Second Edition
Revised November, 2016

Analysis and Synthesis of Pathological Voice Quality

by

Jody Kreiman
Bruce R. Gerratt
Norma Antoñanzas-Barroso

Bureau of Glottal Affairs
Department of Head/Neck Surgery
UCLA School of Medicine
31-24 Rehab Center
Los Angeles, CA 90095-1794

This research was supported by grant DC01797
from the National Institute on Deafness and Other Communication Disorders.

Table of Contents

I. INTRODUCTION

Voice quality is an important topic of study in many disciplines, but knowledge of its nature is limited by a poor understanding of the relevant psychoacoustics. This document describes software for voice analysis and synthesis designed to test hypotheses about the relationship between acoustic parameters and voice quality perception. This formant synthesizer provides experimenters with a useful tool for creating and modeling voice signals. In particular, it offers an integrated approach to voice analysis and synthesis and allows easy, precise, time- and spectral-domain manipulations of the harmonic voice source. The synthesizer operates in near real-time, using a parsimonious set of acoustic parameters for the voice source and vocal tract that a user can modify to accurately copy the quality of most normal and pathological voices. For example, the ability to copy synthesize a voice also allows users to modify acoustic parameter at a time while holding all others constant, in order to evaluate its perceptual importance.

This document describes three integrated programs: software for inverse filtering (invf.exe), voice synthesis (synthesis.exe), and voice analysis (sky.exe). This software was developed at the UCLA Bureau of Glottal Affairs, with support from the National Institute on Deafness and Other Communication Disorders (grant DC01797). The software is distributed as shareware, and the code is available on request on an open source basis. Executable files and documentation are available for download from http://www.linguistics.ucla.edu/faciliti/facilities/acoustic/acoustic.html. Two sample voices (one male and one female) are also available at that site. All software is best suited for Windows computers with 1 GB or more of memory.

Users are implored to report any bugs they find in any of this software to Norma Antoñanzas-Barroso (nab@ucla.edu) or to Jody Kreiman (jkreiman@ucla.edu). Suggestions for modifications, additions, and clarifications are also very welcome, but technical support is not available beyond the information provided in this document.

Organization of the Manual

The organization of this manual follows the steps of the analysis/synthesis process. The usual first step in this process is estimating the vocal source and vocal tract transfer functions for the voice under consideration, and accordingly, Section II of the manual describes inverse filtering software developed for these purposes. (It is also possible to omit this step by opening and modifying one of the sample cases, as described in the synthesizer documentation.) Subsequent analyses and synthesis are conducted using the synthesizer software, which is described in Section III. Finally, a number of specialized tools for voice analysis are described in Section IV.

Each section of the manual begins with a brief introduction to some of the relevant theoretical and technical considerations, followed by step-by-step instructions for completing typical analyses. The final part of each section describes additional features and the function of each menu command. We have assumed some previous knowledge of speech acoustics and signal processing, especially in the introductory sections. In particular, a basic understanding of the acoustic theory of speech production is needed to understand much of what follows. Users without such background may wish to skip the introductory sections of each chapter and proceed directly to the “cookbook” sections that provide step-by-step instructions for using the software.

Technical Credits

Inverse filtering, voice analysis, and voice synthesis algorithms were written primarily by Norma Antoñanzas-Barroso in C++ to run under Windows. The software has been developed and compiled with the Visual Studio 2010 from Microsoft. Algorithms for source and noise modeling and interactive synthesis were originally programmed by Brian Gabelman in MATLAB, and subsequently were significantly revised and adapted for Windows by Norma Antoñanzas-Barroso and Diane Budzik. Significant technical advice has been provided by Lloyd Rice and Michael Döllinger, whose help we gratefully acknowledge.

II. INVERSE FILTERING

Part 1: Background

Introduction

Estimates of the shape of the harmonic part of the glottal source can be obtained by inverse filtering the voice signal (Figure 1). In source-filter theory, the vocal tract is modeled as an all-pole filter shaping the input glottal source, and radiation at the lips (which increases the output sound energy level by 3 dB/octave) is modeled by a differentiator. To recover the glottal source, these factors must be canceled out. The vocal tract transfer function is canceled by applying an all-zero filter (the inverse of the vocal tract model) to the speech signal. This process removes the effects of the transfer function from the signal, leaving behind an estimate of the glottal flow derivative. If the radiation characteristic is also canceled, an estimate of the actual glottal pulse shape is generated.

This introductory section describes some of the theoretical and practical issues involved in inverse filtering, along with the technical details of the algorithms. Step-by-step instructions appear in Part 2.

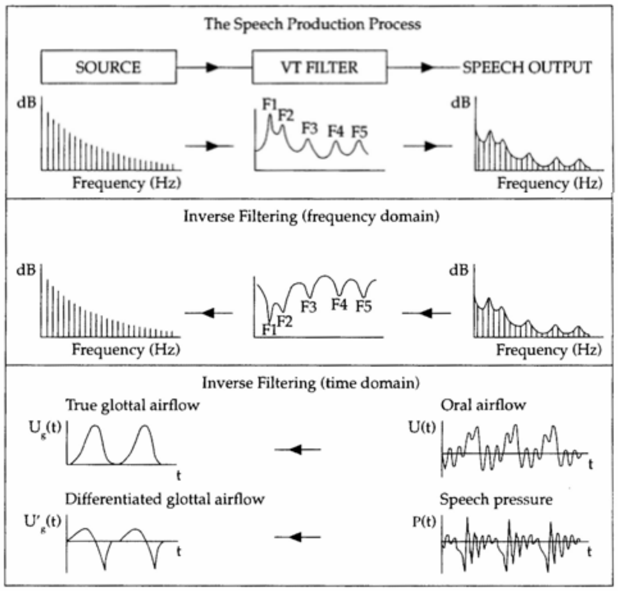


Figure 1. The inverse filtering process. From A. Ní Chasaide & C. Gobl, "Voice source variation," in W.J. Hardcastle, J. Laver, & F. Gibbon, *The Handbook of Phonetic Sciences* (Second edition, Oxford, Blackwell, 2010), p. 381.

Recording Voice Samples for Inverse Filtering

Recording techniques are not particularly critical in investigations of the spectral characteristics of the voice source, as long as good quality equipment is used in a controlled environment. However, when the goal of the analysis is to recover the shape of the glottal pulse (or its derivative, usually referred to as the flow derivative) accurately in the time domain, voice recording for inverse filtering must preserve phase relationships among the different spectral components. Two recording methods theoretically can preserve spectral phase characteristics (and thus pulse shapes): direct digitization from a precision condenser microphone with an appropriate frequency response, or recording the flow signal with a pneumotachographic mask and a differential pressure transducer, as described by Rothenberg (1973; 1977). Standard audio tape recorders do not preserve phase information.

Each recording method has advantages and drawbacks (Javkin et al., 1987). Recording in free field with a condenser microphone provides an excellent high-frequency response. High fidelity ½ inch condenser microphones, like those manufactured by Brüel & Kjær, can transduce acoustic energy down to about 0.1 Hz. However, a microphone cannot capture the low frequency components of the airflow that arise when the glottis fails to close completely, so information about any constant DC offset is generally lost when this method is applied (although it may be possible to use calibration techniques to estimate the DC airflow without use of a flow mask; see Alku et al., 1998, for details). Finally, the effect of radiation from the lips, equivalent to a differentiation of the signal, must be taken into account with microphone signals to recover actual glottal pulse shapes, although this is not an issue when the goal is to recover the glottal flow derivative. To recover the glottal pulse shape, the signal must be integrated to remove radiation effects, producing a high frequency de-emphasis of 3 dB per octave. This de-emphasis has the effect of enhancing any low frequency noise in the signal.

Airflow masks preserve the DC component of the signal and give a calibrated, quantitative measurement of actual glottal flow. The mask also eliminates radiation effects. Thus, very low frequency noise is less of a problem with a flow mask system. However, the flow mask has a poor high frequency response (only up to about 1200 Hz; Rothenberg, 1973, 1977), which may cause significant errors in estimation of the flow waveform shape. In particular, the glottal airflow waveform often has the most abrupt changes during the closing phase, and high frequency information is needed to represent these fast changes (Alku & Vilkman, 1995). In addition, the filtering effects of the mask placed over the face make it difficult to estimate voice formant frequencies accurately.

The particular recording method selected thus depends on the specific application. In our case (where the goal is to derive parameter values for synthesis), loss of high frequency information and difficulties estimating vocal tract resonance frequencies have proven far more problematic than contamination by low frequency noise, so recordings are made with a condenser microphone rather than a flow mask system. Voices for our studies are transduced with a 1/2" Brüel and Kjær condenser microphone (model 4193) and directly digitized. Signals are sampled at 20 kHz, with 16-bit resolution. They are subsequently downsampled to 10 kHz for analysis.

Estimating the Vocal Tract Filter

Success in inverse filtering is usually defined as an output pulse with minimal residual formant ripple, indicating that most of the effect of the formants has been canceled, and a smoothly decreasing source spectrum conforming to theoretical expectations (Fant, 1979). A successful result depends mostly on the correct specification of formants and bandwidths. In particular, the frequency and bandwidth of F1 must be determined rather precisely to avoid distorting the glottal pulse shape. The frequency and bandwidth of the formants above F3 do not have a large effect on the overall source pulse shape (Ní Chasaide & Gobl, 2010), but are important for correct modeling of glottal closure and vocal tract excitation, as discussed below (Alku & Vilkman, 1995).

Because of interactions between the vocal tract and the source, formant frequencies and bandwidths modulate during the open phase of the glottal cycle. For this reason, the most accurate estimate of vocal tract parameters should be obtained during the glottal closed phase, which can be detected from the LPC residual signal (Childers et al., 1983, 1990; Childers & Krishnamurthy, 1985; Childers & Lee, 1991). The closed phase (to the extent that there is one) begins one sample after the residual peak. In LPC analysis, the "error" left over after the vocal tract filter has been estimated approximates the source component of the signal (assuming a linear source-filter theory). Thus, a noisy residual signal indicates that the LPC model of the resonances leaves variance unaccounted for. In theory this may signal a need to adjust the formants later in the modeling process, although in practice we have not noticed a correlation between a noisy residual signal and a “poor” inverse filtering result.

Once the closed phase has been detected, formant frequencies and bandwidths are estimated using a closed phase covariance LPC analysis of 30-56 points, depending on F0 (40 is typical). The number of poles is not restricted, to assure the smoothest possible flow derivative and source spectrum slope. When there is no closed phase, or when the result is unsatisfactory, it is also possible to compute an autocorrelation LPC analysis over a larger window as an alternative to covariance analysis. The variability introduced by the longer window adds its own error to the analysis, but sometimes produces a better result in cases where the assumptions of covariance analysis are violated.

Inverse Filtering Method

Inverse filtering is performed using the method described by Javkin et al. (1987). For signals sampled at 10 kHz, 5-6 zeros are generally appropriate, although the inverse filter software allows up to 10 zeros to be specified. The filter also includes 6 poles (to remove spectral zeros), although in our experience these have not proven particularly useful.

Given that the whole inverse filtering procedure is noisy, an interactive process has been developed that allows the user to manipulate formants and bandwidths to produce the "best" result possible. Use of an interactive filter minimizes the need for precise vocal tract estimation, because a poor estimate can easily be corrected to improve the inverse filtering result. In practice, however, care must be taken because manipulation of the inverse filter to eliminate formant ripple and smooth the source spectrum often simultaneously smooths away perceptually-important details about the shape of the glottal pulse, indicating that the traditional criteria for successful inverse filtering should be applied with caution. This difficulty may be overcome in part by smoothing a theoretically less-than-ideal inverse filter output with a time- or spectral-domain theoretical model instead of attempting to completely model the pulses in the inverse filter. The best approach appears to be limiting intervention in the filter to removing spurious low-frequency poles (a pole at F0, for example) and only enough high-frequency ripple to ensure that the model-fitting algorithm does not crash. Less definitely appears to be more in this case.

Finally, there is no way to know for certain that the inverse filtering process has recovered the “true” or “correct” shape of the glottal pulses, even when the analysis goes smoothly and all traditional criteria for success are met. Depending on the application, different standards for validating the results may be applied. In our case, the recovered source pulses are imported into the synthesizer and then adjusted to produce a synthetic voice that perceptually matches the original natural target voice sample. The source pulse that produces a perceived match to the target voice is considered to be “perceptually correct,” although its relationship to underlying physiological vocal function remains unknown. Individual researchers should be aware of validity issues surrounding the output of the inverse filter, and should take steps to validate their results if they plan to make any claims that require or imply correctness or accuracy.

Part 2: Step-by-Step Procedures

Program Installation

The user can create her/his own working space, either on the Desktop or as a new directory. Copy the file invf.exe into the selected space. Add the inverse filter to the start menu or task bar if desired. The software will automatically create the other directories it needs on first use.

Inverse Filtering Procedure: Introduction

The inverse filter as described here serves as a way of estimating the voice source so that it can be modeled and used to synthesize the voice in question. Obviously, there are other reasons to inverse filter vowels—for example, to gain information about the source to assist in evaluating patients with voice disorders—and the inverse filter can be used for these purposes as well. Procedures will vary slightly depending on the purpose. Major procedural variants are noted below.

This may seem like a complicated process from the number of pages it takes to describe it, but once you get the hang of it you can finish an average analysis in under a minute.

Open a File

First, open a candidate audio file (Figure 2). The default format in the inverse filter is the home-grown .AUD format, in which microphone data have the extension .AUD and flow mask data have the extension .FLO. WAV files can also be used with the command File-Open a WAVE file-filename. ASCII files (.TXT) can also be opened with the command File-Open a text file.

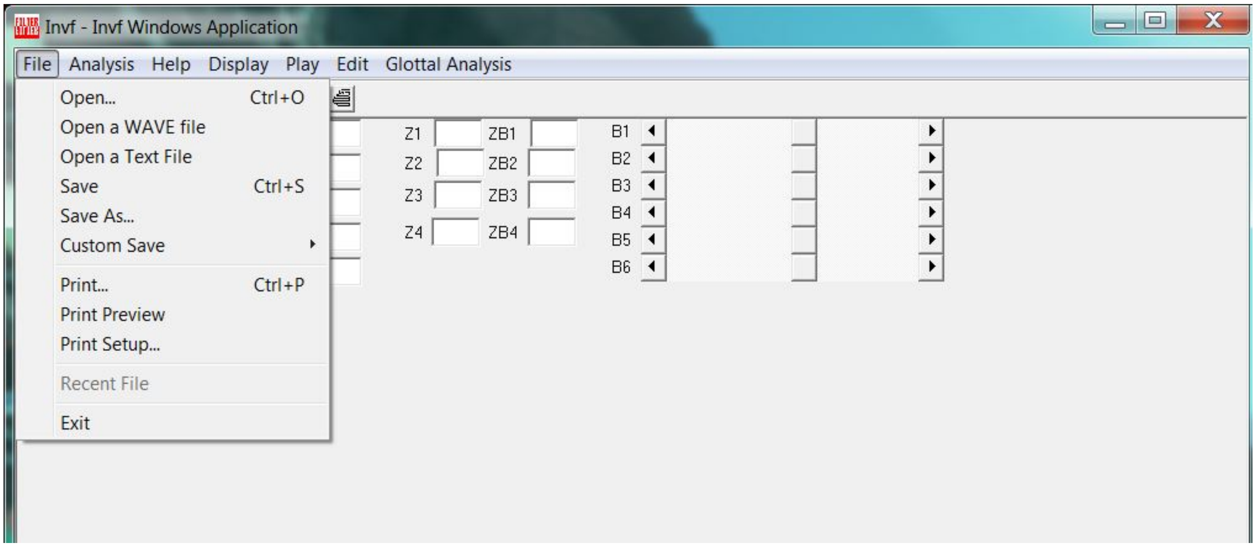
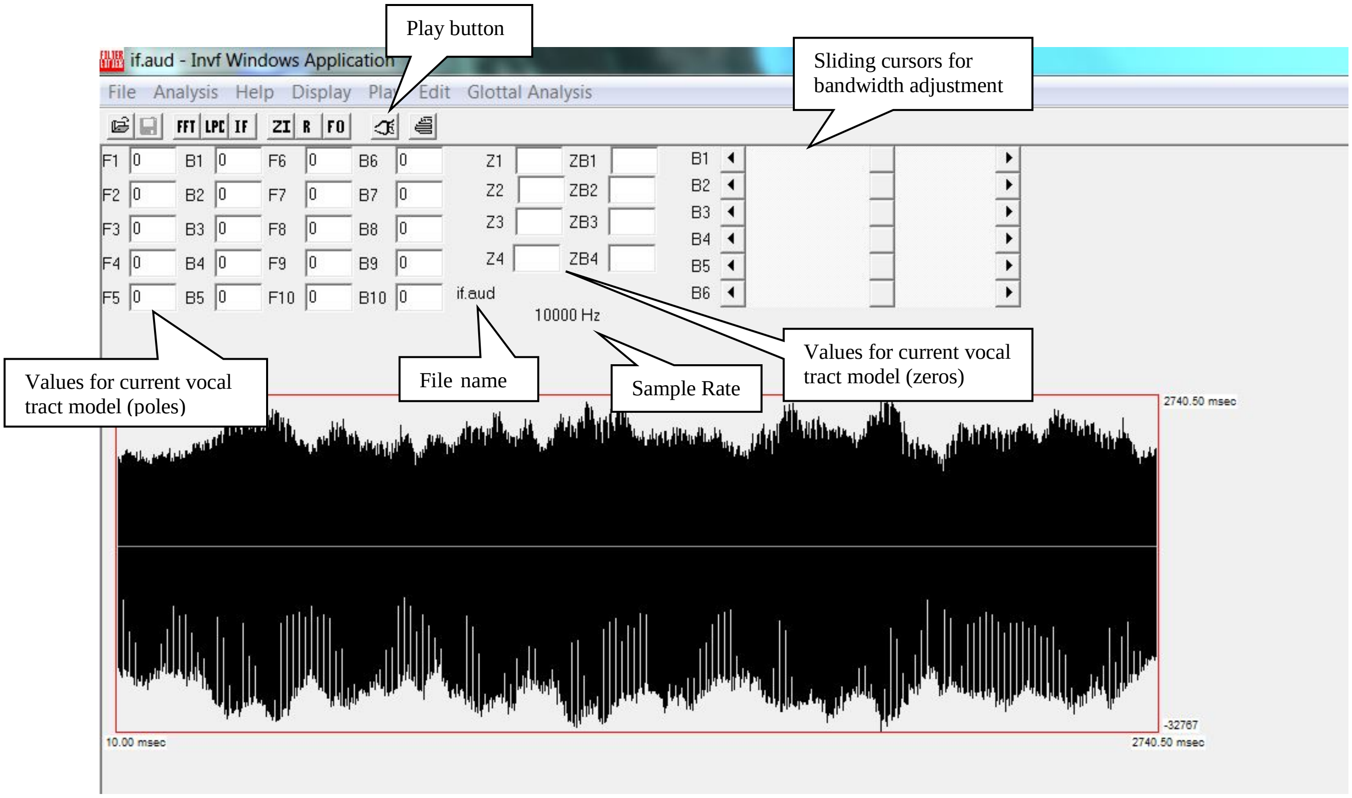


Figure 2. Inverse filter file opening dialog box.

Figure 3 shows the newly opened file. The inverse filter is optimized for a sample rate of 10 kHz, so the sound file may need to be downsampled before proceeding further. Files can be downsampled in the inverse filter itself to rates that are integer submultiples of the original sampling rate, but non-integrally-related sample rates must be converted in some other utility (for example, SoundForge or Praat). To downsample, execute the command Edit-Downsample (Figure 4). The rate defaults to 10,000 (which is what you want). Clicking 'ok' replaces the current file with the downsampled file, and also saves a copy of the downsampled file with the character 'd' appended (e.g., 'if.aud' is saved as 'ifd.aud'; Figure 5).


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
if.aud
10000 Hz
2740.50 msec
-32767
10.00 msec
2740.50 msec

F1 0 B1 0 F6 0 B6 0
F2 0 B2 0 F7 0 B7 0
F3 0 B3 0 F8 0 B8 0
F4 0 B4 0 F9 0 B9 0
F5 0 B5 0 F10 0 B10 0
Z1 ZB1
Z2 ZB2
Z3 ZB3
Z4 ZB4
B1 B2 B3 B4 B5 B6

Play button

Sliding cursors for bandwidth adjustment

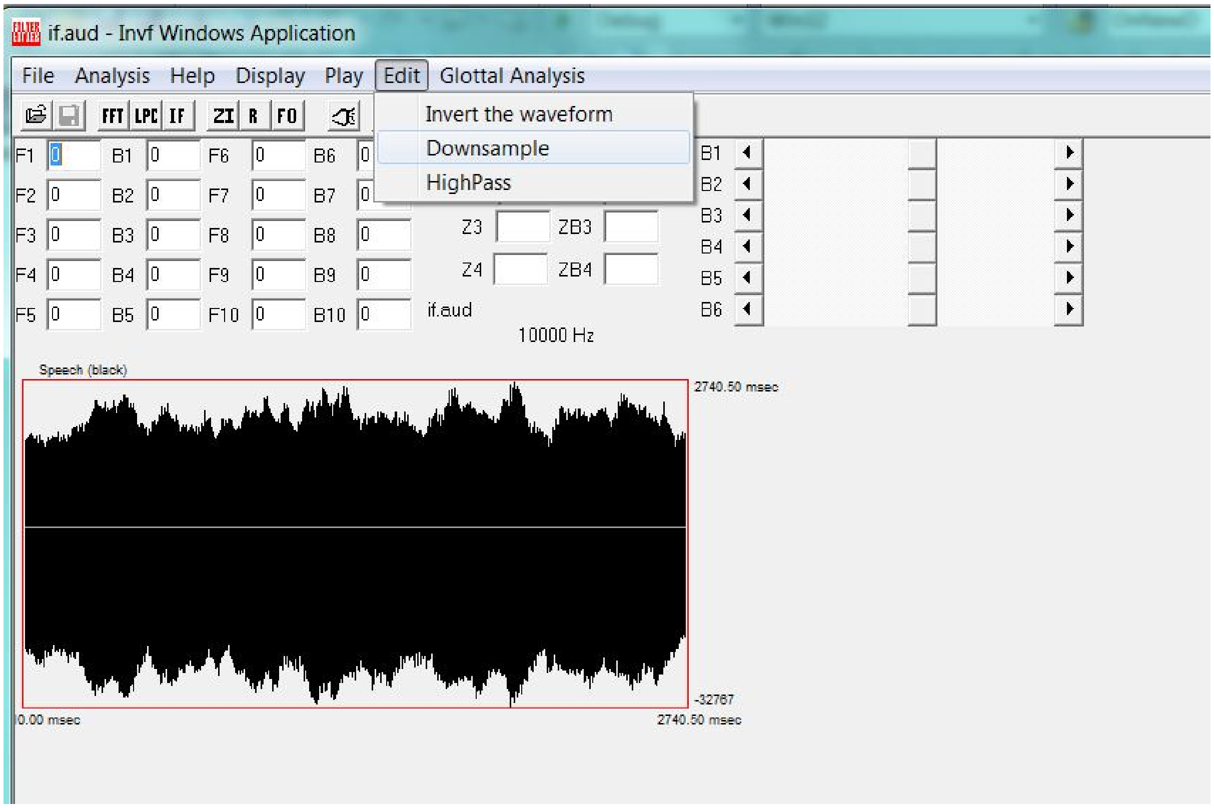
Values for current vocal tract model (poles)

File name

Sample Rate

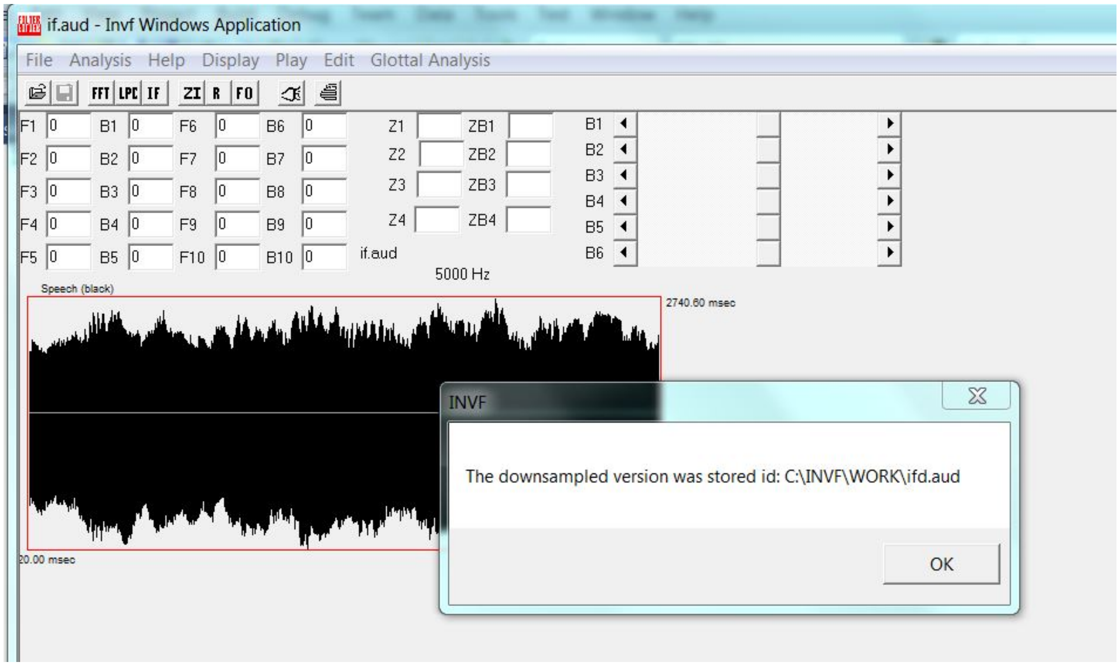
Values for current vocal tract model (zeros)

Figure 3. Inverse filtering window showing open file.



F1 0 B1 0 F6 0 B6 0
F2 0 B2 0 F7 0 B7 0
F3 0 B3 0 F8 0 B8 0
F4 0 B4 0 F9 0 B9 0
F5 0 B5 0 F10 0 B10 0
Z3 ZB3
Z4 ZB4
B1 B2 B3 B4 B5 B6

Figure 4. Downsample dialog box.

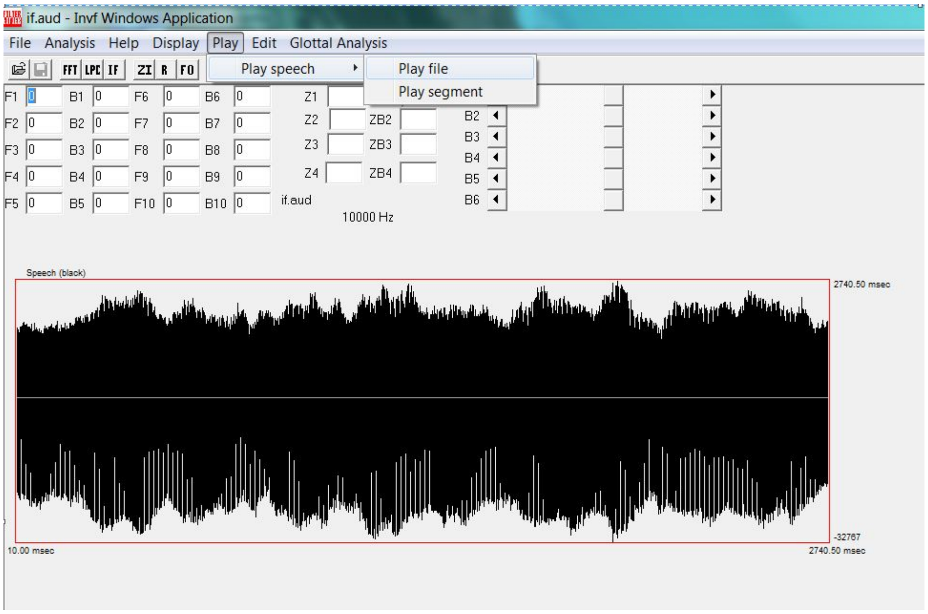

if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
Speech (black)
2740.60 msec
20.00 msec
INVF
OK

F1 0 B1 0 F6 0 B6 0 Z1 ZB1 B1
F2 0 B2 0 F7 0 B7 0 Z2 ZB2 B2
F3 0 B3 0 F8 0 B8 0 Z3 ZB3 B3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4 B4
F5 0 B5 0 F10 0 B10 0 if.aud B5
B6
5000 Hz

The downsampled version was stored id: C:\INVF\WORK\ifd.aud

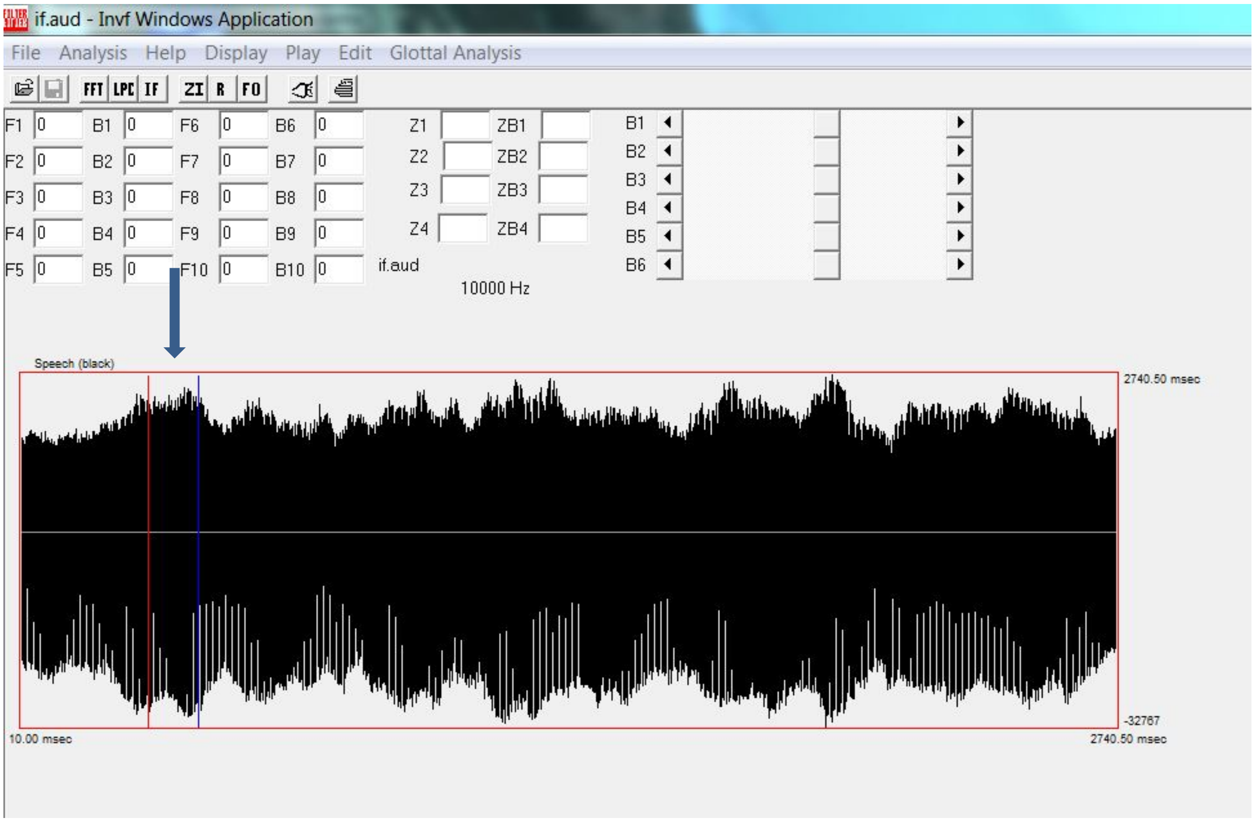
Figure 5. Downsampled file is saved and reopened in active window.

Next, screen your voice sample for any undesirable features (e.g., clipped cycles, unvoiced sections) and identify a segment to work on. Play the whole file by clicking on the sound icon in the toolbar (Figure 3) or use the Play menu option as shown in Figure 6. To select a segment of the file, set the beginning by left clicking, and then right click to set the end (Figure 7). Click the ZI button on the tool bar to **Z**oom **I**n the segment (Figure 8). (Click the R button to **R**estore the original complete file.). You can also use the different commands in the menu option Display-Speech window.



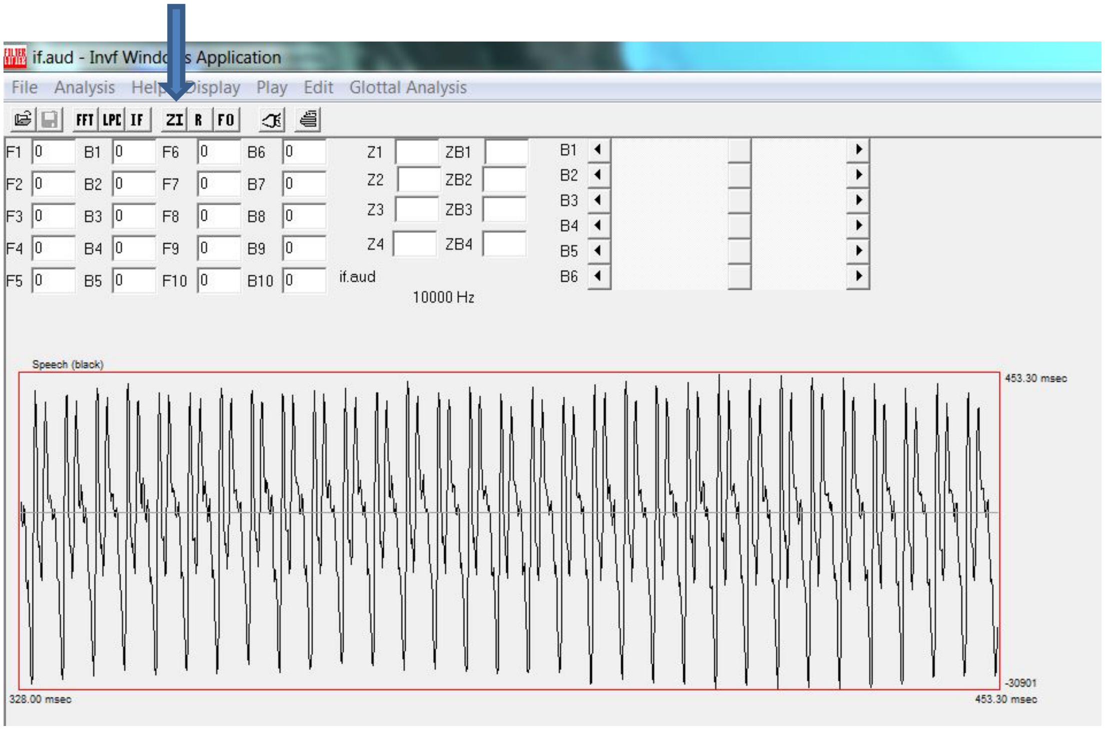
F1 0 B1 0 F6 0 B6 0 Z1
F2 0 B2 0 F7 0 B7 0 Z2 ZB2 B2
F3 0 B3 0 F8 0 B8 0 Z3 ZB3 B3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4 B4
F5 0 B5 0 F10 0 B10 0 if.aud B5
B6
10000 Hz

Figure 6. Use the menu to play a segment of speech; use the sound button on the toolbar (hidden by the expanded menu in this figure) to play the whole file.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
Speech (black)
2740.50 msec
-32767
10.00 msec
2740.50 msec

F1 0 B1 0 F6 0 B6 0 Z1 ZB1 B1
F2 0 B2 0 F7 0 B7 0 Z2 ZB2 B2
F3 0 B3 0 F8 0 B8 0 Z3 ZB3 B3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4 B4
F5 0 B5 0 F10 0 B10 0 if.aud B5
10000 Hz B6

Figure 7. Define a speech segment (indicated by the arrow) by left and right clicking.

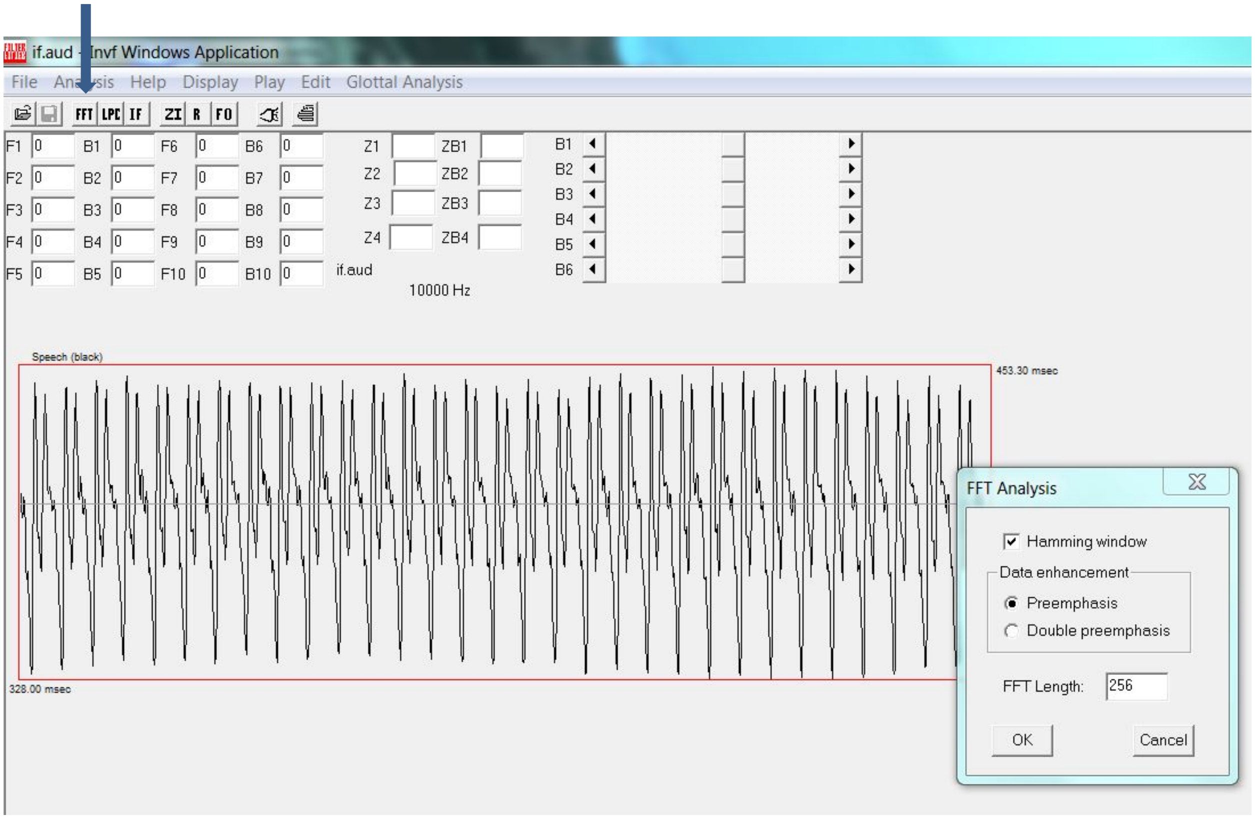

if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
Speech (black)
453.30 msec
-30901
328.00 msec
453.30 msec

F1 0 B1 0 F6 0 B6 0 Z1 ZB1 B1
F2 0 B2 0 F7 0 B7 0 Z2 ZB2 B2
F3 0 B3 0 F8 0 B8 0 Z3 ZB3 B3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4 B4
F5 0 B5 0 F10 0 B10 0 if.aud B5
10000 Hz B6

Figure 8. Zoom in on a defined segment by clicking the ZI button (indicated by the arrow).

Play the selected segment (zoomed or not) by using the command Play-Play speech-Play segment (Figure 6). Continue zooming and playing until you have isolated a segment at least 8 cycles long that meets your analysis criteria. In most cases, this segment will be fairly steady in quality, representative of the overall sample, and free of recording artifacts.

To begin the analysis, click the FFT button on the toolbar (Figure 9). An options window will open. Normally, the default choices of Hamming Window and Preemphasis are appropriate. The default window size is 256 points, which should cover about 2.5 periods. If it doesn't (because F0 is less than 100 Hz), increase the window size to 512 points. Choice of window size often involves compromise. A longer window will give a better analysis, but if the window is too long, variability in the vocal tract configuration may introduce errors. 2.5 periods has proven to be a reasonable compromise in the past. Click OK to continue.



