



**HAL**  
open science

## The WinSnoori user's manual version 1.32

Yves Laprie

► **To cite this version:**

| Yves Laprie. The WinSnoori user's manual version 1.32. 2002. inria-00107630

**HAL Id: inria-00107630**

**<https://inria.hal.science/inria-00107630>**

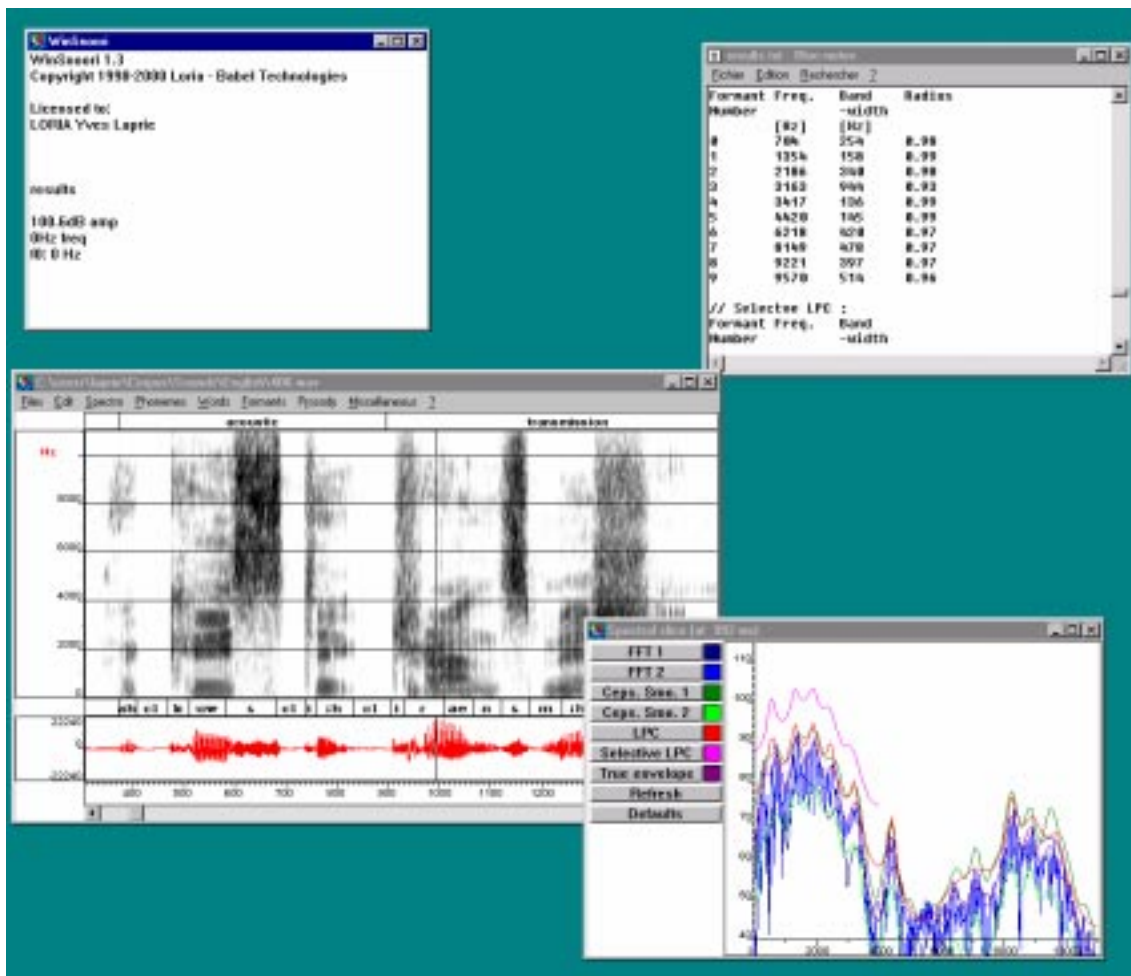
Submitted on 19 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# The WinSnoori User's Manual version 1.32





## **Forward**

This manual together with the on-line help gives a complete description of how to use WinSnoori. The information presented in the on-line help is not covered in full detail in the present manual. Thus, it is the manual and the on-line help together which constitute the complete WinSnoori user documentation.

The breakdown of the documentation between the above two documents is as follows:

- The on-line help contains all the necessary information for its use along with a complete description of all the WinSnoori commands (the menu and the mouse) accompanied by a full glossary of WinSnoori terms.
- The present document covers the rest of documentation, and in particular the necessary generalities for understanding the functions of the WinSnoori software.

The term Windows as used in the present manual, denotes the Microsoft system of multiple windows. The manual itself was elaborated using Windows standard tools. In particular:

- The Print-Screen key for screen and window images.
- Paintbrush for retouches to screen and window images.

As a WinSnoori user, you can also make use of WinSnoori to elaborate your own documents and incorporate screen images as you proceed.

## Contents

1	Starting.....	6
1.1	Hardware configuration .....	6
1.2	Installation.....	6
1.3	Starting WinSnoori .....	6
2	Presentation.....	6
2.1	Generalities .....	6
2.2	The screen .....	7
2.3	Phonetic and orthographic labels .....	9
2.4	File formats .....	9
2.5	The editor .....	13
3	Using WinSnoori .....	14
3.1	The on-line help .....	14
3.2	The mouse.....	14
3.3	The commands of the main menu.....	14
3.3.1	<b>File Menu</b> .....	14
3.3.2	<b>Edit Menu</b> .....	15
3.3.3	<b>Spectro Menu</b> .....	15
3.3.4	<b>Phoneme Menu</b> .....	16
3.3.5	<b>Word Menu</b> .....	16
3.3.6	<b>Formant Menu</b> .....	16
3.3.7	<b>Prosody Menu</b> .....	16
3.3.8	<b>Misc. Menu</b> .....	17
3.4	Advanced editing tools .....	17
3.5	Spectral slicing.....	19
3.6	Formants F1 F2 and F3 .....	20
3.7	Detection of peaks in cepstrally smoothed spectra .....	21
3.8	LPC root detection .....	22
3.9	The graphical interface of the Klatt synthesizer .....	23
3.10	Decorating the spectrogram .....	25
3.11	Prosody .....	26
3.11.1	<b>F0 contour modifications</b> .....	26
3.11.2	<b>Time scale modifications</b> .....	27
3.11.3	<b>Pitch marks</b> .....	27
3.11.4	<b>Energy</b> .....	27
4	Miscellaneous .....	27
4.1	Activities .....	27
4.2	Algorithms used.....	28
4.3	The journal and results.....	29
4.4	Multilingual WinSnoori.....	30
4.5	The phonetic alphabet .....	30
4.6	Write file protection.....	30
4.7	Reading and writing speech and label files.....	31
4.8	The WinSnoori environment .....	32
4.9	General options .....	32
4.10	Play back.....	33

4.11	Windows and graphical objects .....	33
4.12	Printing with WinSnoori.....	35
5	Main changes between version 1.3 and version 1.32.....	35

# 1 Starting

## 1.1 Hardware configuration

WinSnoori can run on a PC multimedia machine having the following configuration:

- Processor: Pentium and above,
- Screen: 256 color VGA screen (the strict minimum),
- Memory space: 128 Mbytes at least,
- Sound interface: a multimedia compatible,
- Operating system: Windows 95, Windows 98, Windows NT, Windows 2000 or Windows XP.

The term PC multimedia denotes a standard PC compatible machine which runs on Windows and is equipped with Windows compatible sound card. WinSnoori necessitates a PC multimedia compatible machine and runs on all multimedia PC machines.

## 1.2 Installation

Simply click on the setup.exe icon. WinSnoori is installed in "Program Files\Wsno" and appears in the program menu.

## 1.3 Starting WinSnoori

WinSnoori is launched under Windows via the program menu. You can use WinSnoori in a DOS window with "wsno.exe filename" or use it from an html page.

For more detail on the different ways of launching WinSnoori, refer to the Windows user manual, more specifically to the following items:

- Applications, the paragraph on Launching an application.
- Programme manager, the paragraph on Creating programs.

# 2 Presentation

## 2.1 Generalities

The goal of WinSnoori is to **edit** and **visualize speech files of different formats**.

**Speech files** in the largest sense, since the files can be labeled, i.e., to each speech file can be associated a phonetic labeling (made up of the phonemes of the speech) and an orthographic labeling (made up of words of the speech utterance). The labels can be in the same file as the speech itself or in separate files according to chosen format.

**Different formats:** WinSnoori recognizes around fifteen different speech formats, phonetic label file formats, orthographic labels file formats among which WINDOWS .WAV format.

**Visualizing** with WinSnoori: WinSnoori displays speech files on the screen together with the associated phonetic and orthographic labels

**Editing** with WinSnoori: WinSnoori incorporates standard editing functions of the type Cut/Copy/Paste again in connection with **visualizing**: WinSnoori calculates and displays on the screen a variety of information:

- spectrogram (fft, lpc, ...)
- formants in the F1/F2 plane
- detection of spectral maxima through cepstral peaks or LPC roots
- the pitch

## 2.2 The screen

At the start up, WinSnoori displays the welcome window bearing the “WinSnoori” and the **main window** in which all the speech spectral diagrams will be displayed.

The first page of the manual presents a typical WinSnoori screen showing the welcome window bearing the title “WinSnoori”, and the main window containing the speech signal, the associated spectrogram and phonetic and orthographic labels, as well as the menu.

