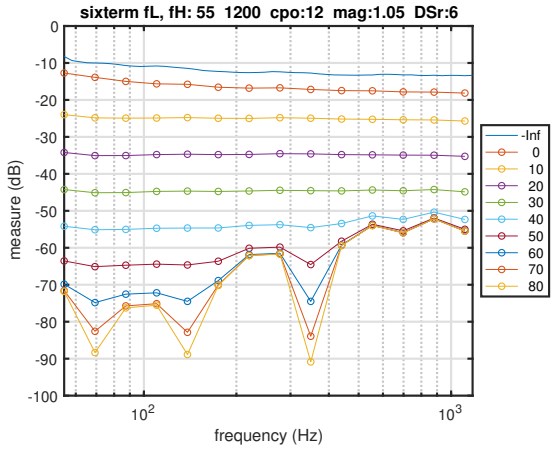

Figure 39: *Fo dependency test using the six-term series for envelope. The smoothing function is Hann windowing function. This version uses smaller downsampling rate.*

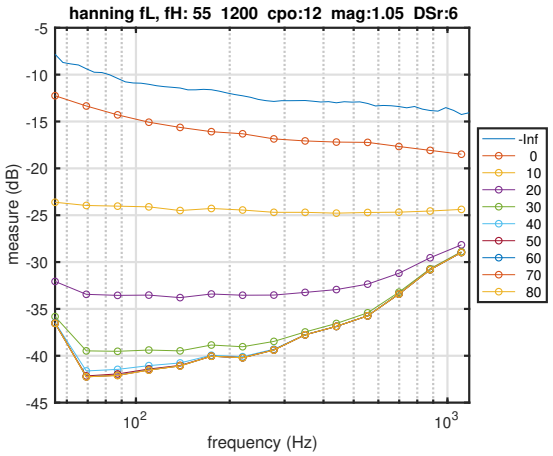

Figure 41: *Fo dependency test using Hann windowing function for envelope. The smoothing function is Blackman windowing function.*

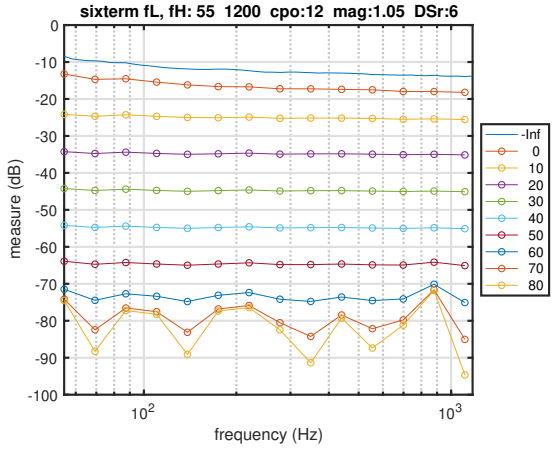

Figure 40: *Fo dependency test using the six-term series for envelope. The smoothing function is Blackman windowing function. This version uses the same downsampling rate to Fig. 39.*

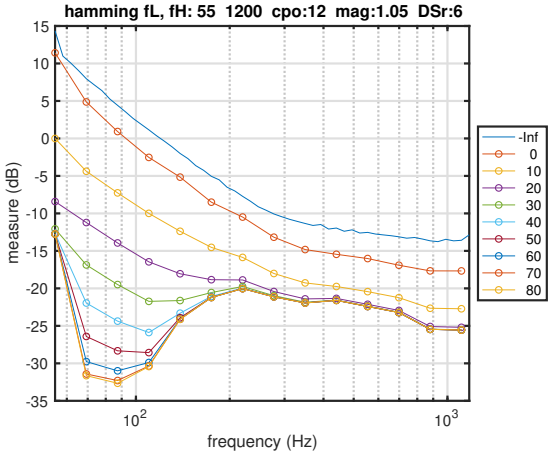

Figure 42: *Fo dependency test using Hann windowing function for envelope. The smoothing function is Blackman windowing function.*