F1 0 B1 0 F6 0 B6 0
F2 0 B2 0 F7 0 B7 0
F3 0 B3 0 F8 0 B8 0
F4 0 B4 0 F9 0 B9 0
F5 0 B5 0 F10 0 B10 0

☑ Hamming window

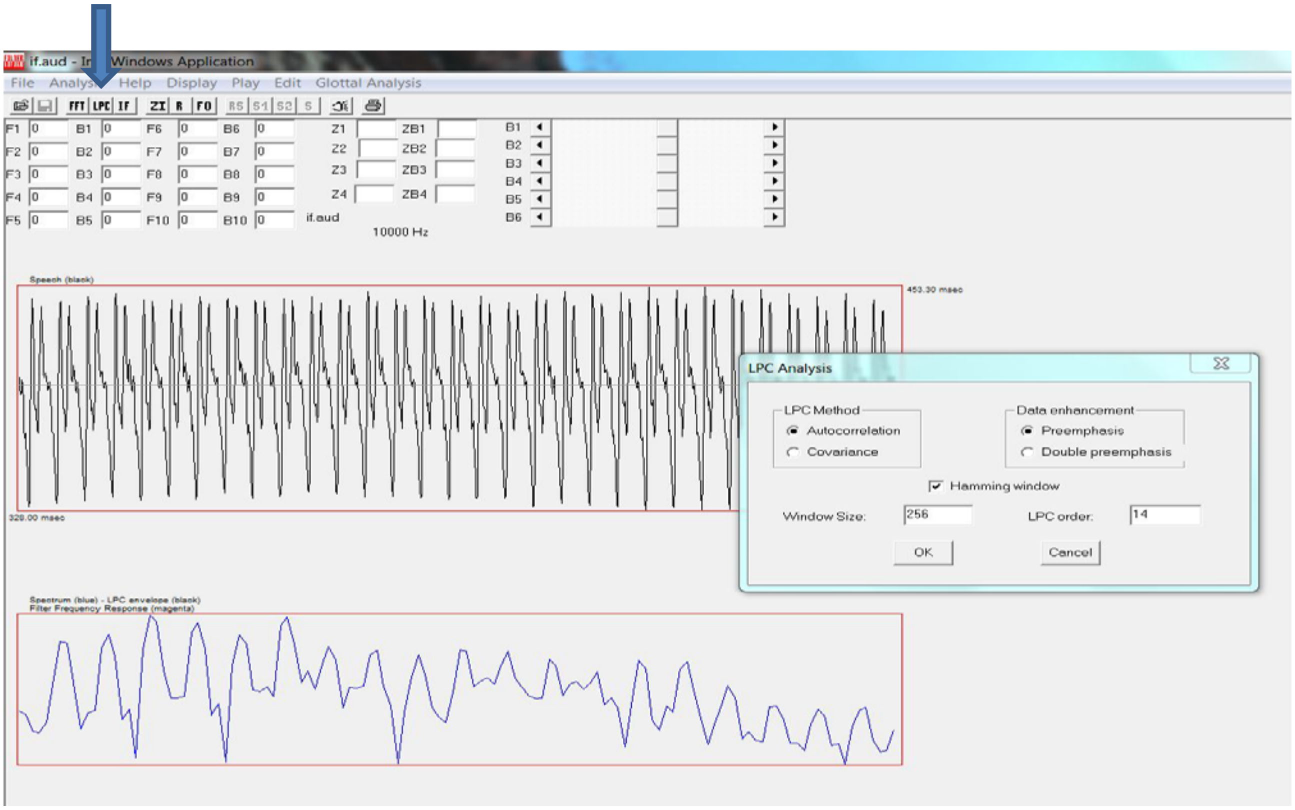
Data enhancement
● Preemphasis
○ Double preemphasis

FFT Length: 256

Figure 9. FFT analysis dialog box.

A spectrum will now appear in the lower left part of the analysis window. Next, click the LPC button (Figure 10). Select autocorrelation and preemphasis, as shown. Window size considerations are as above. Order 14 is usually good for 10 kHz sample rates. If you change the analysis order, the window size will also also change automatically. If necessary, reset the value to the window size you want before clicking OK. Changing the window size will not affect the order unless you set the window size to less than twice the order, in which case the analysis will be rejected.

After OK is clicked, the number of cycles displayed in the window will decrease, an LPC envelope will appear over the FFT spectrum, and numbers will appear in the table of formants and bandwidths in the upper left part of the window (Figure 11). An error signal will also appear under the waveform in the center of the window. Referring to this error signal, mark the beginning and end of a cycle so that F0 can be estimated. To do this, first find the peaks in the error signal. Click the left mouse button near the left peak in the error signal, and click the right button near the right peak. Precision is not critical, and the choice of peaks is not necessarily straightforward. You may have several choices of peak, or you may have to guess at the cycle boundaries. There does not appear to be any particular correlation between the prominence of the peaks and the quality of the inverse filtering—prominent peaks can give a rotten result, and peaks placed by guessing can give a very good result. Also refer to the time series above the error signal. The part of the waveform corresponding to the peaks you choose should look like a complete cycle of phonation, with the left cursor at the beginning and the right cursor at the end. You can reset the cursors if necessary by re-clicking—you may have to move the cursor far to the left or right of the location you want for this to work (because the software looks for a peak near the cursor).



F1 0 B1 0 F6 0 B6 0 Z1 ZB1 B1
F2 0 B2 0 F7 0 B7 0 Z2 ZB2 B2
F3 0 B3 0 F8 0 B8 0 Z3 ZB3 B3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4 B4
F5 0 B5 0 F10 0 B10 0 if.aud B5
10000 Hz B6

LPC Analysis

LPC Method
● Autocorrelation
○ Covariance

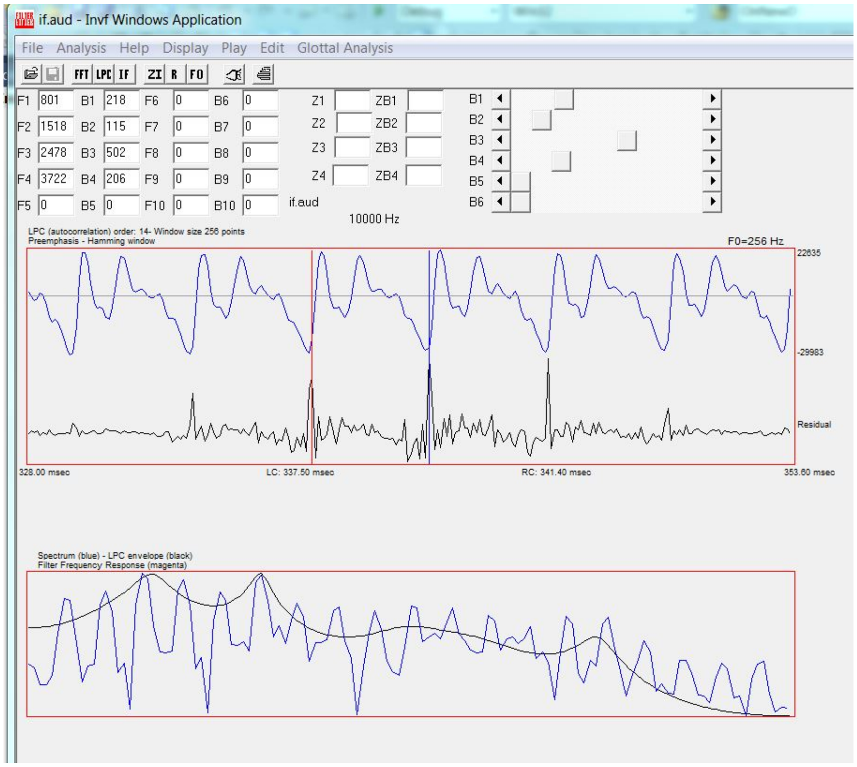
Data enhancement
● Preemphasis
○ Double preemphasis

☑ Hamming window

Window Size: 256
LPC order: 14

Figure 10. Autocorrelation LPC analysis dialog box.

Once you're satisfied and have finished marking a period, click the F0 button on the toolbar to compute F0 for that cycle. The value will appear in the caption at the top right of the frame, just above the time series waveform. Check it to be sure it is sensible given your previous listening to the voice. If it isn't, something is wrong and you should start over.



if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
B1
B2
B3
B4
B5
B6
Residual
22635
-29983

F1	801	B1	218	F6	0	B6	0
F2	1518	B2	115	F7	0	B7	0
F3	2478	B3	502	F8	0	B8	0
F4	3722	B4	206	F9	0	B9	0
F5	0	B5	0	F10	0	B10	0

Z1 ZB1
Z2 ZB2
Z3 ZB3
Z4 ZB4

if.aud
10000 Hz

LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window

F0=256 Hz

328.00 msec LC: 337.50 msec RC: 341.40 msec 353.60 msec

Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)

Figure 11. Output of autocorrelation LPC analysis. Error signal appears below waveform in the middle of the figure. Left and right click to define a cycle, as described in the text.

Run the Inverse Filter

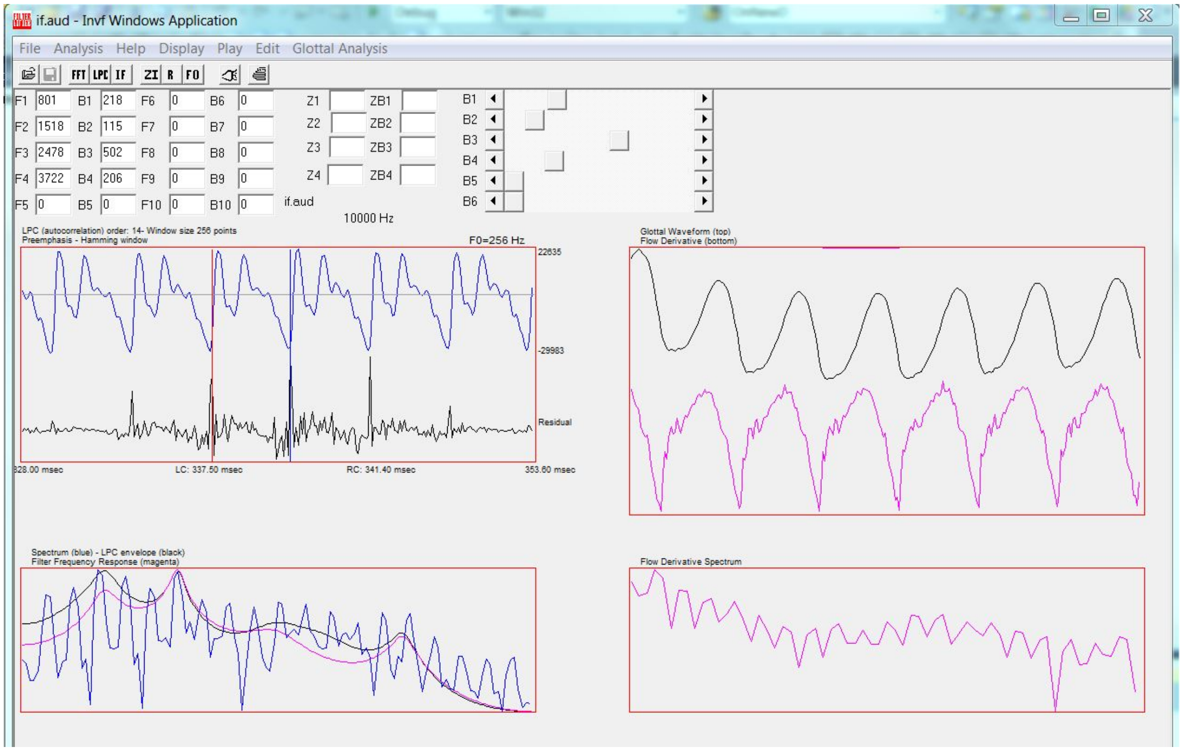
To run the inverse filter using the autocorrelation estimates of vocal tract resonances, just click the IF button on the toolbar (Figure 12). If the extension for the file in use is .AUD, the program assumes that this is a microphone signal and automatically cancels the radiation characteristic. If the file extension is .FLO, the program assumes this is data from a flow mask and does not cancel the radiation characteristic.

The right panels of Figure 12 show the output of the inverse filter. The top tracing in black is the glottal waveform, the second trace in magenta is the flow derivative, and the bottom shows the spectrum of the flow derivative. This result is not very satisfactory, due to the large amount of ripple in the flow derivative. In addition, the flow derivative spectrum is not smoothly decreasing as one would expect from a correctly inverse-filtered signal. The inverse filter allows the user to add or remove resonances and to manipulate bandwidths interactively, as shown in Figures 13-18. To add a new resonance, point the cursor to the appropriate place in the spectrum in the lower left panel of the display and double left click. A resonance will appear in that location (4927 Hz, indicated by an arrow in Figure 13), with a bandwidth of 100 Hz. The formant is added and the inverse filter is automatically reapplied with a new vocal tract model. To remove this resonance, double right click it. Resonances can also be added or deleted manually by editing the values in the table, and then clicking the IF button to apply the new vocal tract model. Note that the resonances in the table may end up "out of order," for example if a formant with frequency 1255 Hz is added to the bottom of the list, below a formant at 3585 Hz, or if a formant in the table is deleted by setting its frequency to 0. The order of the values in the table has no effect on the analysis.

Figure 14 shows the effect on the inverse filtering result of a spurious pole below F1, which is located at 554 Hz (shown by the arrow) in this example. This pole increases the ripples in the inverse filtered waveform and increases the high frequency energy apparent in the flow derivative. Spurious resonances below F1 occur rather commonly when inverse filtering is based on autocorrelation estimates of the vocal tract transfer function, and simply removing them often results in a good result.

Existing formants frequencies can be manipulated in two ways: by typing a new value into the table and clicking the IF button, or by single left clicking the formant in question and dragging it to a new position. Figure 15 shows the result of dragging F1 from its starting value of 801 Hz to a value of 535 Hz; notice the increase in ripple in the flow derivative. As the formant moves, the inverse filter and display update automatically, showing the effect of the new resonance value on the estimated glottal waveform, flow derivative, and flow derivative spectrum. When you are happy with the value, single right click to lock the resonance in place.

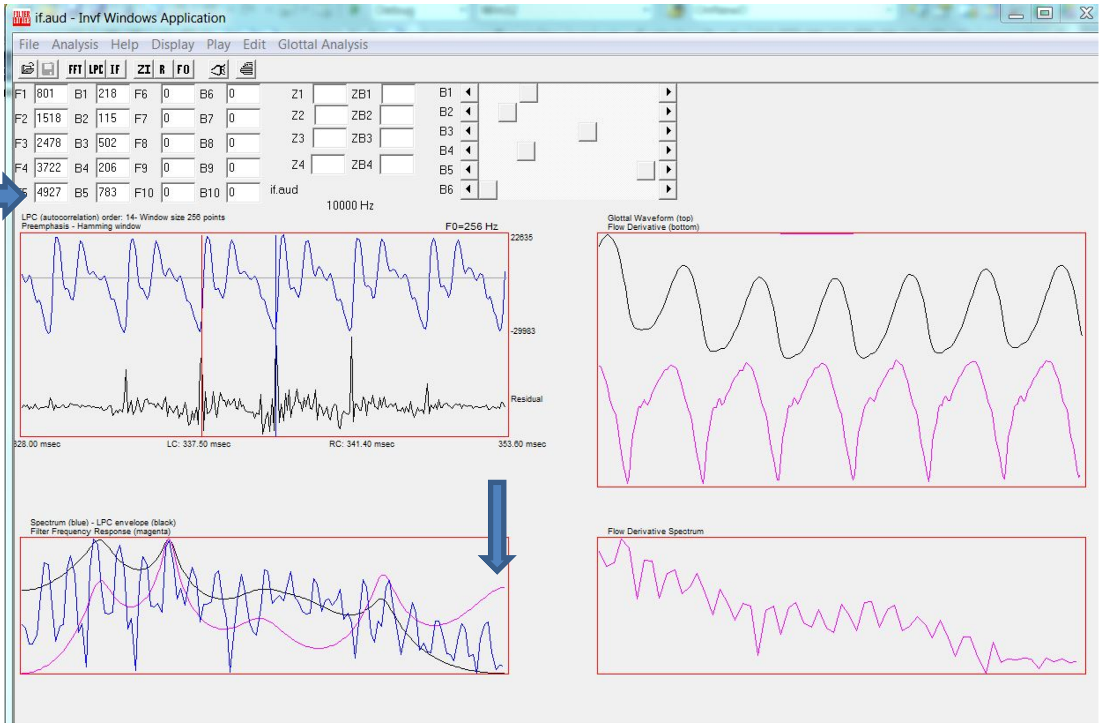
Bandwidths may also be manipulated interactively using the sliders to the right of the table of resonance values. Dragging a slider to the right widens the bandwidth of the resonance in question; in Figure 16, the bandwidth of the first formant has been widened to excess. Dragging the slider to the left narrows the bandwidth. As with the formant values, the effects of changes in slider position are shown immediately in the output display of glottal waveform, flow derivative, and flow derivative spectrum. Bandwidths can be edited directly in the table as well. In this case, the IF button must be clicked to apply the new values. If you are using more than 6 resonances, you will have to edit bandwidth values for the higher resonances in the table, because there are only 6 sliders. Click IF to implement your changes in the filter model.



F1	801	B1	218	F6	0	B6	0	Z1		ZB1	
F2	1518	B2	115	F7	0	B7	0	Z2		ZB2	
F3	2478	B3	502	F8	0	B8	0	Z3		ZB3	
F4	3722	B4	206	F9	0	B9	0	Z4		ZB4	
F5	0	B5	0	F10	0	B10	0	if.aud			

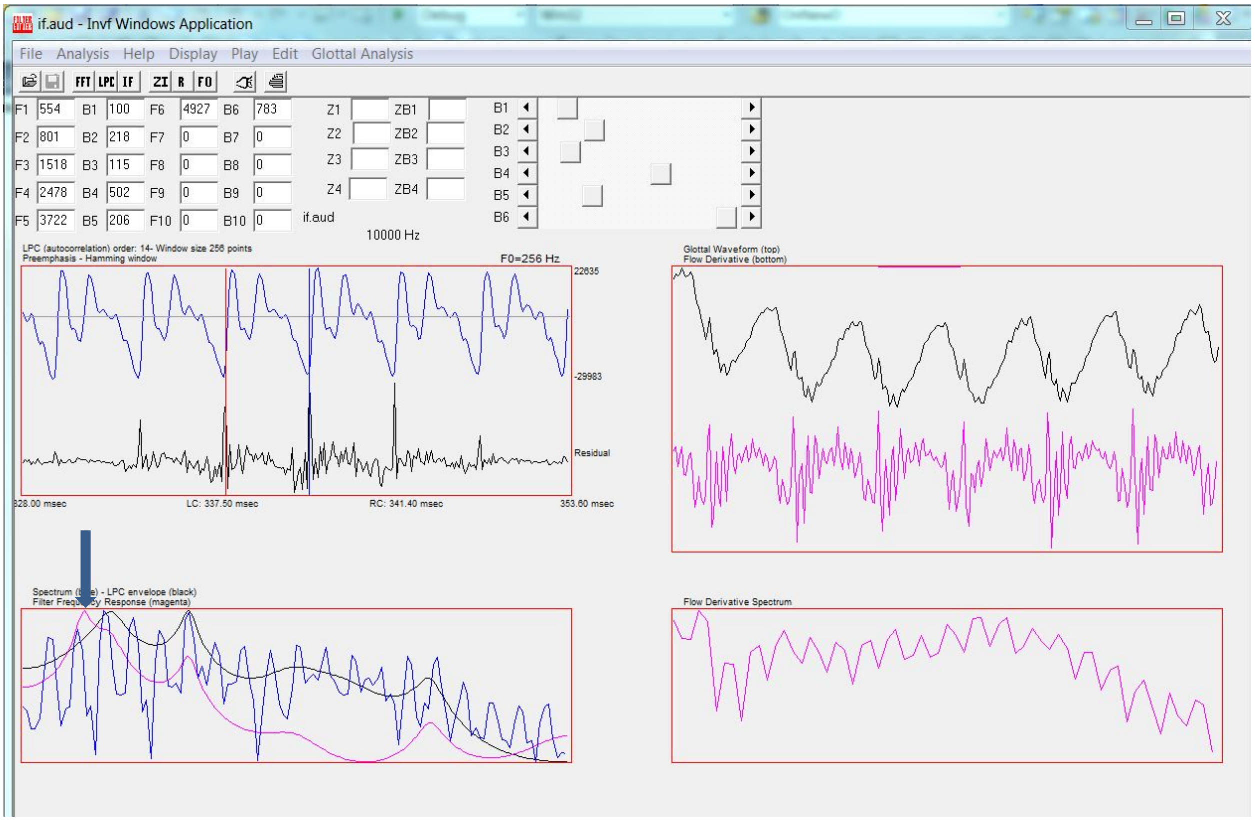
10000 Hz

Figure 12. Initial output of the inverse filter.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
B1
B2
B3
B4
B5
B6
if.aud
10000 Hz
LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window
F0=256 Hz
22635
-29983
Residual
328.00 msec
LC: 337.50 msec
RC: 341.40 msec
353.60 msec
Glottal Waveform (top)
Flow Derivative (bottom)
Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)
Flow Derivative Spectrum

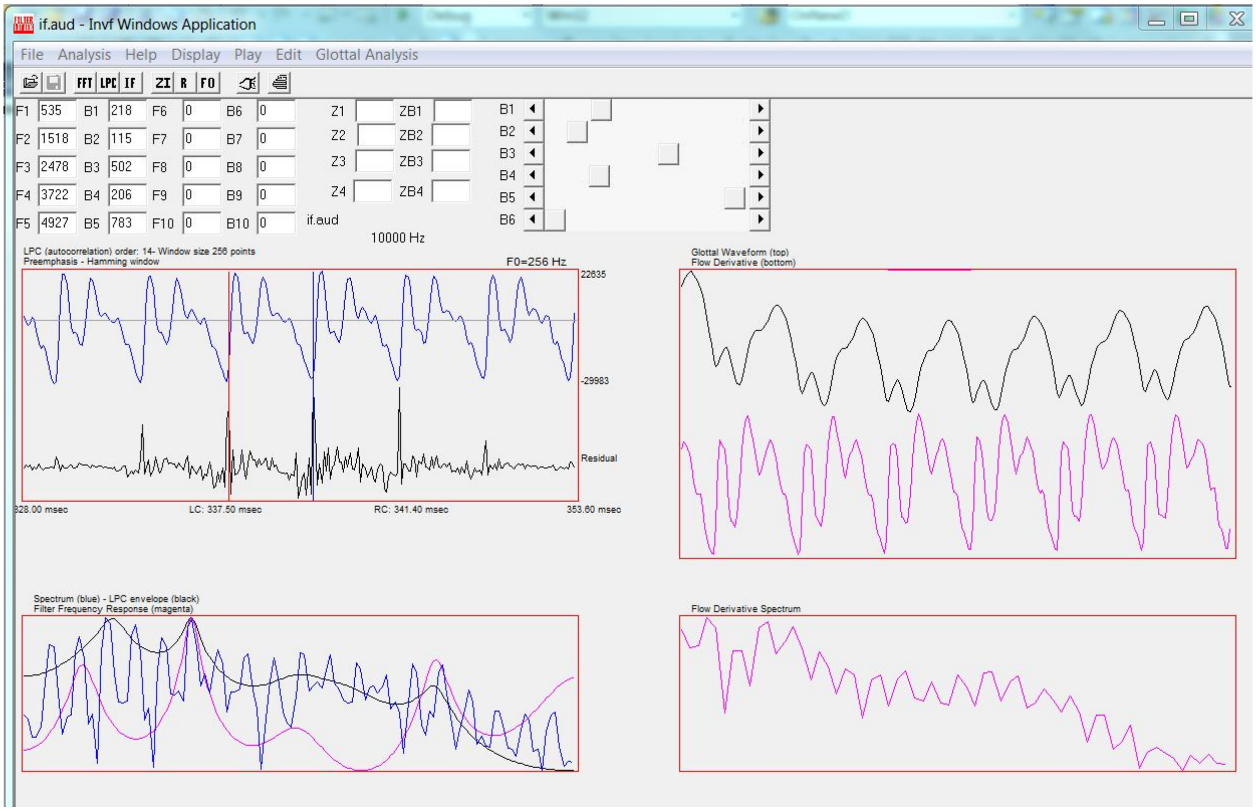
F1	801	B1	218	F6	0	B6	0	Z1		ZB1	
F2	1518	B2	115	F7	0	B7	0	Z2		ZB2	
F3	2478	B3	502	F8	0	B8	0	Z3		ZB3	
F4	3722	B4	206	F9	0	B9	0	Z4		ZB4	
F5	4927	B5	783	F10	0	B10	0				

Figure 13. When the high frequency pole is added, the inverse filtering result improves significantly.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
B1
B2
B3
B4
B5
B6
if.aud
10000 Hz
LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window
F0=256 Hz
22635
-29983
Residual
328.00 msec
LC: 337.50 msec
RC: 341.40 msec
353.60 msec
Glottal Waveform (top)
Flow Derivative (bottom)
Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)
Flow Derivative Spectrum

F1	554	B1	100	F6	4927	B6	783	Z1		ZB1	
F2	801	B2	218	F7	0	B7	0	Z2		ZB2	
F3	1518	B3	115	F8	0	B8	0	Z3		ZB3	
F4	2478	B4	502	F9	0	B9	0	Z4		ZB4	
F5	3722	B5	206	F10	0	B10	0				

Figure 14. Effect of a resonance below F1 to the inverse filter.



F1	535	B1	218	F6	0	B6	0	Z1		ZB1	
F2	1518	B2	115	F7	0	B7	0	Z2		ZB2	
F3	2478	B3	502	F8	0	B8	0	Z3		ZB3	
F4	3722	B4	206	F9	0	B9	0	Z4		ZB4	
F5	4927	B5	783	F10	0	B10	0	if.aud			

10000 Hz

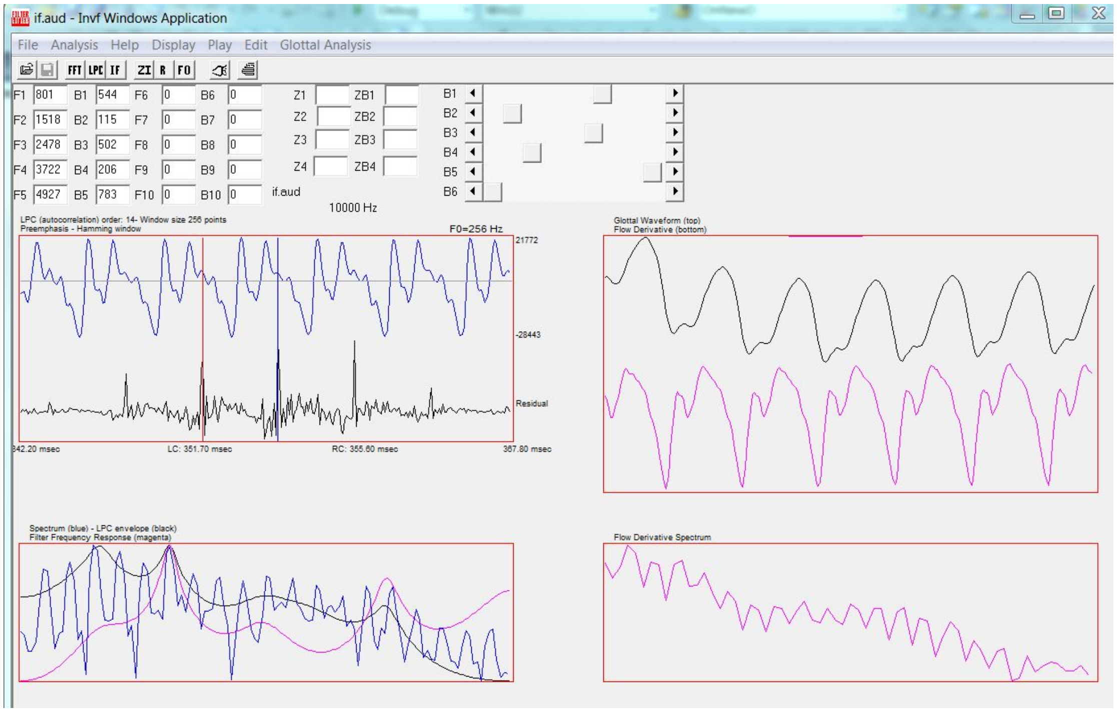
LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window

F0=256 Hz

Figure 15. Result of decreasing the frequency of F1 by dragging the resonance peak to a lower value.

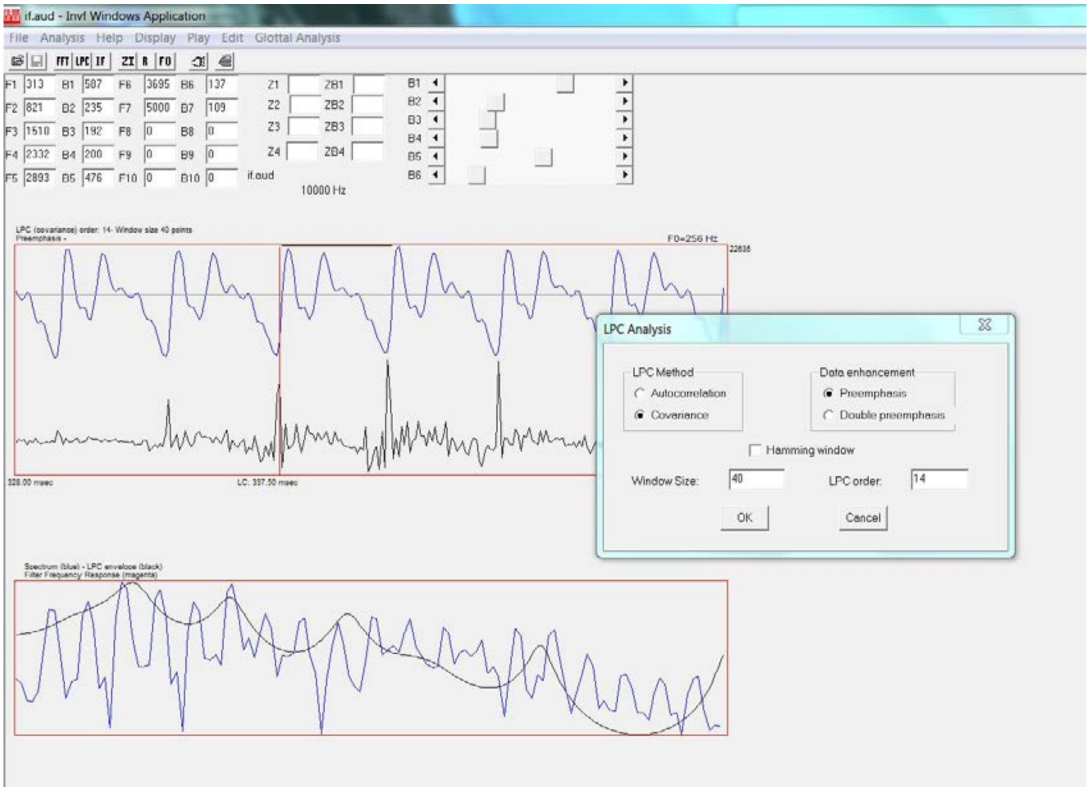
Autocorrelation estimates of vocal tract parameters are robust, but may contain errors in unstable voices due to the long analysis window. Vocal tract parameters may also be estimated in the inverse filter using covariance LPC analysis, which applies a short window but assumes complete or near-complete glottal closure. To estimate the vocal tract using covariance analysis, begin by windowing the signal, calculating an FFT, and using autocorrelation LPC analysis to select a cycle and calculate F0, as described above. Then click the LPC button on the toolbar again, and this time select covariance (Figure 17). The default window size of 56 points is usually too long. Depending on F0, adjust this value so that the analysis just includes the most-closed phase of the cycle (usually this is the section with the largest excitation ripples). If you use fewer than 29 points, change the order to 12. When you click OK, a bar will appear above the time series waveform showing the position and size of the window applied in estimating the vocal tract (as indicated by the arrow in Figure 18).

When you are satisfied with the window size, click the IF button to proceed with the analysis, as above. The output of the inverse filter based on the covariance LPC analysis is shown in Figure 18. The result is very similar to that obtained using autocorrelation LPC, except for a spurious formant at 5 kHz which produces a very steep drop-off in the flow derivative spectrum at high frequencies. This often occurs with covariance analysis inverse filtering unless you fuss with the analysis order ahead of time. Remove this formant by double right clicking or move it to a lower frequency and all will usually be well.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
Z1 ZB1 Z2 ZB2 Z3 ZB3 Z4 ZB4
B1 B2 B3 B4 B5 B6
if.aud
10000 Hz
LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window
F0=256 Hz
21772
-28443
Residual
342.20 msec
LC: 351.70 msec
RC: 355.60 msec
367.80 msec
Glottal Waveform (top)
Flow Derivative (bottom)
Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)
Flow Derivative Spectrum

F1	801	B1	544	F6	0	B6	0
F2	1518	B2	115	F7	0	B7	0
F3	2478	B3	502	F8	0	B8	0
F4	3722	B4	206	F9	0	B9	0
F5	4927	B5	783	F10	0	B10	0

Figure 16. Result obtained when the bandwidth of F1 is increased by dragging the sliding cursor.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
Z1 ZB1 Z2 ZB2 Z3 ZB3 Z4 ZB4
B1 B2 B3 B4 B5 B6
if.aud
10000 Hz
LPC (covariance) order: 14- Window size 40 points
Preemphasis -
F0=256 Hz
22638
328.00 msec
LC: 337.50 msec
Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)
LPC Analysis
OK
Cancel

F1	313	B1	587	F6	3695	B6	137
F2	821	B2	235	F7	5000	B7	109
F3	1510	B3	192	F8	0	B8	0
F4	2332	B4	200	F9	0	B9	0
F5	2893	B5	476	F10	0	B10	0

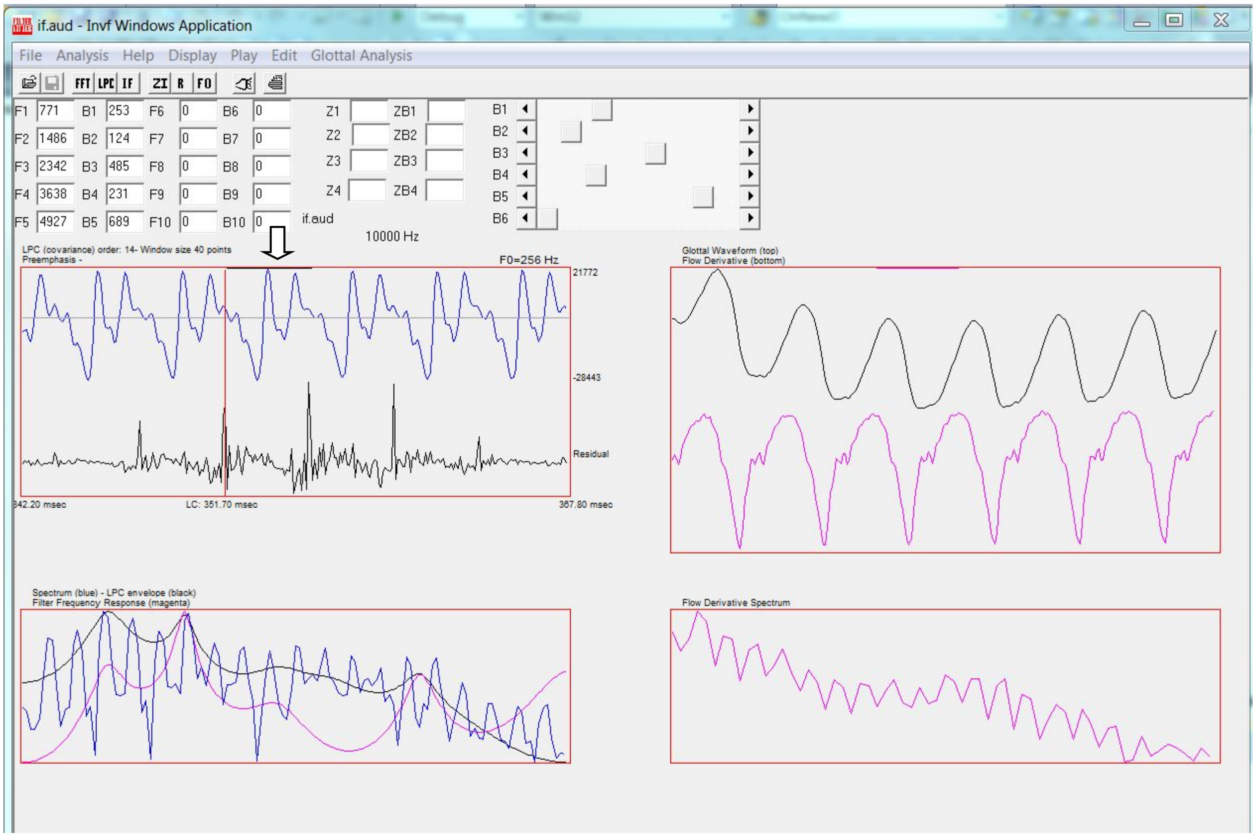
LPC Method
○ Autocorrelation
● Covariance

Data enhancement
● Preemphasis
○ Double preemphasis

□ Hamming window

Window Size: 40
LPC order: 14

Figure 17. Covariance LPC analysis dialog box.



F1	771	B1	253	F6	0	B6	0	Z1		ZB1	
F2	1486	B2	124	F7	0	B7	0	Z2		ZB2	
F3	2342	B3	485	F8	0	B8	0	Z3		ZB3	
F4	3638	B4	231	F9	0	B9	0	Z4		ZB4	
F5	4927	B5	689	F10	0	B10	0				

LPC (covariance) order: 14- Window size 40 points
Preemphasis -
F0=256 Hz

Figure 18. Result of covariance LPC analysis. Arrow shows black window size indicator.

The inverse filter also includes a provision for canceling apparent spectral zeros by adding a pole to the model. To do this, type the frequency and bandwidth into the table between the formant values and bandwidth sliders. This is fun to play around with, but we have never found it to be particularly helpful.

Print and Save Files for Use in the Synthesizer

Once you are satisfied with your result, you can print the inverse filter window by clicking on the printer icon. (Be sure to use landscape format.) You can also save the files needed to import this case into the synthesizer. To save, select File-Custom save-Save for synthesizer-Windows (Figure 19). This command creates the directory \synthesis\work\filename (in this case, filename= if), into which it places 3 files: filename.lv, filename.par, and filename.s. Filename.lv is a 1-second sample of the original voice used as a standard of comparison for later modeling efforts. Listen to this file (it's in ASCII format; convert to .wav or .aud in the Sky utility program if necessary) to be sure that it is representative and suitable. Filename.par contains various parameter values needed by the synthesizer; and filename.s contains several cycles of the inverse filtered source waveform.

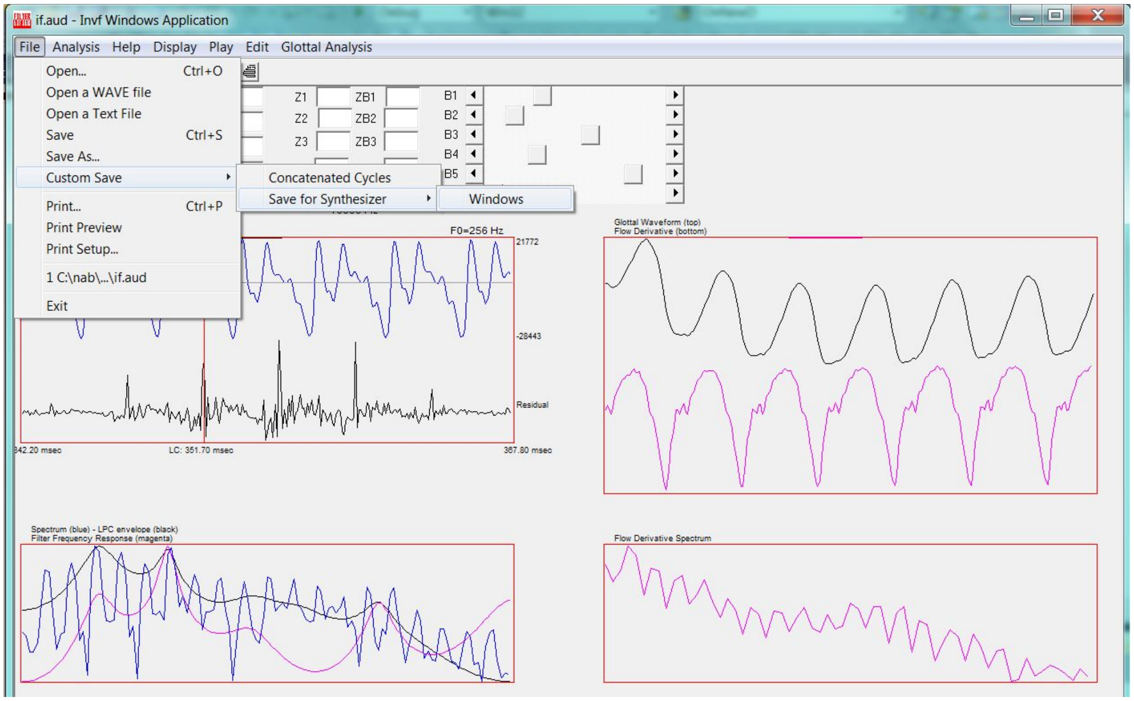


Figure 19. Dialog boxes used to save files needed for input to the voice synthesizer.

If you are not completely happy with the inverse filtering results, you can try repeating the analysis on a single cycle of phonation. This removes variability in resonances across the sample, and can improve the outcome in cases where formant estimation is particularly difficult. To do this, once you have finished analyzing the connected speech as described above, use the command File-Custom save-Concatenated cycles (Figure 19). This creates a file, filenamec.aud (e.g., IFc.aud), with the selected cycle repeated a number of times. (The c stands for concatenated.) (All files created from now on will have the form filenamec.xxx, where filename = the original filename.) This pulse will form the basis for source modeling in the synthesizer, so be sure you like it.