When WinSnoori is running, the welcome window displays the following information on the screen:

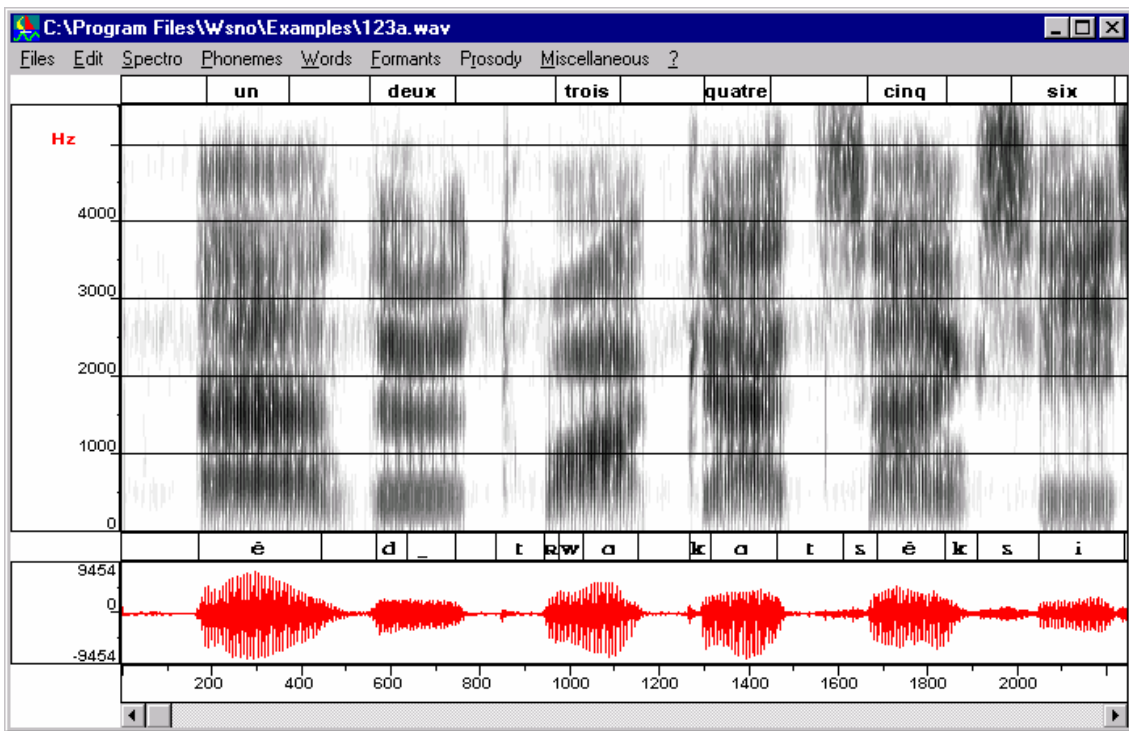
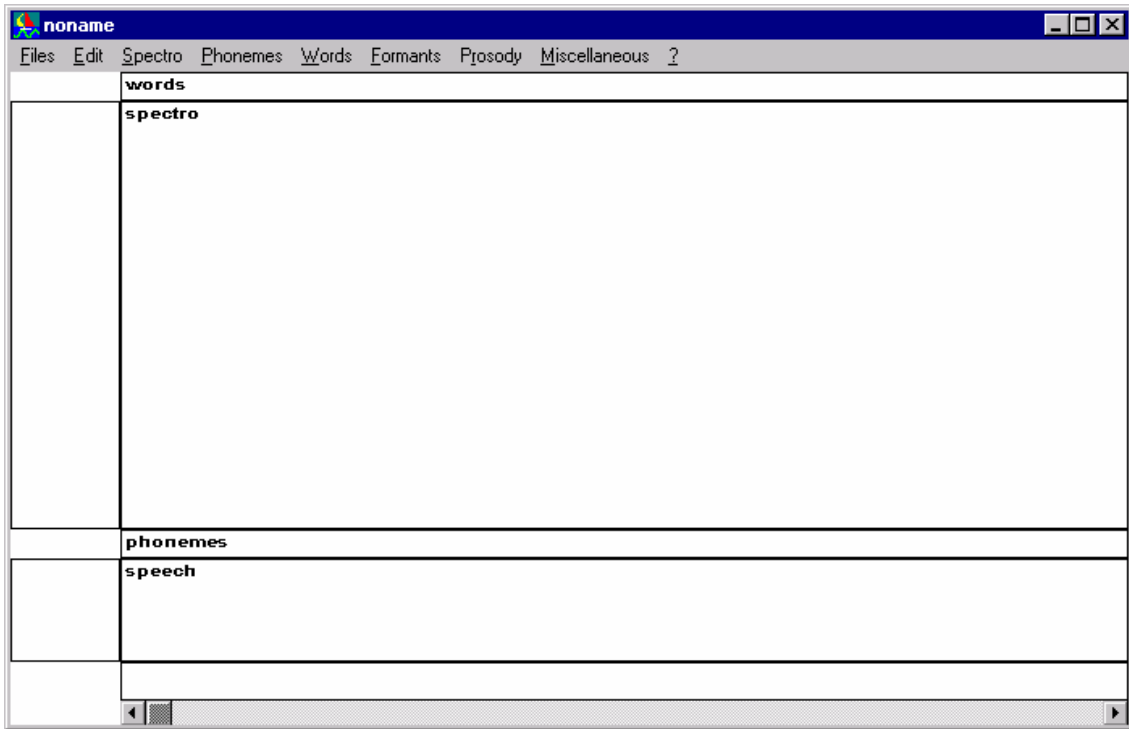
- activity information for lengthy processing, essentially indications about the portion of the task hitherto accomplished (in %).
- statistics about screen displayed items selected by the mouse cursor, e.g., time and frequency for spectrograms.
- an indication of the presence of information in the result file signaled by displaying the message “results” inside the welcome window.

The WinSnoori **main window** is made up of a menu, 4 windows or subwindows and a scroll bar:

- the menu presents the different WinSnoori commands to the user,
- the **speech window** displays the speech signal on a scale graduated in milliseconds,
- the **phoneme window** presents the phonetic labels,
- the **word window** presents the orthographic labels,
- the **spectrogram window** displays the spectrogram,
- the scroll bar allows right and leftward movements and thus facilitates browsing or skimming through the speech file.

Two illustrations are presented below. The first presents how the WinSnoori **main window** looks like at the start up and the second presents a file editing session:





Attention is drawn to the following points:

- the duration of the signal displayed is at most 4 seconds. So, only a portion of long files can be visualized at a given instant. Zooming facilities are provided for visualizing portions of the speech signal shorter than 4 seconds in which case we speak of “zoom in” mode.

- under zoom in, the speech signal is displayed on the speech window to maximum amplitude. On the other hand, when the zoom is off, the signal is displayed in raw amplitude as presented in the speech file without any correction.
- a mark on the “**File•zoom in**” command of the menu indicates that the zoom in mode is on while a mark on the “**File•zoom out**” command of the menu shows that the zoom in mode is off.
- signal frequencies are indicated in the spectrogram window in the form of horizontal lines every 1000 Hz. And as the mouse cursor moves over the signal, the time and the frequency of the cursor position are displayed in the welcome window.
- the speech window, the phonemes window, the words window just like the scroll bar, all have a fix height while the size of the spectrogram window is variable, increasing or decreasing in keeping with the size of the main window as it is modified.
- each time the size of the main window is modified:
  - the 4 windows comprising the speech, the spectrogram, the phoneme and the words windows, are all re-drawn,
  - the duration of the signal displayed is not changed,
  - the spectrogram is simply dilated or contracted without being re-calculated.

Other windows appear during the running of WinSnoori. These are described in the sequel along with the associated commands.

## 2.3 Phonetic and orthographic labels

### Definitions

The *phonetic label* contains the speech file phonemes and is displayed in the *phoneme window*.

The *orthographic label* contains the words of the speech file. It is displayed in the *word window*.

These labels are made up of texts and associated boundaries. The texts specify the phonemes and words while the boundaries specify the start and the end of phonemes or words as the case may be.

The phoneme and the word windows incorporate an editor with facilities to:

- add, move or delete boundaries,
- add, move or delete text.

This editor is operated through the mouse. Refer to the on-line help for complete detail.

## 2.4 File formats

### Generalities

The goal of WinSnoori is to edit and visualize speech files. These files are of different formats, basically of the two types:

**format .WAV**

## **format non.WAV**

The essential features are:

- in a .WAV file, a single file contains the speech signal, the phonetic and the orthographic labels as well as sample format indications such as the sampling frequency and the size of samples expressed in bits.
- in a non.WAV format file , the foregoing information are separated, and the speech signal is defined by:
  - a speech file,
  - a phonetic label file,
  - an orthographic label file,
  - format of the speech file,
  - format of the phonetic label file,
  - format of the orthographic label file.

These last three format files are shared by all the speech files of a same format. It is up to the user to specify all the six speech signal components mentioned above since there is no super-set which assembles all the six elements.

### **Complementary features of the .WAV format**

The name of the file is of the form \*.WAV.

It is both Windows and MPC (Multimedia PC) compatible. WinSnoori knows how to read all .WAV one-channel, 16 bits files whatever the signal frequency. WinSnoori generates files which can be re-read by other .WAV format processing software subject to the following remarks:

- it is possible the frequency of the file generated is not accepted by the other software, particularly when the non.WAV file is the format of the initial file converted to a .WAV file.
- phonetic and orthographic labeling information is lost when other software modifies the .WAV file generated by WinSnoori.

### **Complementary features of the format non.WAV**

The extension to the speech file, to the phonetic and orthographic label files is different from .WAV representing the discriminatory characteristics from which a file is judged as being of the .WAV format type or not.

The above three files have the same name but the directory and the extension could be different, e.g.:

```
c:\Program Files\Wsno\Examples\jb102  speech file with an empty extension
c:\Program Files\Wsno\Examples\jb102.man  phonetic label file
c:\ Program Files\Wsno\Examples\jb102.key  orthographic label file
```

Each non WAV speech format is described in a file (\*.fmt in the directory C:\Program Files\Wsno\format\), and the user can add his own format through a .fmt file (see further down in the sequel).