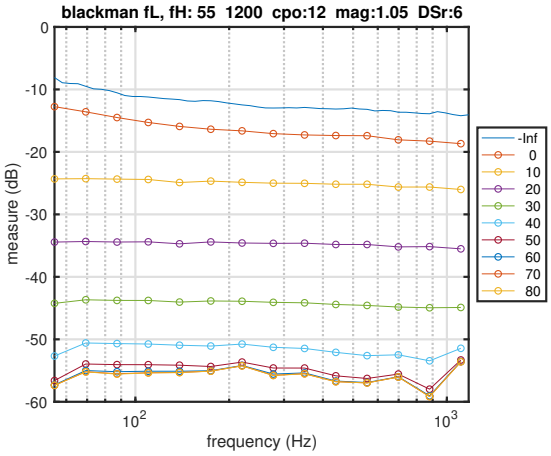

Figure 43: *Fo dependency test using Blackman windowing function for envelope. The smoothing function is Blackman windowing function.*

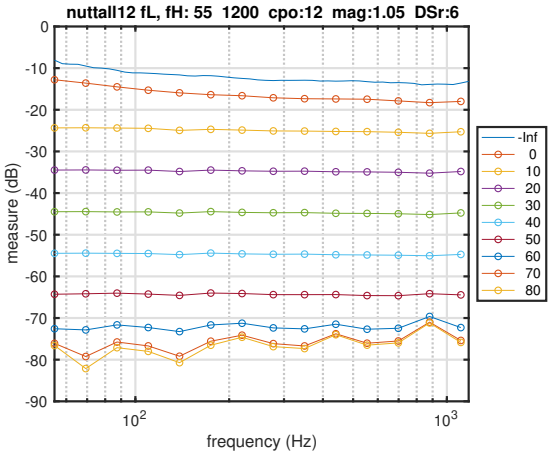

Figure 44: *Fo dependency test using Nuttall's four term windowing function for envelope. The smoothing function is Blackman windowing function.*

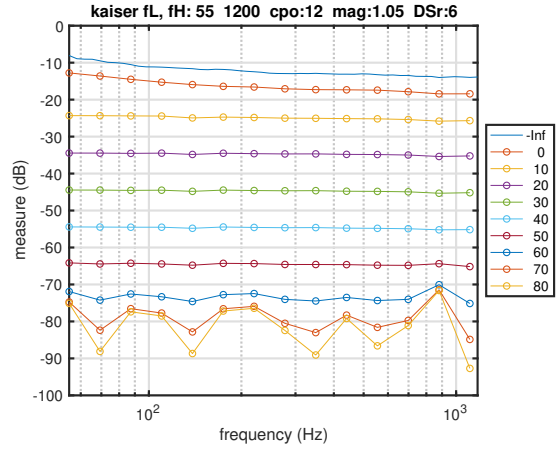

Figure 45: *Fo dependency test using Kaiser windowing function for envelope. The smoothing function is Blackman windowing function.*

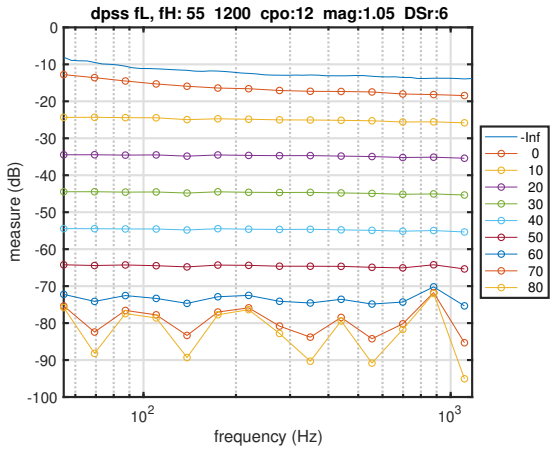

Figure 46: *Fo dependency test using Kaiser windowing function for envelope. The smoothing function is Blackman windowing function.*