After you have saved the concatenated cycles, the inverse filter closes the original audio file and opens filenamec.aud (the new file containing a single concatenated cycle of phonation; Figure 20). Repeat the inverse filtering process on this concatenated cycle:

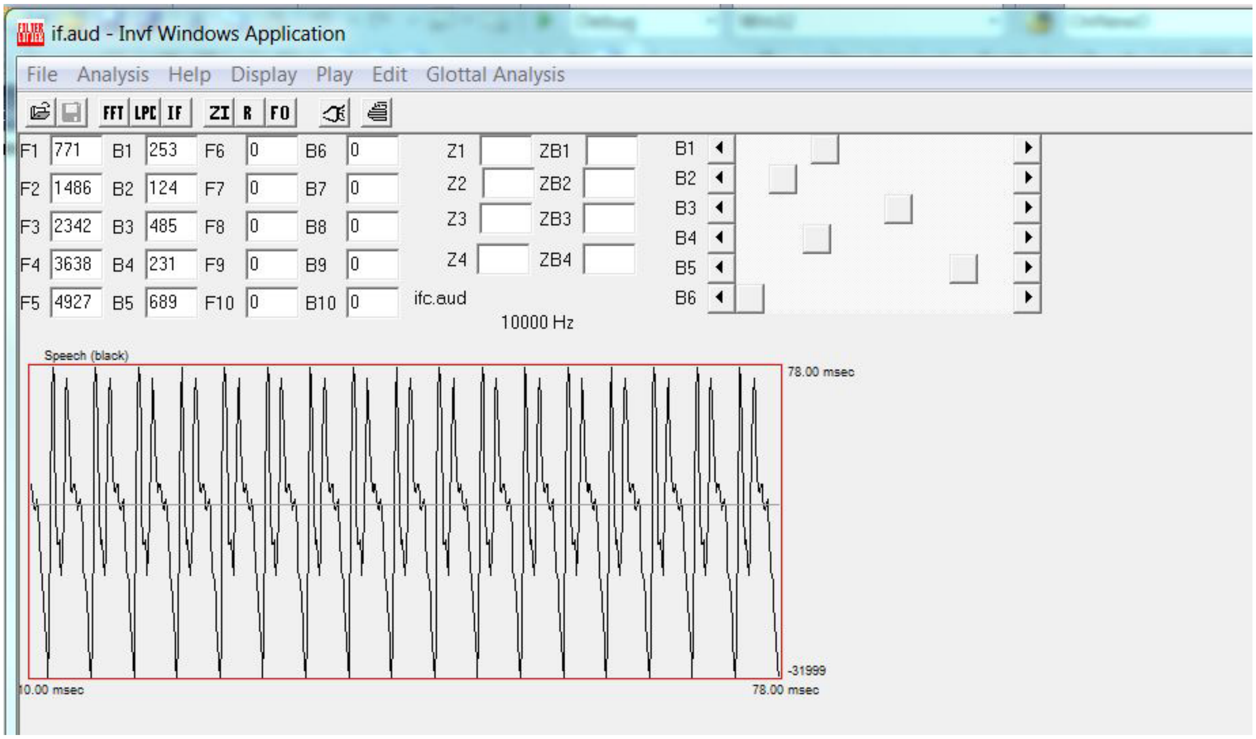
1. Click the FFT button on the toolbar; adjust analysis length if necessary.
2. Click the LPC button on the toolbar; perform an autocorrelation analysis, adjusting window length if necessary.
3. Right and left click to define a cycle; click the F0 button on the toolbar.
4. If desired, click the LPC button again and perform a covariance LPC analysis; adjust window size and analysis order if necessary.
5. Click the IF button to inverse filter the signal. Adjust formants and bandwidths until satisfied with the output (Figure 21).

Filter parameters can now be adjusted as desired to alter the output of the inverse filter, as described above. The result of this process for the example voice is shown in Figure 21.

We offer these final suggestions regarding the use of the inverse filter. In general, we have found that it is better to undermodel than to overmodel. Remember, you will get rid of extra ripples and bumps if you LF fit the source pulse or smooth the source spectrum (see Section III);

you don’t have to do all the work here. Achieving a satisfactory result may require several trials over different cycles. Don't worry if the formants and bandwidths seem strange. The only purpose of the vocal tract model used here is to make a good inverse filter, not to model vowel quality, and many other factors (including but not limited to source-tract interactions) will influence the filter characteristics that you finally settle on.

Also, remember that you are working on a noise-free simulation of a noisy signal. Error is built into the process right from the start, so obsessing about getting the "right answer" is usually misguided. You will get a chance to derive a perceptually corrected answer when you model this voice in the synthesizer, so try to be patient now.



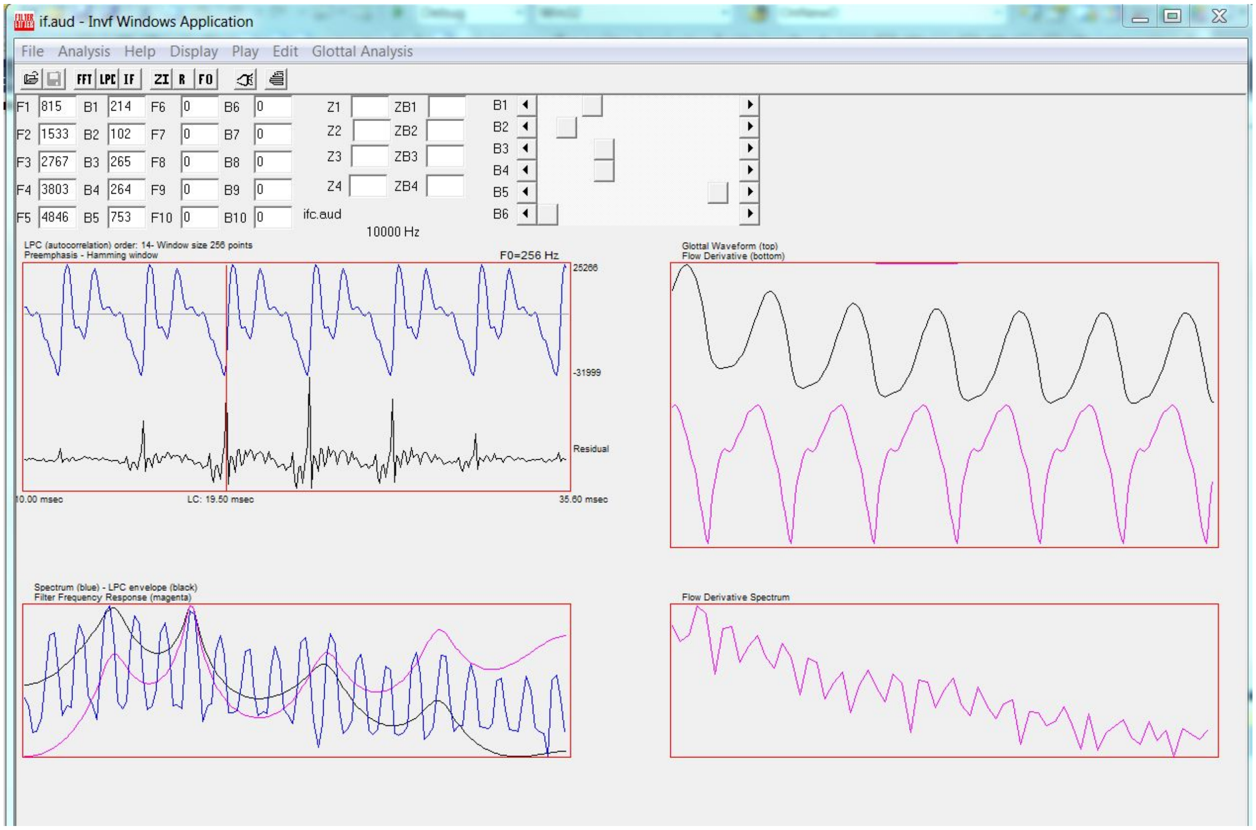
F1	771	B1	253	F6	0	B6	0	Z1		ZB1	
F2	1486	B2	124	F7	0	B7	0	Z2		ZB2	
F3	2342	B3	485	F8	0	B8	0	Z3		ZB3	
F4	3638	B4	231	F9	0	B9	0	Z4		ZB4	
F5	4927	B5	689	F10	0	B10	0				

ifc.aud

10000 Hz

B1
B2
B3
B4
B5
B6

Figure 20. Result of the File-Custom save-Concatenated cycles command, which concatenates a single cycle and places the result in the analysis window.



if.aud - Invf Windows Application

F1	815	B1	214	F6	0	B6	0
F2	1533	B2	102	F7	0	B7	0
F3	2767	B3	265	F8	0	B8	0
F4	3803	B4	264	F9	0	B9	0
F5	4846	B5	753	F10	0	B10	0

ifc.aud

10000 Hz

LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window

F0=256 Hz

Figure 21. Results of inverse filtering process for a single concatenated cycle. These serve as input to the synthesis process.

Part 3: Other Features of the Inverse Filter

Introduction

This section describes additional functions available in the inverse filter that are not listed in the preceding section. Features are listed according to the menu in which they occur.

File Menu

File-Open a text file: Use this command to open a sound file in ASCII format.

Help Menu

Not currently helpful.

Display Menu

The Display menu is shown in Figure 22. The following additional commands are available.

Display-Glottal window-Display zero line in flow derivative-Insert: This command inserts a line at the current zero value in the display for reference. If there is no constant DC offset in the signal, this will align with the closed portions of the flow derivative and flow pulse. It is particularly useful to check this if you are going to fit the pulses with the LF model in the synthesizer, because the LF fitting procedure can go awry if the pulses are substantially offset from zero. If this is the case for your data, proceed to the next command.

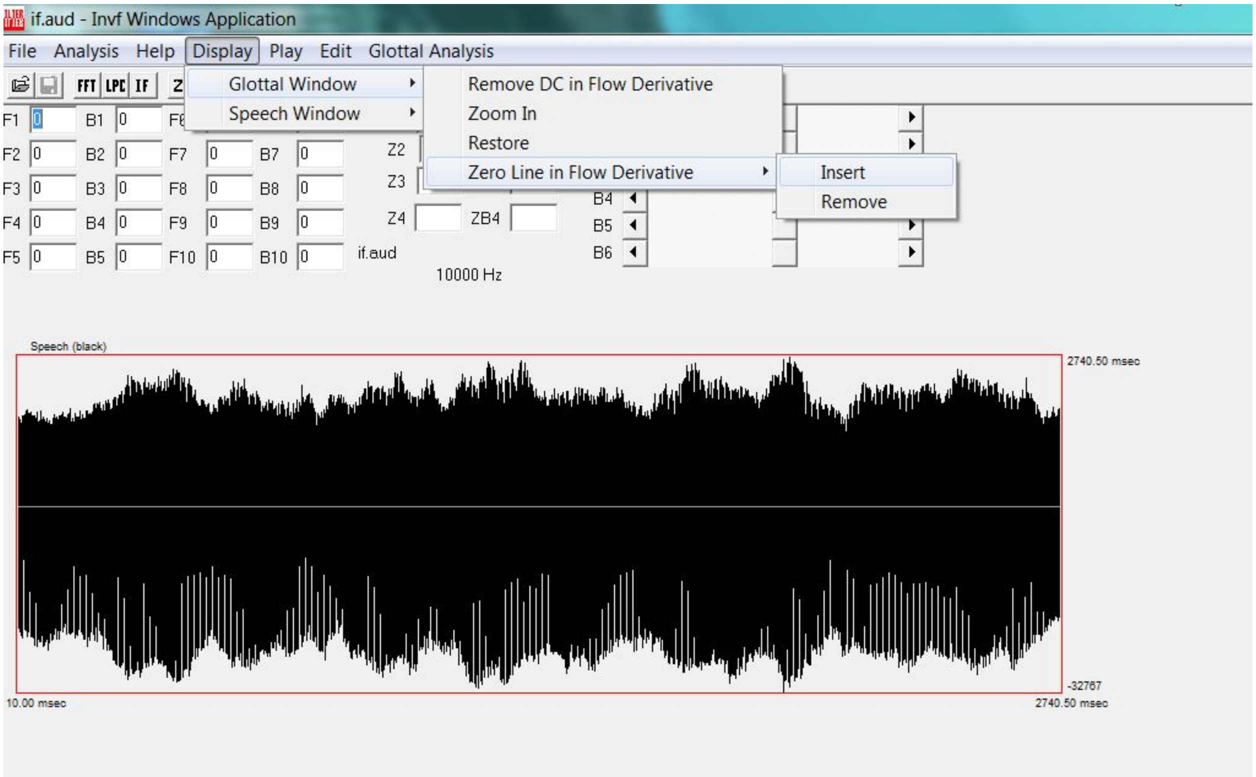

if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF Z
Glottal Window
Speech Window
Remove DC in Flow Derivative
Zoom In
Restore
Zero Line in Flow Derivative
Insert
Remove
F1 0 B1 0
F2 0 B2 0 F7 0 B7 0 Z2
F3 0 B3 0 F8 0 B8 0 Z3
F4 0 B4 0 F9 0 B9 0 Z4 ZB4
F5 0 B5 0 F10 0 B10 0 if.aud
10000 Hz
B4
B5
B6
Speech (black)
2740.50 msec
-32767
10.00 msec
2740.50 msec

Figure 22. Display menu options.

Display-Glottal window-Remove DC in flow derivative: Figure 23 illustrates this process. To recenter the display around the “true” zero line, first left-click to set a cursor at the desired zero point. Then use this command to remove the DC offset and rezero the data. This command can be repeated if you are not satisfied with the first point you select.

Display-Glottal window-Display zero line in flow derivative-Remove: Removes the zero line from the display. This does not affect the zero location; it just hides the line.

Edit Menu

The Edit menu includes the following additional commands.

Edit-Invert the waveform: Multiplies the acoustic signal by -1.

Edit-Highpass: Applies a linear phase high pass filter to remove baseline noise caused by air currents in the recording suite. This is only a problem when the microphone has a very good DC response. Figure 24 shows a typical signal before and after use of this feature. A center frequency of 6 Hz and a transition band of 10 Hz usually work pretty well. Because the filter is linear phase, this does not affect the output of the inverse filter.

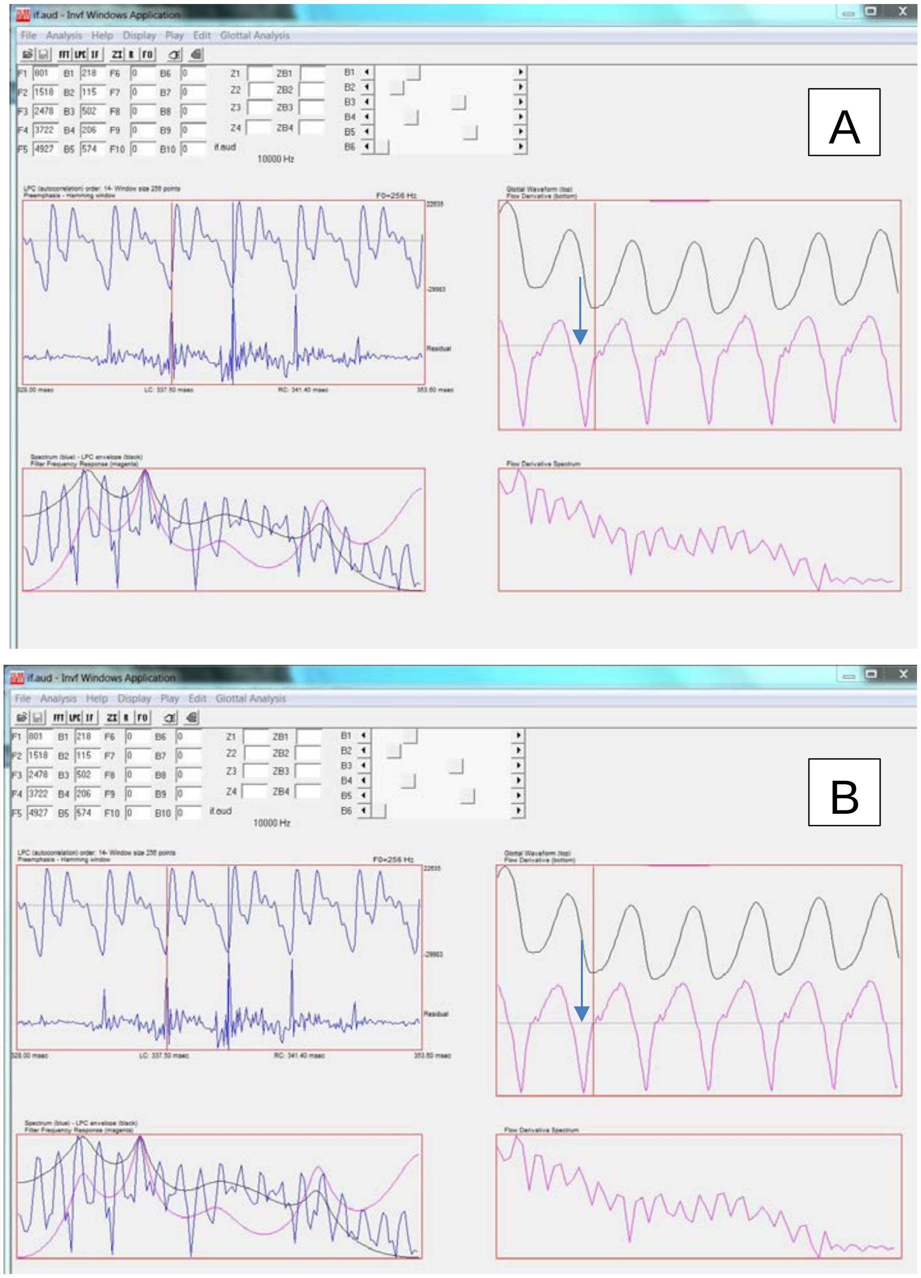
Glottal Analysis Menu

The Glottal Analysis menu allows the user to make preliminary estimates of glottal timing features.

Mark features: Marks and displays the instants of closing, opening, maximum flow, and maximum closing velocity for the current glottal waveform, as shown in Figure 25.

Compute: Using the marks shown in Figure 25, this command calculates the open quotient, speed quotient, speed index, rate quotient, DC offset, and peak flow. The last two measures are not meaningful for audio data.

Delete glottal marking: Erases the marked glottal features from the display.


if.aud - Invf Windows Application
File Analysis Help Display Play Edit Glottal Analysis
FFT LPC IF ZI R F0
A
B
LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window
F0=256 Hz
22635
-29983
Residual
328.00 msec
LC: 337.50 msec
RC: 341.40 msec
353.60 msec
Glottal Waveform (top)
Flow Derivative (bottom)
Spectrum (blue) - LPC envelope (black)
Filter Frequency Response (magenta)
Flow Derivative Spectrum

F1 801 B1 218 F6 0 B6 0 Z1 ZB1
F2 1518 B2 115 F7 0 B7 0 Z2 ZB2
F3 2478 B3 502 F8 0 B8 0 Z3 ZB3
F4 3722 B4 206 F9 0 B9 0 Z4 ZB4
F5 4927 B5 574 F10 0 B10 0 if.aud
10000 Hz

Figure 23. Resetting the zero line to remove DC offset from the flow derivative. The arrow shows the zero line before and after resetting. A: Inverse filtered file with zero placed after the apparent instant of glottal closure/above the zero point. B: Corrected zero line after removal of the DC offset.

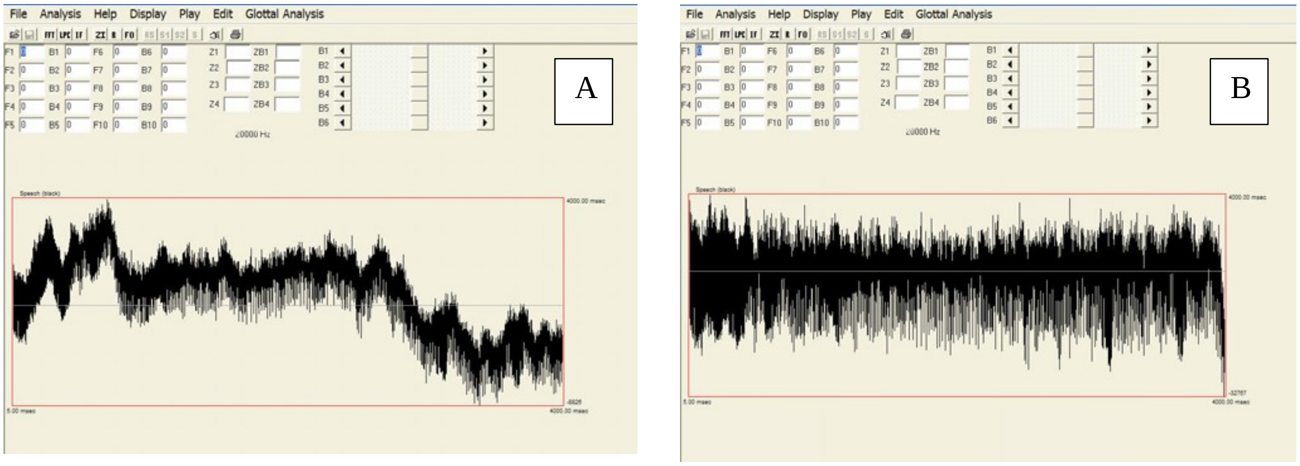
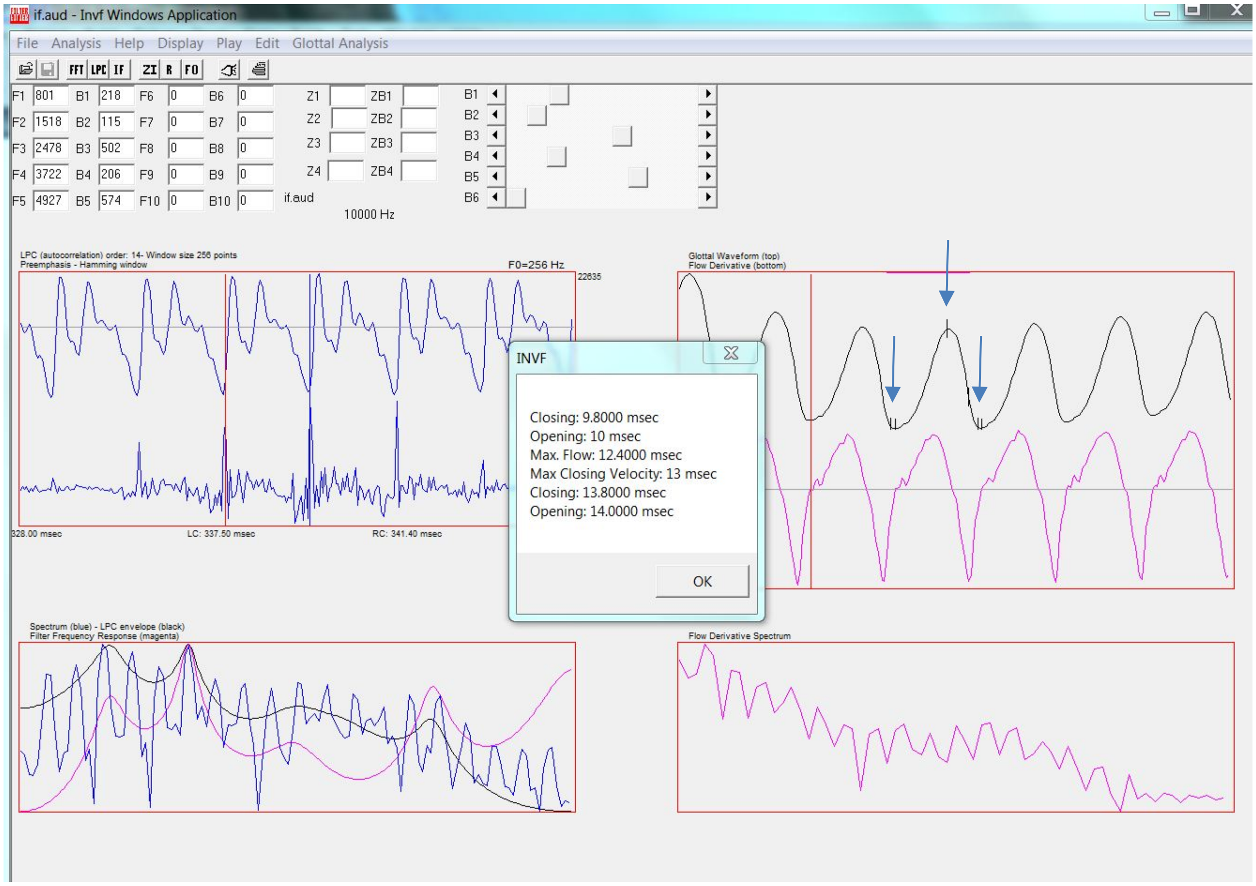


Figure 24. Removing baseline drift by high pass filtering the file. A: Audio signal including significant baseline drift. B: The same file after application of a linear phase filter at 6 Hz.



F1	801	B1	218	F6	0	B6	0
F2	1518	B2	115	F7	0	B7	0
F3	2478	B3	502	F8	0	B8	0
F4	3722	B4	206	F9	0	B9	0
F5	4927	B5	574	F10	0	B10	0

LPC (autocorrelation) order: 14- Window size 256 points
Preemphasis - Hamming window

F0=256 Hz

INVF

Closing: 9.8000 msec
Opening: 10 msec
Max. Flow: 12.4000 msec
Max Closing Velocity: 13 msec
Closing: 13.8000 msec
Opening: 14.0000 msec

Figure 25. Timing features marked in the glottal waveform.

III. VOICE SYNTHESIS SOFTWARE

Part 1: Introduction

About Voice Synthesis and the UCLA Voice Synthesizer

This introduction reviews the features of the UCLA voice synthesizer and some of the issues surrounding voice synthesis in general. It also describes the algorithms used in the synthesizer. The second part of the documentation describes step-by-step procedures for basic analysis and synthesis, and the third section provides details of some additional features of the synthesizer that aren't necessarily needed for every case.

The voice synthesizer is a formant synthesizer, based on the source-filter theory of speech production (Fant, 1960). Accordingly, users must model the vocal source, which is then filtered through a cascade of resonators that models the vocal tract response (e.g., Klatt, 1980). This synthesizer differs from most other formant synthesizers in the precision with which the source can be modeled, and in the degree of interactivity. It also differs from other synthesizers in that it is limited at present to modeling of vowels with steady-state resonances, although instabilities in the source functions can be modeled in some detail, and linguistic changes in voice quality can be modeled using cut-and-paste techniques (see Garellek et al., 2013, for an example).

The synthesis process begins by generating an estimate of the shape of the harmonic part of the source. When the goal is to copy a specific voice, this can be accomplished through inverse filtering (as described in the previous chapter of this manual). Alternatively, a source can be imported from outside the program; for example, a source pulse with the desired characteristics can be created from scratch in another program (a text editor or other program), converted to ASCII format, and then imported. Finally, one of the supplied sample voices can be opened in the synthesizer and its source can be edited until it has the desired characteristics.

The inharmonic part of the source (noise excitation) is estimated through application of a cepstral-domain comb lifter like that described by de Krom (1993). Noise analysis takes place within the synthesizer, as described below, or noise with a specific spectrum can be imported from outside the synthesizer.

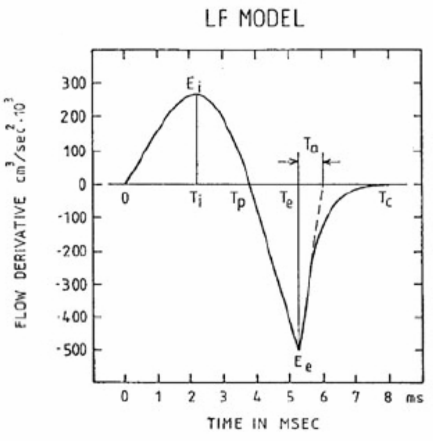
The third step in voice modeling is assessment of the patterns of F0 and amplitude modulation (vocal tremors or pitch contours). Several approaches are available. F0 and amplitude can be tracked within the synthesizer and then smoothed, with the degree of smoothing specified by the user. Alternatively, pitch tracks can be imported from outside the synthesizer. Finally, users can model instabilities using two synthetic tremor models, one that models sinusoidal modulations and one that provides "random" contours. Jitter and shimmer are also modeled directly or by increasing the amplitude of low frequency noise in the inharmonic spectrum, as described below.

Finally, users model the vocal tract response by specifying formant frequencies and bandwidths. Again, these can be based on analyses of a specific voice to be copied; a desired configuration can be created from scratch; or the sample cases can be imported and then manipulated as desired.

Issues in Source Modeling

Accurate modeling of the voice source is an essential part of accounting for variations in voice quality (e.g., Ananthapadmanabha, 1984; Karlsson, 1991). Inverse filtering is commonly used to estimate the shape of the voice source, but despite an experimenter's best efforts the recovered glottal flow waveform often includes ripples, bumps, and other theoretically undesirable but in practice unavoidable features. It is hard to tell if these wiggles and bumps are errors or if they're real features of the voice source, and we often lack the data from imaging or aerodynamics to disambiguate this issue. However, synthesizing a voice without removing at least some of these bumps and lumps typically provides a terrible-sounding result, suggesting that at least some of them are in fact errors.

One common approach to coping with this situation is to fit the output of the inverse filter with a theoretical model of the glottal flow pulse. In practice, substituting the modeled flow for the experimentally derived flow eliminates errors, wiggles, bumps, and excess high-frequency formant ripple and attendant high-frequency distortion, while preserving most of the (hypothetically) important features of the pulse shapes. Experiments with synthetic voices have further shown that smoothing with a theoretical model increases the accuracy with which various parameters of the glottal source can be estimated (Strik, 1998). The synthesizer also allows users to model the source in the spectral domain, and smoothing the source spectrum has a similar effect to model fitting in the time domain without requiring a priori assumptions about which time-domain source features might be perceptually important (see Kreiman et al., 2015, for review).



$$E(t) = E_0 e^{\alpha t} \sin \omega_g t$$
$$(t < T_e)$$
$$E(t) = \frac{-E_e}{\epsilon T_a} \cdot \left[e^{-\epsilon(t-T_e)} - e^{-\epsilon(T_c - T_e)} \right]$$
$$(T_e < t < T_c)$$

$$\omega_g = 2\pi F_g \qquad F_g = \frac{1}{2T_p} \qquad T_c = T0 = \frac{1}{F0}$$
$$R_g = \frac{F_g}{F0} \qquad R_k = \frac{T_e}{T_p} - 1 \qquad R_a = \frac{T_a}{T0}$$
$$O_q = \frac{T_e + T_a}{T0} \qquad O_q' = \frac{T_e}{T0} \qquad F_a = \frac{1}{2\pi T_a}$$

Figure 26. The LF model of the voice source. From G. Fant & Q. Lin, "Frequency domain interpretation and derivation of glottal flow parameters [*STL-QPSR* **2-3**, 1-21 (1988)].

1. Time domain source modeling

Many time-domain source models have been proposed (Ananthapadmanabha, 1984; Imaizumi et al., 1991; see Fujisaki & Ljungqvist, 1986, Ní Chasaide & Gobl, 2010, or Kreiman et al., 2015, for review), including physiological models (e.g., Ishizaka & Flanagan, 1972; Cranen & Schroeter, 1996), models of the glottal flow pulse (Rosenberg, 1971; Fant, 1979), and models of the glottal flow derivative (Fant et al., 1985; Fujisaki & Ljungqvist, 1985). The most common choice, and the one implemented in the present synthesizer, is the LF model (Figure 26; Fant et al., 1985). This model of the glottal flow derivative is well-documented and includes a relatively small number of parameters, which can be estimated from inverse filtered waveforms. Modeling the flow derivative pulse, rather than the volume velocity pulse, has the advantage that rapid changes in pulse shape around the time of glottal closure are emphasized in the derivative domain, compared with other representations. In particular, the rate and moment of glottal closure and the moment of maximum airflow are easy to specify in this model, all of which are important determinants of the source spectral slope. In our implementation, the LF model is fitted to the output of the inverse filter by iterative least squares minimization performed on major features of the time domain LF curve. The spectrum of the LF-fitted pulse is calculated and displayed along with the raw output of the inverse filter and the LF-fitted pulse. These data serve as starting values for source modeling.

The LF model is composed of two segments. The first is the product of a growing exponential and a sinusoid, and models the glottal flow from the point of glottal opening (t_0) to the point of main excitation, t_e. The second segment is a decaying exponential, modeling the flow from the main excitation to the point of glottal closure, t_c. The model comprises 5 direct synthesis parameters. E_0 is a scale factor; alpha = $-B\Pi$, where B is the negative bandwidth of the sinusoid (so that the larger the alpha, the faster the increase in amplitude); $\omega_g = 2\Pi F_g$, where $F_g = 1/2t_p$; t_a, the so-called time constant of the return phase, is the projection of the geometrical tangent of the return curve at time te onto the time axis; and F0 is the fundamental frequency (Fant & Lin, 1988; see also Ní Chasaide & Gobl, 2010). However, a large variety of different sets of 5 parameters can be used to define the functions, and in common practice a set of such parameters is estimated using major features of the glottal flow derivative (see e.g. Fant et al., 1985; Fant & Lin, 1988; Lalwani & Childers, 1991). This practice facilitates interpreting the model in terms of physiologically and acoustically significant events. The critical features most often applied are:

t_c = the length of the entire pulse
t_p = the length of time that U' > 0
t_e = the time of the maximum negative value of U'
E_e = the value of the maximum negative U'
t_a = the effective duration of the return phase

where U' is the derivative of the glottal flow.

Estimation of LF parameters from flow derivative waveforms produces hypothetical but useful correspondences between the model and glottal events or wave shapes (Figure 27; e.g., Childers & Lee, 1991; Childers & Wong, 1994; see Ní Chasaide & Gobl, 2010, for review). For example, t_e is analogous to the instant at which the vocal folds achieve maximum speed of closure, and E_e (the negative peak of the flow derivative, corresponding to the maximum speed of closure) is the "excitation strength," and is correlated with vocal intensity (Fant et al., 1997; Gauffin & Sundberg, 1989). Point t_p is the time of maximum flow through the glottis (or maximum glottal opening); and t_a reflects the abruptness of glottal closure, with larger t_a equaling more gradual vocal cords approximation. (For review see Fant, 1995; Ní Chasaide & Gobl, 2010; or Kreiman et al., 2007.)

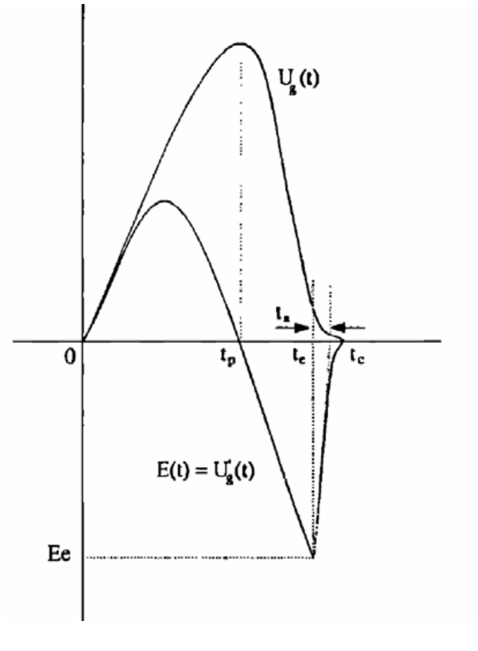


Figure 267. Correspondence between LF model points and the original glottal pulse. From Y. Qi and N. Bi, "A simplified approximation of the four-parameter LF model of voice source" [J. Acoust. Soc. Am., 96, 1182-1185 (1994)].

The LF model makes two important assumptions. First, for computational convenience it assumes that $t_c = t_0$, so that the end of one pulse is the same as the time of opening for the next cycle. Thus, as usually implemented it does not formally model the closed phase, although when t_a is small (as it often is for normal voices), the return phase of the pulse fits closely to 0, providing what amounts to a closed phase while saving a parameter (Ní Chasaide & Gobl, 2010).

Secondly, the original LF model requires the assumption that the positive area of the pulse always equals the negative area. In terms of flow, this assumption means that the baseline of consecutive pulses is constant (although that value need not equal 0; Ní Chasaide & Gobl, 2010). This is often not true for highly-variable pathological voices, and in many cases returning flow derivatives to 0 at the end of the cycle conflicts with the need to match the experimental data and with the requirement for equal areas under positive and negative curves in the flow derivative (the "equal area constraint"). In practice, the combination of these constraints often results in a modeled flow derivative that steps sharply to 0 before the end of a pulse. This introduces significant high frequency artifacts into the signal.

These conflicts between model features, constraints, and the empirical data were handled via two additional modifications. First, a linear term was added to the second (exponential) segment of the model to force the flow back to 0. This has the effect of flattening out the return phase somewhat relative to the original LF model, but improves fit to many pathological voices, for which this segment is nearly linear. Second, the present implementation of the LF model abandons the equal area constraint. Point t_c is not constrained to equal point t_0 for the following cycle, so the closed phase is formally modeled in this implementation. Third, point t_a was replaced by point t_2, defined as the time increment to 50% decay in the return phase. The new equation for the return phase is:

$$E_2(t) = -E_e e^{-E(t-t_e)} + m(t - te), \text{ where}$$

$$m = -E_e \frac{e^{-\varepsilon}(t_c - t_e) + \frac{1}{2}}{t_c - t_e}$$

Values of t_a can be calculated from t_2 by the relation:

$$ta = \frac{E_e}{\varepsilon E_e - m}, \text{ where}$$

$$\varepsilon = \frac{\log\left(\frac{m(t_e - t_2)}{E_e} + \frac{1}{2}\right)}{t_e - t_2}$$

2. Spectral domain source modeling

It is often desirable to evaluate the harmonic voice source in the spectral domain as well as (or instead of) in the time domain. A number of combinations or functions of LF parameters can be used to interpret the time-domain pulse shapes in terms of their spectral content (Gobl, 1988; Fant & Lin, 1988). For example, RA (defined as t_a/t_0), a variant index of the sharpness of glottal closure, measures the amount of airflow through the glottis after the time of maximum excitation, and is related to the spectral slope of the source pulse by the relation FA = F0/(2 ΠRA). Thus, a large RA reflects an increase in the slope of the source spectrum at higher frequencies; alternatively, the larger the t_a, the steeper the spectral tilt and the less energy in the high frequencies. Parameter RG equals [1/(2 Π t_p)]/F0; high values of RG reflect a second harmonic that is large relative to the fundamental, and a low RG reflects a relatively large H1. The open quotient (OQ), defined as [1 + [(te-tp)/tp]] / 2RG, corresponds to the energy in the lower frequencies, with a larger OQ reflecting more energy in the lowest harmonics than a small OQ. (See Fant & Lin, 1988, or Ní Chasaide & Gobl, 2010, for derivations, discussion and examples.)

Although these correspondences are useful and theoretically motivated, in practice it is difficult to manipulate combinations of LF parameters in the time domain to achieve a desired spectral change. Further, while the LF model is flexible and can accommodate a wide variety of source functions, our experiments with pathological voices have shown that it does not permit accurate modeling of all the different kinds of vocal qualities that occur in the clinic. Further, our studies comparing the perceptual adequacy of different source models show that we do not know what events in the time domain are responsible for what listeners hear (Kreiman et al., 2015). This lack of perceptual validity is a serious limitation to time-domain approaches to source modeling.

For these reasons, the synthesizer permits the source to be modified in the frequency (spectral) domain as well as in the time domain. Manipulation in the spectral domain allows source functions to be created that match the natural voices, but that violate the shape constraints imposed by the LF equations. This increased flexibility with respect to source shapes greatly improves accuracy and ease of modeling voice quality, particularly when pathology is present. The spectrum of the voice source is derived from the time-domain source estimate via pitch synchronous Fourier Transform. Note that in order to avoid the effect of leakage on harmonic amplitudes, we do not use a fast algorithm that restricts the length of the analysis window, so the true amplitude of each harmonic is accurately captured in the resulting source spectrum. The user can manipulate the amplitudes of the harmonics until the desired spectral shape is obtained, after which the program generates a new time-domain source via inverse Fourier transform.

Modeling the Inharmonic Part of the Source

1. General considerations

As mentioned above, most voice sources contain significant aperiodic (noise) components in addition to their harmonic components. These noise components contribute substantially to acoustic excitation of the vocal tract, and are an important part of a complete source model. Traditionally, two sources of spectral noise are distinguished, following models of voice production. The first is noise related to irregularities in vocal fold vibration (jitter and shimmer, representing random variability in the period and amplitude of glottal pulses, respectively; see e.g. Baken, 1987, for review). Noise also emerges due to turbulence generated during the open phase of phonation and/or flow through a persistent glottal gap (especially for normal female or pathological voices). Noise is often measured separately from jitter and shimmer with a harmonics-to-noise ratio, or with a bundle of measures representing the relative noise components in different frequency bands (e.g., de Krom, 1993; Michaelis et al., 1997; Qi & Hillman, 1997; Yumoto et al., 1982). Although jitter and shimmer are usually described in the time domain and noise in the frequency domain, any of these parameters can be measured in either domain, as convenient.

The difficulty with this approach is that jitter and shimmer add noise to the acoustic signal, and noise adds jitter and shimmer (e.g., Hillenbrand, 1987; de Krom, 1993). While in theory jitter and shimmer contribute relatively low frequency noise to the spectrum, and turbulent flow through the glottis contributes relatively high frequency noise, in practice it is impossible to separate these two sources of noise. Thus, modeling the independent contributions of perturbation and turbulence to spectral noise (as precursors to synthesizing them) presents significant problems. Further, accurate measurement of jitter and shimmer as a preliminary to noise modeling is technically difficult or impossible in the presence of significant aperiodicity (e.g., Gerratt & Kreiman, 1995; Titze, 1995). Several authors have proposed methods of disentangling measures of spectral noise from perturbation (e.g., Michaelis et al., 1997; Qi, 1992), but the proposed measures are typically still correlated with measures of perturbation, and they have not received wide acceptance. Only a few perceptual studies have appeared (Hillenbrand, 1988; Yanagihara, 1967), and the relative perceptual importance of these hypothetically different sources of noise remains poorly understood, although data (Kreiman & Gerratt, 2005) indicate that jitter and shimmer have little perceptual importance apart from the overall level of aperiodicity present in the voice.