Under WinSnoori, the formats of non WAV phonetic label files cannot be modified, therefore the user must choose the format among those defined.

The formats of non WAV orthographic label files are defined in WinSnoori and cannot be modified.

### A summary of the different acceptable speech formats

- for the .WAV format
  - any frequency
  - one-channel
  - 16 bits
- for the non.WAV format
  - all two byte sampled speech files
  - with a fixed length file header (the header is not analyzed)
  - and with the speech signal occupying the entire file except the header space.

### The non.WAV speech file formats

All the acceptable formats are individually described in a \*.fmt file specifying:

- the frequency
- the number of bits per sample: up to 16 bits with every sample occupying 2 bytes in all the cases
- the address of the first byte of sound in the file (expressed in bytes)
- the length of the signal (in bytes). This is in fact always equal to:  
*sound-file-length - address-of-1st-byte*
- swapping indications about the two byte components of a sample
  - “swap”: the lower byte first and the higher byte last
  - “normal”: the higher byte first and the lower byte last.

The following table summarizes the different formats recognized by WinSnoori:

	the frequency	the number of bits/sample	the address of the first sound byte in the file	the length of the sound	swap
CDROM	16000	16	1024	SOF-1024	swap
CRIN	16000	12	2	SOF-2	normal
EUROMO	16000	16	0	SOF	swap
GOUPIL	16000	16	0	SOF	swap
JCJ	10000	16	256	SOF-256	normal
LABISE	16000	16	2	SOF-2	normal
PUR12	16000	12	0	SOF	normal
PUR16	16000	16	0	SOF	normal
SONY	18900	16	2048	SOF-2048	normal
TEXAS	20000	16	1024	SOF-1024	swap

Remarks:

- SOF denotes: sound file length
- corresponding formats are in a .fmt file of the same name  
e.g.: CRIN is in .\format\CRIN.fmt

### **The different non.WAV formats for phonetic labels files**

Three different formats exist: CRIN, NIST, SAM.

For further details, see the illustrative examples supplied with WinSnoori.

### **The different non.WAV formats for orthographic labels files**

Four different formats exist: VECSYS, NIST, SAM, NOISEX

For further details, see the illustrative examples supplied with WinSnoori.

### **Remarks**

File read and write formats are independent, i.e., when opening a file, the user must specify its format, and when writing a file, the write format (which may be different from the read format) must also be specified. In other words, WinSnoori knows how to convert a file. WinSnoori can thus read a file under given format and save the same file under a different format and that with speech files of any sort, including phonetic and orthographic label files.

When reading a non. WAV speech file, WinSnoori ignores the file header (the portion preceding the first sound byte). When WinSnoori writes a file, it is the current or active file which supplies the file header. It must be noted that the header that is recorded is not perfect (since it is simply recopied and not generated). It may even be altogether wrong in cases where the format of the written file is different from that of the active file.

The c:\Program Files\W sno\Examples\ directory contains all the examples for speech, phoneme and word files embracing all possible formats. Read the file \Examples\readme.txt giving the list of files and their formats.

The .WA\$ file extension is WinSnoori reserved and as a result, the user cannot use this extension for files created by himself. WinSnoori interprets such files as being Windows .WAV format speech files (irrespective of the Windows WAV files option). WinSnoori needs a number of such files for the clipboard and for searching directories for phonemes or words (scrhphon.wa\$ and srchword.wa\$).

## 2.5 The editor

### Lay-out

The editor allows the visualization and modification of a file being edited. It has its own clipboard and often works on a portion of the speech defined by the user and visualized on the screen in an inverted video form referred to as “highlighted region”. It incorporates the following functions:

- Displacements (by using the scroll bar or by going to a specific time)
- Moving about (using the speech window scroll bar)
- Edition functions (in association with the clipboard): *cut, copy, paste, erase.*
- File editing functions (file to file operations): *copy to file, paste file.*

All the preceding functions edit at the same time the speech signal itself and the associated phonetic and orthographic labels. In other words, whenever a speech signal is cut, the associated phonetic and orthographic labels are also cut automatically.

The editor is able to handle very long files (up to 1 gigabyte of speech signal, 32000 phonetic labels and 32000 orthographic labels). It is however only able to visualize a maximum of 4 second slots of signal at a time. It can also skim through the file and edit or modify any portion not exceeding 4 seconds in duration.

The editor comprises advanced tools for editing speech signals:

- Left attenuation,
- Right attenuation
- Central attenuation
- Addition of noise
- FIR Filtering over a rectangle time x frequency region
- OLA Filtering over an arbitrary number of time x frequency trajectories

### The buffer

The editor knows how to process large size speech files (from several megabytes up to the gigabyte) but is able to display no more than 4 second speech segments. Before being displayed, the signal is temporarily stored in a 8 second **buffer memory**. The length of the buffer is thus bigger than the maximum signal segment that can be displayed.

When a speech file is being analyzed, the signal is divided into three parts:

- the buffer portion
- the portion speech file awaiting to transit through the buffer (this portion can be empty in which case the beginning of the buffer coincides with the beginning of the file itself).
- the portion speech file having already transited the buffer. This portion can also be empty.

Immediately after loading the speech file, the buffer is the copy of the corresponding portion of the file. During editing the buffer changes however and can differ from the

corresponding portion of the file (e.g., after cut/paste operations). WinSnoori handles this and when necessary, recopies the buffer into its location in the speech file (e.g., when one skims through the file with the speech window scroll bar). Before every copy, a panel prompts the user for confirmation.

As regards labels, every label file is loaded as a whole into memory at the moment of reading.

### **3 Using WinSnoori**

This is fully described in the WinSnoori on-line help. The presentation here is thus limited only to general explanations. The reader is therefore advised to consult the on-line help for full details.

#### **3.1 The on-line help**

The WinSnoori on-line help is activated through “**?•help**” menu command which presents all the commands in detail and explains the way to use the mouse.

#### **3.2 The mouse**

The mouse, in addition to selecting menu commands, is used to do a variety of things depending on the command and the windows opened:

- it measures the point under the cursor when it is moved around graphical windows such as the spectrogram and the formants F1/F2 windows. As it is moved about, the co-ordinates of the cursor position are displayed in the welcome window.
- the mouse is used in selecting a text or a boundary point in the phonetic or orthographic window for modification.
- the mouse performs particular actions like drawing or selecting a formant in the graphical interface of the Klatt synthesizer.

In all cases, only the left button is used, sometimes in association with the “CAPS (shift)” key and/or the “CONTROL (ctrl)” key.

#### **3.3 The commands of the main menu**

Preliminary remarks:

- A number of commands work on speech segments delimited by a highlighted region of the screen. If no such region appears on the screen, the command applies to the entire signal displayed.
- The editor clipboard is WinSnoori specific and is thus related to the Windows clipboard in no way whatsoever.
- The user is advised to refer to the on-line help for more details. Presentation below is limited to the list of WinSnoori commands.

The commands of the main menu are as follows :

##### **3.3.1 File Menu**

**File**

**New**

- Open**
- Save**
- Save as**
- Record**
- Copy to file**
- Paste from file**
- Register**
- Quit**

### **3.3.2 Edit Menu**

**Edit**

- Cancel**
- Cut**
- Copy**
- Paste**
- Delete**
- Editing tools**
- Play**
- Play many**
- Zoom in**
- Zoom out**
- Go to**
- Resampling**
- Frequency**

### **3.3.3 Spectro Menu**

**Spectro**

- Wide band**
- Narrow band**
- LPC**
- Cepstral smoothing**
- Draft**
- Smoothed draft**
- Clear**
- Keep decoration**
- Remove decoration**
- Options**
  - Upper disp. frequency**
  - Wide band**
  - Narrow band**
  - LPC**
  - Cepstral smoothing**
  - Rough draft**
  - Smoothed draft**
  - Defaults**



### **3.3.4 Phoneme Menu**

#### **Phonemes**

- Read**
- Save as**
- Search for a phoneme**
- Explore directory**
- Mouse help**

### **3.3.5 Word Menu**

#### **Word**

- Read**
- Save as**
- Search for a word**
- Explore directory**
- Mouse help**

### **3.3.6 Formant Menu**

#### **Formants**

- Formants F1 F2 F3**
- Cepstral peak display**
- LPC root display**
- Spectral slice**
- Edit formants**
- F0**
  - Male speaker**
  - Female speaker**
  - Any**
- Options**

### **3.3.7 Prosody Menu**

#### **Prosody**

- Type of speaker**
- F0 for the screen**
- FO for the whole file**
- F0 for a corpus**
- Display F0 scale**
- Open a pitch file**
- Display pitch values**
- FO contour modifications**
  - Import the F0 contour**
    - From the current file**
    - From a pitch file**
  - Display the F0 contour**
    - To see it**
    - To modify it**
- Save the F0 contour**
- Re-synthesize waveform**

**Mouse help**  
**Time scale modifications**  
    **Parameters**  
    **Modify**  
**Energy**  
    **Energy options**  
    **Energy for the whole file**  
    **Display energy**  
**Marks**  
    **Pitch marker**  
    **Display marks**