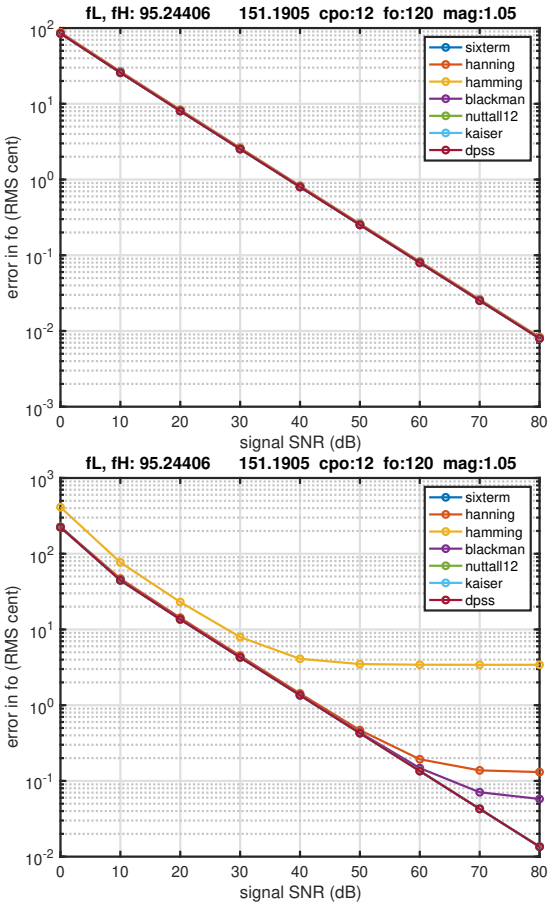

Figure 47: *Signal SNR and the standard deviation of the yielded instantaneous frequency. The upper panel shows results for the power weighted smoothed frequency. The lower panel shows results of raw frequency output. The fo of the test signal is 120 Hz.*

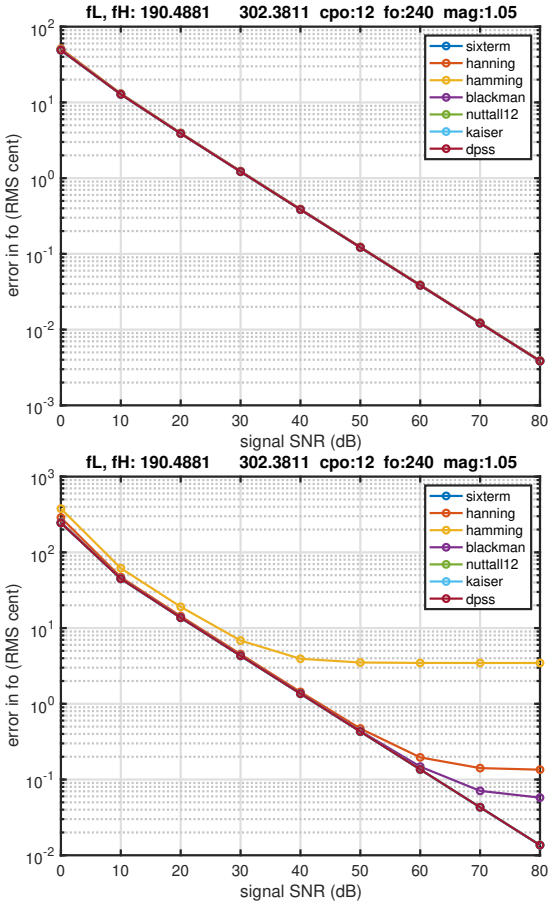

Figure 48: *Signal SNR and the standard deviation of the yielded instantaneous frequency. The upper panel shows results for the power weighted smoothed frequency. The lower panel shows results of raw frequency output. The fo of the test signal is 240 Hz.*