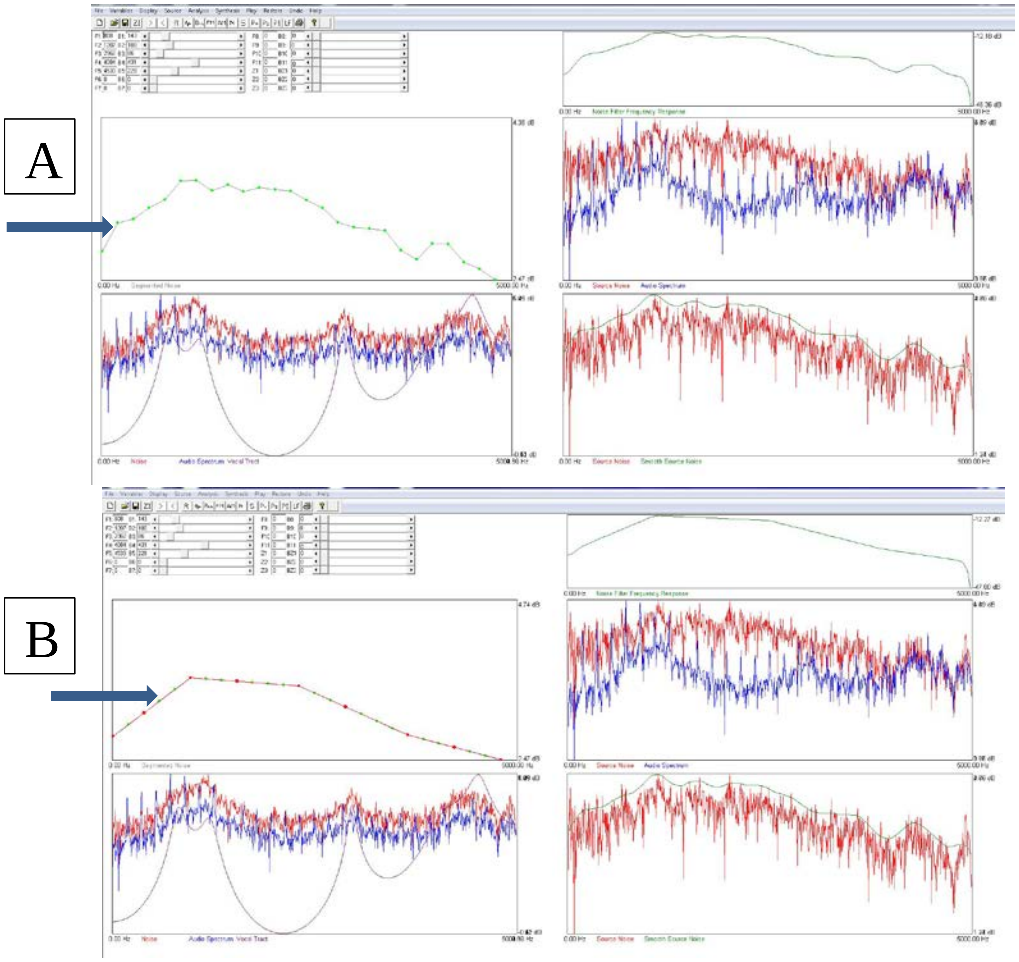


Figure 28. The four-piece noise model. A: The noise filter fit with the default 25-piece model. B: The same filter fit with the 4-piece model.

Recent studies also point to the importance of interactions between the harmonic and inharmonic sources in the perception and discrimination of voice quality (e.g., Kreiman & Gerratt, 2012; Shrivastav & Sapienza, 2006; Shrivastav & Camacho, 2010). However, the specifics of these interactions are still not very well understood. Kreiman & Gerratt (2012) studied the interaction between harmonic and inharmonic aspects of the voice source and the influence on the perception and discrimination of voice qualities. They found that the perception of the harmonic spectral slope and of noise levels in the overall voice quality pattern are significantly influenced by the interaction between the shape and relative levels of the harmonic and inharmonic energy in the voice source, so that the entire perceptual impact of either harmonic or inharmonic components of voice quality depends on both aspects of the voice source excitation.

To facilitate investigation of harmonic/inharmonic interactions in voice perception, the synthesizer now allows users to smooth the noise spectrum with any desired window. Ongoing research (Signorello et al., in preparation) suggests that most of the variance in noise spectral shapes can be accounted for by dividing the spectrum into 4 segments: 0 Hz – 961 Hz, 961 Hz – 2307 Hz, 2307 Hz – 3653 Hz, and 3653 Hz – 5000 Hz. Figure 28 shows this model applied to the spectrum of a female voice. More details about this procedure are given below.

2. Algorithms

Noise analysis follows de Krom (1993; see also Qi & Hillman, 1997). Figure 29 summarizes the analyses. Cepstral analysis is performed on a 204.8 msec segment of the original voice sample. Choice of window length is a compromise between minimizing the spectral effects of windowing (which decrease with increasing window size) and minimizing error due to changes in F0, formants, and noise levels within a window. Effects of window length are greatest when F0 is low, but decrease in importance as F0 increases and harmonics move farther apart, so that side lobes do not overlap. Window effects also interact with noise levels, and are of minimal importance for very noisy signals, because interharmonic energy due to noise exceeds the effects of spectral leakage in these cases (de Krom, 1993).

F0 is estimated using an algorithm based on Pearson correlations between successive cycles, and is used to construct a comb-lifter to remove the rahmonics. The liftered cepstrum is transformed back into the frequency domain, producing the spectrum of the noise component of the voice plus the vocal tract. This signal is then inverse filtered to remove the vocal tract information. (This analysis is updated automatically whenever the vocal tract model is changed in the synthesizer.) Finally, the estimated noise spectrum is smoothed and fit with a piece-wise linear approximation, the number of pieces being specified by the user; the default value is 25, but this can be edited with the Variables-Initialize command, as described below. The pieces will be evenly spaced on the frequency scale. To create unevenly spaced pieces, use the procedures described in Part 2 (see also Figure 28). The synthesizer initializes with the NSR set to -25 dB, but users can adjust this value as desired.

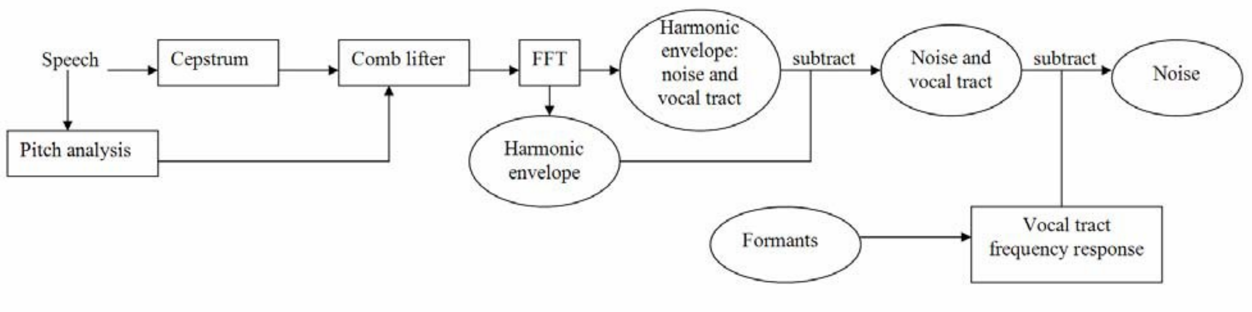


Figure 29. Flow chart showing the noise analysis process.

As discussed above, jitter and shimmer are modeled as part of a general noise component. However, these parameters may be separately manipulated in the synthesizer if necessary (for example, to test hypotheses about the separate contributions of perturbation and turbulence noise to voice quality). Jitter is modeled using a shape-preserving interpolation algorithm. Using the following equation, a rectangularly-distributed set of random values for period durations is generated; the lower limit of this distribution is equal to F0 minus the control variable, and the upper limit is equal to F0 plus the control variable:

$$\text{Jitter} = 1.0 + \frac{\text{newjit}}{100} * 2 * (\text{rand}(1) - .5)$$

where *newjit* is the user control parameter representing the desired percent jitter. Then,

$$\text{Jittered period} = \text{default period} * \text{jitter}$$

where *default period* is the period corresponding to the mean F0. After the period for a cycle has been determined, the LF source pulse is interpolated using the jittered frequency to space the lookup points for the given sample rate. This effectively time compresses or expands the source pulse without altering its shape.

The shimmer parameter varies the amplitude of each glottal pulse by applying a random gain. Pulse gain is controlled by a parameter equal to +/- the range of gain in dB. The distribution of gain values is also rectangular, with the minimum value equal to the normal gain minus the control variable, and the maximum value equal to the normal gain plus the control value.

Frequency and Amplitude Modulations (Tremor)

Slow modulations of frequency and amplitude ("tremor") occur in all voices, and can be modeled in the synthesizer. Tremor can be modeled in two ways: by tracking or importing F0 and amplitude contours, or via parameters that control the rate, extent, and pattern of frequency modulation to create synthetic tremors. In practice, importing the pitch and amplitude contours works very well for most voices. When F0 cannot be tracked, or when it is desirable to modify or manipulate the tremor pattern for a voice, the synthetic tremor models can produce a wide variety of pitch and amplitude contours.

Two different algorithms are available for synthesizing frequency modulation (the acoustic correlate of tremor). The first sinusoidally modulates the vocal F0 above and below the mean F0 specified in the synthesizer (Figure 30a). This algorithm provided a good perceptual match to about one third of the voices we have studied (Kreiman et al., 2003). However, the frequency modulation for the other two thirds of voices studied was non-sinusoidal and irregular in rate. For this reason, a "random tremor" model has also been implemented (Figure 30b). In this model, the pattern of variation in F0 is generated by passing white noise through an FIR Kaiser window low-pass filter with cutoff frequency equal to the maximum modulation rate.

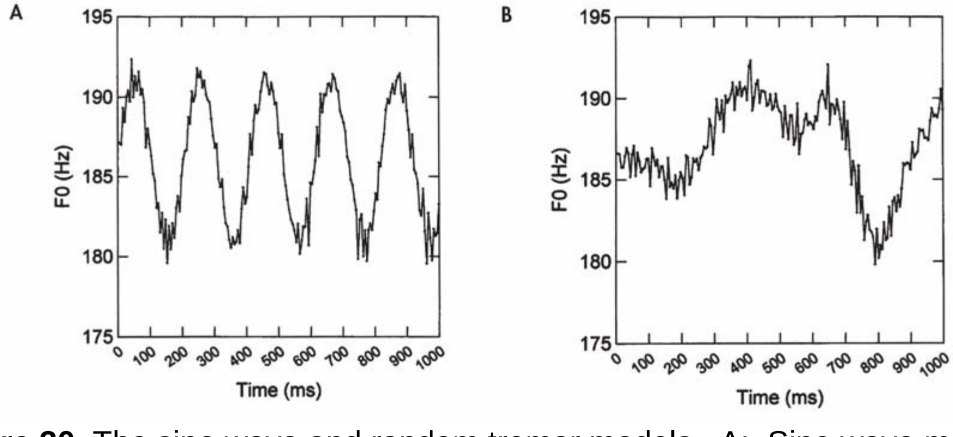


Figure 30. The sine wave and random tremor models. A: Sine wave model; B: Random tremor model. From J. Kreiman et al., "Perception of vocal tremor" [J. Speech Lang. Hear. Res., 46, 203-214 (2003)].

In the sine wave model, the frequency for each cycle of phonation is calculated as:

$$F0(t) = F0_{nom} + DHz \cdot \sin(2\pi \cdot THz \cdot t)$$

where t is time, $F0_{nom}$ is the mean F0 specified in the synthesizer, DHz is the peak amplitude of the modulating sinusoid (the amplitude of the tremor, in Hz), and THz is the repetition rate (the modulation frequency, also in Hz) of the tremor. Frequency modulation in the irregular tremor model follows the following equation:

$$F0(t) = F0_{nom} + \frac{DHz}{D_{\max}} * \left[r(t) * H(THz, t) - \frac{1}{2}\right]$$

where * denotes time domain convolution, H is the impulse response of an FIR Kaiser window low pass filter with cutoff frequency THz, r(t) is white noise uniformly distributed on [0,1], and D_{max} is the maximum excursion of r * H from 0.5. Neither tremor model includes parameters to vary amplitude, although amplitude modulations do emerge from these models, presumably due to movement of harmonics toward and away from resonance peaks as F0 varies. These models also do not specify tremor phase, so that the initial and final points of the modeled tremor do not necessarily match those of the original voice samples.

Modeling Effects of Source/Filter Interactions

Although source-filter theory assumes that the glottal source and vocal tract transfer function are linearly separable and do not interact, interactions do occur (see Childers & Wong, 1994, for review). In the present synthesizer (as in most parametric synthesizers), such effects are modeled by altering the (non-interacting) source and filter functions in such a way that the effects of interactions are incorporated, rather than by attempting to produce a genuinely interactive model (e.g., Childers & Lee, 1991; Childers & Wong, 1994). For example, glottal pulse skewing that results from subglottal coupling is modeled by appropriate variations in source parameters. Although this practice limits the extent to which the synthesizer parameters can be interpreted physiologically, it greatly facilitates the process of mapping between an acoustic signal and perceived quality. These source/filter interactions are normally transparent to the user, requiring no knowledge of these interactions or special manipulation of synthesizer parameters.

The Synthesis Process

The synthesizer sample rate is fixed at 10 kHz. To overcome quantization limits on modeling F0, the source time series is synthesized pulse by pulse. Within each pulse, samples are interpolated at exact sample instants as follows. A plot of F0 vs. time is generated for the duration of the 1-second token to be synthesized. Source pulses with frequencies dictated by the F0 vs. time plot are calculated, then concatenated to produce a synthetic token. To eliminate phase error, the absolute beginning and ending times of each pulse are tracked and used in the interpolation of succeeding pulses. At the beginning instant of each new pulse (which can occur at any time, including between samples), the F0 curve generated above is interpolated to find the frequency used for the duration of this LF pulse. This F0 sets the abscissa spacing for sample points, which is used for LF curve time domain interpolation. Because F0 and the sample rate are constant for each cycle, this procedure effectively sets the instantaneous frequency of the pulse. Using the instantaneous F0 and current sampling rate, the sample instants are converted to phase values (radians) for the abscissa of the LF curve. This is done assuming the last point in the LF curve corresponded to 2 Π radians (one complete cycle). Phase corrections are then made to these abscissae to take into account the fact that a cycle start time might not correspond to a sample instant. Given the abscissa values, interpolation is performed on the LF curve to generate the final lookup values of the source pulse to be concatenated to the growing source time series. The overall effect is equivalent to digitizing an analog pulse train with pulses of the exact desired frequencies at the fixed 10k sample rate.

A 100 tap FIR filter is synthesized for the noise spectrum, and a spectrally-shaped noise time series is created by passing white noise through this filter. The LF pulse train is added to this noise time series to create a complete glottal source time series. The ratio of noise to LF energy is adjusted so that the noise-to-periodic energy ratio approximates the value calculated from the original voice sample. Finally, the complete synthesized source is filtered through the current vocal tract model (estimated through LPC analysis, as described above) to generate a preliminary version of the synthetic voice.

Within the synthesizer, the operator is able to adjust all parameters to create the best possible synthetic match to the original voice sample. Procedures and suggestions for this process are given in the next section.

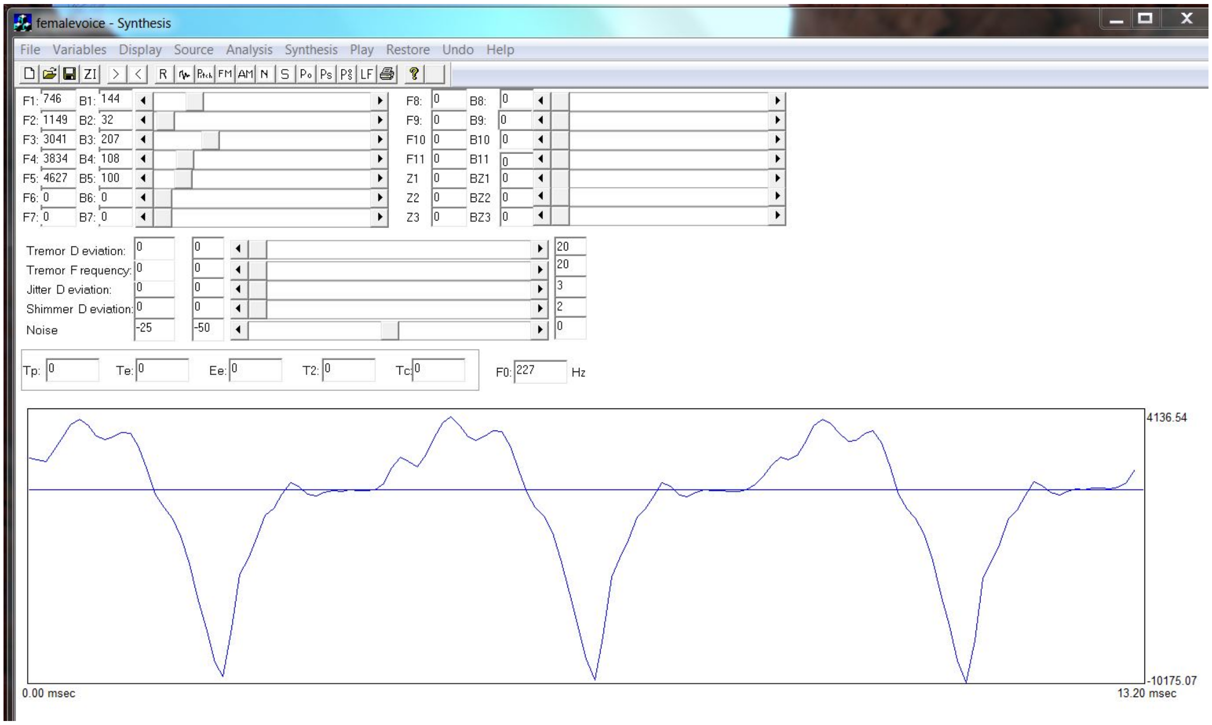
Part 2: Step-by-Step Synthesis Procedures

This section describes the procedures necessary to synthesize a typical voice. It assumes that the voice has either been inverse filtered using our software, or that it is one of the two sample cases included with the software. If some other method is used, you will need to create the three ASCII files needed by the synthesizer (filename.lv, filename.par, filename.s) and save these into the directory C:\synthesizer\work\filename. The synthesizer will not run unless these files are present.

The process of generating the initial synthetic version of a voice is nearly automatic, and begins with estimates of the acoustic characteristics of the voice. The first step is fitting an time- or spectral domain model to the output of the inverse filter, to smooth the pulse and to generate parameters that can be manipulated. Next, F0 is tracked on the natural voice sample and the track is imported to model slow changes in F0 and amplitude (tremor/vibrato). Next, the spectral shape of the inharmonic part of the source (the noise excitation) is estimated and modeled. Finally, parameters are combined with a vocal tract model (imported from the inverse filter) to synthesize a preliminary version of the voice. All the synthesizer parameters can be adjusted to improve the perceptual match to the original voice once a synthetic token has been generated. Thus, the precision of the initial estimate is not particularly important, and automatically generated parameters are usually close enough for a first pass.

Program Installation

To install the synthesizer, copy the file into the desired location on your computer. If desired, add a link to your start menu. To install the sample voices, create the additional directories C:\synthesis\work\sample1 and C:\synthesis\work\sample2 and copy all the files associated with each case into the appropriate directory. (If you have analyzed voices using the our inverse filtering software, the C:\synthesis\work directory was created automatically, along with the appropriate subdirectories for the voices you have analyzed. Create additional subdirectories for the sample voices if desired.) All the files associated with a given voice have the same name; only the extensions differ. The following files should appear in each subdirectory: **filename.lv** (a 1-sec sample of the original voice), **filename.par** (a list containing F0, the sampling rate, and the formant and bandwidth values from the inverse filter), and **filename.s** (a series of inverse-filtered glottal flow derivative pulses, produced by the inverse filter), although other files may also be present depending on the analyses that have been run. All files are in ASCII format and can be edited with a text editor or imported into a graphics package.



F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0
F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise -25 -50 0

Tp: 0 Te: 0 Ee: 0 T2: 0 Tc: 0 F0: 227 Hz

Figure 31. Synthesizer interface as it appears when a file is first opened. Glottal flow derivative waveform, formant frequencies, and bandwidths are all imported from the inverse filter.

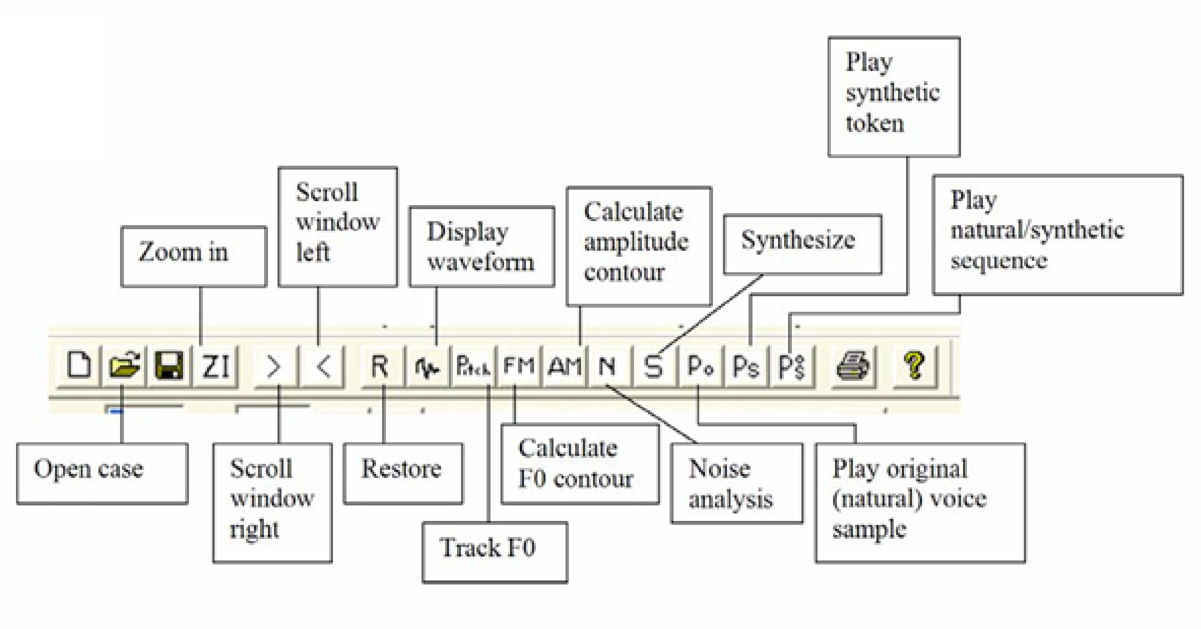


Figure 32. Details of the synthesizer toolbar.

The Synthesizer Interface

Figure 31 shows the synthesizer interface, and Figure 32 shows details of the synthesizer toolbar. Below the menu and toolbar are spaces for specifying up to 11 formants, 3 zeros, and bandwidths, along with sliding cursors that can be used to adjust bandwidths interactively. When a case is opened, this list will be populated automatically with initial values imported from the inverse filter and saved in the file **filename.par**. Below the resonance display are additional sliding cursors that are used to model vocal tremors and to control the levels of jitter, shimmer, and aspiration noise (the NSR). At the bottom of the display is an editable table showing the current LF model parameters.

Step 1: Open a File

To open a file, click the Open case icon (the second button) on the toolbar. Alternatively, use the menu to issue the command File-Open case. Enter the case name (“femalevoice” in this example) and select the appropriate file format. If the files have been created in the inverse filter, select ASCII. The .wav option is not yet implemented. Several cycles of the inverse filtered flow derivative will appear, as shown in Figure 31. Notice that this waveform appears misshapen relative to theoretical expectations: It is characterized by lumps and bumps, and it is not entirely clear where the closed phase begins (if in fact there is a closed phase). The values for formants and bandwidths that were used in the inverse filtering operation appear in the table as first estimates for synthesis. The NSR is initially set to -25 dB; other variables are initially set to 0.

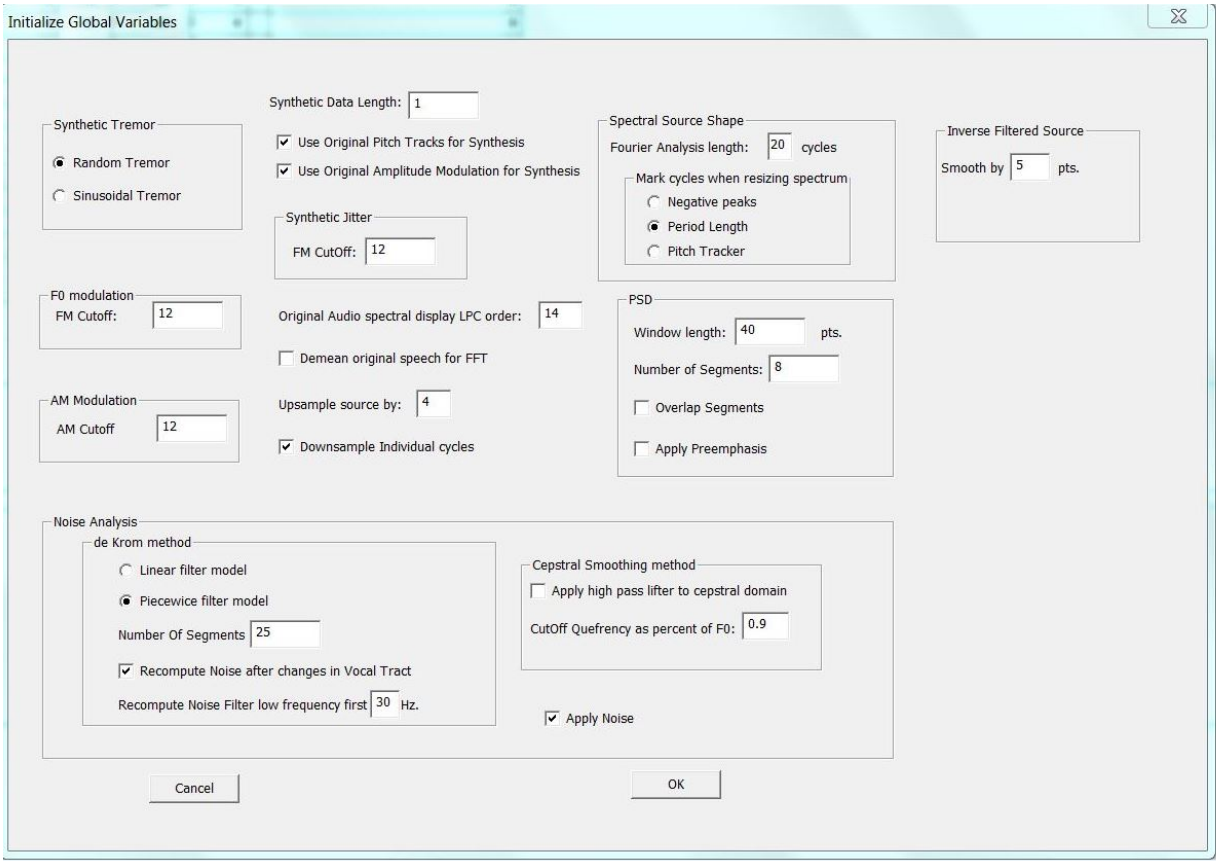
Step 2: Initialize Variables

Next, select the command Variables-Initialize from the menu. A window will open, allowing you to set a number of analysis parameters, as shown in Figure 33.

The first set of options controls the manner in which F0 and amplitude variations are modeled. If you want to create stimuli with constant F0 and amplitude, uncheck Use original pitch track for synthesis and Use original amplitude modulation for synthesis. If you want to model F0 and amplitude by copying the original F0 and amplitude tracks, leave these options checked. If you want to use the synthetic tremor models, select the tremor model you want to use. As described above, the sine wave model modulates F0 in a sinusoidal pattern; the random model creates an irregular pattern of frequency modulation (see Figure 30). Note that amplitude modulation and F0 modulation may be selected independently of one another. Values of the F0 modulation cutoff and AM modulation cutoff establish the amount of smoothing that occurs when F0 and amplitude are imported into the synthesizer. Figure 34 shows the effects of these variables. In the upper panels, both variables are set to their default values of 12 Hz. In the middle panel, both variables have been set to 3 Hz; and in the bottom panel, they have been set to 25 Hz. The default settings are appropriate for most voices, but if you find that you want more (or less) detail when you are modeling the contours based on the original voices, change the values.

Under most circumstances, other variables should be left with their default values. The Synthetic data length option cannot be changed at present, and synthetic stimuli are restricted to a duration of 1 sec. Synthetic jitter FM sets the cutoff frequency for a highpass filter used in the synthesis of jitter. The default value of 12 Hz is usually appropriate. Original audio spectral display LPC order controls the order of the LPC analysis used for the display of the natural and synthetic voice spectra; the default order of 14 is appropriate for a sampling rate of 10 kHz.

Several variables control modeling of spectral noise. First, users can choose between modeling noise very simply by fitting a straight line to the derived noise spectrum, or more precisely by fitting a number of line segments to the spectrum to better approximate its shape. If you choose the piecewise approximation, you can also specify the number of line segments to be fitted. A value of 1 fits a straight line through the center of the spectrum; linear fit fits a regression line. The model defaults to a 25 segment piecewise fit. Note that the pieces will be equally spaced along the frequency axis. It is also possible to smooth the spectrum with unevenly spaced segments, as described in below (see Figure 28).



Synthetic Tremor
● Random Tremor
○ Sinusoidal Tremor

Synthetic Data Length: 1
☑ Use Original Pitch Tracks for Synthesis
☑ Use Original Amplitude Modulation for Synthesis

Synthetic Jitter
FM CutOff: 12

Spectral Source Shape
Fourier Analysis length: 20 cycles
Mark cycles when resizing spectrum
○ Negative peaks
● Period Length
○ Pitch Tracker

Inverse Filtered Source
Smooth by 5 pts.

F0 modulation
FM Cutoff: 12

AM Modulation
AM Cutoff 12

Original Audio spectral display LPC order: 14
□ Demean original speech for FFT
Upsample source by: 4
☑ Downsample Individual cycles

PSD
Window length: 40 pts.
Number of Segments: 8
□ Overlap Segments
□ Apply Preemphasis

Noise Analysis
de Krom method
○ Linear filter model
● Piecewice filter model
Number Of Segments 25
☑ Recompute Noise after changes in Vocal Tract
Recompute Noise Filter low frequency first 30 Hz.

Cepstral Smoothing method
□ Apply high pass lifter to cepstral domain
CutOff Quefrency as percent of F0: 0.9

☑ Apply Noise

Figure 33. Options for initializing the variables that control the synthesis.

Click ok when you have finished making selections. Values of these variables can be changed at any time during an analysis, but the analyses will not automatically update with the new values. All analyses must be rerun after reinitializing a parameter.

Step 3: Fit a Model to the Inverse Filtered Source Pulses

Analyses continue by optionally fitting an LF model to the inverse filtered flow derivative waveform by issuing the menu command Source-LF-Find LF features automatically. Little red points appear on the unsmoothed flow derivative indicating the location of points used to fit the LF model (Figure 35). Next, use the menu to issue the command Source-LF-Fit an LF model. Two panels will be displayed on the screen (Figure 36). The left panel shows the best-fitting LF model (in red) plotted on top of the original inverse filtered waveform (plotted in blue). (The color key is also shown at bottom of the frame.) The right panel shows the superimposed spectra of the inverse filtered and LF fitted pulses, color coded as above.

Alternatively, fit a spectral domain source model by issuing the menu commands Source-

Inverse filter-Mark cycle, which places red dots at the beginning and end of one source pulse (Figure 37a), and then entering the command Source-Inverse filter-Compute source (Figure 37b). It helps if the cycle boundaries are relatively accurate, but exact precision isn't required.

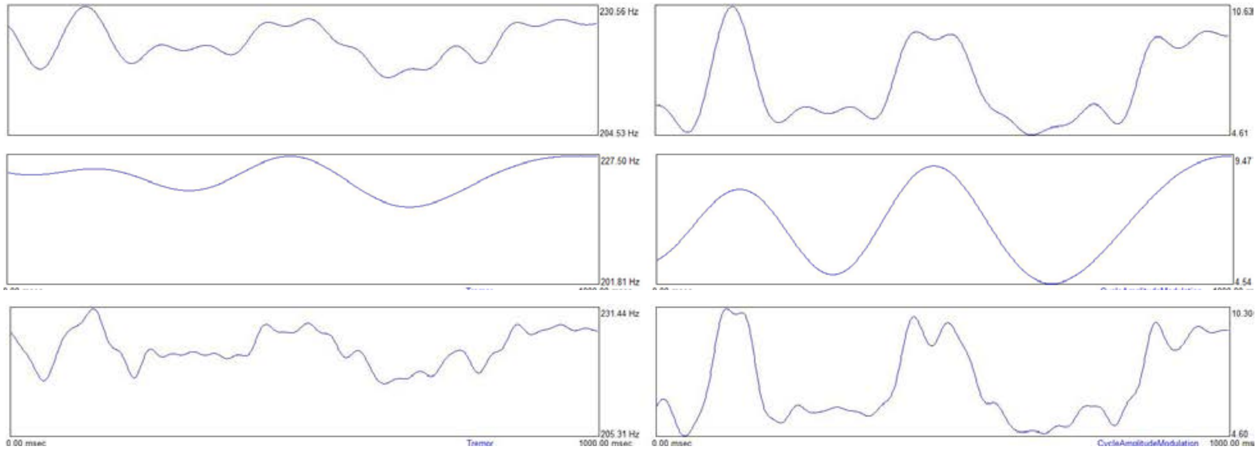


Figure 34. Effect of changing F0 and AM modulation cutoff values. The top panel shows the F0 (left) and amplitude (right) contours that result when the F0 and amplitude tracks are low-pass filtered at 12 Hz. The middle panel shows the same contours after low-pass filtering at 3 Hz, and the bottom panel shows the output of a 25 Hz low-pass filter.

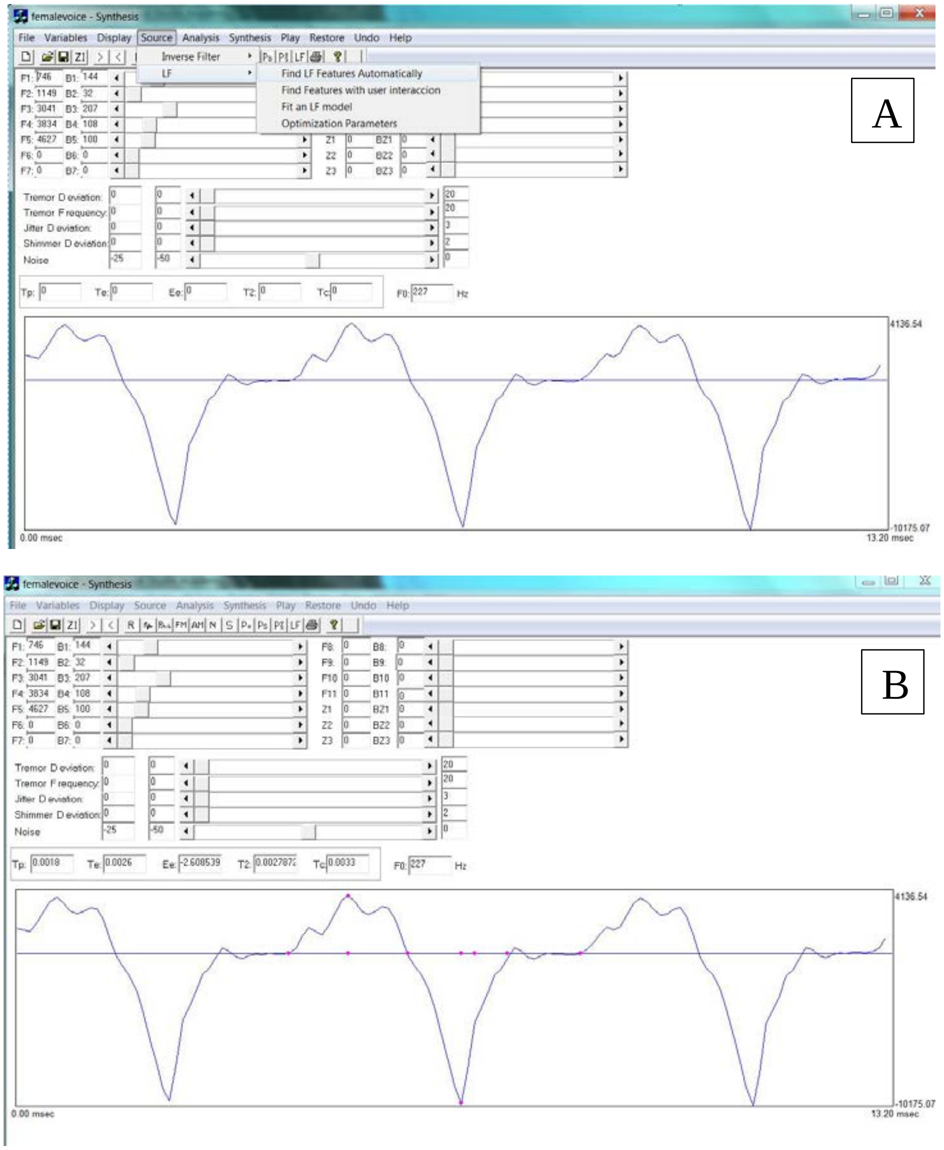
Step 4: Track F0

Next, open the file containing the complete audio signal (femalevoice.lv) by clicking the Display waveform button (#8 on the toolbar, to the right of the R button; see Figure 32), or the Display-audio-display menu option. A window will open showing a portion of the time series waveform for the voice, as shown in Figure 38. Select an event (a positive or negative peak) that repeats relatively reliably throughout the voice. The cursor will be placed at the first significant peak when the window opens. If this peak repeats reliably throughout the file, click the Pitch button to track F0. If the peak does not seem trackable, select another peak and mark it with the cursor by left clicking. Zoom the window if necessary by setting the left and right cursors (by left and right single clicking) and then clicking the ZI (zoom in) button to expand the waveform. You can page through the waveform by using the < and > buttons on the toolbar. Mark your preferred event by left clicking to place the left cursor, and then click the Pitch button (to the right of the Display audio waveform button) to track F0. The program places little red lines to show each cycle boundary, as shown in Figure 39. Page through the file to verify that the marks are placed with reasonable accuracy. Exact precision is not necessary: Because we are only generating the F0 contour and not measuring jitter, errors of 2-3 points are not serious. If the track is grossly inaccurate (marks skip from one peak to an adjacent peak; cycles omitted; etc.), try selecting a different event and repeat the process if necessary. If you continue to have trouble tracking F0, try using the interactive pitch tracker in the Sky utility program and import the contours, or abandon your efforts and model tremor with the synthetic tremor models provided in the synthesizer.

Step 5: Model Frequency and Amplitude Modulations

Once F0 has been tracked, the next step is to compute the F0 and amplitude contours for use in the synthesizer. To model the F0 contour, click the FM button on the toolbar. A new pane opens in the window, as shown in Figure the left panel of Figure 40. The top plot in this panel shows the unsmoothed F0 track for the entire voice sample. The second plot shows the smoothed pitch track. The degree of smoothing can be modified by changing the FM cutoff value with the command Variables-Initialize, as described above. The final plot in the window represents the result of subtracting the smoothed contour (low frequency modulation) from the complete F0 track, leaving the faster frequency changes usually attributed to aspiration noise or jitter.