### **3.3.8 Misc. Menu**

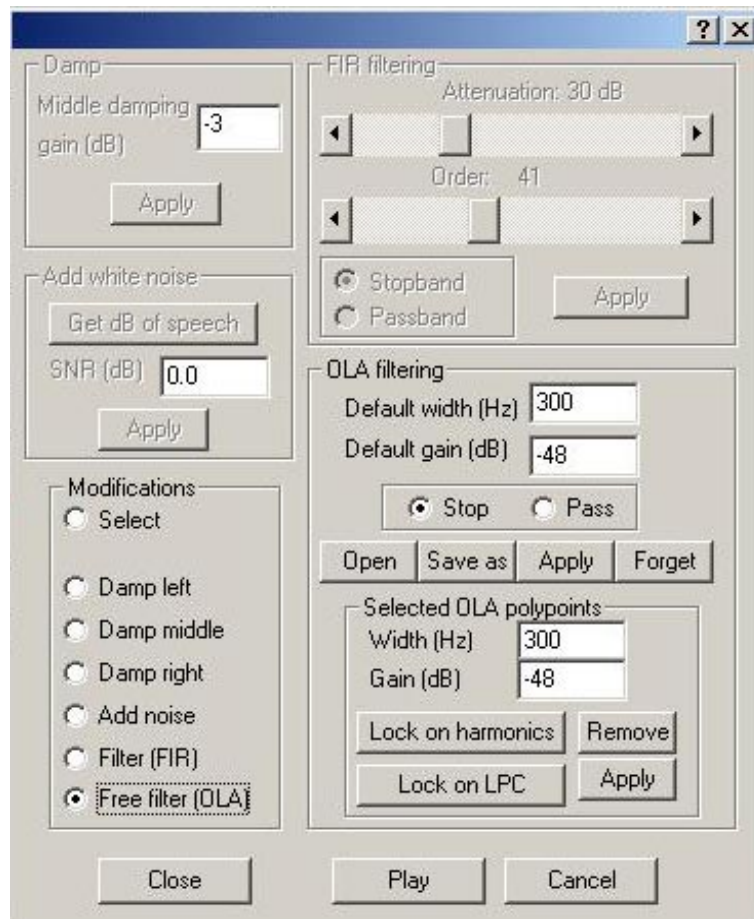
**Misc.**

**Colors**  
    **General options**  
    **Shows results**  
    **Add results to log**  
    **My Functions**  
        **My function 1**  
        **My function 2**  
    **?**  
    **Help**  
    **On Help**  
    **About**

## **3.4 Advanced editing tools**

The command “**Edition•Advanced editing tools**” allows tools specifically designed for speech editing to be used:

- Damping (left, middle or right),
- Adding noise,
- FIR filtering of a time x frequency region,
- OLA filtering of an arbitrary number of time x frequency trajectories.



**Damping (left, middle or right)** enables the attenuation the left, the middle or the right portion of the highlighted region.

1. The attenuation is said to be “**left**” when the multiplying factor starts at 0 at the left end and increases linearly to 1 towards the right end of the highlighted region.
2. The attenuation is said to be “**middle**” when the speech signal of the highlighted region is multiplied by a coefficient the gain (in dB) of which is to be specified in the panel. The coefficient decreases, or increases, progressively from 1 to the user specified value over a portion extending over 20 milliseconds but not exceeding the third of the region.
3. This attenuation is said to be “**right**” when the attenuation profile is the symmetrical of that of the left attenuation (the multiplying factor starts at 1 at the left end and decreases linearly to 0 towards the right end of the highlighted region).

**FIR filtering** enables the definition of a FIR filter applied to a given time x frequency region. Order, attenuation (in dB) and type of filter (stop or pass band) can be chosen. Once these parameters are set, the user can draw a rectangle with the pointer to select the time x frequency region where to apply the filtering. Order represents the length of the filter and is expressed in terms of number of samples. Filter applied is the FIR filter while the order represents the number of filtering coefficients. The attenuation reflects the sharpness of the filtering and is expressed in dB.

**Adding noise** enables noise to be added to a signal. Before setting the Signal to Noise Ratio (SNR) the user has to compute the level of the speech signal. If a signal region is selected Snoori computes the level of speech over this region. Otherwise, the level is computed over the signal in the memory of Snoori. Note that this preliminary calculation has a strong influence on the noise that is added to signal. If the signal is zero it receives a default level of 60 dB. Note that the default spectrogram calculation uses pre-emphasis and therefore gives the feeling that high frequencies are more affected than low frequencies.

**OLA filtering** enables arbitrary time x frequency trajectories to be filtered by the overlap and add method. User has to draw trajectories. Then, trajectories can be edited, saved, read and applied to a signal. For each of these trajectories, bandwidth and gain can be specified independently.

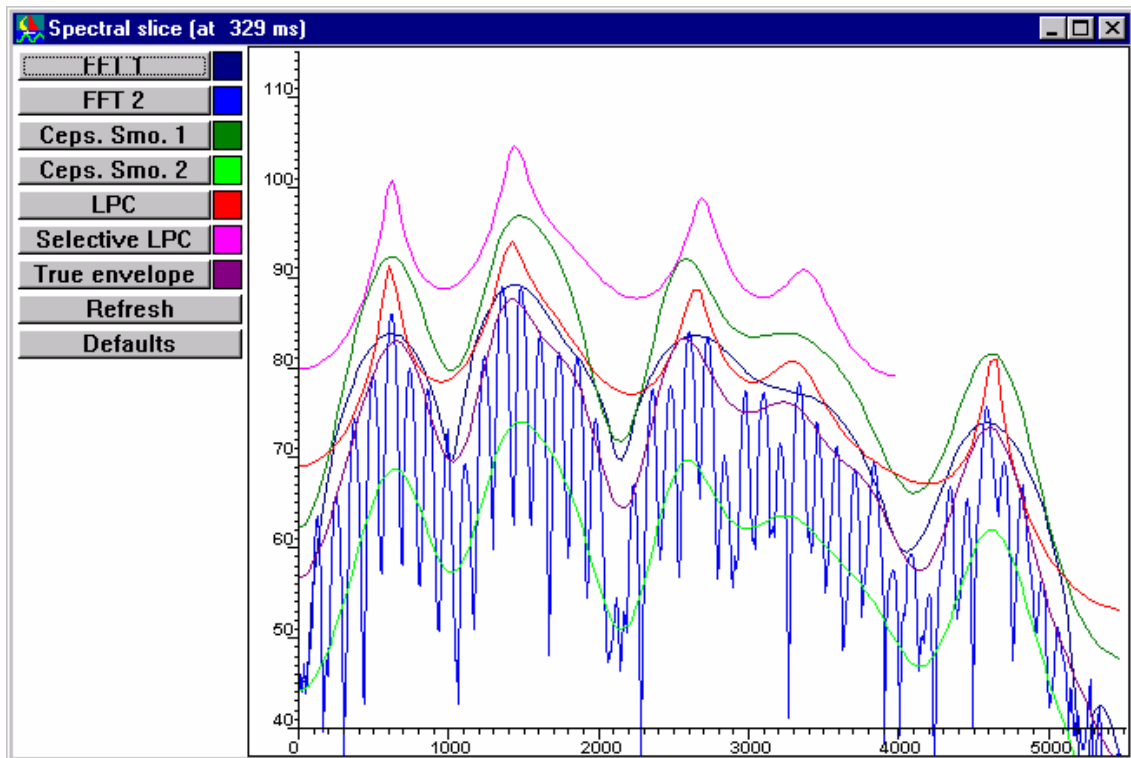
The OLA overlap is 75%, FFT order is 1024 and the signal window (32 ms) is zero padded.

OLA filtering in Snoori was designed with a view of lowering or raising formants or harmonics levels. Therefore, the two commands **“lock on LPC”** and **“lock on harmonics”** can be used to move a trajectory towards the nearest line of LPC roots or narrow band spectrum peaks.

### 3.5 Spectral slicing

This function displays different types of spectra of a point in the speech file. It is activated through the **“Formants•Spectral slice”** command of the main menu which leads to the opening of a spectral slice window comprising:

- a title bar giving the time at which the speech file spectrum is calculated.
- a daughter window in which the calculated spectra are displayed. There are 7 types of spectra:
  - 2 FFT spectra distinguished by the options governing their calculation,
  - 2 cepstrally smoothed spectra also distinguished by the options governing their calculation,
  - 1 LPC spectrum,
  - 1 selective LPC spectrum,
  - 1 true envelope.
- 7 gray tone buttons through which the respective options governing the calculation of the 7 spectra can be modified. To access the options, it suffices to click on the corresponding button.
- 7 colored buttons showing the color of each spectrum and indicating whether the spectrum is displayed or not (displayed if colored, white if not). Clicking on a button activates or deactivates the display of the corresponding spectrum.
- a “refresh” button for re-calculating the spectrum for the same point (indicated on the title bar).
- a “defaults” button for resetting the options to their WinSnoori booting values.



Once the spectral slice window is opened, one only needs to drag the mouse on the appropriate point of the spectrogram for the spectra to be calculated and displayed. The coordinates of the cursor position in the slice is displayed in the welcome window as the mouse is moved over the spectra.

The gray button corresponding to the name of a spectral analysis allows the user to modify the calculation parameters through a window containing:

- a default button for resetting the options to their WinSnoori booting values.
- a CheckBox type button for enabling or forbidding pre-emphasis
- a window length parameter which specifies the portion of the speech signal for the spectral calculation.
- in some cases, parameters specifically governing the spectral calculation.