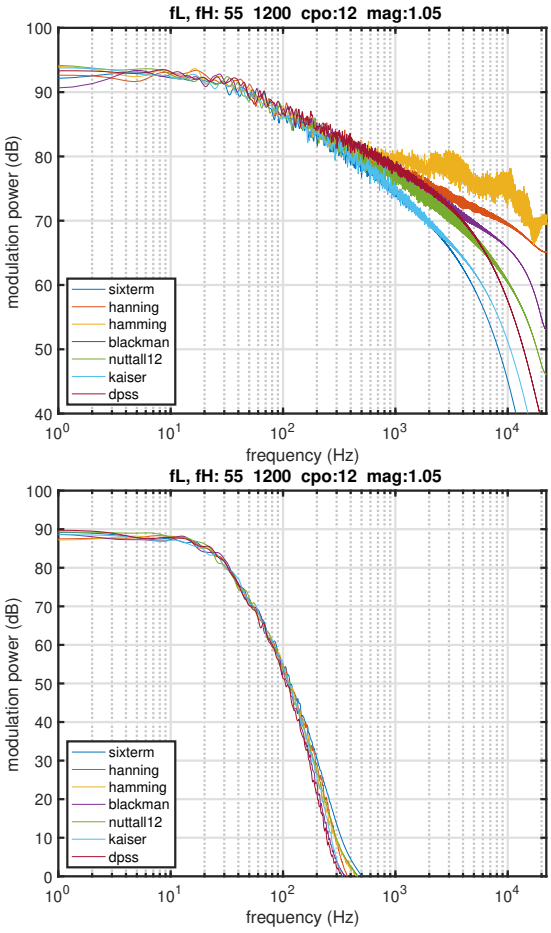

Figure 49: *Power spectra of the output instantaneous frequency trajectories. The upper panel shows the results of the raw trajectory. The lower panel shows the results of the power weighted average. The integration interval is 30 ms using Hann windowing function for weighting.*

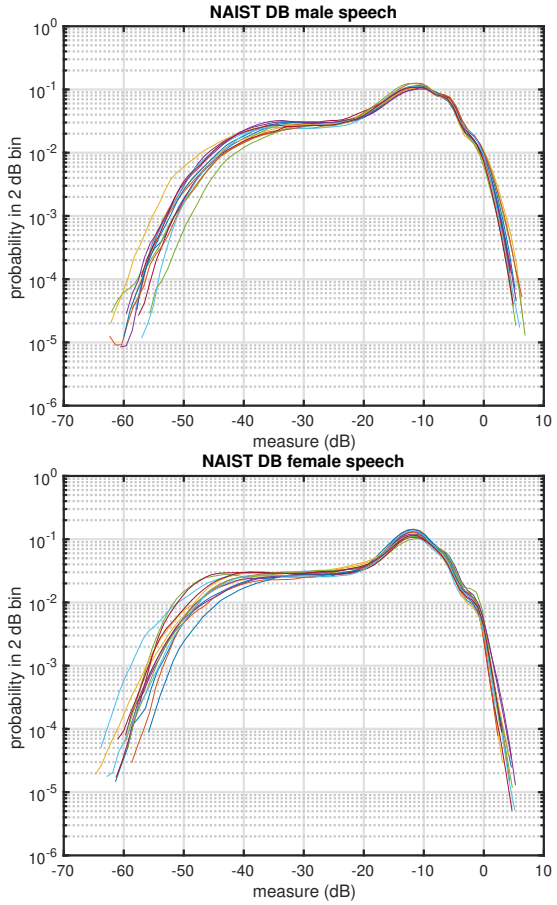

Figure 50: *The measure value distribution of the speech and EGG database prepared by Atake et.al. [20]*

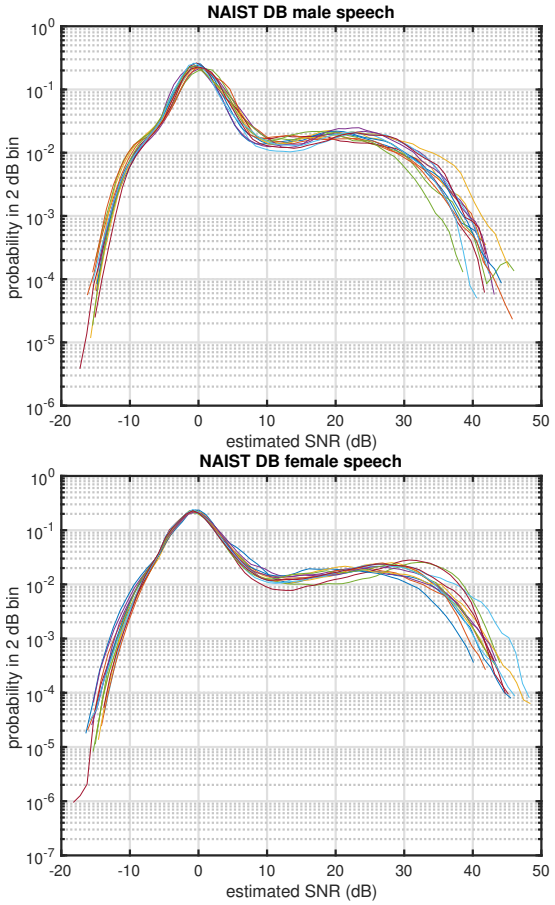

Figure 51: *The estimated SNR distribution of the fo candidates extracted from the speech and EGG database prepared by Atake et.al. [20].*

C. Measure of natural database

We tested the measure value distribution in natural speech database. This section introduces analysis results of several known database samples.