To compute the amplitude contour, click the AM button on the toolbar. A second pane opens in the window, as shown in the right panel of Figure 40. As above, the first plot shows the unsmoothed cycle-to-cycle changes in amplitude for the entire voice sample; the second plot shows the smoothed amplitude contour (with degree of smoothing set using the Variables-Initialize command and the AM cutoff parameter), and the final plot shows the higher-frequency amplitude perturbations that remain after subtracting the slow changes in the smoothed contour.


femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
Inverse Filter
LF
Find LF Features Automatically
Find Features with user interaccion
Fit an LF model
Optimization Parameters
A
4136.54
-10175.07
0.00 msec
13.20 msec
femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
R FM AM N S
B
4136.54
-10175.07
0.00 msec
13.20 msec

A)

F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise -25 -50 0

Tp: 0 Te: 0 Ee: 0 T2: 0 Tc: 0 F0: 227 Hz

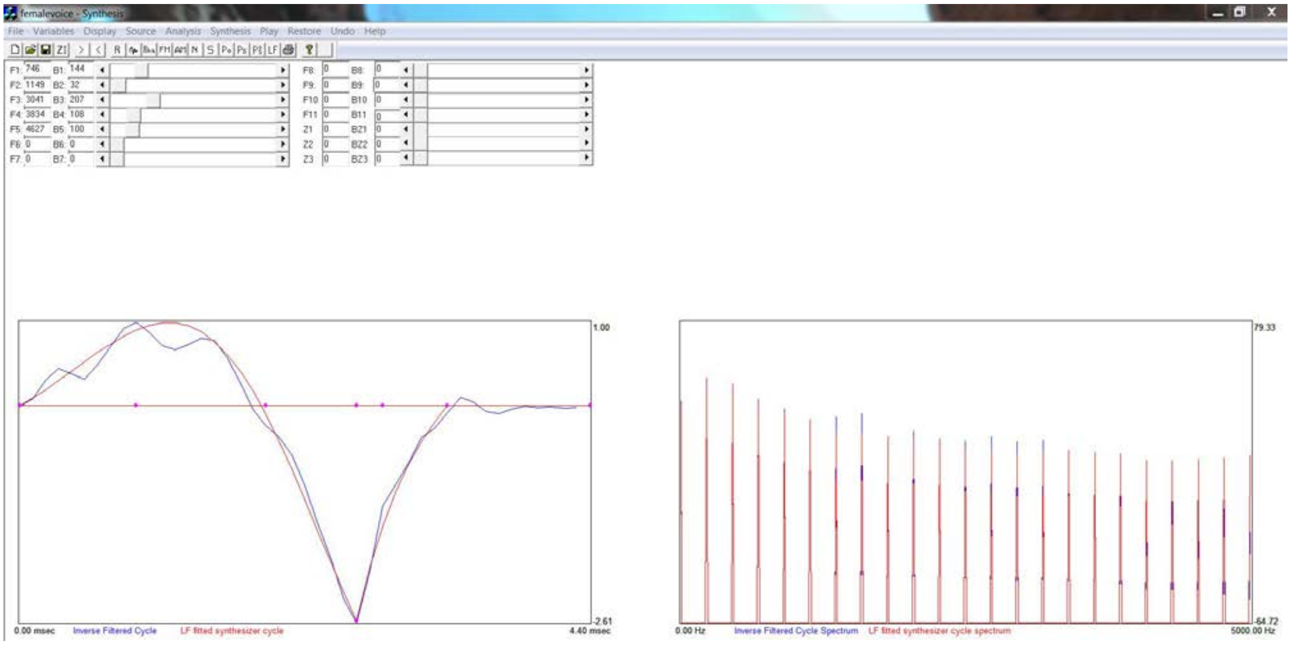
B)

F1: 746 B1: 144 F8: 0 B8: 0
F2: 1149 B2: 32 F9: 0 B9: 0
F3: 3041 B3: 207 F10 0 B10 0
F4: 3834 B4: 108 F11 0 B11 0
F5: 4627 B5: 100 Z1 0 BZ1 0
F6: 0 B6: 0 Z2 0 BZ2 0
F7: 0 B7: 0 Z3 0 BZ3 0

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise -25 -50 0

Tp: 0.0018 Te: 0.0026 Ee: -2.608539 T2: 0.0027872 Tc: 0.0033 F0: 227 Hz

Figure 35. Flow derivative pulses showing A) the Source menu and B) location of points used to fit the LF model.


femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help

F1:	746	B1:	144	F8:	0	B8:	0
F2:	1149	B2:	32	F9:	0	B9:	0
F3:	3041	B3:	207	F10	0	B10	0
F4:	3834	B4:	108	F11	0	B11	0
F5:	4627	B5:	100	Z1	0	BZ1	0
F6:	0	B6:	0	Z2	0	BZ2	0
F7:	0	B7:	0	Z3	0	BZ3	0

1.00
-2.61
0.00 msec
Inverse Filtered Cycle
LF fitted synthesizer cycle
4.40 msec
79.33
-64.72
0.00 Hz
Inverse Filtered Cycle Spectrum
LF fitted synthesizer cycle spectrum
5000.00 Hz

Figure 36. Fit LFmodel.

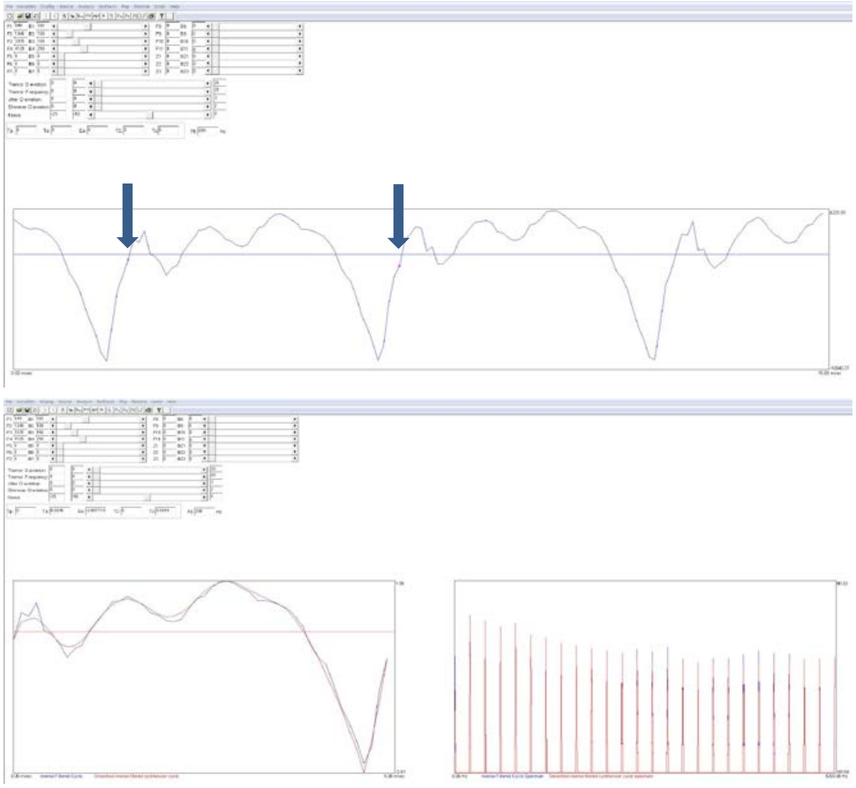
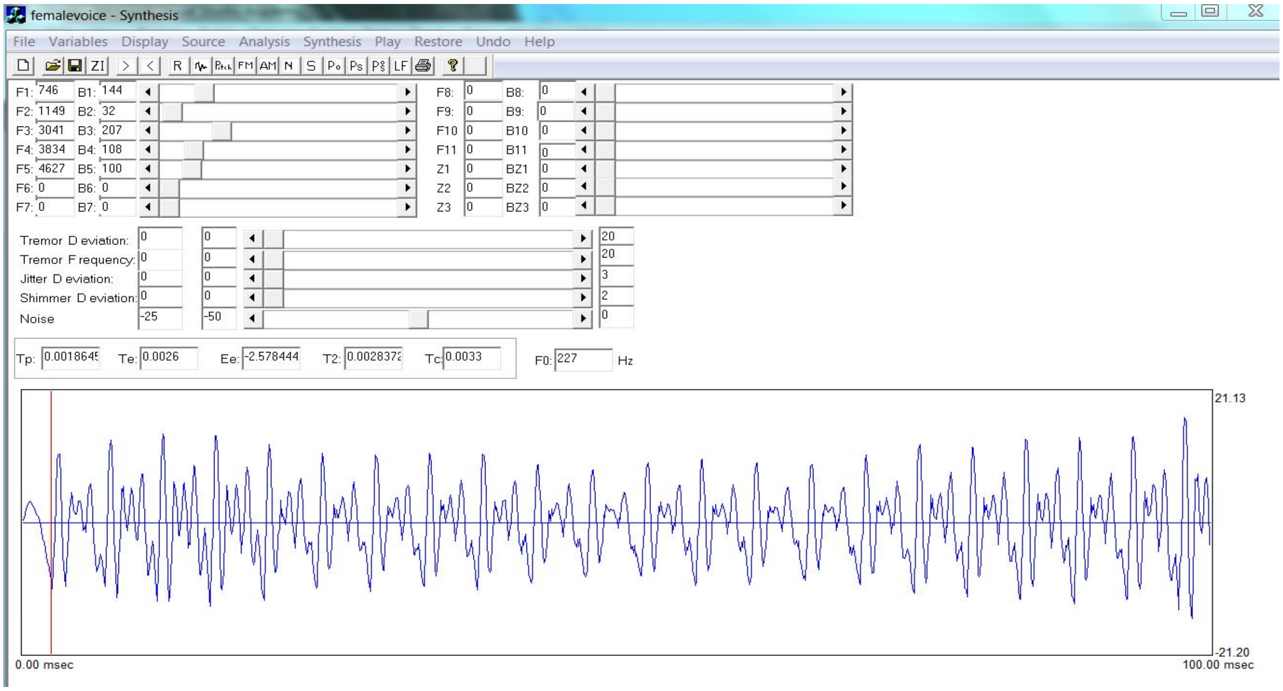


Figure 37. Modeling the harmonic voice source in the spectral domain. A: Marking a cycle. B: Calculating the source spectrum.


femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
ZI > < R Prch FM AM N S Po Ps PS LF
21.13
-21.20
0.00 msec
100.00 msec

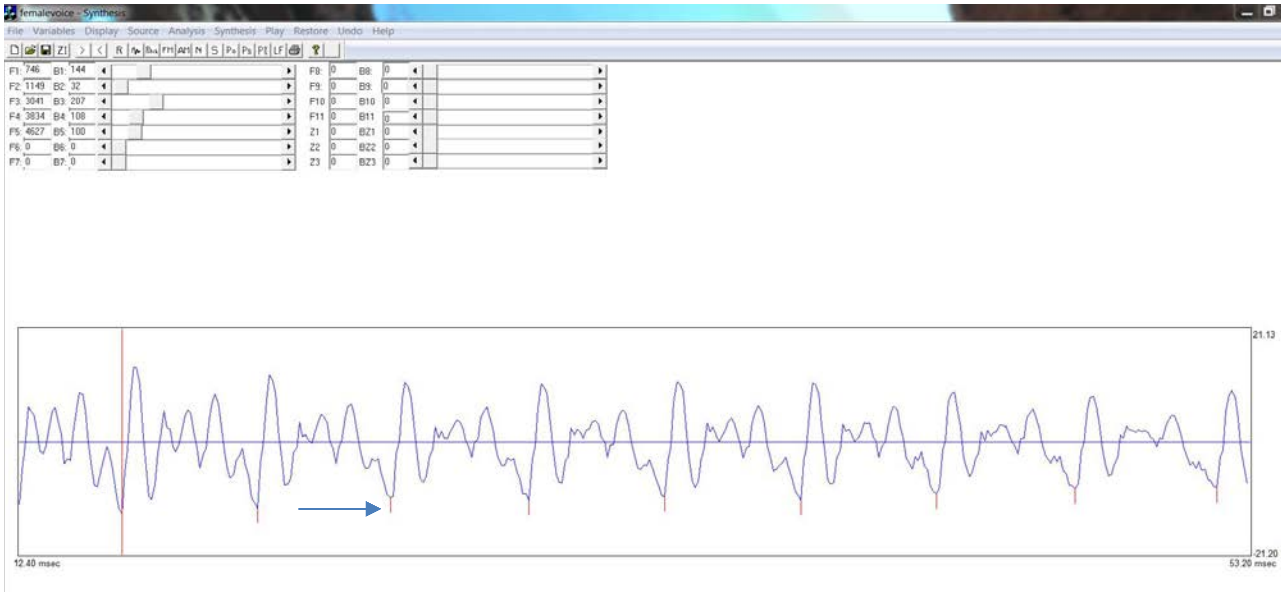
F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0

F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise -25 -50 0

Tp: 0.0018645 Te: 0.0026 Ee: -2.578444 T2: 0.0028372 Tc: 0.0033 F0: 227 Hz

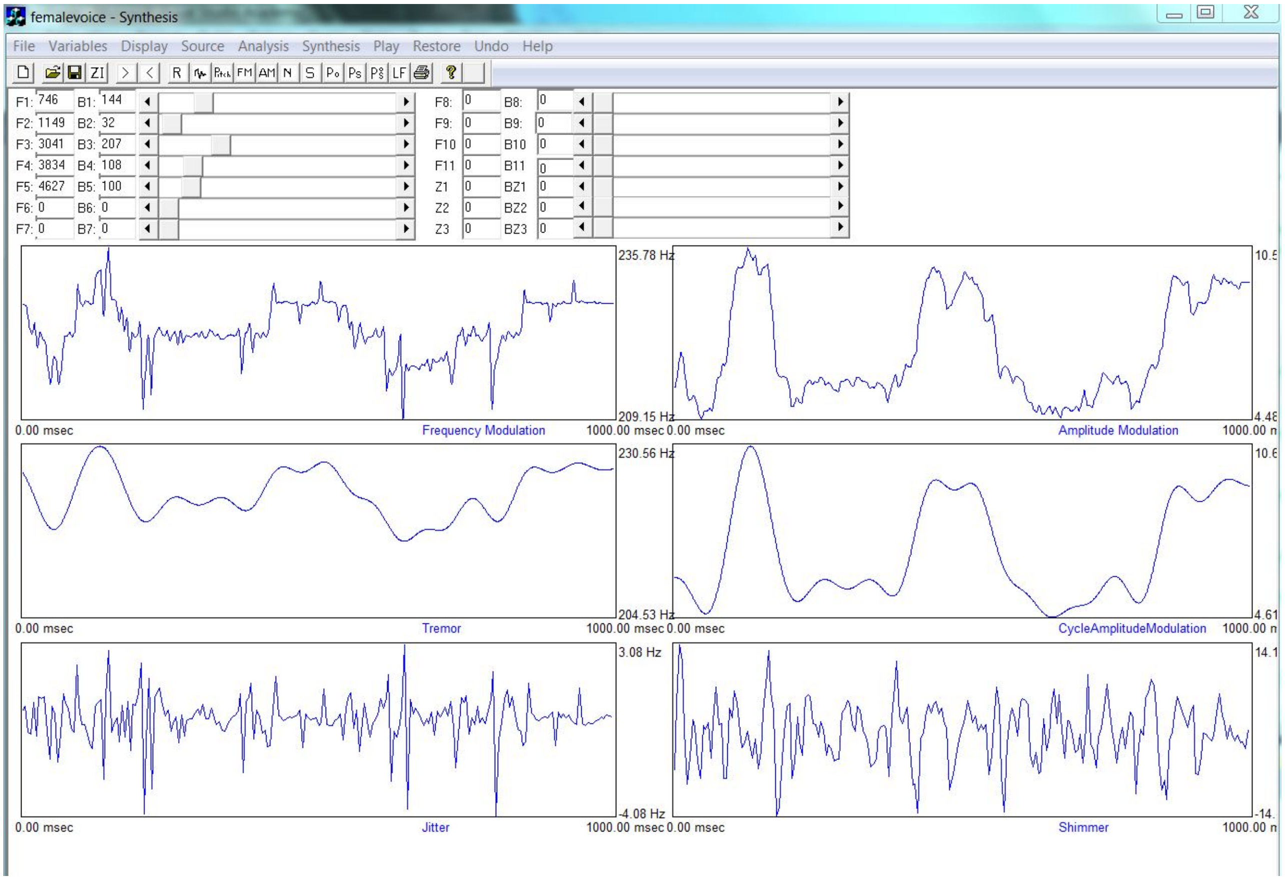
Figure 38. Newly opened waveform file.


femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
ZI > < R Prch FM AM N S Po Ps Pt LF
21.13
-21.20
12.40 msec
53.20 msec

F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0

F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Figure 39. Window showing result of pitch tracking analysis. Window has been zoomed in. Arrow indicates the tracked event.



F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0
F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Figure 40. Results of frequency and amplitude modulation analysis, shown to the right of the frequency modulation results. The top panels show plots of frequency (left) and amplitude (right) vs. time; the second show the same plots after lowpass filtering at 12 Hz, and the bottom panels show the difference.

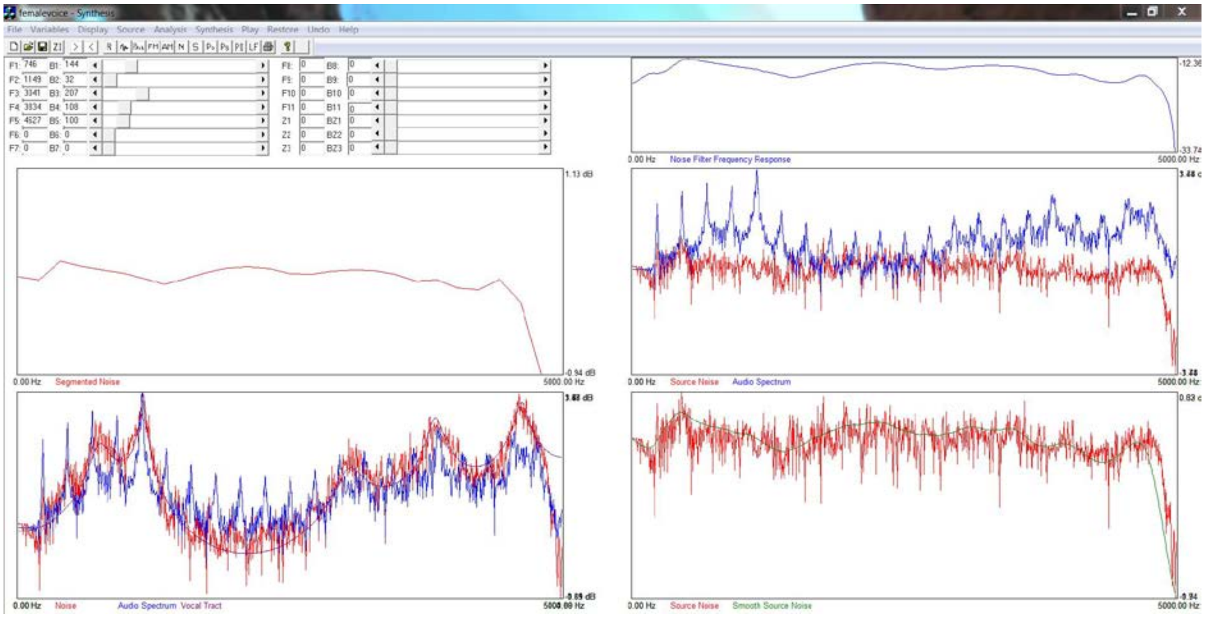
Step 6: Model the Inharmonic Part of the Source (Noise Excitation)

The final step needed to generate a preliminary version of the synthesized voice is modeling the inharmonic part of the source. To do this, click the N button on the toolbar. A new window opens, as shown in Figure 41. The top left panel shows the segmented-piecewise function (in grey), which can be manipulated by the user to change the frequency response of the noise filter, shown in the top right panel in green**.** The blue traces in the left bottom panel represent the spectrum of the natural voice. The red trace in the bottom left panel represents the spectrum after comb liftering but before inverse filtering; it shows source noise and vocal tract resonances with the current vocal tract model superimposed in purple. The red trace in the middle right panel shows the residual noise spectrum after the vocal tract is removed by inverse filtering. Finally, the bottom right trace shows the same noise spectrum with the 25-segment piecewise smoothing function superimposed (the frequency response of the noise filter)**.** This smoothing function is used as the basis for synthesizing spectrally shaped noise in the synthesizer, as described in Part 1. Note that noise analyses update automatically as the vocal tract model is updated in the synthesizer, so errors in resonance and bandwidth values are not fatal at this stage of analysis.

Step 7: Synthesize the Voice

Now that all the analyses are completed, create a synthetic voice by clicking the S button on the toolbar. A new window opens, as shown in Figure 42. In the left panel of this window, the spectrum of the original voice is plotted in red, and the spectrum of the current version of the synthetic voice is shown in blue. The vocal tract frequency response for the synthetic voice is shown in green above the spectra. The top right panel shows the current version of the synthetic LF source pulse (plotted in blue). (Note that users can control what is and is not displayed by checking or unchecking the various boxes above this display.) The current values of the LF control parameters (if an LF model has been fitted) are plotted as magenta dots on the current synthetic pulse. The spectrum of the pulse is plotted at the bottom of the figure. Finally, the segmented noise spectrum and the frequency response of the noise filter are plotted on the center of the window.

To play the original voice, click the Po button; to play the synthetic voice, click the Ps button. Clicking the Pos button plays the stimuli in the sequence [original, original, synthetic, synthetic]. This can be useful when you are making small changes to the stimuli and need to listen very carefully to hear the impact.



F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3341 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0

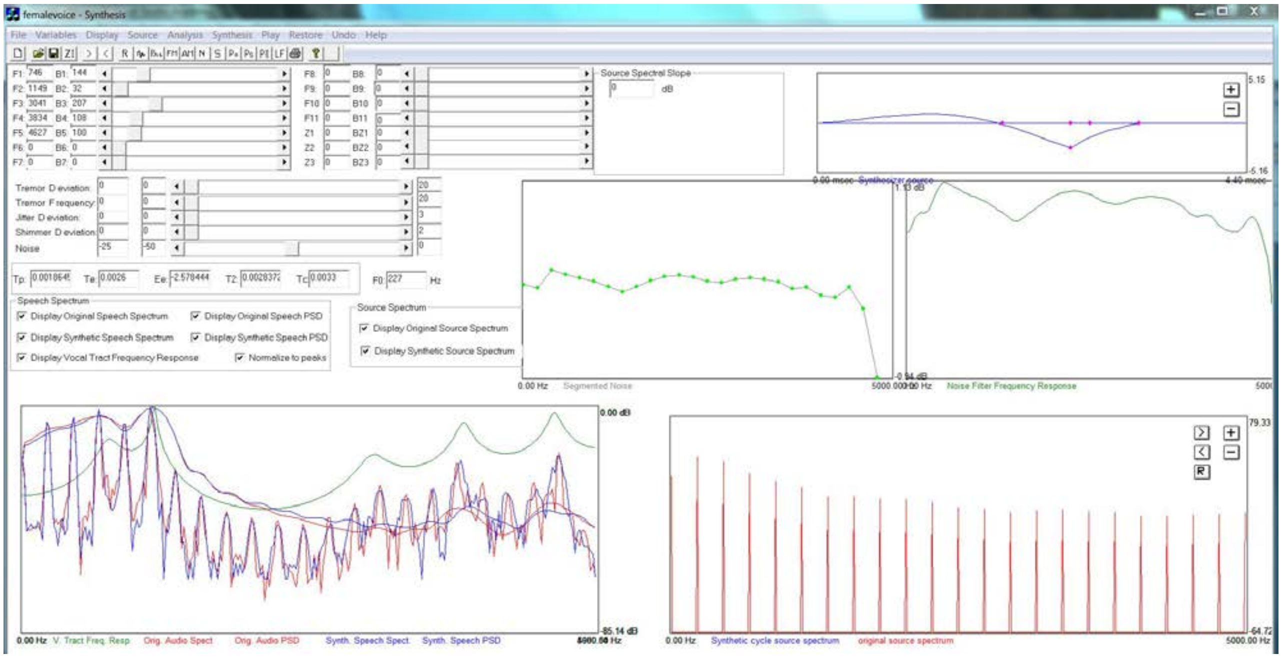
F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Figure 41. Results of noise analyses. See text for discussion of the different spectral displays

Part 3: Making Changes to the Synthetic Voices

Introductory Remarks

It sometimes happens that the synthetic stimuli match the original voices without any intervention beyond the fundamental modeling steps. In other cases, some adjustment of the synthesizer parameters is needed to improve the perceptual match between natural and synthetic tokens. In addition, you may wish to manipulate the synthesizer parameters to create a series of stimuli, a stimulus with specific characteristics, etc. The following section describes how to change the synthesis parameters once the first pass at a voice has been created. The first pass can be a voice you have generated, as described above; alternatively, you can just open one of the sample voices and then edit its properties to create a different voice.


femalevoice - Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
Z1 > < R FH AH N S Po Ps PI LF

F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0
F8: 0 B8: 0
F9: 0 B9: 0
F10: 0 B10: 0
F11: 0 B11: 0
Z1: 0 BZ1: 0
Z2: 0 BZ2: 0
Z3: 0 BZ3: 0

Source Spectral Slope: 0 dB

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise: -25 -50 0

Tp: 0.0018645 Te: 0.0026 Ee: -2.578444 T2: 0.0028372 Tc: 0.0033 F0: 227 Hz

Speech Spectrum
☑ Display Original Speech Spectrum ☑ Display Original Speech PSD
☑ Display Synthetic Speech Spectrum ☑ Display Synthetic Speech PSD
☑ Display Vocal Tract Frequency Response ☑ Normalize to peaks

Source Spectrum
☑ Display Original Source Spectrum
☑ Display Synthetic Source Spectrum

5.15
-5.16
0.00 msec Synthesizer source
4.40 msec
1.13 dB
-0.94 dB
0.00 Hz Segmented Noise
5000.00 Hz
0.00 Hz Noise Filter Frequency Response
5000
0.00 dB
-85.14 dB
0.00 Hz V. Tract Freq. Resp. Orig. Audio Spect Orig. Audio PSD Synth. Speech Spect Synth. Speech PSD
5000.00 Hz
79.33
-64.72
0.00 Hz Synthetic cycle source spectrum original source spectrum
5000.00 Hz

Figure 42. The waveform editing window. The panel at the left shows the spectra of the natural and synthetic tokens, with the current synthetic vocal tract frequency response above. The top panel to the right of the display shows the time domain flow derivative waveform. The bottom panel to the right shows its spectrum, as described in the text. The two panels at the center of the window show the segmented noise filter and the noise filter frequency response, respectively.

The different synthesizer parameters can be manipulated in any order you like, but in our experience, it's best to start by editing the vocal tract configuration to correct errors in vowel quality. Next, adjust the formant bandwidths to correct gross errors in spectral balance. (It's helpful to compare the spectrum of the natural and synthetic voices (bottom left panel) and the source spectra (bottom right panel) when making edits.) Third, edit the source functions (harmonic and inharmonic), and last of all tweak the other features (tremor, jitter and shimmer) if necessary. We will describe the procedures in this order. Note that changes to one domain may affect perception in another; for example, changes to the harmonic part of the source alter harmonic amplitudes, and thus can have surprisingly large effects on vowel quality. Similarly, changing resonance bandwidths can significantly impact perceived voice quality. For these reasons, a certain amount of back-and- forth editing may be necessary before you are satisfied with the quality of your synthesis. Again, the spectral displays can provide a useful tool for understanding what needs to be changed to improve your result. The synthetic and/or natural stimuli can be played at any point in this process by clicking the Po, Ps, and/or Pos buttons, as described above.

Editing the Vocal Tract Configuration

Play the stimuli and listen carefully to the vowel quality. Typical values of F1 and F2 for a variety of American English vowels are shown in Figure 43. The synthesizer allows you to add, delete, or move formants with the mouse. To add a resonance, point the cursor at the location you want in the spectral display. The frequency corresponding to the current cursor position appears in the lower left corner of the window. <Shift> left click to add a resonance; the default bandwidth is 100 Hz. The table of values, the spectral display, and the synthetic speech sample will all update automatically. Alternatively, type the desired formant frequency and bandwidth into the table, and click the S button on the toolbar to update the spectral display and synthetic token. To delete a formant, point the cursor at it and <Shift>right click, or change the formant and bandwidth values in the table to 0 and click the S button. All displays update automatically when resonances are edited with the mouse, but no changes are made until S is clicked when edits are made in the table. The formant frequencies may not be in correct numerical order in the table, but this does not affect the synthesis.

To change the frequency of a formant, first activate it by double left clicking, then drag it to the desired location (the frequency of the current location is shown in the lower left corner of the display) and double right click to lock it in position. Only the frequency response is continuously updated while the resonance is being moved; spectra and the synthetic sample update automatically with the final value when the right mouse button is clicked. Values can also be edited by typing the desired value into the table and then clicking S. Finally, edit bandwidths interactively by manipulating the sliding cursors (drag right to increase bandwidth of a given resonance, drag left to decrease the bandwidth). Display and synthesis update when the mouse button is released. Alternatively, type the new desired value into the box and click S.

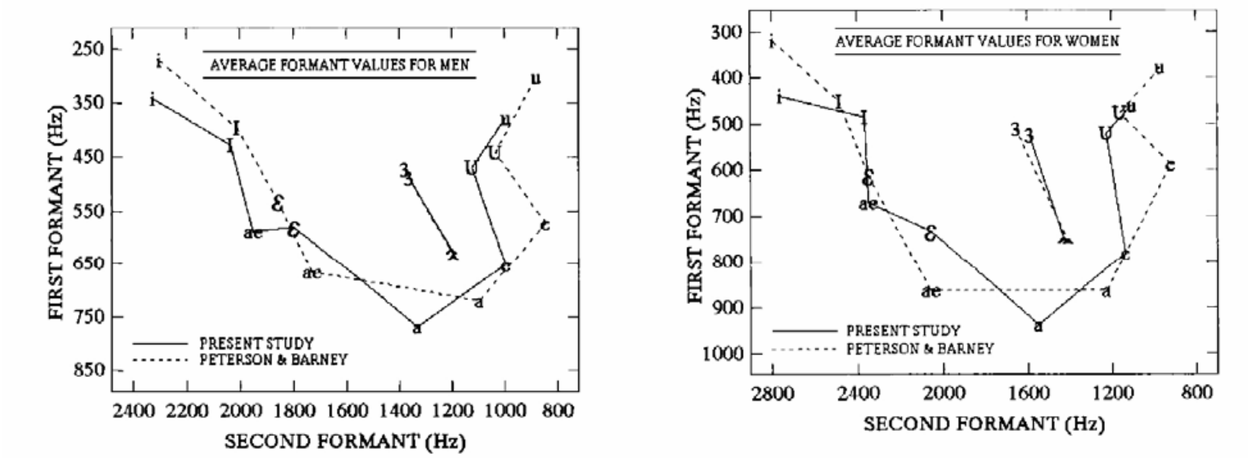


Figure 43. Average values of F1 and F2 for the vowels of male (left panel) and female (right panel) speakers of American English. From J. Hillenbrand et al., "Acoustic characteristics of American English vowels," *J. Acoust. Soc. Am.*, **97**, 3099-3111 (**1995**).

Editing the Source

The synthesizer allows the source to be manipulated in both the time and spectral domains. Time domain manipulations are accomplished by clicking and dragging the different LF parameters, which are plotted with red dots on the flow derivative pulses in the top right panel in the synthesizer window. To activate a parameter, double left click and drag it to the desired location; double right click to lock it in position (Figure 44). The source spectrum in the lower right window frame will update once the right mouse button is clicked, as will the synthesis and the spectrum of the synthetic voice in the left frame. Occasionally you may drag a point to an illegal value, in which case an error message will appear and the source windows will close. To restore the display when this happens (or just to undo your edits), use the menu command Restore-Source-Last LF source.

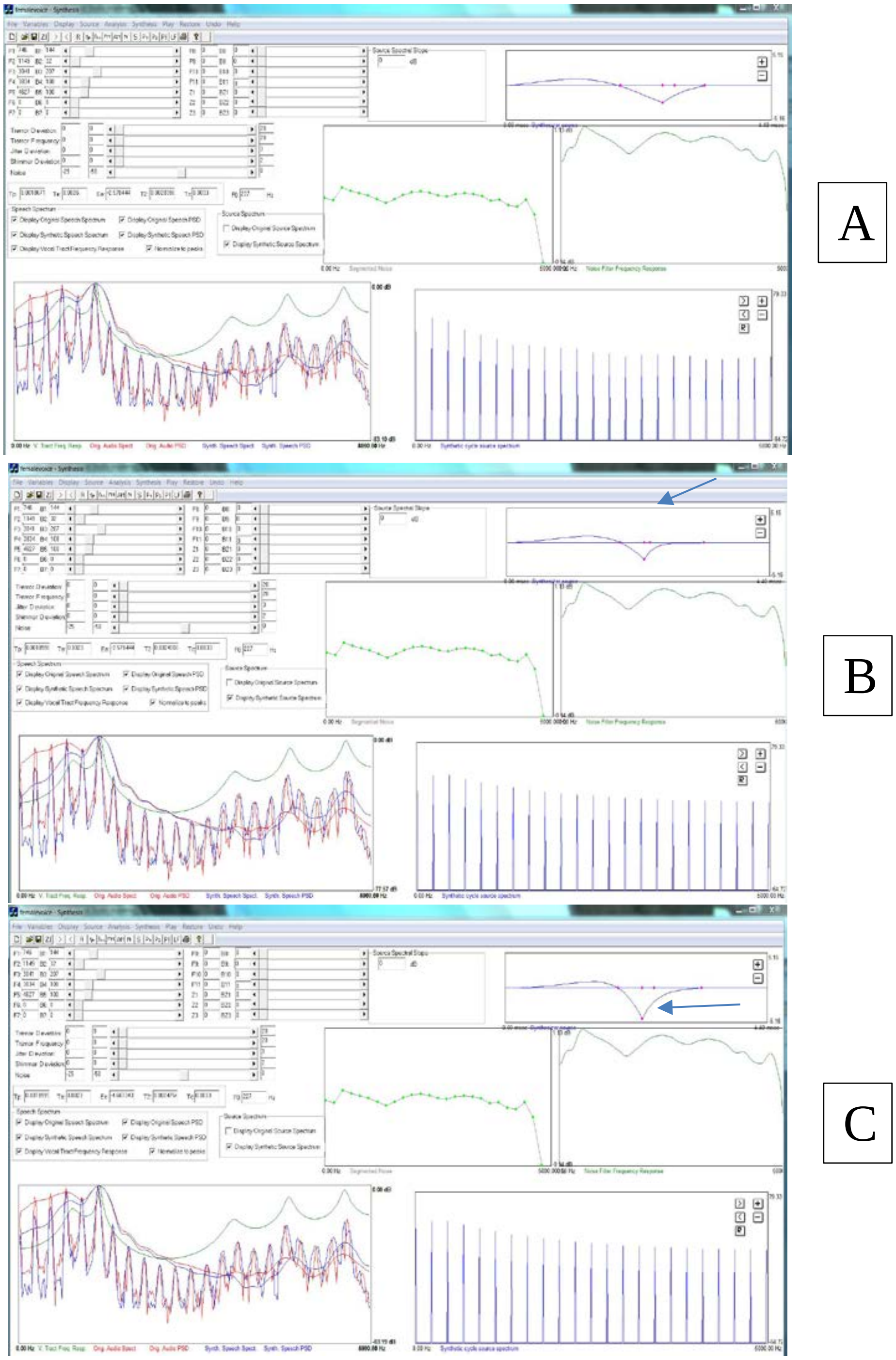


Figure 44. Changing the shape of the time domain flow derivative waveform by moving the points of the LF model. Spectral effects of time-domain changes are shown as the changes are made. A: Original waveform shape. B: Points T_e and T_2 moved to the left. C: Point E_e moved down. Changes are shown by arrows.

In the source spectral domain, users can manipulate the amplitudes of individual harmonics. They can also define a set of harmonics and change the slope of that part of the spectrum, or can increase or decrease the amplitude of those harmonics as a group while preserving the relative amplitudes within the selection. These manipulations allow users to create pulse shapes that violate the constraints on the LF model, which is particularly helpful when dealing with nonmodal or pathological phonation. However, in practical terms this also means that the LF model cannot be fit to these pulses, so that *time domain pulse manipulations must precede spectral domain modeling*. If you want to move points in both domains, finish manipulating the LF model before performing any spectral domain changes. Time domain manipulations are disabled once spectral domain changes have been made.

The synthesizer allows users to define groups of harmonics whose amplitudes can be altered to change the slope or amplitude of that part of the spectrum. To define a segment, <shift> left click to define the left endpoint, then <shift> right click to define the right endpoint. A line segment will appear connecting the two points, along with a third point plotted between the endpoints, as shown in Figure 45. You may specify as many segments as you wish, and segments may overlap or share common endpoints. Although the LF points disappear in the upper display, the time-domain display will update automatically as you manipulate the harmonic spectrum. When you are done, click the S button to update the spectral displays and synthesis.

To change the slope of a segment, double left click on the end you wish to change, and drag it up or down (Figure 46). Double right click to lock in the new location. Corresponding changes to the flow derivative pulse shape are displayed as you manipulate the spectrum, but the speech spectral display and synthesis do not update until you click the S button on the toolbar. To change the amplitude of a single harmonic, define a segment linking it to the previous or following harmonic and move the selected harmonic extreme (Figure 46b). To change the overall amplitude of a group of harmonics, double left click on the dot in the middle of the desired segment to activate it, and then drag the whole segment up or down (Figure 46c). Double right click to lock the segment in place. Display and update conventions are as before.

Note that when you are manipulating the source spectrum, changes to flow derivative pulse shape are not constrained by possible LF pulse shapes, or by any other model constraints. As a result, you can create some very unusual source pulses. Increasing the amplitude of a group of harmonics can also introduce the equivalent of a formant into the synthetic speech (Figure 47). This can be instructive and fun to play with, but be aware of what you are doing.

Segments slopes are shown in the Source Spectral Slope box at the center top of the window. To change the slope of an existing segment to a specific desired value, enter the new value in the box. Slope can be changed by moving either end of the selected segment. <Ctrl> left click the mouse on the beginning of the selected segment to increase or decrease slope by moving that end of the segment, or <ctrl> right click on the end of the segment to change slope by moving the right end.

It is sometimes desirable to smooth the harmonic source spectrum by fitting a number of segments to it and then adjusting the amplitudes of the individual harmonics so that they decrease smoothly according to the applied segments. We call this harmonic smoothing “snapping to” (Figure 48). To snap to, point the cursor at the center of the segment to be snapped, press and hold down the right mouse button, then press and release the left button while keeping the right button down.

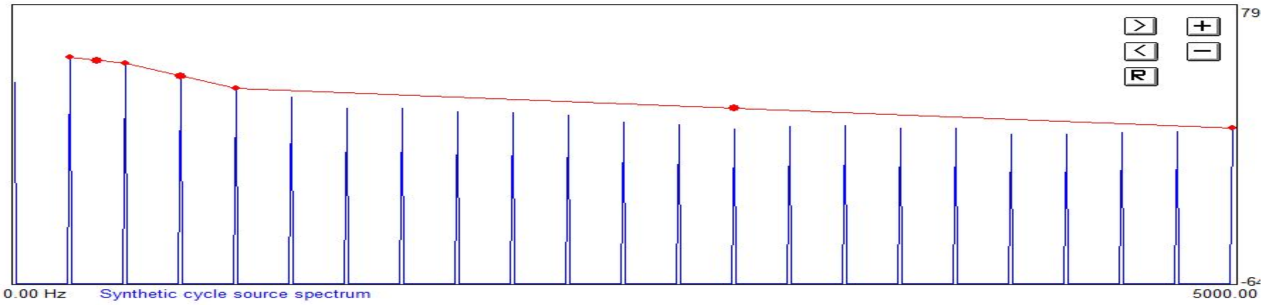


Figure 45. Defining segments on the source spectrum. The first segment connects H1 and H2; the second connects H2 and H4; and the third spans the range from H4 to the top of spectrum (5 kHz). The amplitude and/or slope of each segment can be altered by clicking and dragging, as described in the text.

Editing the Tremor Parameters

The rate and extent of frequency modulation are controlled by continuous parameters in both the sine wave and random tremor models. To change tremor extent—the maximum deviation from the baseline F0—adjust the sliding cursor labeled Tremor Deviation, or type the desired deviation above and below the mean F0 into the box next to the parameter name. To increase scale resolution or re-center the scale on a new value, type a new value for one or both endpoints into the boxes, and press enter. The scale range can be changed as often as desired.

To alter the tremor rate, use the Tremor frequency cursor, or type the desired value into the box next to the parameter name. Neither of these changes has any effect until both are given values greater than 0. To change between the random and sine wave tremor models, use the Variables-Initialize command and click the appropriate button. Also, be sure to uncheck the Use original pitch track for synthesis and Use original amplitude modulation for synthesis options in this menu unless you want the synthesized tremor added to the original pitch and amplitude contours.