### 3.6 Formants F1 F2 and F3

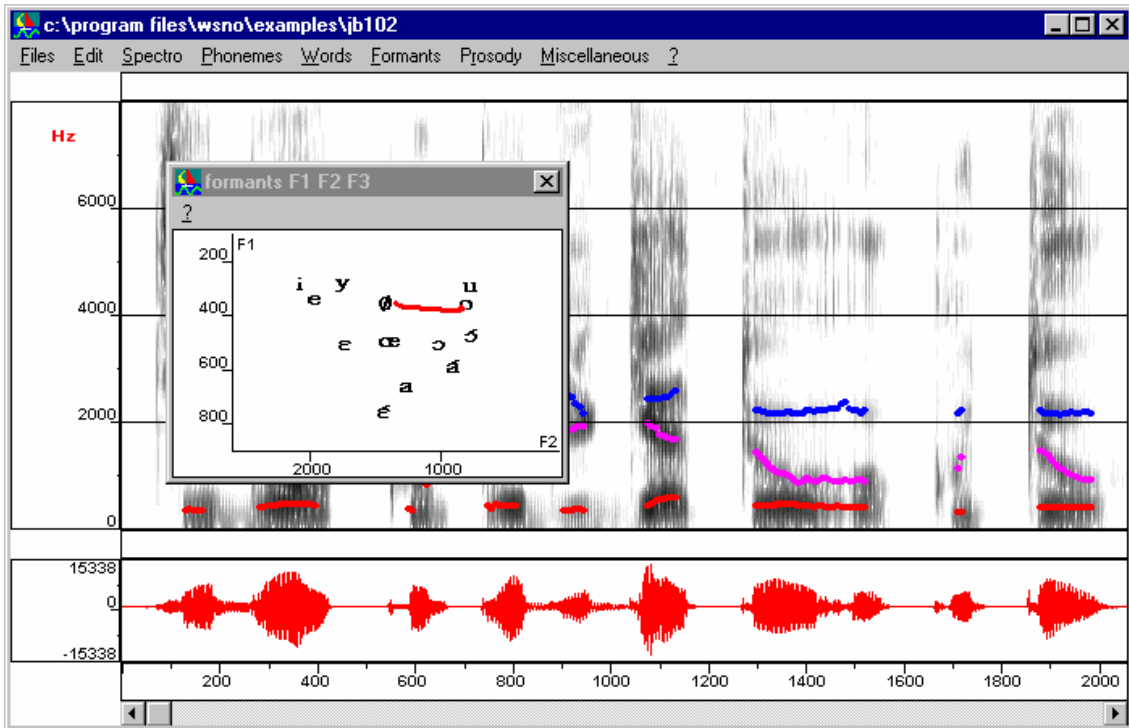
The command **“Formants•Formants F1 F2 F3”** of the main menu calculates formants. To derive the three formants F1 F2 and F3, WinSnoori applies the McCandless algorithm (see references in the sequel) to the spectrogram window highlighted region and then displays the results on the spectrogram. The associated F1/F2 curve is at same time, plotted the in the **“F1 F2 F3 formats”** window opened for the purpose. The highlighted region can be specified either before or after triggering the command.

The example below shows:

- the highlighted region on which the spectral calculation was performed.
- the three formants F1, F2, F3 (respective values around 450 Hz, 900 Hz and 2000 Hz) superimposed on the spectrogram at the “au” of “thibault”. The

spectrogram calculation was limited only to a portion of the signal in order to allow the formants to stand out more clearly.

- the “F1 F2 F3 formants” window with the F1/F2 curve.

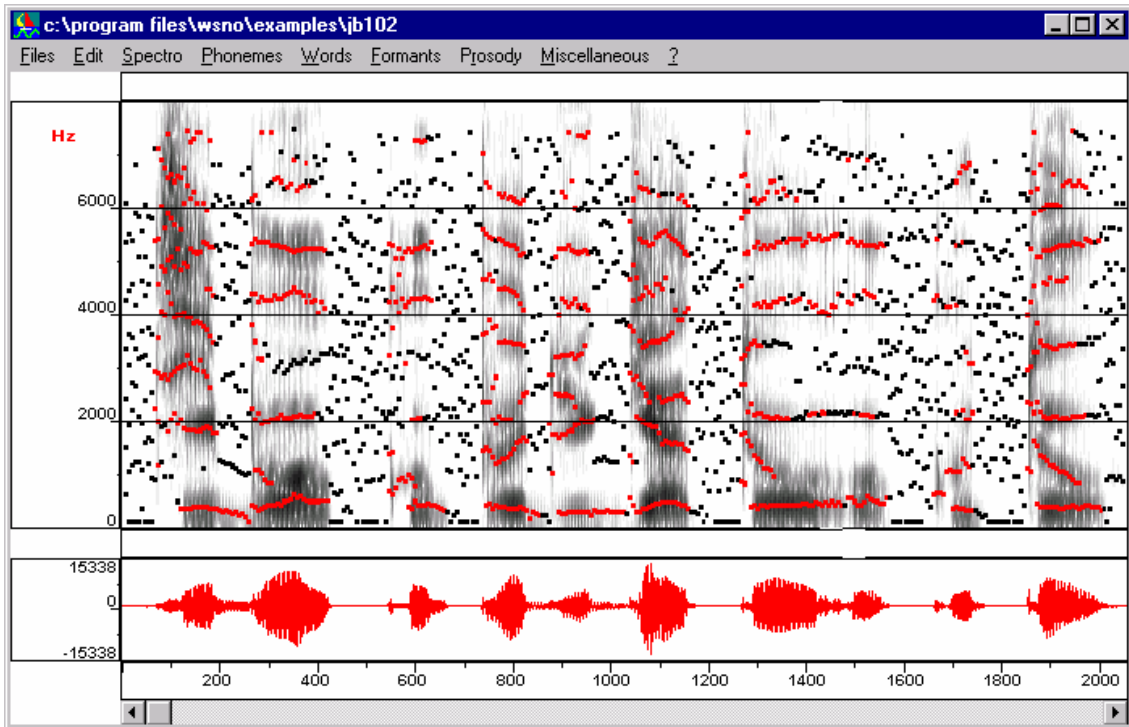


The active window contains only the on-line mouse help which mouse is used here for:

- specifying the highlighted region of the spectrogram window whose F1, F2, F3 formants are to be calculated. The calculation effectively begins with the release of the mouse.
- computing the F1 and F2 of the mouse cursor position in the F1 F2 F3 Formants window.

### 3.7 Detection of peaks in cepstrally smoothed spectra

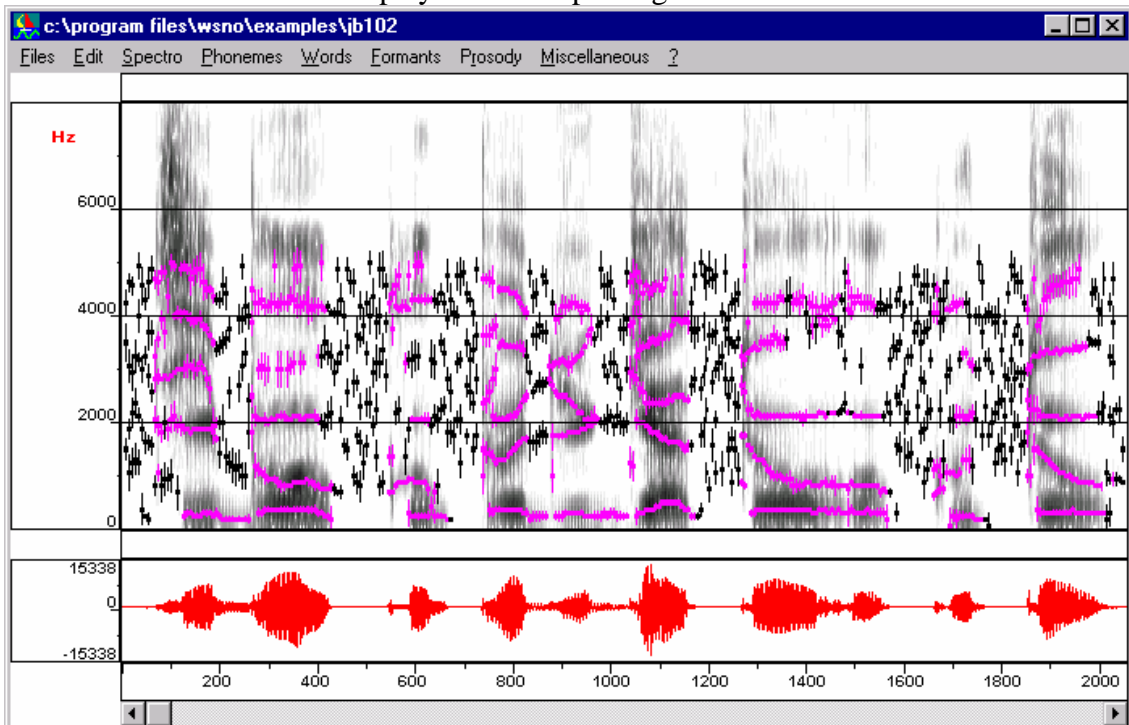
The command “**Formants•Cepstral peak display**” of the main menu calculates the spectral maxima cepstrally. The calculations are performed on highlighted regions of the spectrogram window and the resulting peaks displayed on the spectrogram.



Peaks are displayed in red except when the energy level is too low in which case they are shown in black.

### 3.8 LPC root detection

The command “**Formants•Detect LPC roots**” of the main menu calculates the spectral maxima by LPC. The calculation is performed on the highlighted region of the window and the roots obtained are displayed on the spectrogram.



The LPC roots are shown in dark yellow except when the energy level is too low in which case they are shown in black. The vertical bar gives the bandwidth (broader for large bandwidths, in other words, when the associated peak is little pronounced).

### 3.9 The graphical interface of the Klatt synthesizer

This interface allows users to display and edit Klatt parameters. Most of this interface consists of editing facilities and tools to produce a synthetic copy of original speech. *The Klatt synthesizer itself is not incorporated into WinSnoori but generates speech signals from files of Klatt parameters edited in WinSnoori.* The Klatt synthesizer is that written by Jon Iles and and Nick Ing-Simmons (see C:\Program Files\W sno\Klatt\klatt.htm for a detailed description of this free Klatt synthesizer).

Two modes for editing Klatt parameters have to be distinguished:

- **A speech signal does exist before the Klatt synthesizer interface is launched.** This corresponds to “copy synthesis”, i.e. constructing Klatt parameters which produce a speech signal close to the original. In this case the main window (temporal signal and spectrogram) displays the original signal and the interface window (called “Klatt synthesizer graphical interface”) displays the spectrogram of the synthesized signal as soon it does exist.
- **There does not exist any speech signal before launching the interface.** This simply corresponds to editing a file of Klatt parameters or to creating a new file of parameters from scratch. In this case the main window as well as the interface window display the spectrogram of the synthesized signal if it does exist. If there is a speech file already loaded into WinSnoori, the user has to create a new empty speech file (“new” in the file menu) to edit a file of Klatt parameter without the displaying of the spectrogram of the already loaded file.