C.1. Speech and EGG database by NAIST, 2000

This database consists of the speech signal and the simultaneously recorded EGG signal. Students (four teen male and four teen female students) of NAIST participated in the recording. Each talker read thirty Japanese sentences. The recording was conducted in an unechoic chamber (Room B117) in NAIST from 25 November to 9 December, 1999. A cardioid directional condenser microphone (SONY ECM-23F3) is placed in front of the speaker with about 20 cm from one's lip. The simultaneous recording equipment was LARYNGOGRAPH BS5724. The signals were recorded on a DAT recorder (SONY DTC-200ES) at 48 kHz and 16 bit condition.

Figure 50 shows the distribution of the measure value inside ± 8 semitone from the center channel of voicing.

Figure 51 shows the distribution of the estimated SNR for extracted fo candidates. A fixed point analysis of mapping from the wavelet carrier frequency to output instantaneous frequency yielded the fo candidates [4]. The periodicity measure value conversion to the estimated SNR uses the simulation results at 20 dB SNR.

C.2. Speech and EGG database: Bagshaw DB

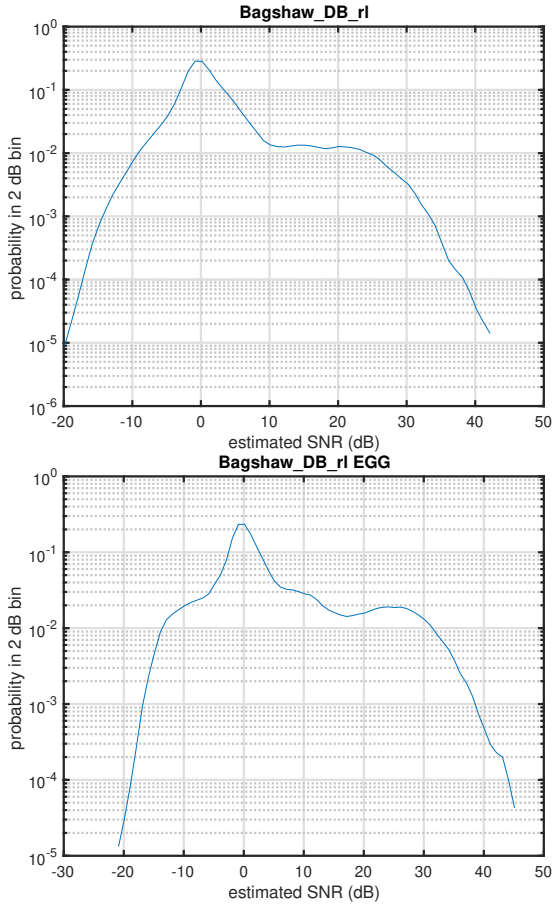

Figure 52: *The estimated SNR distribution of the fo candidates extracted from the speech and EGG database prepared Bagshaw for speaker rl. These plots uses 30 ms integration time.*