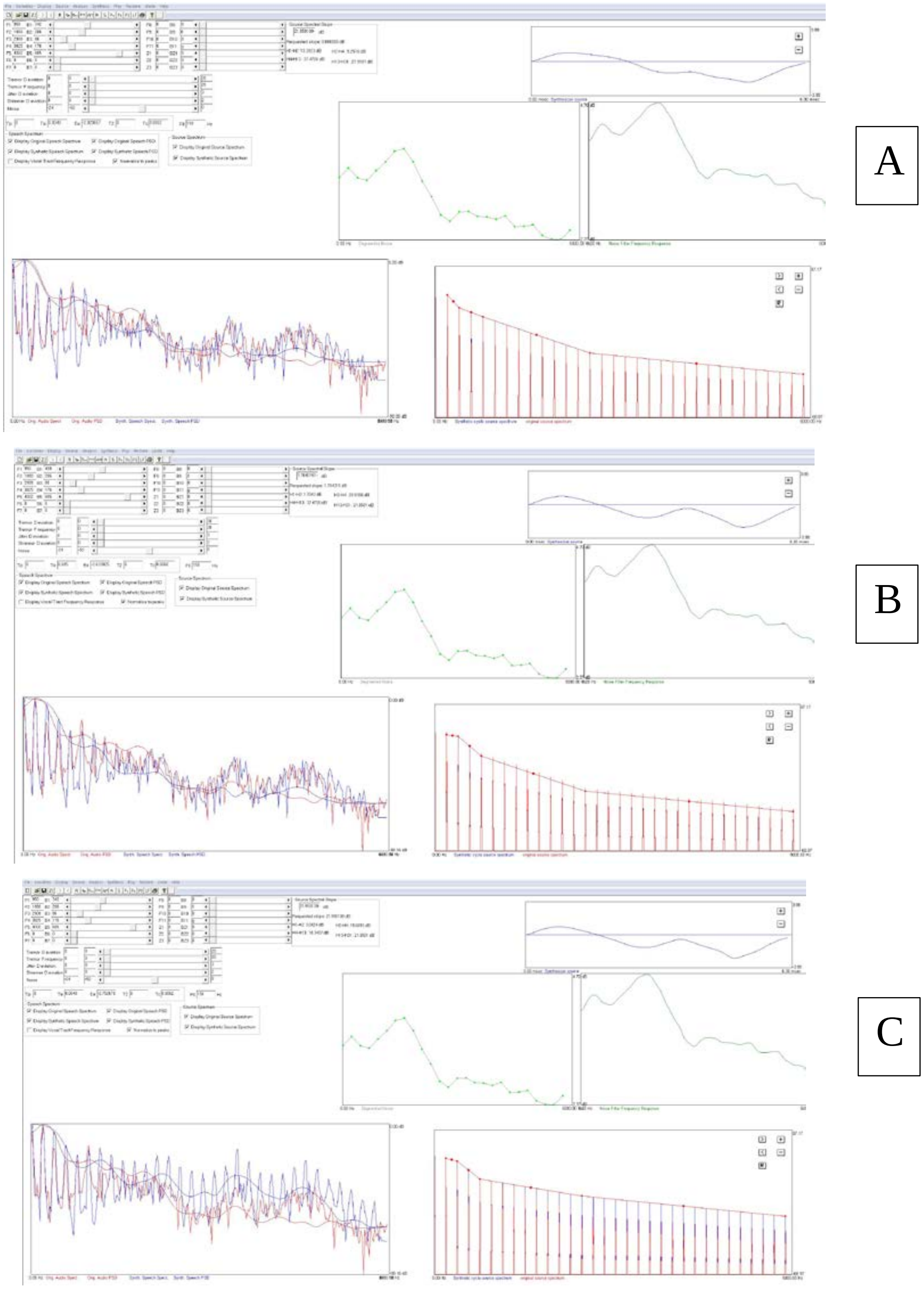
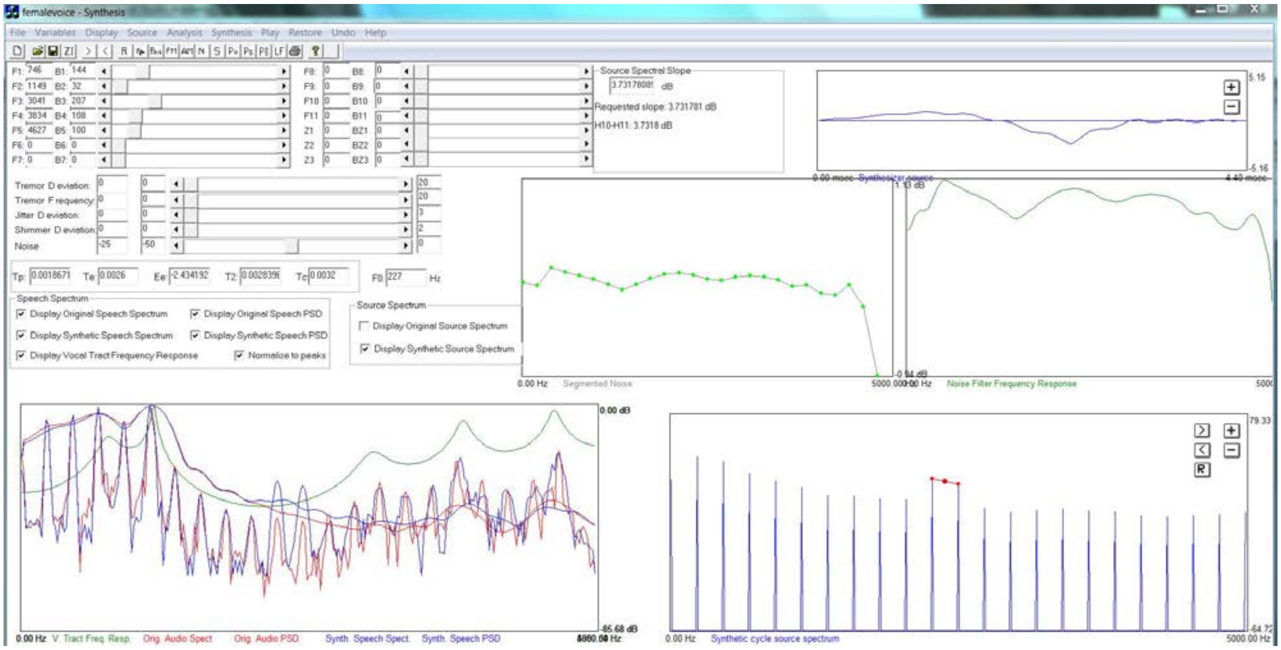


Figure 46. Spectral effects of increasing and decreasing the amplitude of individual harmonics. A: The original voice spectrum. B: The same voice after an increase in the amplitude of H2. C: The same voice after increasing the amplitude of all harmonics above 2 kHz in frequency.



femalevoice - Synthesis

File Variables Display Source Analysis Synthesis Play Restore Undo Help

F1: 746 B1: 144
F2: 1149 B2: 32
F3: 3041 B3: 207
F4: 3834 B4: 108
F5: 4627 B5: 100
F6: 0 B6: 0
F7: 0 B7: 0

F8: 0 B8: 0
F9: 0 B9: 0
F10 0 B10 0
F11 0 B11 0
Z1 0 BZ1 0
Z2 0 BZ2 0
Z3 0 BZ3 0

Source Spectral Slope
3.73178081 dB
Requested slope: 3.731781 dB
H10-H11: 3.7318 dB

Tremor Deviation: 0 0 20
Tremor Frequency: 0 0 20
Jitter Deviation: 0 0 3
Shimmer Deviation: 0 0 2
Noise -25 -50 0

Tp: 0.0018671 Te: 0.0026 Ee: -2.434192 T2: 0.0028398 Tc: 0.0032 F0: 227 Hz

Speech Spectrum
☑ Display Original Speech Spectrum ☑ Display Original Speech PSD
☑ Display Synthetic Speech Spectrum ☑ Display Synthetic Speech PSD
☑ Display Vocal Tract Frequency Response ☑ Normalize to peaks

Source Spectrum
☐ Display Original Source Spectrum
☑ Display Synthetic Source Spectrum

Figure 47. Increasing the amplitude of bands of harmonics adds the equivalent of an unwanted vocal tract resonance (in this case, at 2276 Hz) to the synthetic voice signal. Notice "formant" ripple in the flow derivative waveform in the top right panel of the figure.

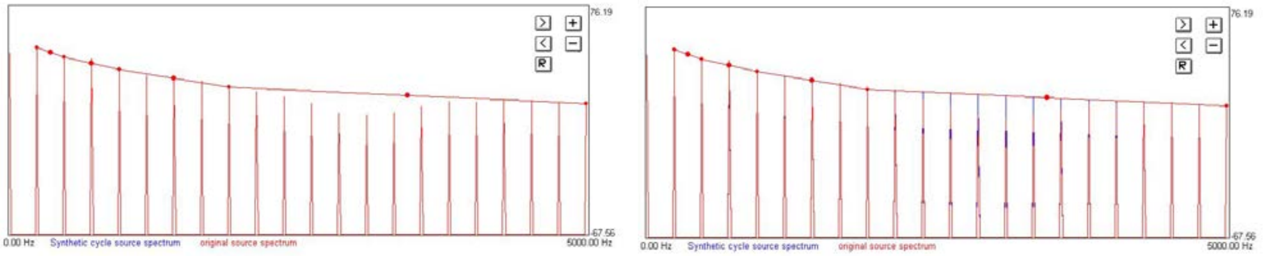
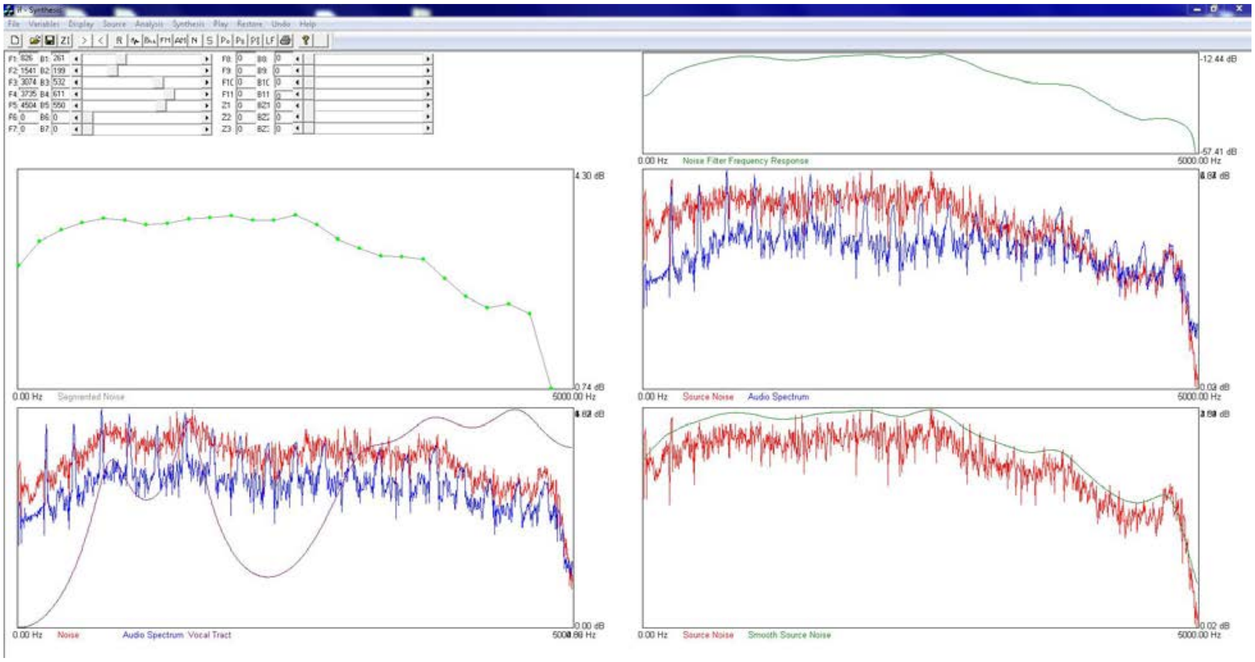


Figure 48. "Snapping to" the harmonic source spectrum to fit it with a smoothed spectral model. A: A source spectrum before snapping to. B: The same spectrum after snapping to. Notice the change in harmonic amplitudes, especially in the higher frequencies.

Adjusting the Noise Spectrum and Levels of Jitter, Shimmer, and Noise

The synthesizer allows users to smooth the noise spectrum with any desired level of detail. The analysis window is shown in Figure 49. As described previously, the topmost plot on the left shows the segmented- piecewise function (in grey), which can be manipulated by the user to change the frequency response of the noise filter, shown in the top right panel in green**.** The blue traces in the left bottom panel represent the spectrum of the natural voice. The red trace in the bottommost left panel represents the spectrum after comb liftering but before inverse filtering; it shows source noise and vocal tract resonances with the current vocal tract model superimposed in purple. The red trace in the middle right panel shows the residual noise spectrum after the vocal tract is removed by inverse filtering. Finally, the bottom right trace shows the same noise spectrum with the 25-segment piecewise smoothing function superimposed (the frequency response of the noise filter). When the default settings are used, the noise spectrum is estimated as described above, and a filter is created by fitting a 25-segment piecewise approximation to the estimated spectrum. All segments are of equal length in this case; the number of equally-spaced segments can be changed by editing the appropriate value in the Initialize-Variables menu, as described above.

Alternatively, users can specify any desired number of segments of any size as follows. To create a new filter shape for the noise, click the N button to open the noise analysis window. Now point the cursor just to the right of the beginning point for the desired segment and <shift> left click; set the endpoint for this segment by pointing the cursor slightly to the right of the desired endpoint and <shift> right clicking. A red line segment will appear connecting the two points, smoothing the spectrum between them (Figure 50a, b). Users can establish as many segments as desired, of any size.



Synthesis
File Variables Display Source Analysis Synthesis Play Restore Undo Help
ZI > < R B FH AH N S P P P LF
-12.44 dB
-57.41 dB
0.00 Hz Noise Filter Frequency Response 5000.00 Hz
4.30 dB
0.74 dB
0.00 Hz Segmented Noise 5000.00 Hz
8.84 dB
0.03 dB
0.00 Hz Source Noise Audio Spectrum 5000.00 Hz
4.82 dB
0.00 dB
0.00 Hz Noise Audio Spectrum Vocal Tract 5000.00 Hz
3.80 dB
0.02 dB
0.00 Hz Source Noise Smooth Source Noise 5000.00 Hz

F1: 826 B1: 261
F2: 1541 B2: 199
F3: 3074 B3: 532
F4: 3735 B4: 611
F5: 4504 B5: 550
F6: 0 B6: 0
F7: 0 B7: 0
F8: 0 B8: 0
F9: 0 B9: 0
F10: 0 B10: 0
F11: 0 B11: 0
Z1: 0 BZ1: 0
Z2: 0 BZ2: 0
Z3: 0 BZ3: 0

Figure 49. The noise analysis window. See text for description of the different plots.

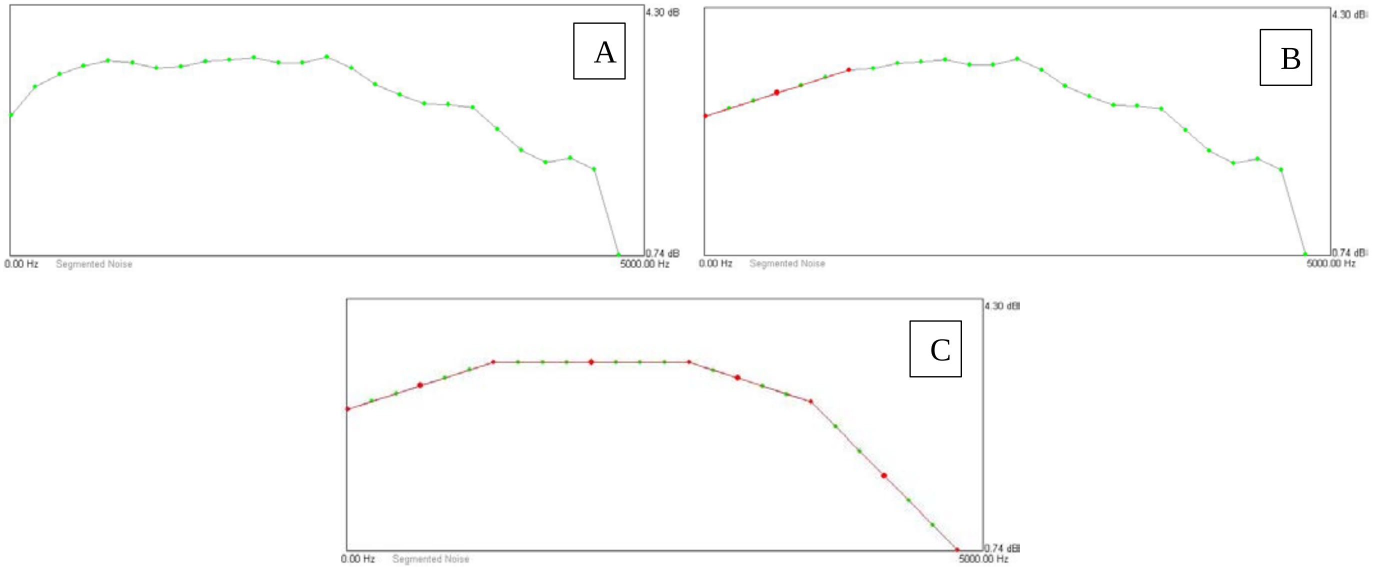


Figure 50. A: The noise filter smoothed by a 25-piece approximation. B: The same filter after the first 6 pieces have been linearized. C: The same filter fit with a 4-piece approximation.

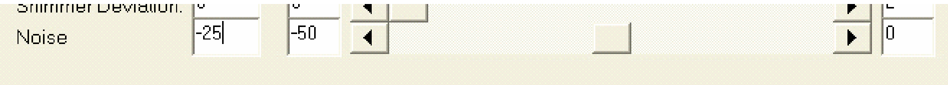
To change spectral shape by altering the vertical position of a given point, select a vertex to move and <ctrl> <shift> left click to activate it. Continue to hold down the <ctrl> and <shift> keys, point the cursor at the desired location and right click. The segment endpoint will move to the new location. To drag a point to a new location, double left click to activate the desired vertex, then drag it to its new location and double right click to lock it in position. Note that relocating any point will cause the slope of any adjoining spectral segments to also change, so that no discontinuities occur in the window (Figure 50 c). The noise spectrum is recalculated whenever the user makes a change, using the newly-specified spectral envelop to filter white noise as described above. Changes can be seen in the filter frequency response plot, at the top right in Figure 49. This plot may rescale as a result of your manipulations, making it look as if you have done something entirely different than you intended (you haven't). To remove the last change go to the Undo menu, and select Noise frequency response.

The number and frequency location of the points that are available to select as part of these operations are determined by the number of segments specified in the Variables-Initialize menu. The default is 25, which is plenty in our experience, but increasing this number will increase resolution if desired. It is also possible to zoom the display by setting cursors in the segmented source filter response (<ctrl> left and <ctrl> right click to set the left and right cursors, respectively) and then selecting Segmented source noise filter frequency response – Zoom in from the Display menu. To remove the cursors, click Display-Segmented source noise frequency response – Delete cursors.

Once you are happy with the noise filter's shape, be sure to File-Save segmented noise, or your work will not be saved. Then click S to synthesize the voice using the new noise spectrum. If more work is needed, go to the Analysis menu, select Noise-Back to last recomputed noise, to restore the Noise window with the last modifications. Clicking the N key will restore the original 25-piece noise spectrum (as will the command Analysis–Noise–Compute noise). If you do this by accident, you can restore the segmented noise by using the command File-Read segmented noise *provided you have saved it*.

Once you have finished smoothing the noise spectrum as desired, the overall noise-to-signal ratio (NSR) can be manipulated with the sliding cursor shown in Figure 51. (The ability to manipulate the NSR in individual bands should be available early in 2017.) The boxes nearest the sliding cursor show the range of the scale, which is initially set to -50 dB (essentially noise-free) to 0 dB (extremely noisy), as shown in the figure. To increase scale resolution or re-center the scale on a new value, type a new value for one or both endpoints into the boxes. The box to the left of the figure shows the current NSR value, which is initialized to -25 dB. To change this value, click and drag the sliding cursor labeled Noise. Dragging the cursor to the right increases the amount of noise present; dragging it left decreases the NSR. The synthetic voice and spectral displays are automatically updated when changes to the NSR are made with the cursor. Alternatively, type the desired NSR value into the leftmost box and press enter. The cursor will move to indicate the new value, but in this case the spectrum and synthetic voice do not update until you press the S button to resynthesize. The synthetic and/or natural stimuli can be played at any point in this process by clicking the Po, Ps, and/or Pos buttons, as described above.

As described in Part 1, jitter and shimmer are normally modeled as part of the overall noise component of the voice, rather than as individual dimensions (Kreiman & Gerratt, 2005). However, each can be added separately to the voice if desired. To add jitter and/or shimmer to the synthetic voice, move the sliding cursors (located above the noise cursor and below the tremor cursors) to the desired level, or type in the precise level of jitter (in percent) or shimmer (in dB) that you want. The synthetic voice and spectral display are automatically updated when the cursors are used; when a value is typed into the box, you must use the S button to resynthesize before any changes take place. As for noise adjustments, the display endpoints may be reset to change scale resolution by simply typing in new values in the boxes nearest the cursors. It is also possible to model jitter and shimmer by increasing the amplitude of the noise spectrum in the lower frequencies. This provides more precise control and a generally better result, in our experience.



Shimmer Deviation:

Noise -25 -50 0

Figure 51. Sliding cursor for adjusting the noise-to-signal ratio (NSR) in the synthesizer. The first box shows the NSR value (set at the default of -25 dB in the figure), which can be changed by dragging the cursor or by typing a new value into the box. The second and third boxes show the current scale limits. Scale resolution can be increased or decreased by editing these values, as described in the text.

Saving Your Work and Creating Stimuli

The Save Menu is shown in Figure 52. The command File-Save all opens a box asking users if they want to create a new case. If yes, the program creates a new directory with the new name supplied by the user where it saves the files, together with the files filename.lv, filename.par, and filename.s. If no, it will save the files to the current directory and overwrite any previous version. The Save all command saves the current slider settings, formant and bandwidth values, and all the other variables into two files, a binary file filename.var and an ASCII file filename.vtxt. It also saves these additional files in ASCII format: a single pulse of the current version of the flow derivative vocal source into file filename.sou; a 1 sec synthetic voice sample filename.syn; synthetic jitter (filename.sjitt), synthetic tremor (filename.strem), and synthetic shimmer (filename.shimm) if they are available; original tremor in filename.otrem, tremor index file in filename.otremix, noise in filename.noise, segmented noise in filename.segnoise, amplitude modulation in filename.am, and the necessary files for importing pitch tracks: filename.imi, filename.ami, and filename.ape. The command Save variables saves the slider settings, formant and bandwidth information, and all the other variables in filename.var, and also creates an ASCII file filename.vtxt containing the same information.

The command File-Save synthetic speech creates a new file named filename.syn, which contains the 1-sec synthetic voice sample in ASCII format to the same location. To create series of stimuli, change the name of this file as necessary prior to saving.

The command File-Save source-LF source saves the LF source in a file filename.LfSou, which is written to the directory for this case. This file contains a single cycle of the current flow derivative source pulse, in ASCII format. File-Save source- others saves a single pulse of a non-LF voice source in ASCII format to the file filename.sou in the directory for the current case.

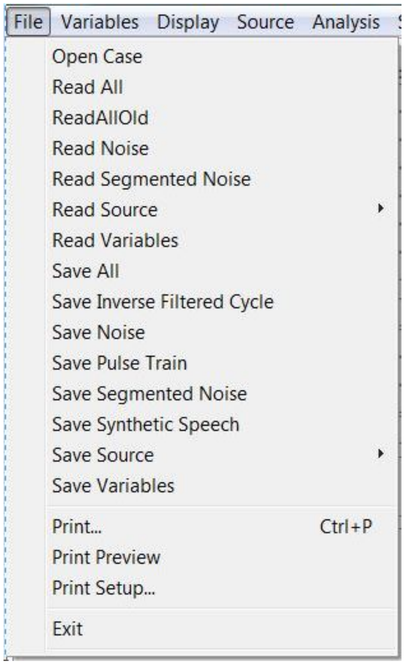


Figure 52. The Save menu.

Part 4: Synthesizer Variables

Table 1. Complete list of synthesizer variables

LF parameters	The set of LF parameters: E_e, T_e, T_2, T_c, T_p
F0	Fundamental frequency
Number of Formants	Number of poles in the vocal tract model
Formants Frequencies	Central frequencies of the poles in the vocal tract model

Formants Bandwidths	Bandwidths of the poles in the vocal tract model
Number of Zeros	Number of zeros in the vocal tract model
Zeros Frequencies	Central frequencies of the zeros in the vocal tract model
Zeros Bandwidths	Bandwidths of the zeros in the vocal tract model
Sampling Rate	Synthetic signal sampling rate, which equals the sampling rate of input waveform filename.lv
Use Original Tremor	Use original tremor computed from a pitch tracker analysis of the original waveform filename.lv
Random Tremor	Use synthetic random tremor model
Sinusoidal Tremor	Use synthetic sinusoidal tremor model
Synthetic Tremor Deviation	Value of the synthetic tremor frequency deviation from F0 selected with the slider
Synthetic Tremor Frequency	Value of the synthetic tremor cut off frequency selected with the slider
FM CutOff	Cutoff frequency for the low pass filter used to compute jitter in the original audio signal
Synthetic Jitter Deviation	Value of the synthetic jitter deviation selected with the slider
Add Synthetic Jitter	Add synthetic jitter as opposed to the jitter calculated from the audio waveform
Synthetic FM Cutoff	Value of the high pass filter cutoff frequency for the synthetic jitter computation, as a percent of F0
AM Cutoff	Cutoff frequency for the low pass filter used to separate shimmer from amplitude modulation in the original audio signal
Synthetic Shimmer Deviation	Value of the synthetic shimmer deviation selected with the slider
Add Synthetic Shimmer	Add synthetic shimmer as opposed to the shimmer calculated from the audio signal
Use Original Amplitude Modulation	As opposed to a list of amplitude modulation values
Synthetic Data Length in seconds	Length of the input audio waveform, which determines the length of the synthetic waveform
Output Synthetic Data Length in seconds	Output synthetic data length
Upsample Factor	The source is upsampled by this factor and concatenated; the program adds tremor, jitter, and shimmer, and builds

	the pulse train
Downsample individual cycles	The resulting pulse train is downsampled to the original sampling rate, either each of the individual cycles, then concatenated, or the entire pulse train
Synthetic Noise	Value of the synthetic noise selected with the slider. This affects the gain of the noise applied to the synthetic signal.
Noise Filter Approximation Model Type	Linear or piecewise
Noise Filter Approximation Number of Segments	Number of segments for a piecewise noise filter model
Noise Correction Approximate in Hz	Correction to the low frequency noise filter frequency response
Recompute Noise	If checked, program recomputes noise after modifications to the vocal tract.
Apply Noise	Apply noise to the synthetic speech, or not.
High Pass Lifter	A high pass filter in the cepstral domain, not currently used in noise analyses
CutOff Quefrency	Cutoff frequency for the unused high pass lifter
LPC order	By default the LPC order for a sampling rate of 10kHz should be 14.
PSD Window length	Power Spectrum Density window length
PSD Overlap Segments	Overlapped segments as opposed to continuous segments for the computation of the PSD
PSD Number of Segments	Number of segments to compute the PSD
PSD Preemphasis	Apply preemphasis to the signal prior to PSD analysis.
Mark IFFT Cycles Type	After spectral manipulation of the source mark IFFT cycles at the negative peaks, after running a pitch tracker or using the period length, to select the best source.
DFT Number of cycles	Number of cycles to concatenate to compute the DFT for spectral source manipulation
Demean Speech previous to FFT computation	Demean the audio signal before Fourier analysis
Inverse Filtered Source Smoothing window length	Determines the extent by which the selected inverse filtered source needs to be smoothed before use as an initial source model

Part 5: Menu Commands and Other Features of the Synthesizer

This section lists all the menu commands currently available in the synthesizer and documents the functions of those that are not described elsewhere in this manual.

File Menu

Open case

Opens a dialog box into which users enter the name and file format for the case under study.

Read all

Used to restore a session that was stored with File-Save all, as explained above.

Read all old

Restores a session that was stored in an old format.

Read noise

Reads only the noise component that is to be added to the source.

Read segmented noise

Reads the segmented noise. It can be manipulated by the user to create a new frequency response for the noise filter.

Read source

LF Fitted

Parameters: Reads the LF model parameters Tp, Te, Ee, T2, Tc, and the cycle length from the saved ASCII file filename.LfSouPar in the directory for this case. This file is created by saving the source with the command File-Save source-LF fitted, as described above.

Cycles: Reads a previously-saved LF source pulse from the ASCII file filename.LfSou in the directory for this case, and then computes the best- fitting set of LF parameters for that pulse. File is created using the command File-Save source-LF fitted, as described above.

Others

Reads an ASCII file representing a source with the same sampling rate as the inverse filtered waveform.

Read variables

Used to restore the saved values for all the variables from previous analyses.

Save all

Saves the current session as described above.

Save inverse filtered cycle

Saves the inverse filtered cycle as it came from the inverse filter program in Filename.invf. This cycle is used for the LF approximation.

Save noise

Saves the noise time series to be added to the synthetic speech in Filename.noise

Save pulse train

Saves the concatenated source pulses after jitter, shimmer, and tremor have been added, but before applying the vocal tract filter, in Filename.pulses.

Save segmented noise

Saves the segmented noise filter in Filename.segnoise.

Save synthetic speech

This command creates a new file named filename.syn, which contains the 1-sec synthetic voice sample in ASCII format and then saves it to the directory for this case. If you are creating multiple versions of the same basic voice, change the name as necessary prior to saving.

Save source

LF fitted

Saves the file filename.LfSou to the directory for this case. This file contains a single cycle of the current flow derivative source pulse, in ASCII format. Also saves the file Filename.LfSouPer, which contains a list of the current LF parameter values.

Others

Saves a single pulse of a non-LF voice source in ASCII format to the file filename.sou in the directory for the current case.

Save variables

Saves the new formant and bandwidth values into an ASCII file, plus all the synthesizer variables The default name for this file is filename.par, and the default save location is C:\synthesis\work\filename. Change these if you want to avoid overwriting the original .par file.

Print
Print preview
Setup
Exit
[list of recently used files]

Variables Menu

Initialize

Allows users to set initial values for variables controlling analyses spectral noise, frequency and amplitude modulation, and all the other variables used in the synthesis process.

Display Menu

Inverse filtered cycles

Move zero line

Changes the location of the zero line in the display of the glottal flow derivative pulses. Set the left cursor (by single left clicking) at the desired location, then select this command to move the zero line.

Erase cursors

Erases the cursors from the display of the flow derivative waveform.

Source spectrum

Logarithmic display

The default display mode. Selecting this command displays the source spectrum and the spectrum of the voice sample on a logarithmic scale.

Linear display

Switches the source spectral displays from a logarithmic to a linear scale.

Delete Segment

Deletes segments that have been created for manipulating spectral slopes, in the reverse order that they were set (deletes most recent segment first, and first segment last.)

Vertical zoom in

Vertical magnification of the synthetic source.

Vertical zoom out

Vertical shrinking of the synthetic source

Audio

Display

Same effect as clicking the Display audio waveform button on the toolbar. Displays the time domain voice waveform.

Hide

Hides the waveform display.

Zoom in

Same effect as clicking the ZI button on the toolbar. Zooms the window to the cursors. Set left and right cursors by left and right single clicking.

Page right

Same effect as the > button on the toolbar. Scrolls the waveform display one page to the right.

Page left

Same effect as the < button on the toolbar. Scrolls the waveform display one page to the left.

Synthesizer source

Vertical zoom in

Vertical magnification of the synthetic source.

Vertical zoom out

Vertical shrinking of the synthetic source

Segmented source noise filter frequency response

Zoom in

Zoom in between cursors.

Restore

Restore the display to its original.

Erase cursor

Erase cursors.

Delete segment

Delete the last segment created.

Noise Window

Bring back the noise window with the current noise filter frequency response, which has a larger display of the filter.

Source Menu

Inverse filter

Mark cycle

The program automatically marks the cycle to smooth.

Compute source

The program applies a moving average to the selected cycle. The user can select the smoothing window size in points in the menu Variables-Initialize-Inverse Filtered Source.

LF

Find LF features automatically

Finds major features of the flow derivative waveform for use during LF model fitting. Plots these features with small magenta points on the flow derivative waveform in the synthesizer window.

Find features with user interaction

The user marks beginning and end of a cycle on the flow derivative waveform, the program finds the major features for LF model approximation and plots them as magenta dots.

Fit an LF model

Fits an LF model to the flow derivative waveform using the features that were marked using the previous command.

Optimization Parameters

Allows user to optionally increase weight given to points Ee and Tc during LF model fitting. Increase the values of these variables to increase the precision with which the model fits these precise points.

Analysis Menu

Pitch

Track

Tracks F0 across the complete 1 sec natural voice sample. To use, first select an event (a positive or negative peak) that repeats relatively reliably across the entire waveform. Mark this peak by left single clicking to set the left cursor. Next, issue the command. The program places a small red mark at each recurrence of the event it finds. Page through the file to verify marking if desired.

Import tracks

Allows the user to import an F0 track from the Sky analysis program (which allows interactive F0 tracking) or another source.

Import ASCII F0 file

Read a list of F0 values arranged in a column from a text file.

F0 modulation

Same function as the FM button on the toolbar. Computes a smoothed pitch track, with the degree of smoothing specified using the FM cutoff value in the Variables-Initialize window. The pitch tracker must be run before this command will work.

Amplitude Modulation

Compute: Same function as the AM button on the toolbar. Computes a smoothed amplitude track, with the degree of smoothing specified using the AM cutoff value in the Variables- Initialize window. The pitch tracker must be run before this command will work.

Import from ASCII file: Reads a user-supplied list of amplitude values. Values must be in a column with a sampling rate equal to the original speech sampling rate.

Noise

Compute Noise

Same effect as the N button on the toolbar. Computes the spectrum of the aspiration noise present in the voice and synthesizes a filter that models the noise spectrum. The degree of precision with which the spectral shape is modeled is controlled using the Variables-Initialize command and the Piecewise filter model/number of pieces parameter (default is 25 pieces).

Synthesis Menu

Synthesis

Same effect as the S button on the toolbar. (Re-)synthesizes the voice using the current set of parameters.

Compute LF source

Compute the LF approximation with the parameters listed on the screen.

Play Menu

Play original

Same effect as the Po button on the toolbar. Plays the natural (original) voice sample one time.

Play synthetic

Same effect as the Ps button on the toolbar. Plays the current synthetic voice sample one time.

Play sequence

Same effect as the Pos button on the toolbar. Plays the natural voice token two times, followed by two repetitions of the synthetic voice token.

Restore Menu

Audio signal

Restores the audio display to its original size after zooming the window. This only works when the audio signal has been displayed.

Source

Original LF source

It restores the original LF fit source parameters and recomputes the original LF model from them. This command undoes all edits (time domain or spectral domain) to the synthetic voice source. Nothing is saved.

Last LF source

It restores the last computed LF source.

Original other source

It restores the last source; it could be an LF fitted source or any other source, by example a source whose spectrum was manipulated.

Vocal tract

Undoes all edits to the vocal tract model (formants and bandwidths) and restores the configuration to the initial startup values for this case. Nothing is saved.

UnDo Menu

This commands only works when issued from the synthesis display window.

Vocal tract

Undo last change to the vocal tract, and synthesize.

LF parameters

Undo the last change to the LF parameters, recompute the source, and synthesize.

Source

Undo the last change to the source.

Shimmer deviation

Undo the last change to shimmer deviation, and synthesize.

Jitter deviation

Undo the last change to jitter deviation, and synthesize.

Noise

Undo the last change to the NSR, and synthesize.

Tremor deviation

Undo the last change to tremor deviation, and synthesize.

Tremor frequency

Undo the last change to tremor frequency, and synthesize.

Last change

Undo the last change, whatever in can be, and synthesize.

Noise frequency response

Undo the last segment movement. This command can be applied only once.

Help Menu

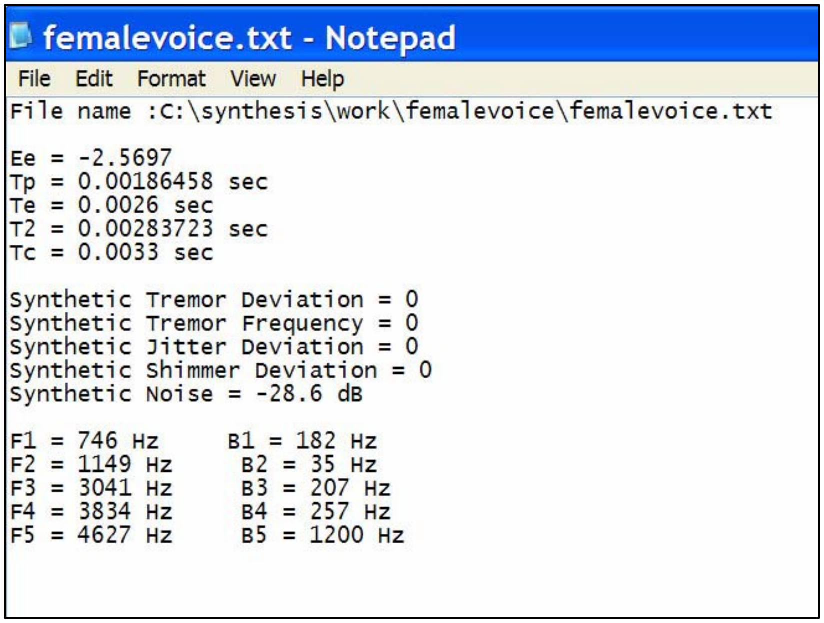
Not currently helpful.

Part 6: Index of File Names and What They Mean

This section lists the origin and purpose of all the various files that may be created during the course of voice synthesis efforts. Files should be found in the directory for the case under consideration. Not every file will be created for every case.

File name	Definition
Filename.am	An ASCII file listing the evenly-spaced amplitude values.
Filename.ami	An ASCII file listing the location of each marked cycle boundary in points (one location/line). Created by the SKY pitch tracking algorithm. You will need to copy this file into the directory for this case in order to import a pitch track.
Filename.ape	An ASCII file listing the parabolically-interpolated period lengths for the case. Created by the SKY pitch tracking algorithm. You will need to copy this file into the directory for this case in order to import a pitch track.
Filename.imi	An ASCII file listing the location of the cycle boundaries after parabolic interpolation. Created by the SKY pitch tracking algorithm. You will need to copy this file into the directory for this case in order to import a pitch track.
Filename.invf	An ASCII file containing the inverse filtered cycle that was used as a model for the LF approximation.
Filename.LfSou	An ASCII file containing the y coordinate values for a single LF source pulse. X coordinate values are determined by the sampling rate. The file is created by the synthesizer when the source is saved with the command File-Save source-LF fitted.
Filename.LfSouPar	An ASCII file containing the LF model parameters Tp, Te, Ee, T2, Tc, and the cycle length. This file is created by the synthesizer when the source is saved with the command File-Save source-LF fitted.
Filename.lv	A one sec sample of the original natural voice centered around the cycle that was inverse filtered. This file is played whenever you select Play original or click the Po button. The file is created by the inverse filter and is in ASCII format.
Filename.noise	An ASCII file listing the time series of the source noise component.
Filename.otrem	An ASCII file listing the original tremor.
Filename.otremix	An ASCII file listing the original tremor index.
Filename.par	An ASCII file listing F0 on the first line, the sampling rate on the second line, and the frequencies of each formant and its associated bandwidth on the third and subsequent lines. This file is created by the inverse filter, and values are those that were used in inverse filtering. Values do not reflect any changes made in the synthesizer.

Filename.pulses	The sequence of pulses that are the input to the vocal tract.
Filename.s	An ASCII file containing a series of at least three inverse-filtered glottal flow derivative pulses. Produced by the inverse filter.
Filename.segnoise	A binary file with the information about the segments, the noise filter.
Filename.shimm	An ASCII file listing the evenly spaced shimmer values.
Filename.sjitt	An ASCII file listing the evenly spaced synthetic jitter values
Filename.sou	An ASCII file containing the y coordinates for a single synthetic source pulse. The x coordinates are determined by the sampling rate. Unlike filename.LfSou, this pulse does not have to be LF-fitable, and can have any shape. The file is created by the synthesizer with the command File-Save source-Others.
Filename.strem	An ASCII file listing the evenly spaced synthetic tremor values
Filename.syn	An ASCII file containing a 1-sec sample of the synthetic voice. The file is created by the synthesizer with the commands File-Save synthetic speech or File-Save all.
Filename.vtxt	An ASCII file containing all the synthesizer variables in the format shown in Figure 53. This file is created by the synthesizer with the commands File-Save variables or File-Save all.
Filename.var	A file containing the information shown in Figure 53, but in binary format. The file is created at the same time as filename.txt, using the synthesizer commands File-Save variables or File-Save all.



```
File name :C:\synthesis\work\femalevoice\femalevoice.txt

Ee = -2.5697
Tp = 0.00186458 sec
Te = 0.0026 sec
T2 = 0.00283723 sec
Tc = 0.0033 sec

Synthetic Tremor Deviation = 0
Synthetic Tremor Frequency = 0
Synthetic Jitter Deviation = 0
Synthetic Shimmer Deviation = 0
Synthetic Noise = -28.6 dB

F1 = 746 Hz     B1 = 182 Hz
F2 = 1149 Hz     B2 = 35 Hz
F3 = 3041 Hz     B3 = 207 Hz
F4 = 3834 Hz     B4 = 257 Hz
F5 = 4627 Hz     B5 = 1200 Hz
```

Figure 53. Contents of the file filename.txt.

IV. SKY VOICE ANALYSIS PROGRAM

Part 1: About Sky

Sky is an analysis package specialized for acoustic analysis of voice. It is constantly evolving as new measures and analysis approaches appear in the literature. This documentation is as accurate and complete as possible, but care should be taken that results make sense and that algorithms are applied appropriately. References to the papers originally describing the algorithms are included when possible, as are descriptions of major deviations our implementation makes from the original descriptions.

Part 2: Program Installation

To install the program, copy the .exe file to the desired location on your system. Sky creates all other directories it needs. In particular, results of many analyses will be written to the directory C:\sky\work\[today's date]. These files are not automatically deleted, and you may wish to clean up your files from time to time.

Part 3: Menu Functions

File Menu (Figure 54)

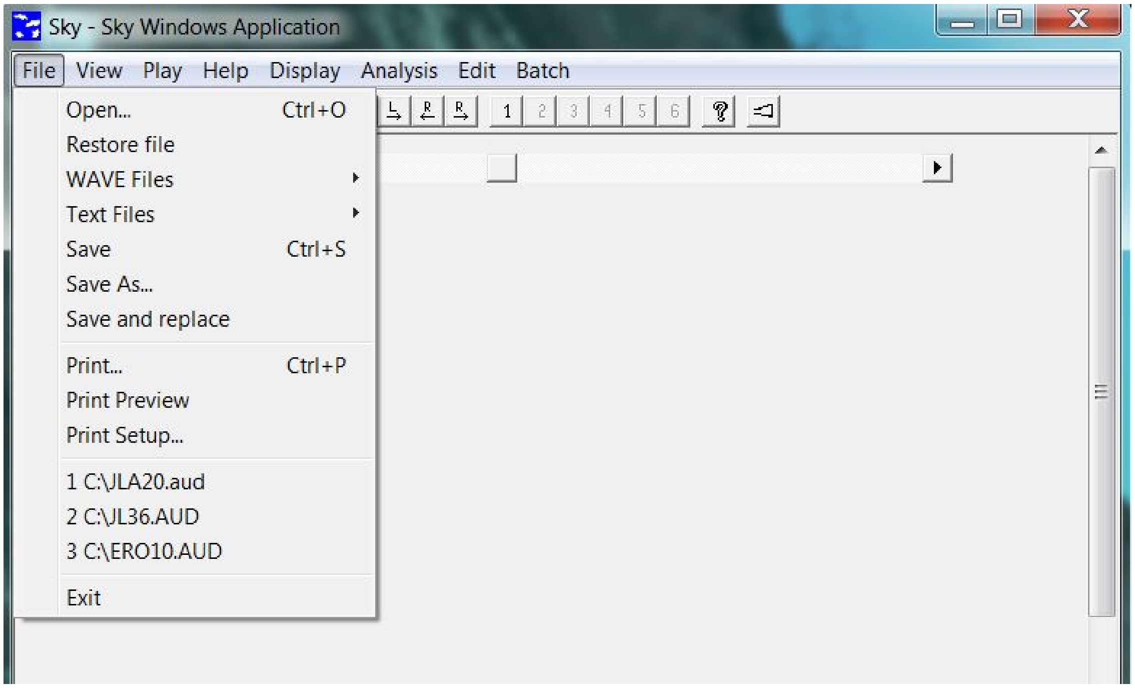


Figure 54. File menu options.