Besides editing facilities and more specific tools, users can examine the synthesized signal by zooming on or playing some signal regions, and more importantly by displaying spectral slices by moving the pointer in either the main window or the interface window. This last possibility enables the comparison of original and synthetic spectra and can guide users to improve Klatt parameters.

The overall task of users consists of drawing parameters trajectories with the mouse (Shift + left button pressed). Users can edit trajectories by clicking (Ctrl + left button) on a trajectory to be corrected, and then by moving points (Ctrl + left button pressed). In order to avoid discontinuities which can lead to synthesis “accidents” users should merge (“merge” in the Edit menu of the Klatt window) neighboring trajectories corresponding to the same parameter.

Parameters are organized according their dimension and values :

- **Frequency (from 0 to half the sampling frequency) (Hz):** formant frequencies (F1..F6, nasal zero and pole)
- **Bandwidth for the cascade branch (Hz):** F1..F6 and nasal zero
- **Bandwidth for the parallel branch (Hz):** F1..F6 and nasal pole



- **Amplitude for the cascade branch and source parameters (dB):** Amplitude of voicing for the cascade branch, Aspiration, Turbulence, Spectral tilt, Amplitude of frication, Amplitude of bypass frication , Overall gain.
- **Amplitude of the parallel branch (db):** F0..F6 and nasal pole.
- **Frequency of F0**
- **Numerals:** Open quotient of voicing waveform, Spectral skew

This organization is intended to reduce the number of trajectories displayed simultaneously on the spectrogram. “Scale” menu in the Klatt window allows the dimension to be selected.

The default scale is that of formant frequencies (from 0 to half the sampling frequency). This offers a “naive” use of the Klatt synthesizer since only formant frequencies are visible. However, switching to another scale allows users to edit all the Klatt parameters. In order to reduce the number of trajectories displayed, only those different from default values are displayed in the form of trajectories. “Default parameters” in the “Setup” menu of the Klatt window allows these values to be modified.

There are two cases for editing formant parameters:

- Frequency is a major parameter. That means that it is possible to create new trajectories by drawing onto the spectrogram when the scale is Frequency of F0 frequency.
- On the other hand when the scale selected is not the frequency (Bandwidth for instance) only corrections are allowed. Trajectories of parameters specifying the source (frication, tilt, skewness...) can be created when the scale is not Frequency because they do not contain any frequency information.

Note that the Klatt synthesizer is completely independent from WinSnoori. That means that no “semantic” control is performed before synthesizing a waveform. Therefore users are highly recommended to read carefully the help of the Klatt synthesizer and the papers of Dennis Klatt about his synthesizer (C:\Program Files\Wno\Klatt\klatt.exe):

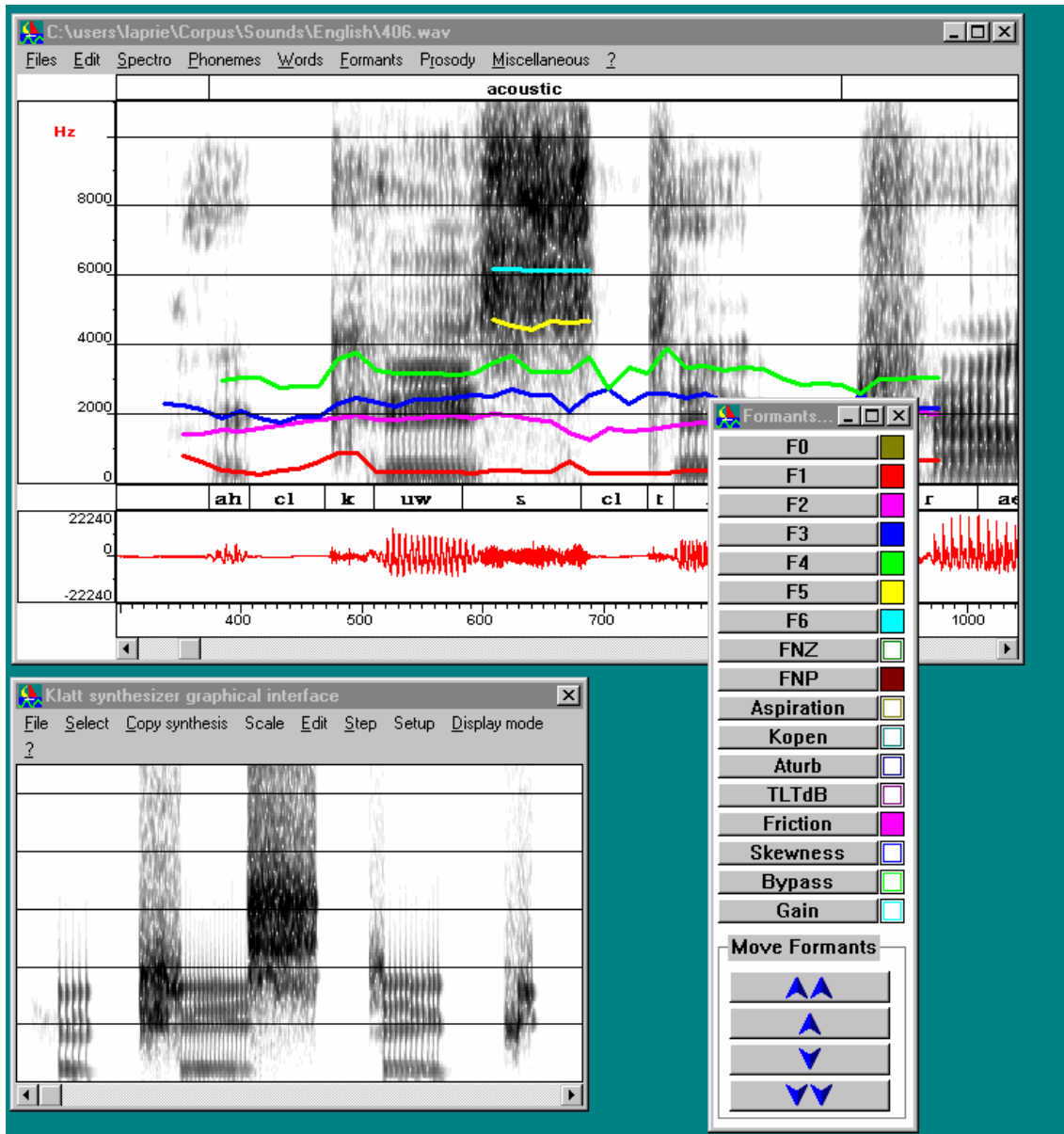
- **Klatt,D.H.** Software for a cascade/parallel formant synthesizer, in the *Journal of the Acoustic Society of America*, pages 971-995, volume 67, number 3, March 1980.
- **Klatt,D.H. and Klatt L.C.** Analysis, synthesis, and perception of voice quality variations among female and male talkers, in the *Journal of the Acoustic Society of America*, pages 820-857, volume 87, number 2, February 1990.

The current synthesizer is the GPL version developed by Jon Iles and Nick Ing-Simmons (You can download the windows version at <http://www.loria.fr/equipes/parole/Html/klatt.html>).

When it happens that parameters are not correct, synthesis fails, stops and thus generates only a partial waveform, possibly empty if the failure happens at the beginning. A floating point exception can happen if the Klatt parameters lead to huge

values of synthetic samples. In these cases the spectrogram is partially or completely white because the signal does not exist.

The example below shows how the graphical interface of the Klatt synthesizer can be used for rough copy synthesis. The original spectrogram is in the main window, the synthetic waveform is in the window of the Klatt synthesizer. The vertical window allows the status of parameters to be chosen (hidden, inactive or active) and trajectories to be move upwards or downwards.



### 3.10 Decorating the spectrogram

Formants, spectral peaks and other spectral maxima that the “Formant” menu adds to the spectrogram constitute the latter's decorating items and are simply referred to as “decorations” for short. WinSnoori possesses a copy of the spectrogram window,

named “**spectroBis**” which facilitates the saving of decorations. Here is how spectroBis window is treated:

- a copy of every spectrogram diagram or curve is drawn both in the spectrogram and spectroBis window.
- the diagrams of decorations are drawn only in the spectrogram window.
- the “**Spectro•keep decoration**” menu command copies the spectrogram window into the spectroBis window.
- the “**Spectro•clear decorations**” menu command copies the spectroBis window into the spectrogram window (the symmetrical operation to the preceding).
- all the “Edition” menu commands which modify the screen implicitly preserves decorations (i. e., involves an operation which is identical to the “**Spectro•keep decoration**” command).

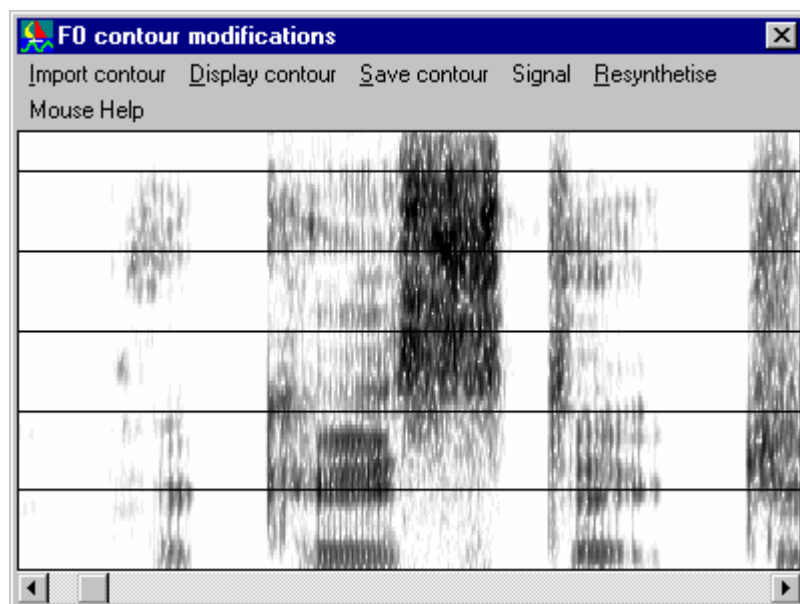
It is to be noted that the Klatt interface works on non decorated spectrograms since it recopies the spectroBis window into the spectrogram window before drawing its formants. In order to draw formant trajectories easily it can be useful to compute “LPC roots” on a selected region and then to “Keep decoration” so that LPC roots are displayed when the Klatt interface is open.