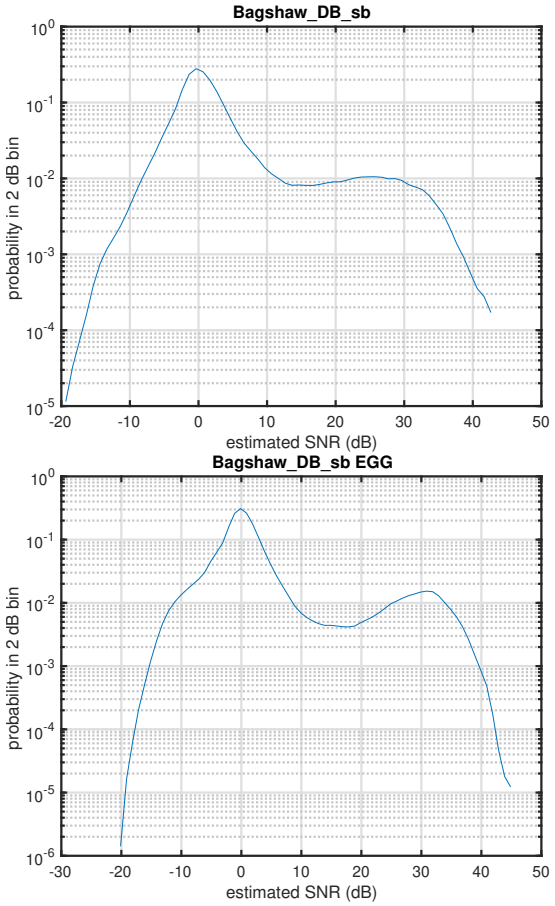

Figure 53: *The estimated SNR distribution of the fo candidates extracted from the speech and EGG database prepared by Bagshaw for speaker sb. These plots uses 30 ms integration time.*

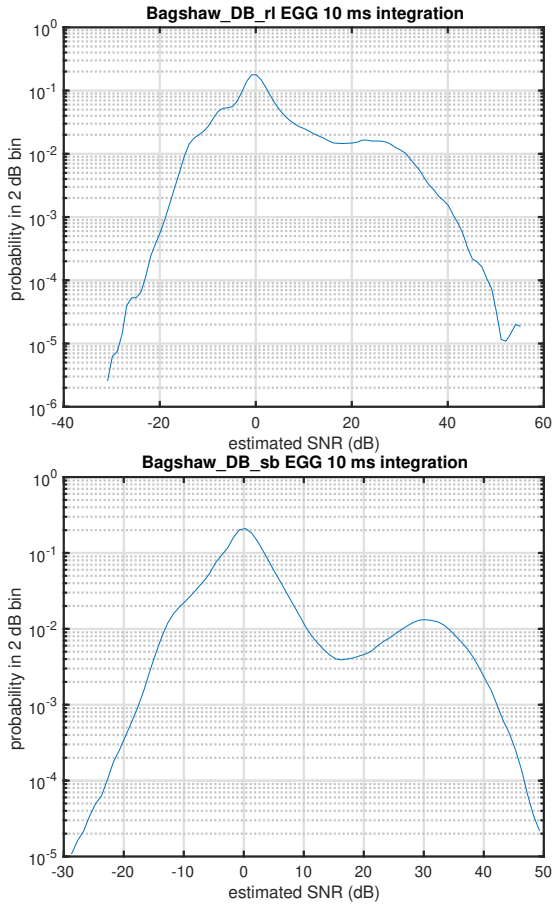

Figure 54: *The estimated SNR distribution of the fo candidates extracted from the speech and EGG database prepared by Bagshaw. These plots uses 10 ms integration time.*

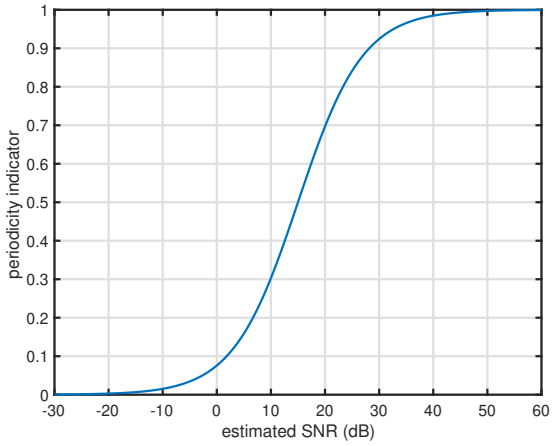

Figure 55: *Designed mapping from the estimated SNR to periodicity indicator.*

D. Implementation details

This section provides detailed information about implementation and underlying evidences.

D.1. realtimeFoViewerR

Figure 55 shows designed nonlinear mapping fro the estimated SNR to periodicity indicator of the realtime application. The probability (density) distribution of the estimated SNR calculated from the speech database indicates that 15 dB corresponds to the most relevant boundary of random signal and periodic component. The mapping is a sigmoid given below:

$$\xi = \frac{1}{1 + \exp\left(-\dfrac{r_{SNR} - 15}{6}\right)}, \tag{17}$$

where the constants 15 and 6 are iteratively determined to make the indicator to behave perceptually consistent and linear.