Open…

Opens a file that is in the UCLA-specific .aud format. If the file is in another format, the program returns the error message "This may not be a VAS file." Files used in Sky must be mono only. Any sampling rate is ok.

Figure 55 shows the features of the Sky workspace, and Figure 56 shows the toolbar. A maximum of 6 channels can be open at once. Only one channel at a time is "active." The window containing the active channel is outlined in green; the inactive channel(s) are outlined in magenta. To activate a different channel, click once anywhere in its frame, or use the toolbar and click on the number corresponding to the desired channel, or use the menu command Display-Active channel. The frame will turn green when the channel is activated.

To close a channel, double right click just outside the frame, to its right. (You can also close a channel using the Display-Deactivate channel command described below.) A channel does not have to be active to close it. Pressing the <delete> key closes all open channels.

Beware: The program does not prompt you to save before closing a channel, so be careful. If you close the active channel, you will need to click on another channel to activate it. This does not happen automatically.

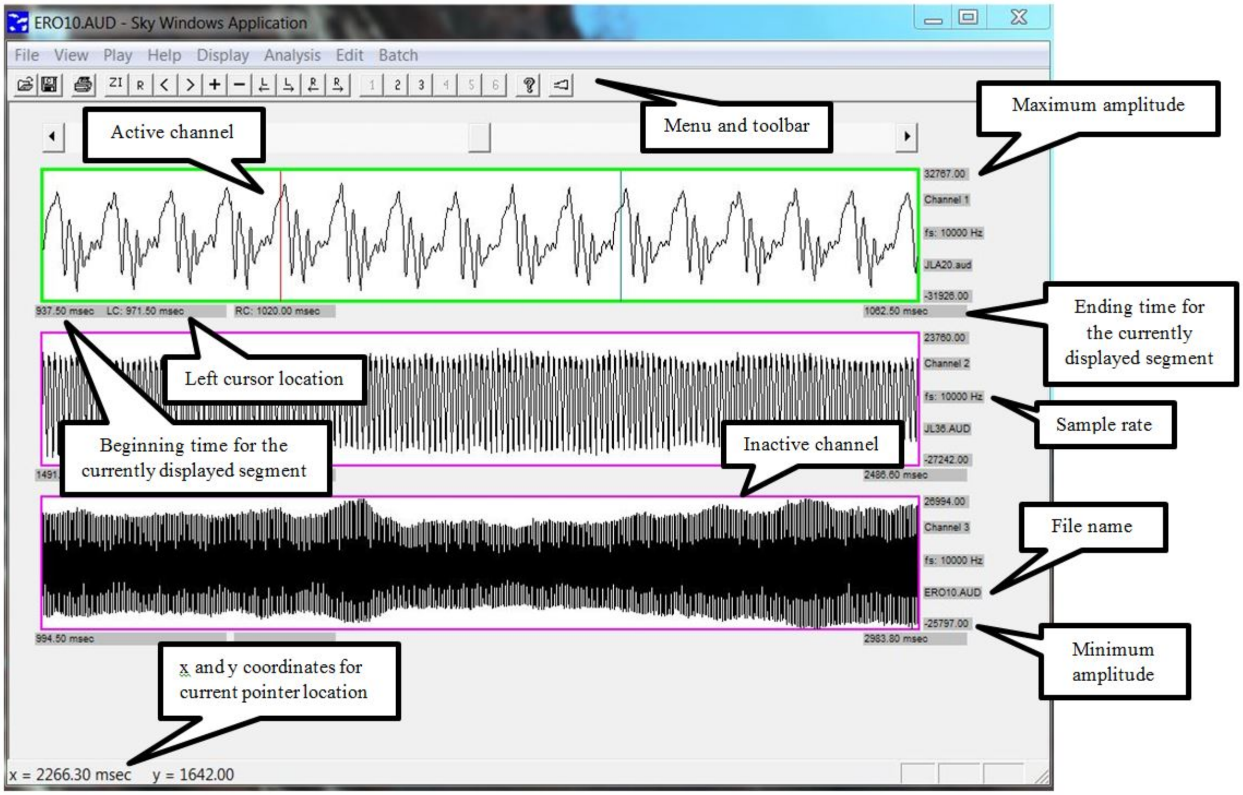


Figure 55. Sky workspace.

Restore file

Opens a copy of the last file to be opened in a new channel. Replaces the file if you've accidentally closed it; opens a second copy if it is already open.

WAVE files

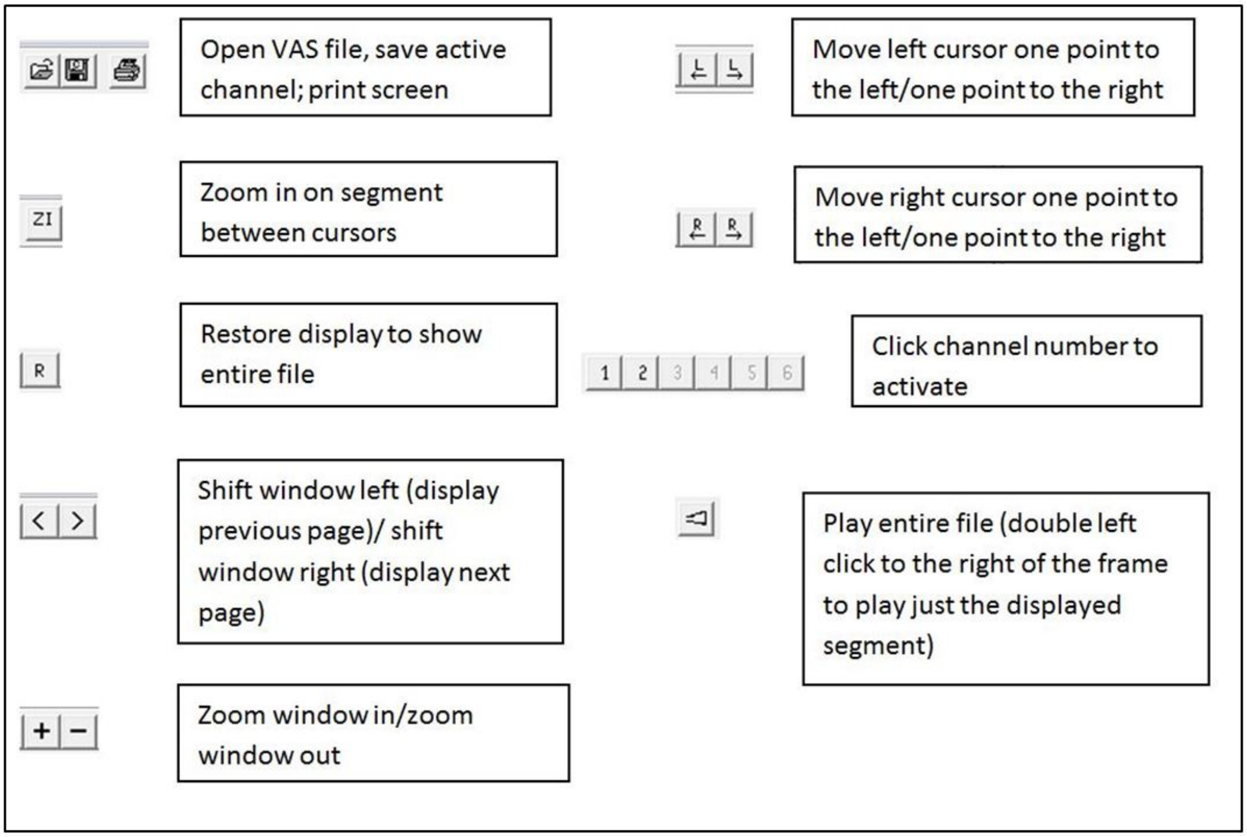
Use this menu to open a .wav file, or to convert a file to the .wav format.

Save

Saves the signal displayed in the active window in the .wav format. You must type the extension .wav as part of the name when you save, or the file will be saved in the .wav format but with no file extension. If this file is not a .wav file, it will be converted. This command only saves the portion of the file that is displayed, not the entire file. It does not affect the active file, which can be redisplayed in its entirety by clicking the R (restore) button on the toolbar. Be sure the window is zoomed appropriately so that you save what you actually mean to save.

Open

Opens a .wav file. Returns an error message window if the target file is not in the correct format.



Open VAS file, save active channel; print screen

Zoom in on segment between cursors

Restore display to show entire file

Shift window left (display previous page)/ shift window right (display next page)

Zoom window in/zoom window out

Move left cursor one point to the left/one point to the right

Move right cursor one point to the left/one point to the right

Click channel number to activate

Play entire file (double left click to the right of the frame to play just the displayed segment)

Figure 56. Sky toolbar components.

Text files

Save

Saves the signal displayed in the active window in ASCII format. You must type the extension .txt as part of the name when you save, or the file will be saved with no extension. If the file is not an ASCII file, it will be converted. As above, this command only saves the portion of the file that is displayed, not the entire file.

Open

Opens an ASCII file.

Save

Saves the portion of the signal displayed in the active window in the UCLA-specific .aud format. Extension .aud must be specified or the file will be saved with no extension.

Save as…

As above, but with the option to change the name of the file to avoid overwriting the original copy. Extension .aud must be specified or file will be saved with no extension.

Save and replace

Saves the portion of the signal currently displayed in the active window and replaces the open file with the displayed segment. Preserves file format, and appends 0 to the filename to prevent overwriting (e.g., file.aud becomes file0.aud).

Print…

Orientation must be landscape for this to work correctly.

Print Preview

Print Setup…

Sky print functions may or may not work properly, depending on your screen resolution. A frame grabber may be used in place of the print functions in a pinch.

[list of last 3 files used]

Exit

Using the File Menu to Convert File Formats

To convert file formats, open the file in its current format, and save it using the SAVE option under the desired target format. For example, to convert an ASCII file to .wav, first open the ASCII file using the command File-Text files-Open, then convert using the command File-Wave files-Save. Specify the extension you want (.txt, .wav), as the program does not append one automatically.

File formats can also be changed for multiple files at once using the BATCH menu, as described below.

View Menu

Toolbar

Status bar

Check or uncheck these options as desired.

Play Menu

These commands allow the user to play the entire file, the displayed segment, or the segment between the cursors (whether displayed or not). You can also play the displayed segment in a channel (whether or not the channel is active) by double left clicking in the margin to the right of the file; and you can play the entire file using the sound icon on the toolbar.

Help Menu

Not currently helpful.

Display Menu

The Display menu includes a variety of options for changing the way files are displayed, placing and moving cursors, and so on (see Figure 57). It also includes tools for marking, editing, and displaying events, which are the basis for analyses of F0 and F0 perturbation.

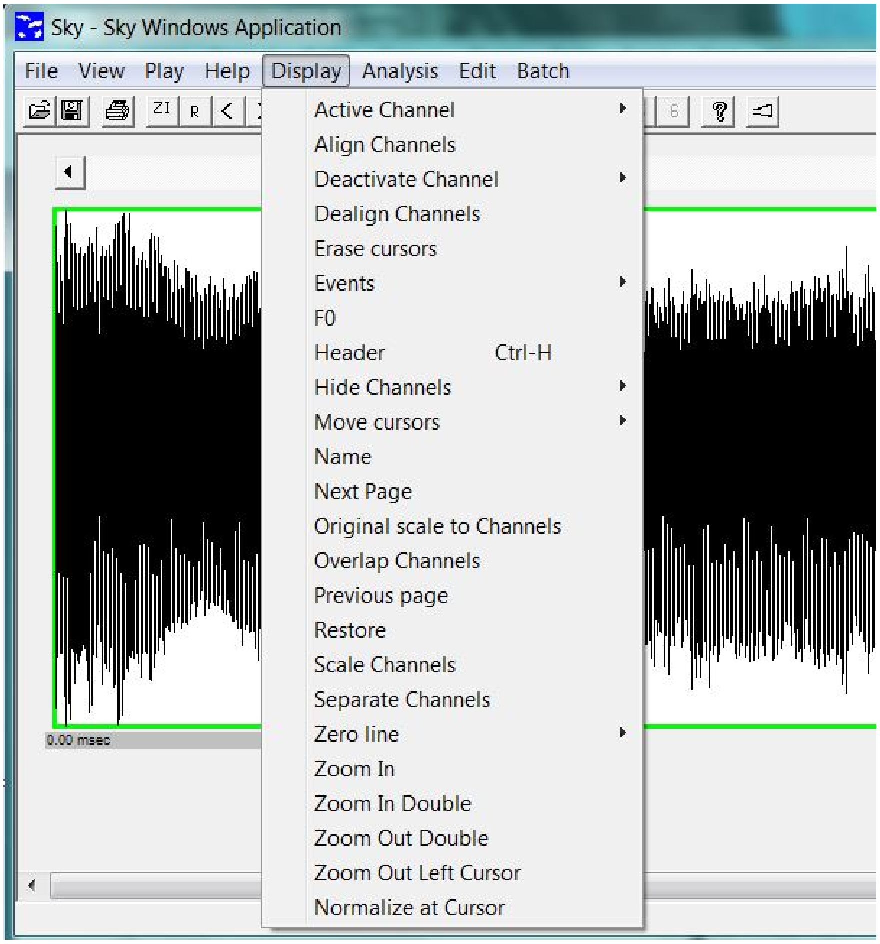


Figure 57. Display menu options.

Many functions in this menu require setting cursors. Cursors can only be set in the active channel, although existing cursors remain in place when a channel is deactivated. To place the left cursor, left single click the desired location. A red line will appear, indicating the cursor position. To place the right cursor, right single click in the desired location. The right cursor is displayed in blue. To move the cursors, single click again in the new desired location, or use the L and R buttons on the toolbar.

Active channel

Produces a list of channels in use (in dark type) and empty channels (in grey type). Highlight the channel you wish to activate. Its frame will turn green. It is also possible to activate a channel by clicking its number on the toolbar, or by single left clicking somewhere within the appropriate frame in the waveform display.

Align channels

Temporally aligns the active channel and the channel or channels you select. Once channels are aligned, they stay aligned until you select Dealign Channels. Setting cursors in one channel will place cursors at the same points in time in any aligned channels, and zooming in or out one window will zoom the other or others in the same manner.

Deactivate Channel

Closes the file on the specified channel without saving, whether or not the file is active. You can also close all open channels at once by selecting Deactivate Channel- All or by pressing the <delete> key. Individual channels can also be closed by double right clicking outside and just to the right of the frame.

Dealign Channels

Turns channel alignment off so that channels can be zoomed or scrolled independently.

Erase Cursors

Removes both cursors from the active channel.

Events

“Events” are markers that indicate the beginning and end of each cycle of phonation in a file. They form the basis for analyses of F0, F0 perturbation, and some measures of source spectral slope. To mark events automatically, first select a waveform landmark (a peak or zero crossing) that recurs relatively reliably across the entire file. Mark a cycle of phonation by placing the left and right cursors on adjacent instances of this landmark. The cursors don’t have to be placed exactly—within a few points is fine. Next, execute the command Display-F0. Finally, execute the command Analysis-Pitch. A dialog box will open. Indicate whether a peak or zero crossing is to be marked, and change the event code number if desired. (It doesn’t really matter what number you use.) Finally, check the Save tracks to file option if you want to create the files necessary to import the pitch track for this sample into the synthesizer. Uncheck the option if you do not want the pitch data to be saved. Click ok when you’re done. The program places a small mark at each cycle boundary. Page through the file to be sure the tracking is accurate. Events are saved automatically in the file c:\sky\work\[today’s date]\filename.ev.

If the program is unable to track the event through the whole file, the process will pause and ask you if you wish to continue. If you do, place the cursor at the beginning of the next cycle and select Analysis-Pitch a second time to continue the analysis.

Sky allows you to edit events in various ways, to correct errors in automatic tracking, to provide approximate pitch contours for voices that are too irregular for automatic pitch tracking, or to handle pitch tracking for period doubled or diplophonic voices. The options available for working with events in Sky are shown in Figure58.

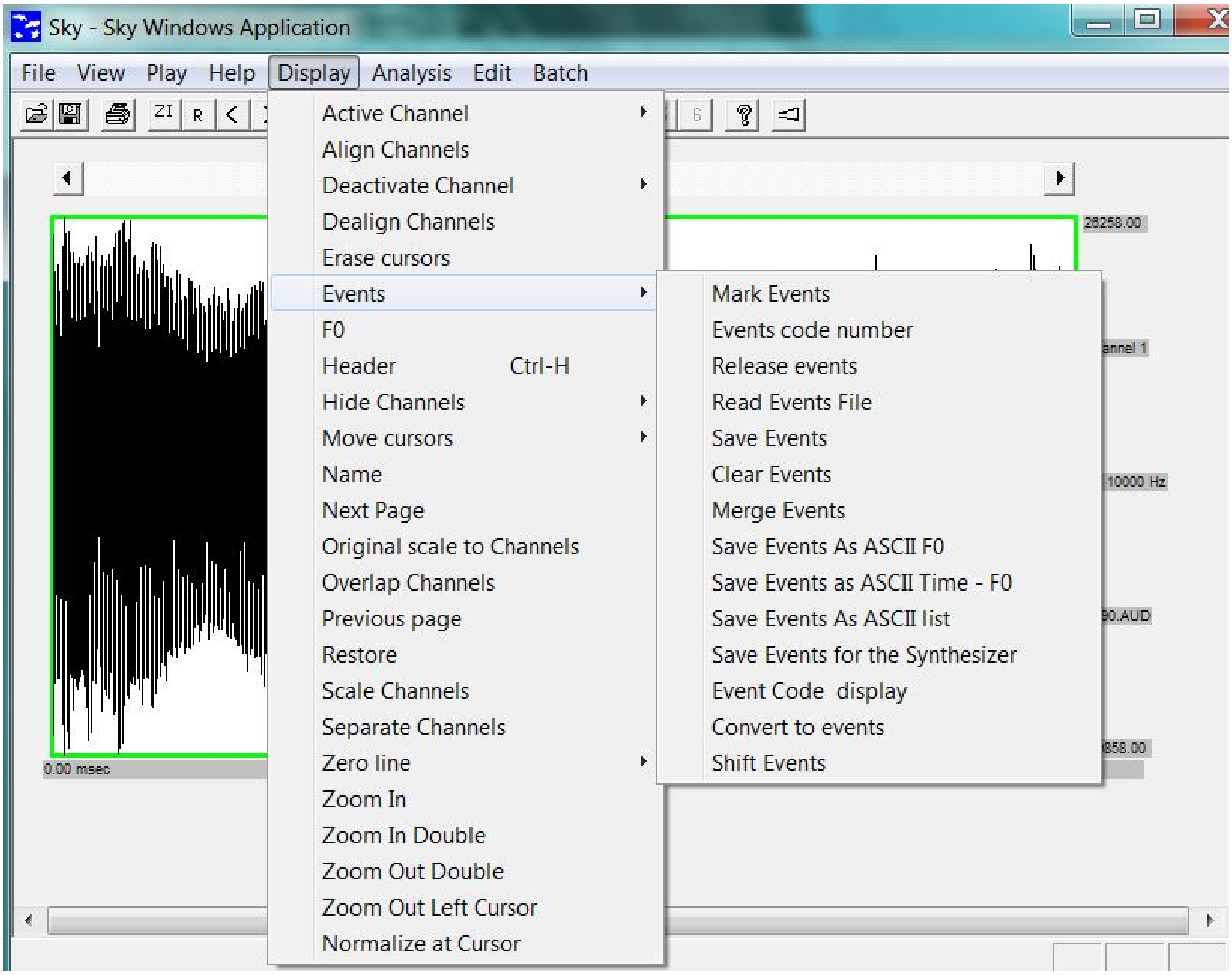


Figure 58. Options for editing and managing events.

Mark Events

After you have run Analysis-Pitch, use this command to edit and manipulate events in various ways. Selecting Display-Events-Mark events turns off the cursors so that mouse clicks add and delete event markers. Double left click to place an event marker; double right click to delete one. Zoom the window (by clicking the + key on the toolbar) as needed to make this easier. Placing an event creates a small marker on the waveform, and a small rectangle containing the event number appears below the waveform. Sometimes the events hop around, so keep an eye on those numbered rectangles to be sure that you don't accidentally mark two events in the same place. (Just delete the second one if this happens.) When you are finished editing events, select Display-Events-Release events as described below to reinstate the cursors. You may wish to save the events file as well; see Display-Events-Save events command.

Event code number

Enter an integer code number for this set of events. Default used during automatic marking is 4.

Release events

Stops the event marking process and activates the cursors. Use when you are done marking events, or the program will think every click is a marking attempt and funny things will happen.

Read events file

Reads the (previously saved) events file for the voice sample in the active channel and displays the events.

Save events

Saves the events. The original events generated by Analysis-Pitch are saved automatically, and it is not necessary to resave unless you have changed them.

Clear events

Erases all events from the screen display (and from memory), but does not erase the saved events file. To restore events after clearing them, use Display- Events-Read events file.

Merge events

This command was implemented to automate tracking of F0 in voices with period doubling. The pitch tracker will often crash on period doubled voices if you try to track the A and B pulses at the same time, but it can often track the double cycle accurately. To use this command to track both the A and B cycles, first open two copies of the same file. Track one double cycle (A+B) in one file; track the second long cycle (B+A) in the second. Then select Display-Events- Merge events. The program will combine the events from channel 1 and channel 2, and plot them both on a third copy of the voice sample in channel 3. The events in this channel should mark the A and B cycles correctly (Figure 59). Events are not saved automatically; execute the command Display-Events-Save events command if desired.

Save events as ASCII F0

Saves an ASCII file named filename.apr in directory c:\sky\work\[today's date]. This directory is created automatically when it is needed. The file contains the time of each event in the active channel and the F0 (reciprocal of the period length) of the associated period. The file can be imported into a statistics package or graphics program for analyses of F0. (Traditional F0 analyses can also be calculated in Sky—see **Analysis Menu** below.)

Save events as ASCII list

Saves an ASCII file named filename.aev in directory c:\sky\work\[today's date]. This file contains the event code for each event in the active channel, the offset in points, and the time of the event in msec.

Event code display

Toggles the event code display (the little numbers in the rectangles) on and off.

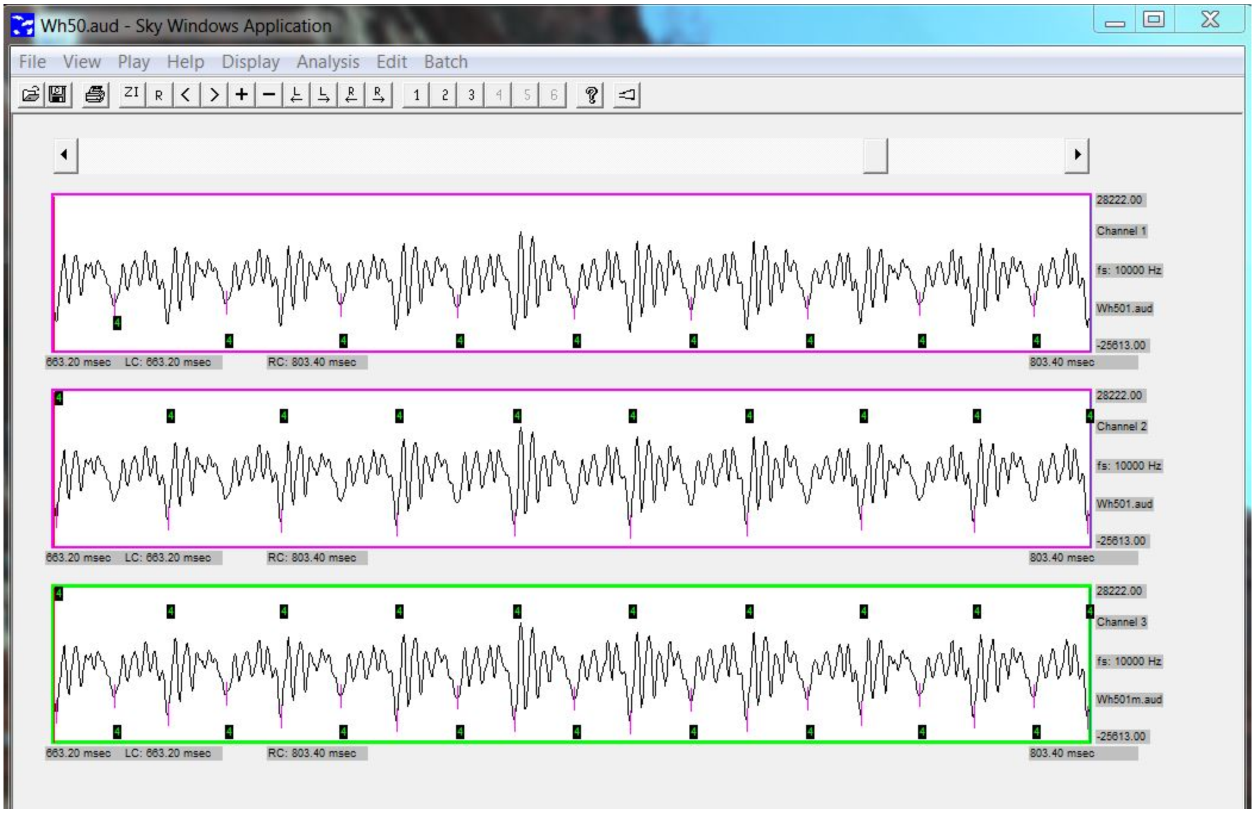


Figure 59. Merging events to track F0 in period-doubled phonation. The waveform being analyzed shows the typical pattern of two cycles repeating in an A/B/A/B pattern. Cursors in the first channel mark one A/B pattern repeat; cursors in the second channel mark one B/A repeat. The waveform in the third channel shows the merged events, with each A and B cycle correctly marked (scale has been changed to show detail).

Convert to events

Converts a column of integers stored in an ASCII file into events in the active channel. The integers represent the number of individual points in the file. The file can have any name, but the extension should be .asc.

F0

Displays the reciprocal of the period of the signal between the cursors. When the cursors delimit a single phonatory cycle, this provides an estimate of F0.

Header

Opens a window containing the information in the audio file header. Depending on file format, this may include the maximum and minimum values on the y axis, the sampling rate, the duration of the file, and the number of points in the file.

Hide channels

Hides a channel without deactivating it. A button showing the name of the hidden file will appear at the bottom of the screen. Click this to reactivate the channel.

Move cursors

Left cursor

Increment: Moves left cursor one point to the right. The same goal may be accomplished with the L→ key on the toolbar.

Decrement: Moves the left cursor one point to the left. The same goal may be accomplished with the L← key on the toolbar.

Right cursor

Increment: Moves right cursor one point to the right. The same goal may be accomplished with the R→ key on the toolbar.

Decrement: Moves the right cursor one point to the left. The same goal may be accomplished with the R← key on the toolbar.

Name

Changes the name of the file in the active channel without saving it. Type the new name and extension into the dialog box and click ok. The program does not add an extension automatically—you must specify one either when renaming or when saving the file. If you later save the file or segment, it will be saved with the new name unless you specify otherwise.

Next page

Moves the display of the active channel one page to the right. Page size is determined by the amount of zoom currently applied to the active channel. The same function is served by the > button on the toolbar.

Original scale to channels

Reestablishes the original scale in the open channels.

Overlap channels

Overlaps the data from all channels in a single channel. Data are not rescaled or realigned. The file originally in channel 1 is plotted in red; channel 2 is plotted in green, channel 3 is plotted in grey, and channel 4 is plotted in blue. File names are also color coded in the same manner.

Previous page

Moves the display one page to the left. Page size is determined by the amount of zoom currently applied to the active channel. The same function is served by the < button on the toolbar.

Restore

Restores a zoomed channel so that the entire file (including the cursors) is redisplayed. The same function is served by the R button on the toolbar.

Scale channels

Replots all open channels on the same vertical scale.

Separate channels

Replots overlapped files as separate channels, in their original order.

Zero line

Insert

Inserts a line y=0 into the active channel display.

Delete

Removes the zero line from the display.

Zoom in

Expands the display so that only the segment between cursors is shown. The same function is served by the ZI button on the toolbar. This command has no effect unless the cursors have been set.

Zoom in double

Zooms the window to show only the middle 50% of points. For example, if the file segment begins at 0 msec and ends at 4000 msec, it will display the segment from 1000 to 3000 msec. This command may be repeatedly applied to zoom in further. Zooming is relative to the left and right edges of the window, not to cursor position, which has no effect in this command.

Zoom out double

Zooms the window out doubly (doubles the number of points displayed), keeping the center point of the display fixed.

Analysis Menu (Figure 60)

Unless otherwise specified, output of each analysis is automatically saved to the directory C:\sky\work\[today’s date]. The program automatically creates this directory when needed. Files are not automatically deleted, and you may wish to clean up the work directory from time to time.

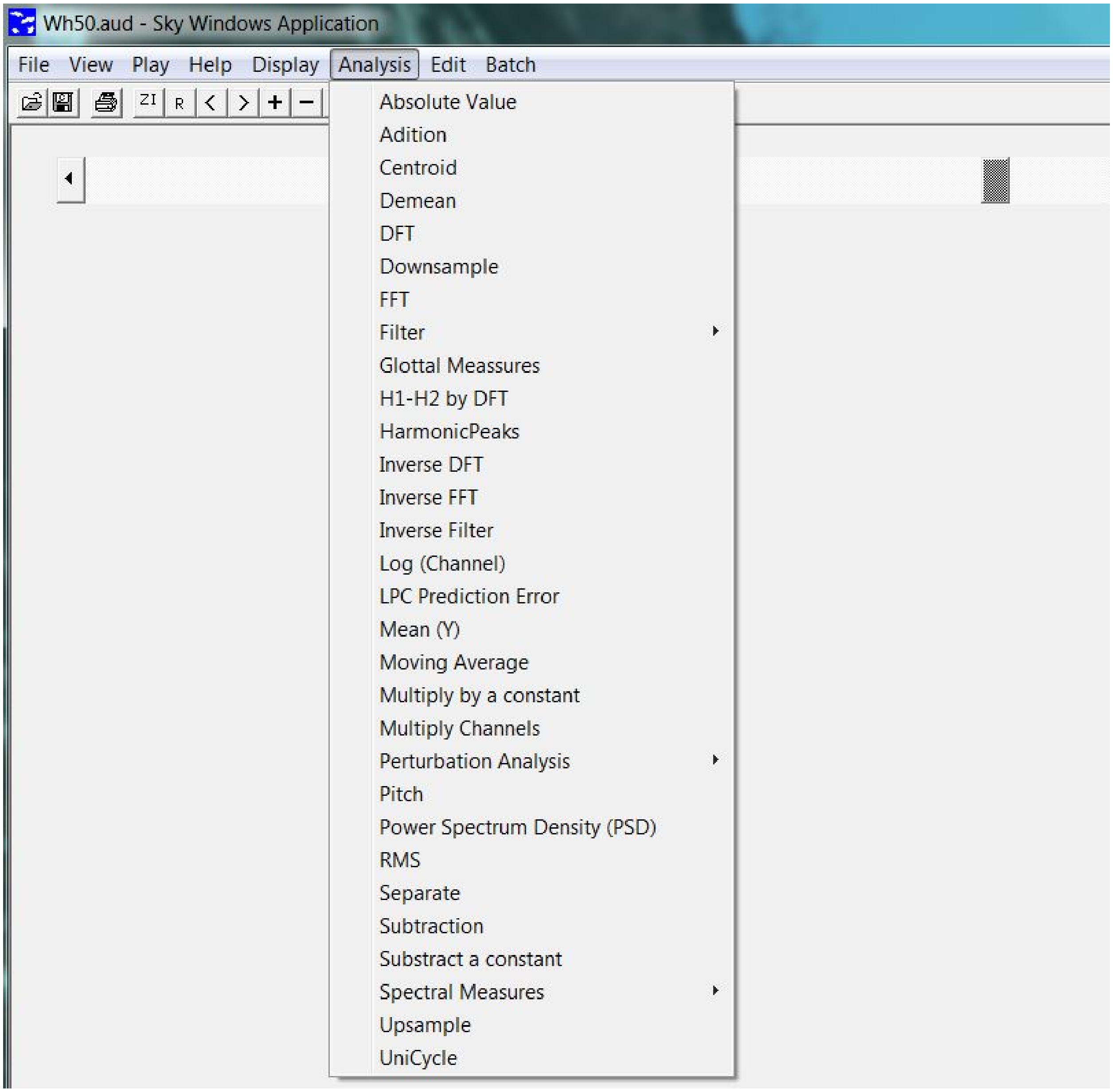


Figure 60. Analysis menu options

Absolute value

Computes the absolute value of each element in the active channel and displays the result in another channel with same name and extension .abs. It is not automatically saved.

Addition

Adds the specified waveforms together point by point. The command opens a dialog box in which the user indicates which files to include. Files must have the same sampling rate. The result is placed in a new channel with the filename of the first file in the sum and with extension .add. The new file has the same duration as the shortest file in the sum

Centroid

Returns the value of the amplitude-weighted mean of the points between the cursors. Set the left and right cursors prior to using this command.

Demean

Calculates the mean amplitude across the file segment displayed in the active window, and subtracts this value from each point in the displayed segment. (If you want to demean the complete file, be sure the window isn't zoomed.) The resultant waveform is opened in a new channel with the original file name and extension .dmn. It is not automatically saved. This is useful for removing a DC offset from a file.

DFT

Computes a Fourier transform using the standard definition. It is useful to compute a transform of any length. If the user sets the length to 0, it performs the transform of the entire file. Phase can be saved for a future inverse DTF computation.

Downsample

Downsamples the file segment displayed in the active window, and places the result in a new channel with the original filename and extension .dwn. (File is not automatically saved.) User specifies the desired target sample rate in a dialog box, which must be a whole-number divisor of the original sample rate. Appropriate anti-aliasing filtering is applied automatically.

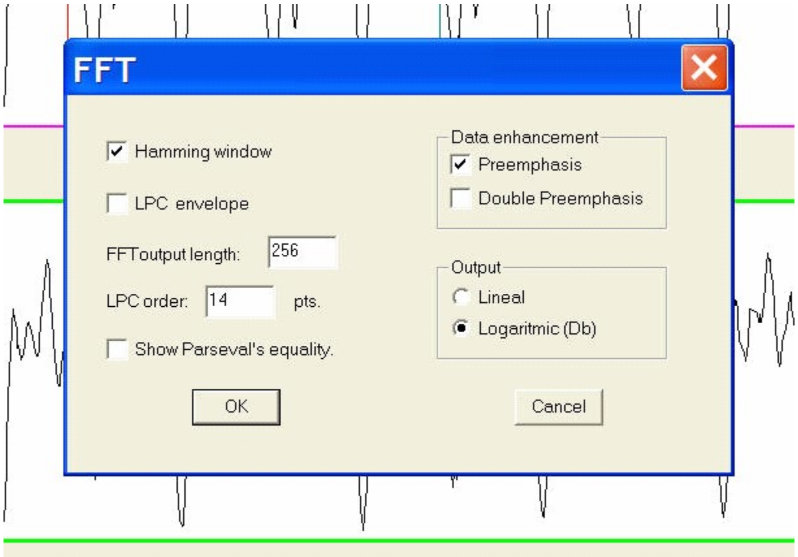
FFT

Calculates a fast Fourier transform. Analysis options are shown in the dialog box in Figure 61. The output spectrum is displayed in a new channel with the original file name and extension .dB (dB spectrum) or .mag (magnitude spectrum). If you place the cursor over a point in the spectrum, the frequency at that point will be displayed in the lower left portion of the analysis window.

Filter

Lowpass

Applies a linear phase low-pass filter with a Kaiser window design to the file in the active channel. The command opens a dialog box in which the user can specify the cutoff frequency, stopband frequency, maximum passband ripple, and minimum stopband attenuation. The filtered signal is placed in a new channel with the name filename.klp.



FFT

☑ Hamming window

□ LPC envelope

FFToutput length: 256

LPC order: 14 pts.

□ Show Parseval's equality.

Data enhancement

☑ Preemphasis

□ Double Preemphasis

Output

○ Lineal

● Logaritmic (Db)

Figure 61. Dialog box showing options for calculating an FFT.

Highpass

Applies a linear phase high-pass filter with a Kaiser window design to the file in the active channel. The command opens a dialog box in which the user can specify the cutoff frequency, stopband frequency, maximum passband ripple, and minimum stopband attenuation. The filtered signal is placed in a new channel with the name filename.khp.

Differentiator

Differentiates the signal in the active channel and places the result in a new channel with the name filename.der. The difference equation is $y(n) = x(n) - x(n-1)$.

Notch (zero)

Applies a second order filter with difference equation $y(n) = \frac{x(n)+b*x(n-1)+c*x(n-2)}{1+b+c}$ to the file in the active channel. User specifies the center frequency and bandwidth of the filter, and optionally the window size. The result is placed in a new channel with name filename.not. To view the frequency response of the filter, the user can optionally send the frequency response to a channel with the selected length.

Pole

Applies a second order resonator to the file in the active channel, with difference equation $y(n) = c * x(n) - a * y(n-1) - b * y(n-2)$. The user specifies the center frequency, bandwidth, and optionally the window size. The result is placed in a new channel with name filename.res. The user can view the frequency response of the filter, as above.

Integrator

Integrates the signal in the active channel. The result is placed in a new channel with name filename.int. The filter difference equation is $y(n) = \big(x(n) + x(n-1)\big) * 0.5 + y(n -$

1) following the description in Hamming (1983).

Glottal measures

Estimates open quotient, speed quotient, speed index, rate quotient, DC offset, and peak flow from the acoustic waveform, and opens a new window with results. To use this command, first run the inverse filter with Mark glottal events selected. This leaves the channel with the glottal pulses as the active channel.

H1H2 by DFT

This function calculates the difference between the amplitudes of H1 and H2 in dB. Before running this command, estimate F0 as described in the Display-F0 menu. When invoked, a window opens requesting an analysis window length. Because this uses the definition for the calculation of a Fourier transform, the analysis length can be any number. To calculate a pitch synchronous analysis, concatenate a cycle several times and then compute the analysis on an exact number of cycles. The algorithm then smooths the spectrum to localize the harmonics before returning a value. The difference between H1 and H2 can also be calculated with the routines described under Spectral analyses.

Harmonic peaks

Computes a Fourier transform of any length of the entire file or a segment of it, and connects the harmonic peaks. You need to mark F0 using Display-F0. It opens a box with some parameters to select. If the analysis length is set to 0 it will take the entire file length.

Inverse DFT

Performs an inverse Fourier transform of a sequence in polar coordinates, with magnitude stored in the active channel, and phase in the channel specified by the user.

Inverse filter

A non-interactive inverse filter. (For interactive inverse filtering, use the program IF.exe described earlier in this document.) Zoom in the window and select a cycle by left clicking (to place the left cursor) and right clicking (to set the right cursor). Select Analysis-Inverse filter. A dialog box opens, as shown in Figure 62. Select audio (assuming the signal was recorded with a microphone), and specify the LPC autocorrelation analysis window length (increase to 512 points if F0 is near 100 Hz) and LPC covariance analysis window length (decrease if F0 is over 200 Hz). When you click ok, a second dialog box opens with proposed formants and bandwidths, as shown in Figure 63. If the values look ok, click Finish, or enter new values for formants, bandwidths, or analysis parameters and click Try again. After you click Finish, the results appear as shown in Figure 64. The first channel shows the original signal. The second channel shows the LPC prediction error (filename.err), which indicates the beginning of the most-closed-phase of the signal. The third channel shows the glottal pulses (filename.glot), and the fourth channel shows the flow derivative waveform (filename.flwder). **Warning:** This routine only runs for a sampling rate of 10 kHz. Downsample before running if necessary.

Log

Computes the $\log_{10}$ of the elements of the channel which must be larger than zero.

LPC prediction error

Calculates the LPC prediction error for the file segment displayed in the active window. User specifies analysis window size, analysis order, preemphasis, and windowing options in a dialog box that opens when the command is executed. Analysis result is placed in an open channel with name filename.perr.

Mean (Y)

Returns a value equal to the average of the y coordinates for all the points currently displayed in the active window.

Moving average

Calculates a moving average of the file segment displayed in the active window. User specifies analysis window size, decimation, and shaping window in a dialog box that opens when the command is selected. Result of the analysis is opened in a new channel with name filename.mav.

Multiply by a Constant

Multiplies the file segment displayed in the active window by a constant (positive or negative) specified by the user, and opens the resulting file in a new channel with the filename of the multiplicand and extension .cmul.