### 3.11 Prosody

F0 is the voicing frequency. It is referred to as the fundamental frequency or more simply as the fundamental or pitch for short. Commands in the “**Prosody**” menu allows you to study and to change prosody. F0 is calculated by the frequency comb method. The mouse cursor displays the value of F0 in the welcome window as the mouse is moved around the spectrogram window.

#### 3.11.1 F0 contour modifications

The F0 contour modification windows allows waveforms corresponding to modified F0 contours to be synthesized by means of TD-PSOLA.



Once the user has imported a F0 contour (by reading a pitch file or by calculating F0 for the current speech file) the user can edit this contour and synthesize the corresponding waveform. The standard scenario is thus :

1. Import the F0 contour,
2. Display the contour to modify it,
3. Modify the contour with the mouse. The original contour is displayed in the form of red points and allows the comparison between original and new contours. The new contour is drawn by dragging the left button of the mouse while maintaining the “CAPS/shift” key of the keyboard pressed.
4. Resynthesize the waveform to listen to the modified prosody. WinSnoori launches an audio player each time a new waveform is generated. If no audio player is launched correct the wsno.ini file in the Windows directory (see C:\Program Files\Wsno\Presentation\WinSno.htm).

### **3.11.2 Time scale modifications**

Unlike F0 contour modifications which operate locally, these modifications modify the signal globally regarding F0 level and /or speech rate. Parameters greater than one correspond to a decreasing of the speech rate or the F0 level. Modifications are carried out either by PSOLA or by phase vocoder. In the latter case only time modifications are possible.

### **3.11.3 Pitch marks**

F0 contour modifications as well as time scale modifications rely on the PSOLA method and thus require the knowledge of pitch marks which allow the signal to be decomposed into overlapping windows. WinSnoori incorporates a pitch marking algorithm which is called during FO and speech rate modifications but the user can launch this algorithm to have an idea of the marks used. These marks are put in the file result file.

### **3.11.4 Energy**

The command “**Prosody/Energy/Display energy**” calculates the energy in dB after Fourier Transform. The user can set several parameters used during energy calculation. The most important parameters are the minimal and maximal energy levels which define the range of energy which is taken into account. Outside these two thresholds the energy is not considered.

## **4 Miscellaneous**

### **4.1 Activities**

Activities are functions informing the user on the sequence of past or current events in the course of a long processing. These are indicated through information displayed in the welcome window or inside temporary windows opened for the purpose. Activities are of three kinds:

- the “percent” activities which indicate what percentage of the process underway is so far terminated. A window opens for lengthy treatments to give percentage of the sequence of events and at times the name of the current stage of the treatment.

- the “results” message indicates that a command has put its results in the result file for eventual visualization through the “**Miscellany•show results**” command of the main menu.
- “measurement” activities:
  - When the mouse is moved about the graphic window, the coordinates of the cursor position are displayed. In the spectrogram window for instance, these coordinates are simply the instant and the frequency of the cursor position.
  - Moreover, as the mouse is moved about the spectrogram window, the F0 value is displayed if it has already been calculated.
  - When a highlighted region (either of the spectrogram or speech window) is being drawn its duration is displayed in milliseconds under form  $1/freq$ . This facilitates the measurement of the fundamental frequency with the mouse. All that is in fact needed is to zoom on a one period voicing interval and to draw the corresponding highlighted region.  $freq$  then gives the value of the fundamental.

WinSnoori uses the first line of the welcome window to communicate information about the current treatment to the user (speech signal, the names of the files read or written...). The information presented, in addition to being fugitive, is scantily documented.

## 4.2 Algorithms used

Before presenting the different algorithms used, it is worth drawing attention to the following points:

- every calculation uses the **signal window**. This gives the length of the signal used in the calculation.
- all spectrum calculations (FFTs and LPCs) apply the **Hamming Window** on the signal window.
- exceptions aside, the signal is always **pre-emphasized** before the application of the Hamming window to it. The Hamming window is applied not the signal  $s[i]$  itself but to the function  $s[i]-s[i-1]$  where  $s[i]$  is the  $i$ th sample of the signal  $s$ .

### FFT

This is no other than the well-known Fast Fourier Transform.

### Cepstral smoothing

The spectra are cepstrally smoothed according to the Rabiner and Schafer method as described in “**Digital processing of speech signals**” (Prentice-Hall 1978, p. 355).

The liftering window given by `begin_liftering` and `end_liftering` is a user modifiable parameter. The liftering windows is applied on the cepstrum. The aim of cepstral smoothing is to suppress the contribution of voicing emanating from the portion right to `end_liftering` in the pseudo-temporal domain and to keep only the vocal tract contribution from the portion left to `begin_liftering`.

### LPC

This is no other than the classical Linear Predictive Coding by auto-correlation.

## **Selective LPC**

LPC spectral analysis is applied to a portion of the spectrum as J. D. K. Markel & A. H. Gray describe it in their book **“Linear Prediction of Speech”** (Springer-Verlag Berlin Heidelberg New-York 1976, pp 146-151). The typical band selected for doing so is the frequency band 0Hz-4000Hz (or 0Hz-5000Hz) containing the significant vocal information.

### **4.2.1.1 True envelope**

Cepstral smoothing is applied in an iterative way so to minimize the difference between the FFT spectrum and the cepstrally smoothed spectrum as P. Halle describe it in his paper, **“Techniques cepstrales améliorées pour l'extraction d'enveloppe spectrale et la détection du pitch”**, Actes du séminaire “Traitement du signal de parole”, pp 83-93, Paris, 1983

### **Formant F1 F2 F3**

The extraction of formants F1 F2 and F3 is as described in **“An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra”** by S. McCandless (IEEE ASSP-22, n° 2, April 1974, pp. 135-141).

### **4.2.1.2 Cepstral peak detection**

Calculates the spectrum by cepstral smoothing to extract the best spectral peaks.

### **Detecting LPC roots**

Calculates the LPC coefficients to extract the spectral maxima using the LPC coefficients (for the frequency and the bandwidth) and the LPC spectrum for the amplitude.

### **Fundamental frequency**

The fundamental frequency, F0 is extracted according to the frequency comb method described in **“Extraction de la fréquence fondamentale par intercorrélation avec une fonction peigne”** by P. Martin (JEP 1981 pp. 223-232) and completed by H. Ney in **“A dynamic programming algorithm for nonlinear smoothing”** (Signal Processing 5-1983 pp. 163-173).

## **4.3 The journal and results**

The results are produced by the calculations performed by a number of commands. The latter save their results in the file results.txt in the c:\Program Files\Wsn0\temp\ directory. The message “results” is then displayed in the welcome window to announce to the user that results have just been put into this file. The menu command **“Misc. • shows results”** allows the user to see these results while the command **“Misc. • add results to log”** allows him to transfer them to the journal.

The **journal** stores all the results that the user wishes to transfer into it. It is the log.txt file of the c:\Program Files\Wsn0\temp\ directory. The menu command **“Misc. • add results to the journal”** allows the user to transfer the results to the journal while the command **“Misc. • show results”** allows him to see the journal.

## 4.4 Multilingual WinSnoori

WinSnoori can be adapted to any language as regards:

- the user interface, by changing the menu command names as well as the messages (texts) displayed in all the windows.
- the phonetic alphabet.

The adaptation of the user interface can be done with the help of a file containing all the texts employed by WinSnoori. This file is declared in the wsno.ini file as follows:

```
CountryFile=english.str
```

in which “english.str” can be replaced by another file name. The file must be in the .\format\ directory. The standard WinSnoori French package comes with francais.str and the English package with english.str.

The definition of the phonetic alphabet is done by means of a file containing all the phonemes employed by WinSnoori. This file is declared in the file wsno.ini as follows:

```
LanguageFile=english.pho
```

in which “english.pho” can be replaced by another file name (francais.pho for the phonemes used in the SAM corpus) which must be included in the .\format\ directory. Care must be taken to ensure coherence between this file and the phonetic labelling files since the phonemes are saved in text form. The english package is delivered with the file english.pho containing English phonemes and the standard WinSnoori french package is delivered with the file francais.pho containing french phonemes. The creation of another file for a new language is within the reach of the standard user. The necessary knowledge is contained in the present manual and in the on-line help as well as the example given in the file francais.pho to which must be added the practical experience gained in the use of the WinSnoori software.

## 4.5 The phonetic alphabet

The phonetic alphabet contains all the phonemes that can be used in the course of phonetic labeling. As previously indicated, necessary information can be found in the .txt file the name of which is specified in the wsno.init file. Each phoneme is defined by a string of 1 or 2 characters.

The search for phonemes can be done using either phonemes or phonetic classes. A **phonetic class** is made up of several phonemes to which is assigned a single collective name (e.g., the class of “voiceless stops” is made up of the phonemes p, t, k). Phonetic classes are also defined in the phoneme file. The same phoneme can belong to several phonetic classes and a phonetic class can comprise several other phonetic classes (e.g., the “consonant” class embraces the “fricative”, the “stop” and the “sonorant”). Phoneme manipulations are proposed to the user through hierarchical menus.

## 4.6 Write file protection

WinSnoori respects the write protect access rights to the speech, phonetic and orthographic files. It proscribes writing on protected files and invalidates menu commands which can modify such files.

WINDOWS allows the individual protection of each file. For that, use the File Manager to modify the file attributes by activating the “read only” option. The necessary details are provided in the WINDOWS user manual, in the paragraph on file attribute definition

of the chapter “The File Manager”. Write protection of files is of particular interest when one is working on speech files received from elsewhere and which one wishes to save without modification. Protection of files guards the user against eventual erroneous manipulation.

## 4.7 Reading and writing speech and label files

### Opening label and speech files

A speech file is opened using the menu command “**File•Open**”. WinSnoori then opens a panel from which the user can select the file to open. Two cases are encountered:

- the case of .WAV files: WinSnoori opens the speech file, then takes possible phonetic and orthographic labels that may be included in the file. If the file is write protected, WinSnoori then invalidates all commands which can modify the file.
- the case of non.WAV files: WinSnoori proposes a panel from which the user can select:
  - the format of the speech file (this is obligatory)
  - the name of the speech file; the user doesn't have to modify this name as WinSnoori had already asked him for it.
  - the name of the (optional) phonetic label file
  - the format of the (optional) phonetic label file. Select NONE if you do not wish to read the phonetic label file.
  - the name of the (optional) orthographic label file
  - the format of the (optional) orthographic label file. Select NONE if you do not wish to read the orthographic label file.

If the speech file is write protected, WinSnoori proceeds to invalidate all the commands whose effect can modify the file.

### Writing speech and label files

Writing a speech file is effected:

- through the “**File•save**” menu command,
- through the “**File•save as**” menu command,
- or on the initiative of WinSnoori itself in some cases such after “Cut/Paste” command where the user skims through the file with the main menu scrolling bar. The write operation is the same as the “**File•save**” menu command.

In the case of the “**File•save**” menu command, WinSnoori automatically writes the speech file as well as any existing label files (i.e., if any has previously been loaded).

In the case of the “**File•save as**” menu command, WinSnoori opens a panel which allows the user to select the file into which to write. Two cases can arise:

- the case of .WAV files: WinSnoori writes the speech file as well as the phonetic and orthographic label files into the selected file.
- the case of non.WAV files: WinSnoori, opens a panel from which the user can select the format of the speech file (this is obligatory) and the name of the speech file; the user doesn't have to modify this name as WinSnoori had already asked for it. He can select:
  - the name and the format of the phonetic label file (optional). Select NONE if you do not wish to save the phonetic label file.
  - the name and the format of the (optional) orthographic label file. Select NONE if you do not wish to read the orthographic label file.



- WinSnoori writes the speech file as well as phonetic and orthographic label files.

The phoneme and the word menus contain the “**Phonemes•save as**” and “**Words•save as**” commands. Only the phonetic or orthographic labels can be saved on files having a non.WAV format with these commands.

## 4.8 The WinSnoori environment

The environment here is no other than the directories and the files that WinSnoori uses. The **explicit environment** is specified in the wsno.ini file and embraces:

- the name of the WinSnoori directory which in turn contains:
  - the .\format\ directory containing in its turn:
    - the phoneme file,
    - the message text file,
    - the format files of the non. WAV sound files,
  - the temporary directory .\temp\ which contains all the WinSnoori generated temporary files (clipboard, journal, ...) the precise knowledge of which is of little utility to the user.
- the name of the phoneme file (without its directory, see above),
- the name of the message text file (without its directory, see above),
- the name of the audio player,
- the name of the phonetic font,
- the name of the user.

The explicit environment is modified by modifying the wsno.ini which is a text file .txt, for example with the WINDOWS notepad.

The **implicit environment** is made up of the speech directory, the phonetic label file directory and the orthographic label file directory. The names of these directories are left to the user to decide since files are open interactively under WinSnoori using the mouse and dialogue panels. There can be several speech file and label file directories.

## 4.9 General options

The general options can be accessed through the main menu “**Misc.•general options**” command. They can be displayed in a “variable modifier object” (see in the sequel) and can be modified by clicking to the right button of the object.

- **waveOutDevice**: gives the WinSnoori active sound output device. It is useful only when the PC has several output devices.
- **autospectro**: If yes, the spectrogram is up to date and is automatically re-drawn whenever necessary. If NO, the spectrogram is re-drawn only when the user activates a calculation in the menu.
- **WindowsWAVfiles**: if YES (the normal case), WinSnoori processes the WINDOWS compatible .WAV files. If NO, WinSnoori processes them as non.WAV files.
- **graymode**: grayed spectrogram mode.
  - “with gray”: the normal case, the spectrogram is drawn in gray tones. This mode is not available on screens having only 16 colors.

“color”: the spectrogram is drawn to the best with the 16 basic colors.

## 4.10 Play back

In WinSnoori, “play” sends the signal back to a loudspeaker. Though this operation is simple, it requires closer consideration since it can get involved in situations requiring conversion of signals.

Indeed the WinSnoori internal format codes every sample on 16 bits; now, some sound peripherals are able to accept only 8 bits whence the need for conversion. WinSnoori accepts several sound frequencies (16 KHz in particular); some devices however accept only 11025 Hz and the multiples 22050 Hz and 44100 Hz. In this case too there is need for conversion.

Here is the strategy applied by WinSnoori:

if possible, output sound at 16 bits, if not convert it to 8 bits and then try to output the signal at its own frequency, otherwise perform frequency conversion and output the sound at 11025 Hz.

The frequency conversion operation is quite long and breaks down as follows. Let *Freq* be the frequency of the sound to output and *Freqout* be the output frequency. WinSnoori chooses a frequency which is roughly a multiple of *Freq* and *Freqout*, say *Mulfreq*. WinSnoori then transforms the sound raising its frequency to *Mulfreq* and filters the sound with a low-pass band, using *freqout/2* as the cut off frequency. Finally, WinSnoori performs a final transformation of the sound by lowering its frequency to *Freqout*.

## 4.11 Windows and graphical objects

### windows

This paragraph gives some generalities about the presentation and the manipulation of windows.

Texts in the title bar ending with numbers such as 7.5 or 12. or .8 give the window numbers. When several windows are opened at the same time, these numbers are useful in establishing a hierarchy between the windows:

- 7.5 means window 7, the daughter of window 5.
- 12. means window 12
- .8 means the window is the daughter of window 8.

### variable modifier objects

Variables which are user modifiable are presented on the screen in the form of “variable modifier objects”. A window with a button displays the name of the variable.

The button provides a means for initiating modifications; the object changes shape becoming bigger when one clicks on the button and can be moved. An object generally contains a scroll bar through which the variable can be modified.

The variable modifier object has two states: a closed state (the standard initial state) and an open state in which the variable can be modified. The button to the top right of the object facilitates switching from one state to the other by clicking on it. When a closed object does not have any button, its variable cannot be modified. The letter appearing in the button gives the variable type and hence the form the object will take when opened:

- i** integer
- f** floating number
- c** choice
- w** the output device of the sound

An open object in the **i** and **f** form presents a scroll bar for modifying the variable and displays the possible extreme values.

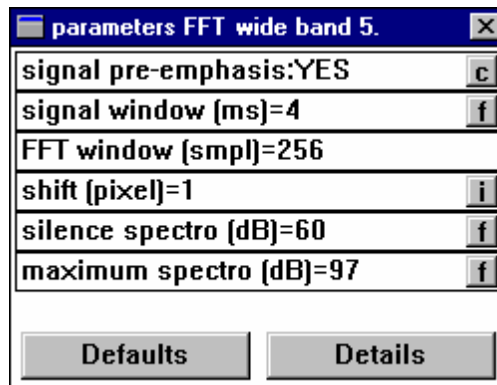
In **c**, the object does not open but changes to a new value on a pre-recorded range each time one clicks on the button.

In **w**, the object displays the characteristics of the output device used and offers a “next” button for moving to the next device. Of all the characteristics displayed, only the name of the device shown in clear is of use to the standard user.

The example below shows a window with variable modifier objects:

The window “Parameter FFT large 5” is a panel opened by the menu command “**Spectro•option•Wide band**”. It contains several variable modifier objects as can be seen and in particular an object for the signal window variable shown above and opened below.

The window “signal\_window (ms).5” is the object we have previously opened. The number 5 recalls that this object is defined in window 5 (i.e., the one in which the number appears). The letter **f** in the button of the object indicates that its variable is of the floating type (in the C or C++ programming sense).



## 4.12 Printing with WinSnoori

Printing is done through WINDOWS facilities:

- the “Print screen” key copies the screen into the WINDOWS clipboard.
- the keyboard combination “alt+Print screen” copies the active window into the WINDOWS clipboard.
- the clipboard is then pasted on a text processing document (WRITE, WORKS, WORD, ...) or painting documents (Paintbrush, ...), and then printed.
- programs like Paintbrush allow adding retouches one's own commentaries to the screen copy or saving only some parts of the screen.

The present document is a typical example that can be printed on WinSnoori since contains screen copies as well as copies of windows.

## 5 Main changes between version 1.3 and version 1.32

- New organization of the editing tools
- Addition of white noise ("Editing tools" in the "Edit" menu)
- OLA filtering along several time x frequency trajectories ("OLA filtering" in the "Editing tools")
- Changing the highest frequency of the spectrogram (In "Options" from the "Spectro" menu)
- Possibility of changing the voicing of F0 points by moving them point by point (CTRL + left button of the mouse)
- Recording at 22050 Hz ("File" menu)
- Resampling of signals ("Edit" menu)
- Computation of F0 for a corpus