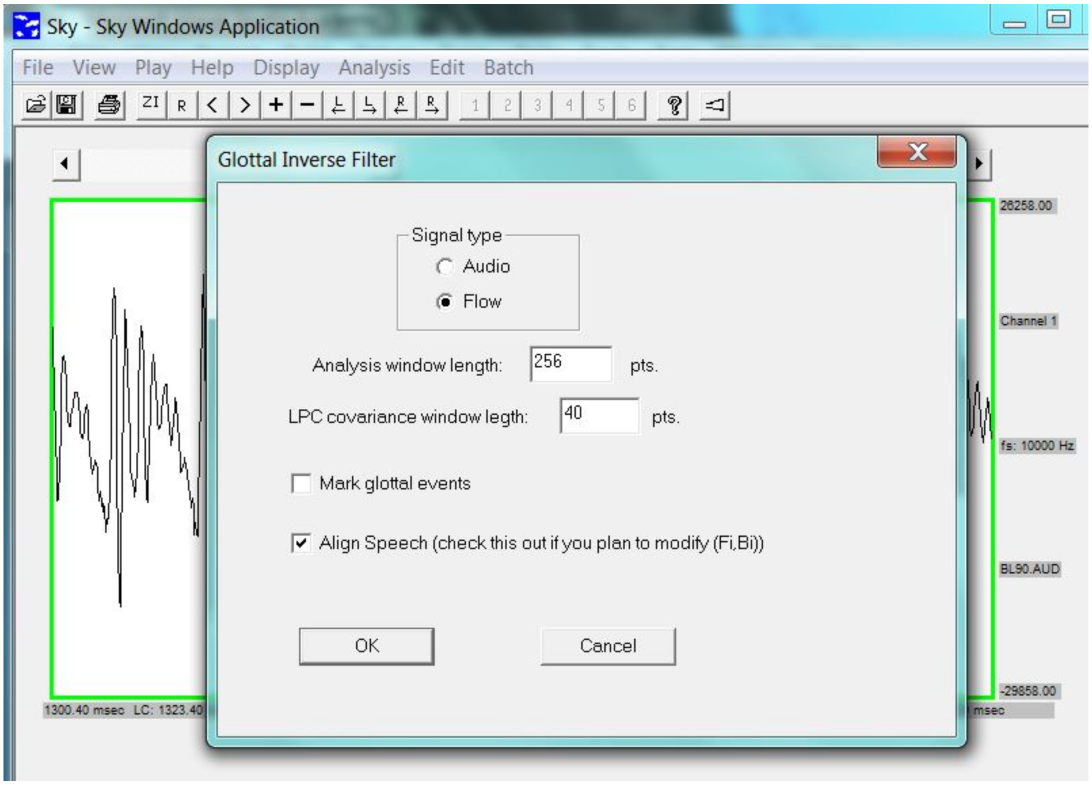
Perturbation analysis

Jitter

Calculates a variety of measures of frequency perturbation, as shown in Figure 65. Interpolation is applied automatically. Pitch analysis must be run first. If you are using events from a previous pitch analysis, read the events file and then save the events into today’s directory under c:\sky\work, or copy the events file (filename.ev) into that directory. The analysis will not run if the events file is not in today’s directory.

Shimmer

Calculates mean shimmer, the standard deviation of shimmer, and directional perturbation, and prints values in a new window along with the number of cycles analyzed and the analysis length in msec. Interpolation is applied automatically. Pitch analysis must be run first, and the events file must be in today’s directory under c:\sky\work, as described for jitter analyses.



Glottal Inverse Filter

Signal type

○ Audio

● Flow

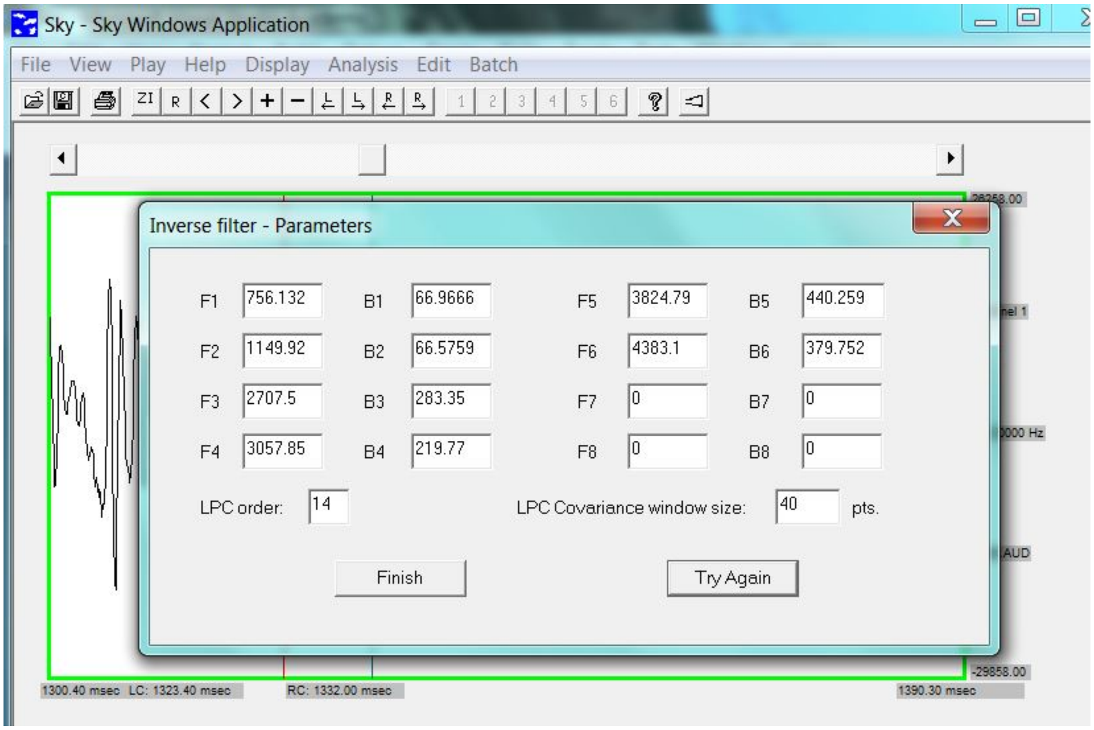
Analysis window length: 256 pts.

LPC covariance window legth: 40 pts.

☐ Mark glottal events

☑ Align Speech (check this out if you plan to modify (Fi,Bi))

Figure 62. First dialog box for inverse filtering; specifies parameters for LPC analysis.


Sky - Sky Windows Application
File View Play Help Display Analysis Edit Batch
ZI R
1 2 3 4 5 6
26258.00
nel 1
000 Hz
AUD
-29858.00
1300.40 msec LC: 1323.40 msec
RC: 1332.00 msec
1390.30 msec
Finish
Try Again

Inverse filter - Parameters

F1	756.132	B1	66.9666	F5	3824.79	B5	440.259
F2	1149.92	B2	66.5759	F6	4383.1	B6	379.752
F3	2707.5	B3	283.35	F7	0	B7	0
F4	3057.85	B4	219.77	F8	0	B8	0

LPC order: 14

LPC Covariance window size: 40 pts.

Figure 63. Second dialog box for inverse filtering, used for entering or correcting resonance parameters.

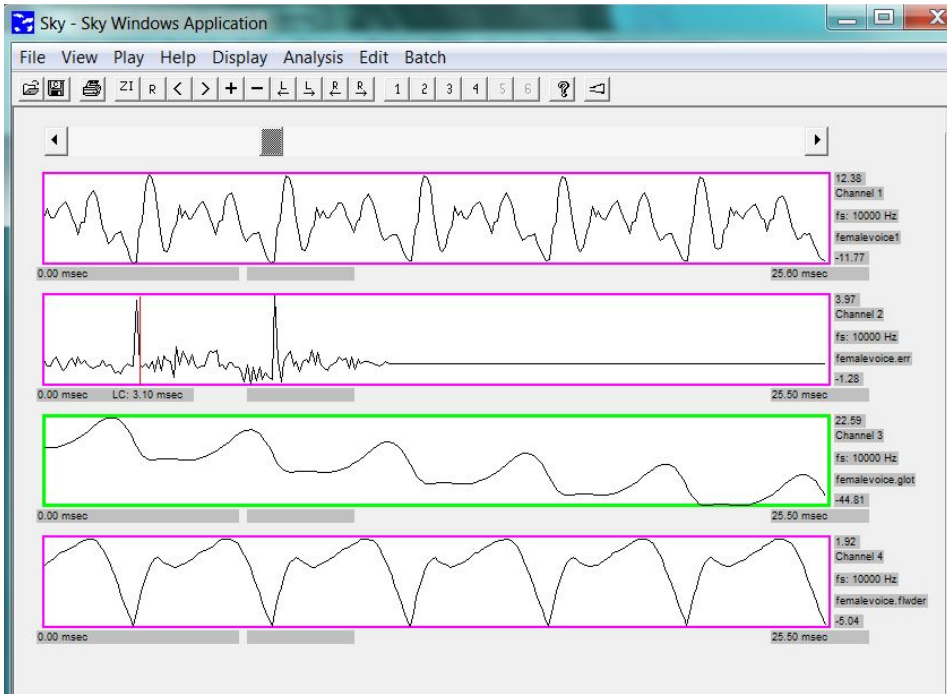
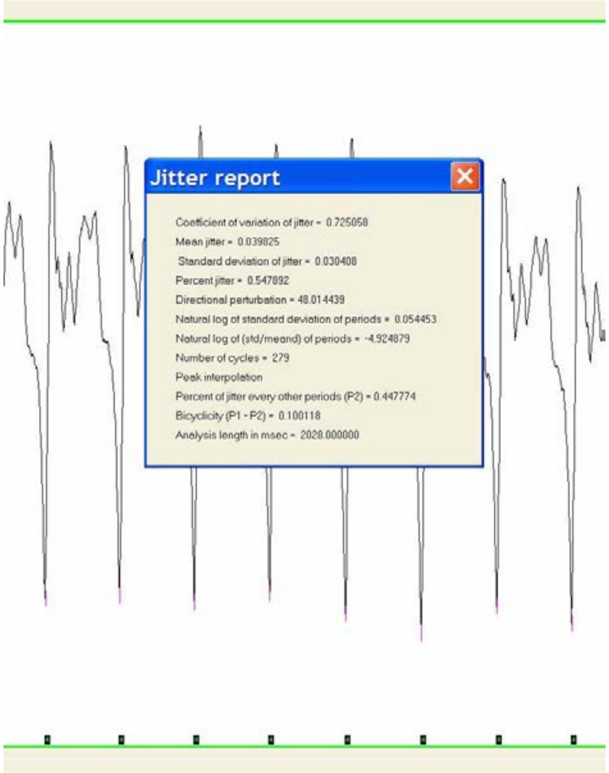


Figure 64. Inverse filtering results. The top waveform shows the acoustic time series; the second shows the LPC prediction error; the third shows glottal pulses, and the fourth shows the glottal flow derivative.



Jitter report

```
Coefficient of variation of jitter =  0.725058
Mean jitter =  0.039825
 Standard deviation of jitter =  0.030408
Percent jitter =  0.547892
Directional perturbation = 48.014439
Natural log of standard deviation of periods =  0.054453
Natural log of (std/meand) of periods =  -4.924879
Number of cycles =  279
Peak interpolation
Percent of jitter every other periods (P2) = 0.447774
Bicyclicity (P1 - P2) =  0.100118
Analysis length in msec =  2028.000000
```

Figure 65. Results of Sky jitter analyses calculated using the **Analysis-Perturbation analysis-Jitter** command.

HTN

Calculates the harmonics-to-noise ratio, as described by Yumoto et al. (1982). Pitch analysis must be run first, and the events file must be in today's directory under c:\sky\work, as described for jitter analyses.

Pitch

Pitch analysis places event markers at the boundaries of each cycle of phonation. It is a preliminary to many acoustic analyses, as described in the various sections. To run the analysis, first identify a waveform landmark (peak or zero crossing) that repeats relatively reliably across the entire signal to be analyzed, and mark a cycle using this landmark by placing the left and right cursors. Cursor placement doesn't have to be exact. Next, execute the command Display-F0. Finally, do Analysis-Pitch. A dialog box will open. Indicate whether a peak or zero crossing is to be marked, and change the event code number if desired. (It doesn't really matter what number you use.) Finally, check the Save tracks to file option if you want to create the files necessary to import the pitch track for this sample into the synthesizer, and update the save location if necessary. Uncheck the option if you do not want the pitch data to be saved. Click ok when you're done. The program places a small mark at each cycle boundary. It's a good idea to page through the file and verify that events are placed accurately and consistently. Events are saved automatically in the file c:\sky\work\[today's date]\filename.ev. Tools for editing errant event markers or for tracking pitch manually are described under Display-Events.

Power spectrum density (PSD)

The average of consecutive FFT's of the signal with the same length N. The segments can be taken one after another or overlapping. Each segment is multiplied by a shaping window. If the segment length is smaller than N, it is padded with zeros. The algorithm was taken from Press et al., "Numerical Recipes in C++" (2005).

RMS

For each selected point in the file in the active channel, this command performs an RMS analysis of length *window*, jumping *decimation* points to the next analysis. Both window length and decimation value are entered in a dialog box that opens when the command is selected. The resulting signal is opened in a new channel with the name filename.rms.

Separate

This command separates sections of a signal that have been marked with events. Example: run Pitch to mark the cycles, then run Separate to create a file with each cycle.

Subtraction

Subtracts the file or file segment displayed in one channel from that displayed in a second channel. User specifies the channels in a dialog box. The resulting signal is opened in a new channel with the filename of the minuend and extension .sub.

Subtract a constant

Subtracts a constant value (positive or negative) from each point in the file segment displayed in the active channel. The user enters the desired constant in a dialog box. The resulting signal is displayed in a new channel with the filename of the minuend and extension .csub.

Spectral measures

Sky calculates a variety of measures of the slope of the spectrum, as described in this section. Analyses can be applied to extracted source pulses or to the entire waveform. T

Deviation from horizontal

This algorithm calculates the deviation of the spectral slope of the file in the active channel from an “ideal” source spectral slope of -12 dB/octave in 4 bands: 0-1 kHz, 1-2 kHz, 2-3 kHz, and 3-4 kHz (Sundberg & Gauffin, 1979; Ní Chasaide & Gobl, 2010). Because vocal tract resonances affect the spectral slope, these measures are normally calculated on the glottal flow derivative, rather than on the entire audio waveform, although it will work either way. Slopes are calculated from the entire length of the file. Thus, if a single cycle is being analyzed, it should be concatenated to the appropriate length or the file padded with zeros. Pitch synchronous inverse filtering can also be used to extract a sequence of source pulses. The file must have sample rate equal to 10 kHz or the analysis will return all zeros.

To run the analysis, first mark a cycle. Then select Analysis-Spectral measures-Deviation from horizontal. The program returns the difference in dB between the ideal slope in each frequency band and the slope estimated for the current signal. Results are also saved to an ASCII file in the directory c:\sky\work\[today’s date] with name filenameext.dev (e.g., if the original file is filename.wav, the saved file will be named filenamewav.dev).

Frequency bands

Computes the ratio of spectral energy in two frequency bands (typically one low frequency, for example from 0-1 kHz, and one higher-frequency, for example from 1-5 kHz or 5-8 kHz; e.g., Lofqvist & Manderson, 1987). Open a file containing a single source pulse. Concatenate it, and make it the active channel. Select Analysis-Spectral measures- Frequency bands. A dialog box will open. Enter values for the frequency bands you wish to compare (defaults are 0-1 kHz and 1-5 kHz). If you are analyzing a single pulse, also enter the file length for the analysis (it must be a power of 2) and the program will automatically pad the file with 0 to that length. (Alternatively, concatenate the source pulse to a length exceeding 1024 points prior to computing the analysis.) Select dB spectrum, or uncheck this option to calculate the energy ratios from a linear spectrum. Finally, select preemphasis or no preemphasis. The analysis returns the ratio of the two energy levels in a new window, and also places the spectrum on which the result is based in a new channel with name filename.db (for a dB spectrum) or filename.mag (for a magnitude spectrum).

Hanson

Calculates measures of H1*-H2*, H1*-A1*, and H1*-A3* as described by Hanson (1997), with the modifications described by Iseli and Alwan (2004). To use this command, you will need to know the values of the poles and bandwidths. If this is a synthetic signal, or if you have inverse filtered this signal, use the values from the synthesizer or the inverse filter. Otherwise, derive your best estimates using your preferred method. Mark a cycle with the cursors before selecting the analysis. When the dialog box opens, enter the values of the poles and bandwidths into the top table to calculate H1*-H2*. To calculate H1*-A1* or H1*-A3*, enter the values of the appropriate poles and bandwidths in the second table. Two cells are provided for each resonance in this table, because occasionally matching the spectral shape during voice synthesis requires placing two resonances very close together or on top of each other. If this is the case with your sample, enter all the values in the table.

Spectral slope

Computes the slope of a straight line fitted to the peaks of the first n harmonics of a power spectrum (Jackson et al., 1985), where n is an integer specified by the user. Analyses are computed from single glottal pulses. To analyze a single cycle, open the file and mark the beginning and end of the pulse with the left and right cursors. Select Analysis-Spectral measures-Spectral slope from the menu. In the dialog box, specify the number of harmonics to which a line should be fit, and whether the spectrum should be calculated on a dB or linear scale. Pitch synchronous analysis is recommended. The frequency and amplitude of each harmonic from dB (filename.dbampl) and magnitude (filename.linampl) spectra are written to ASCII files in directory c:\sky\work\[today’s date], and the concatenated cycles are saved to filename.concat, which is displayed in a new channel. The program also returns the slope of the line fit to those harmonics, the value of r associated with the regression, and the number of harmonics used in the analysis.

Spectral smoothing

This algorithm calculates the long-term average spectrum for a source time series by averaging together pitch-synchronous FFT spectra across the entire sample, and then calculates the relative amounts of energy in the resultant average spectrum in two frequency bands selected by the user. Each cycle is padded with zeros for a FFT of 1024 points. Analyses can be done on logarithmic or linear scales, with or without preemphasis. Run the pitch tracker first to mark cycle boundaries. This is not a pitch synchronous analysis.

PSP

The parabolic spectral parameter (PSP) is calculated as described by Alku et al. (1997) from the spectrum of a single glottal cycle. The algorithm fits a parabola to the spectrum of a single glottal flow pulse and then calculates the steepness of that parabola along with the correlation between the empirical spectrum and the parabola. The input to the PSP is a single glottal cycle whose beginning and end points must have the same amplitude value. The following procedure ensures that this is the case when the input is a flow derivative pulse. First, concatenate the differentiated waveform with itself 20 times. Demean the result, and then integrate the demeaned waveform. (Be sure you have the correct channel active for each step of this analysis.) Select one cycle towards the end of the file, mark it with cursors, zoom in as necessary to ensure precision, and finally Cut and replace (see Edit menu) the selected cycle in the same channel. Save the new pulse data in case of crashes, and then apply the command Analysis-Spectral measures-PSP. The program returns the value of the PSP and the correlation between the fitted parabola and the spectral data.

Upsample

Upsamples the signal in the active channel by a multiple of its sampling rate by zero stuffing and lowpass filtering with a Kaiser window FIR (Proakis & Manolakis, 1988). This upsample is a linear phase process. The upsampled signal opens in a new channel.

Unicycle

Separate a bicyclic (period doubled) waveform with A and B cycles into two signals, one with the A cycles, another with the B cycles. Run the Pitch command to mark the cycles as explained for bicyclic signals. The program places results in two channels with the A and B cycles, respectively.

Edit Menu

Undo delete

Restores the last channel closed.

Concatenate

Concatenates a segment between the cursors in the active channel. User specifies the number of repetitions to be concatenated. The new file opens in a new channel with the name filename.concat.

Concatenate channels

Concatenates the displayed portions of up to 8 files in the order specified by the user. Open the files you want to concatenate, then select Edit-Concatenate channels. This opens a dialog box. Enter the number of the channel to be placed first, second, etc., in the appropriate box and click ok. The concatenated files will be opened in a new channel as a new file with the name of the first file in the series and the extension .conc. If you want a period of silence inserted between files in the sequence, pad each file with zeros with the command Edit-Waveform editing-Pad with 'value' before concatenating.

Cut and replace

Cuts the displayed portion of the file in the active channel from the file, and replaces it in as many as 6 channels, effectively deleting the undisplayed portions of the channel. To use, mark the desired portion of the signal in the active channel with cursors, and zoom in. Select Cut and replace and specify the number of copies of the segment you want. If one copy is selected, the zoomed-in portion of the active channel will be segmented from the original waveform and replaced in the original channel with name filename1. Additional copies, if requested, are opened in new channels with names filename2, filename3…filename6.

Generate waveform

Sinewave

Generates a sine wave. User specifies the length of the signal in points, the amplitude, the sampling rate, the frequency, and the phase of the signal in degrees. The resulting signal is placed in a new channel with name Sfreq (for example, a 100 Hz sine wave would be named S100). The file has no extension.

Hyperbole

Generates the curve k/(x-a), where k is a constant. User specifies k, a, the length of the waveform in points, and the sampling rate. The result appears in a new channel with name Hk (for example, if k=10, the file is named H10). No extension is applied.

Impulse

Generates an impulse. The user specifies the total length of the file in points, the location of the impulse within the file (also in points), the amplitude of the impulse, and the sampling rate. The resulting signal is placed in a new channel with the name “impulse”.

LF

Opens the file filenameNLF (generated by the synthesizer; an ASCII file containing the parameters of an LF source), plots the waveform in a new channel with name filename.lf0, and calculates T_a and R_a (e.g., Fant, 1995). Results are printed in a new window. The NLF file must be on the desktop.

Flip

Flips (time inverts) the file segment displayed in the active window. Opens the reversed signal in a new window with the name filename.rev.

Linear stretching

Stretches the waveform to a new length keeping the shape and the sampling rate. Mark the segment to be stretched with the cursors.

Remove

Deletes the segment between cursors from the file, and reopens the edited file in a new channel with the name filename.ed. Select the segment to be removed from the file by left clicking (to set left cursor) and right clicking (to set right cursor).

Waveform editing

Change value

Changes the y value at the left cursor position.

Enlarge

Assumes a periodic waveform. It adds the first cycle at the beginning of the waveform, and the last cycle at the end. If there is no F0 information in the channel, it looks for it in the *.par file created by the synthesizer. To place F0 info in the channel use Display-F0.

Line

Replaces the data between the cursors with a straight line.

Pad with “value”

Adds a user-specified number of points with a user-specified value at the left cursor position.

Zero stuff

It “stuffs” N zeros in between samples. This is an step to increase the sampling rate by N+1, and is also a form of interpolation (Proakis & Manolakis, 1988, p. 657).

Right shift

This command moves the waveform to the right by adding N points at the beginning. The user can choose the value to be zeros, the maximum or minimum of the waveform, or the value at index 0.

Left shift

This command moves the waveform to the left by adding N values at the end. The user can choose the value to be zeros, the maximum or minimum of the waveform, or the value at index 0.

Batch Menu

Downsample

Bulk downsamples files, with optional simultaneous file format conversion, amplitude normalization, and signal inversion. Options are shown in Figure 66. The program reads a list of the files to be downsampled from a text file that must be in the same directory as the files to be downsampled. Downsampled files are saved by default in the directory c:\sky\work\downsample, with their original filenames and extension .aud (for UCLA’s AUD format), .wav, or .asc (for ASCII files), depending on the output format requested. The downsample command is executed when the Select files button is clicked, so be sure that you are happy with the other options before proceeding.

File format conversion

Bulk converts file formats, without changing the sample rate. Signals may simultaneously be inverted and/or normalized in amplitude. This routine reads the names of the files to be converted from a text file that must be in the same directory as the files being converted. Converted files are saved by default in the directory c:\sky\work\fileformat, with their original filenames and extension .aud, .wav, or .asc. The file format command is executed when the select files button is clicked, so be sure that you are happy with the other options before proceeding.

H1 graph

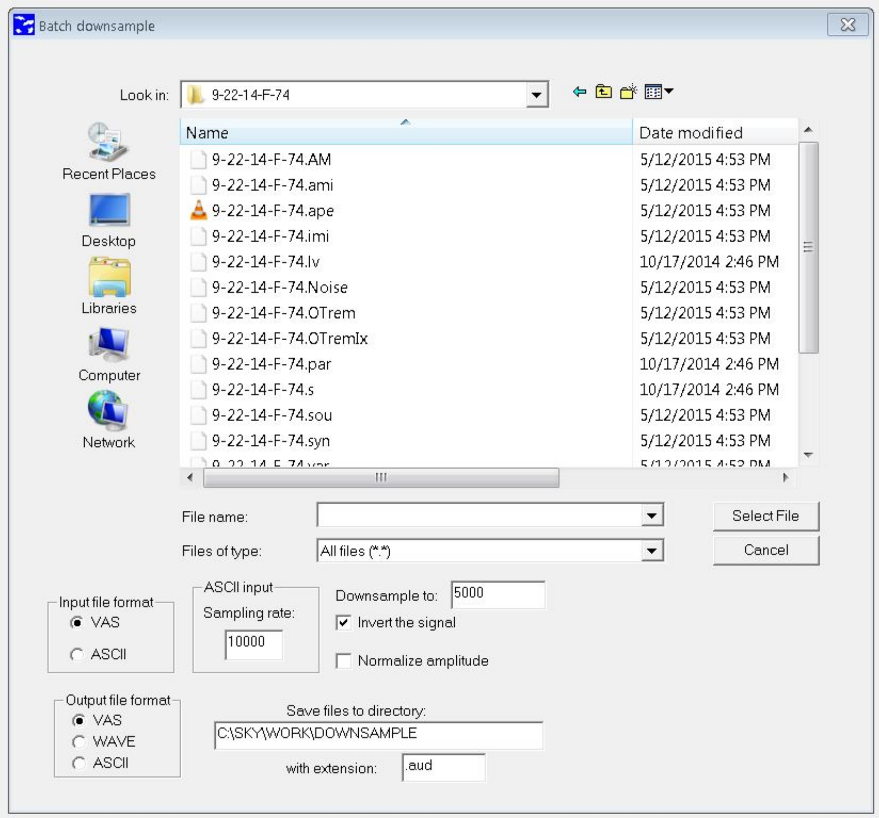
A plot of the H1 value as a function of the Fourier transform length. Concatenate several cycles, and mark F0. Start with a Transform of N cycles and finish with a transform of N+1 cycles.

H1-H2 graph

A plot of the H1-H2 value as a function of the Fourier transform length. Proceed as above.

Juntos

Concatenates the files listed in an ASCII file with extension .lst. (The word “juntos” means “together” in Spanish.) Optionally normalizes amplitudes, inserts a specified amount of silence between files, and/or applies onset and offset ramps of specified duration to each file prior to concatenation. The .lst file and the files to be concatenated should be in the same directory. A new file containing the concatenated files is saved in directory c:\sky\work\juntos, with the name juntos.voi.



Look in: 9-22-14-F-74

Name	Date modified
9-22-14-F-74.AM	5/12/2015 4:53 PM
9-22-14-F-74.ami	5/12/2015 4:53 PM
9-22-14-F-74.ape	5/12/2015 4:53 PM
9-22-14-F-74.imi	5/12/2015 4:53 PM
9-22-14-F-74.lv	10/17/2014 2:46 PM
9-22-14-F-74.Noise	5/12/2015 4:53 PM
9-22-14-F-74.OTrem	5/12/2015 4:53 PM
9-22-14-F-74.OTremIx	5/12/2015 4:53 PM
9-22-14-F-74.par	10/17/2014 2:46 PM
9-22-14-F-74.s	10/17/2014 2:46 PM
9-22-14-F-74.sou	5/12/2015 4:53 PM
9-22-14-F-74.syn	5/12/2015 4:53 PM

File name:

Files of type: All files (*.*)

Input file format
● VAS
○ ASCII

ASCII input
Sampling rate: 10000

Downsample to: 5000
☑ Invert the signal
□ Normalize amplitude

Output file format
● VAS
○ WAVE
○ ASCII

Save files to directory: C:\SKY\WORK\DOWNSAMPLE
with extension: .aud

Figure 66. Dialog box for batch downsampling.

Spectral slope

Reads a list of files from an ASCII list (with extension .lst). Files must be in the UCLA .aud format; convert if necessary prior to using this batch routine. Each file in the list is assumed to be one cycle in length; this cycle is then concatenated the number of times specified by the user, after which an FFT (dB or linear scale, as specified by the user) is calculated. A regression line is fit to the peaks of the harmonics in the FFT, and the slope of the line is written to a file c:\sky\work\spectralslope\regression.asc. A second file, c:\sky\work\spectralslope\filename.dbampl, is also created for each file in the list. This file contains the frequency and amplitude of each harmonic in the FFT spectrum.

References

Alku, P., Strik, H., and Vilkman, E. (**1997**). "Parabolic spectral parameter: A new method for quantification of the glottal flow," Speech Communication **22**, 67-79.

Alku, P., and Vilkman, E. (**1995**). “Effects of bandwidth on glottal airflow waveforms estimated by inverse filtering,” Journal of the Acoustical Society of America **98**, 763-767.

Alku, P., Vilkman, E., and Laukkanen, A.M. (**1998**). "Estimation of amplitude features of the glottal flow by inverse filtering speech pressure signals," Speech Communication **24**, 123-132.

Ananthapadmanabha, T.V. (**1984**). "Acoustic analysis of voice source dynamics," Speech Transmission Laboratory Quarterly Status and Progress Report **2-3**, 1-24.

Baken, R.J. (**1987**). *Clinical Measurement of Speech and Voice* (College Hill, Boston).

Brigham, E.O. (**1988**). *The Fast Fourier Transform and Its Applications* (New York, Prentice-Hall).

Childers, D.G., Hicks, D.M., Moore, G.P., Eskenazi, L., and Lalwani, A.L. (**1990**). "Electroglottography and vocal fold physiology," Journal of Speech and Hearing Research **33**, 245-254.

Childers, D.G. and Krishnamurthy, A.K. (**1985**). "A critical review of electroglottography," CRC Critical Reviews of Biomedical Engineering **12**, 131-161.

Childers, D.G. and Lee, C.K. (**1991**). "Vocal quality factors: Analysis, synthesis, and perception," Journal of the Acoustical Society of America **90**, 2394-2410.

Childers, D.G., Naik, J.M., Larar, J.N., Krishnamurthy, A.K., and Moore, G.P. (**1983**). "Electroglottography, speech, and ultra-high speed cinematography," in *Vocal Fold Physiology: Biomechanics, Acoustics and Phonatory Control*, edited by I.R.Titze and R.C.Scherer (Denver Center for the Performing Arts, Denver), pp. 202-220.

Childers, D.G. and Wong, C.-F. (**1994**). "Measuring and modeling vocal source-tract interaction," IEEE Trans Biomed Eng **41**, 663-671.

Cranen, B. and Schroeter, J. (**1996**). "Physiologically motivated modelling of the voice source in articulatory analysis/synthesis," Speech Communication **19**, 1-19.

de Krom, G. (**1993**). "A cepstrum-based technique for determining a harmonics-to-noise ratio in speech signals," Journal of Speech and Hearing Research **36**, 254-266.

Fant, G. (**1960**). *Acoustic Theory of Speech Production* (Mouton, The Hague).

Fant, G. (**1979**). “Glottal source and excitation analysis,” Speech Transmission Laboratory Quarterly Status and Progress Report **1**, 85-107.

Fant, G. (**1995**). "The LF model revisited. Transformations and frequency domain analysis," Speech Transmission Laboratory Quarterly Status and Progress Report **2-3/95**, 119-156.

Fant, G., Hertegard, S., Kruckenberg, A., and Liljencrants, J. (**1997)**. "Covariation of subglottal pressure, F0 and glottal parameters," Proc. Eurospeech **97**, 453-456.

Fant, G., Liljencrants, J., and Lin, Q. (**1985**). "A four-parameter model of glottal flow," Speech Transmission Laboratory Quarterly Status and Progress Report **4**, 1-13.

Fant, G. and Lin, Q. (**1988**). "Frequency domain interpretation and derivation of glottal flow parameters," Speech Transmission Laboratory Quarterly Status and Progress Report **2-3**, 1-21.

Fujisaki, H. and Ljungqvist, M. (**1986**). "Proposal and evaluation of models for the glottal source waveform," Proceedings of the IEEE International Conference on Acoustical Speech Signal Processing, 1605-1608.

Garellek, M., Keating, P., Esposito, C. M., and Kreiman, J. (**2013**). "Voice quality and tone identification in White Hmong," J Acoust Soc Am **133**, 1078-1089.

Gauffin, J. and Sundberg, J. (**1989**). "Spectral correlates of glottal voice source waveform characteristics," Journal of Speech and Hearing Research **32**, 556-565.

Gerratt, B.R., and Kreiman, J. (**1995**). "Utility of acoustic measures of voice," in D. Wong (ed.), *Proceedings of the Workshop on Standardization in Acoustic Voice Analysis* (National Center for Voice and Speech, Denver), pp. GER1 - GER7.

Gobl, C. (**1988**). "Voice source dynamics in connected speech," Speech Transmission Laboratory Quarterly Status and Progress Report **1**, 123-159.

Hamming, R. W. (**1983**). *Digital Filters* (Prentice Hall, New York).

Hanson, H.M. (**1997**). "Glottal characteristics of female speakers: Acoustic correlates," Journal of the Acoustical Society of America **101**, 466-481.

Hillenbrand, J. (**1987**). "A methodological study of perturbation and additive noise in synthetically generated voice signals," Journal of Speech and Hearing Research **30**, 448-461.

Hillenbrand, J. (**1988**). "Perception of aperiodicities in synthetically generated voices," Journal of the Acoustical Society of America **83**, 2361-2371.

Hillenbrand, J., Getty, L.A., Clark, M.J., and Wheeler, K. (**1995**). "Acoustic characteristics of American English vowels," Journal of the Acoustical Society of America **97**, 3099-3111.

Imaizumi, S., Kiritani, S., and Saito, S. (**1991**). "Perceptual evaluation of a glottal source model for voice quality control," in *Vocal Fold Physiology: Acoustic, Perceptual, and Physiological Aspects of Voice Mechanisms*, edited by J.Gauffin and B.Hammarberg (Singular, San Diego), pp. 225-232.

Iseli, M., and Alwan, A. (**2004**). "An improved correction formula for the estimation of harmonic magnitudes and its application to open quotient estimation," in Proc. ICASSP, pp. 669-672, Montreal, Canada.

Ishizaka, K. and Flanagan, J.L. (**1972**). "Synthesis of voiced sounds from a two-mass model of the vocal cords," Bell System Technical Journal **51**, 1233-1268.

Jackson, M., Ladefoged, P., Huffman, M.K., and Antonanzas-Barroso, N. (**1985**). "Measures of spectral tilt," UCLA Working Papers in Phonetics **61**, 72-78.

Javkin, H., Antoñanzas-Barroso, N., and Maddieson, I. (**1987**). "Digital inverse filtering for linguistic research," Journal of Speech and Hearing Research **30**, 122-129.

Karlsson, I. (**1991**). "Female voices in speech synthesis," Journal of Phonetics **19**, 111-120.

Klatt, D.H. (**1980**). "Software for a cascade/parallel formant synthesizer," Journal of the Acoustical Society of America **67**, 971-975.

Kreiman, J., Gabelman, B., and Gerratt, B.R. (**2003**). "Perception of vocal tremor," Journal of Speech, Language, and Hearing Research **46**, 203-214.

Kreiman, J., Garellek, M., Gerratt, B.R., Chen, G., and Alwan, A. (**2015**). "Perceptual evaluation of voice source models," Journal of the Acoustical Society of America **138**, 1-10.

Kreiman, J. and Gerratt, B.R. (**2005**). "Perception of aperiodicity in pathological voice," Journal of the Acoustical Society of America **117**, 2201-2211.

Kreiman, J., and Gerratt, B.R. (**2012**). "Perceptual interactions of the harmonic source and noise in voice," Journal of the Acoustical Society of America **131**, 492-500.

Kreiman, J., Gerratt, B.R., and Antoñanzas-Barroso, N. (**2007**). "Measures of the glottal source spectrum," Journal of Speech, Language, and Hearing Research **50**, 595-610.

Lalwani, A.L. and Childers, D.G. (**1991**). "Modeling vocal disorders via formant synthesis," Proceedings of the IEEE , 505-508.

Lofqvist, A. and Manderson, B. (**1987**). "Long-time average spectrum of speech and voice analysis," Folia Phoniatrica **39**, 221-229.

Michaelis, D., Gramss, T., and Strube, H.W. (**1997**). "Glottal-to-noise excitation ratio - A new measure for describing pathological voices," Acustica **83**, 700-706.

Ní Chasaide, A., and Gobl, C. (**2010**). "Voice source variation and its communicative functions," in *The Handbook of Phonetic Sciences*, edited by W.J. Hardcastle, J. Laver, & F. Gibbon, (second edition, Blackwell, Oxford), pp. 378-423.

Qi, Y. (**1992**). "Time normalization in voice analysis," Journal of the Acoustical Society of America **92** , 2569-2576.

Qi, Y. and Bi, N. (**1994**). "A simplified approximation of the four-parameter LF model of voice source," Journal of the Acoustical Society of America **96**, 1182-1185.

Qi, Y. and Hillman, R.E. (**1997**). "Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals," Journal of the Acoustical Society of America **102**, 537-543.

Press, W.H., Teulosky, S.A., Vetterling, W.T., and Flannery, B. P. (**2005**). *Numerical Recipes in C++. The Art of Scientific Computing*, second edition (Cambridge University Press, Cambridge).

Proakis, J. G., Manolakis D. G. (**1988**). *Introduction to Digital Signal Processing* (Macmillian, New York).

Ramirez, R.W. (**1985**). *The FFT: Fundamentals and Concepts* (New York, Prentice-Hall).

Rosenberg, A.E. (**1971**). "Effect of glottal pulse shape on quality of natural vowels," Journal of the Acoustical Society of America **49**, 583-590.

Rothenberg, M. (**1973**). "A new inverse-filtering technique for deriving the glottal air flow waveform during voicing," Journal of the Acoustical Society of America **53**, 1632-1645.

Rothenberg, M. (**1977**). "Measurement of airflow in speech," Journal of Speech and Hearing Research **20**, 155-176.

Shrivastav, R., and Camacho, A. (**2010**). "A computational model to predict changes in breathiness resulting from variations in aspiration noise level," J Voice **24**, 395-405.

Shrivastav, R., and Sapienza, C. (**2006**). "Some difference limens for the perception of breathiness," Journal of the Acoustical Society of America **120**, 416-423.

Signorello, R., Rhee, N., Antoñanzas-Barroso, N., Gerratt, B.R., and Kreiman, J. (2017). "A psychoacoustic model of spectral noise in the voice source," manuscript in preparation.

Strik, H. (**1998**). "Automatic parameterization of differentiated glottal flow: Comparing methods by means of synthetic flow pulses," Journal of the Acoustical Society of America **103**, 2659-2669.

Sundberg, J. and Gauffin, J. (**1979**). "Waveform and spectrum of the glottal voice source," in *Frontiers of Speech Communication Research: Festschrift for Gunnar Fant*, edited by B.Lindblom and S.Ohman (Academic Press, London), pp. 301-320.

Titze, I.R. (**1995**). *Workshop on Acoustic Voice Analysis Summary Statement* (Denver: National Center for Voice and Speech).

Yanagihara, N. (**1967**). "Significance of harmonic changes and noise components in hoarseness," Journal of Speech and Hearing Research **10**, 531-541.

Yumoto, E., Gould, W.J., and Baer, T. (**1982**). "Harmonics-to-noise ratio as an index of the degree of hoarseness," Journal of the Acoustical Society of America **71**, 1544-1550

Appendix

Keyboard Shortcuts and Commands for Synthesis Software

ACTION	**COMMAND**
Vocal tract	
Add a resonance	Shift + single left click
Delete a resonance	Shift + single right click
Move a resonance	Double left click and drag
Lock a resonance in place	Double right click
Source	
Define a left segment boundary	Shift + single left click
Define a right segment boundary	Shift + single right click
Modify spectral slope	Double left click near point to be moved and drag
Lock spectral slope	Double right click

Reset spectral slope from left end	Enter desired slope in Spectral Slope cell, <ctrl> + single left click
Reset spectral slope from right end	Enter desired slope in Spectral Slope cell, <ctrl> + single right click
“Snap to” (smooths harmonic amplitudes within a selected segment of the spectrum)	Point cursor near the center of the selected segment; hold down right mouse button; while still holding down right button, single left click and release both mouse buttons.
LF Model	
Mark an event for the pitch tracker	Single left click
Move an LF parameter	Double left click and drag
Lock an LF parameter in place	Double right click
Noise Manipulation	
Select vertex to move	<ctrl> <shift> single left click
Set new vertex position	<ctrl> <shift> single right click
Set left cursor	<ctrl> single left click

Set right cursor	<ctrl> single right click
Set beginning of segment	<shift> single left click
Set end of segment	<shift> single right click
Change segment slope or amplitude by moving the beginning, end, or middle	Double left click
Lock in the new position for a segment	Double right click

Keyboard Shortcuts and Commands for Inverse Filter Software

ACTION	**COMMAND**
Select left segment or cycle boundary	Single left click
Select right segment or cycle boundary	Single right click
Delete a resonance	Double right click
Add a resonance with BW=100	Double left click

Move a resonance	Single left click and drag
Lock a resonance in place	Single right click

Keyboard Shortcuts and Commands for Sky Software

ACTION	**COMMAND**
Activate a new channel	Single left click anywhere in a non-active channel
Play a channel	Double left click near the right edge of a channel
Close a channel	Double right click near the right edge of a channel
Select left segment or cycle boundary	Single left click within the channel frame
Select right segment or cycle boundary	Single right click within the channel frame
Place an event mark	Use Menu to set mark symbol, double left click
Delete an event mark	Double right click