xassp

User’s Manual

Version 1.3.8  (May 2000)

Michel Scheffers and Tobias Rettstadt

Institut für Phonetik und digitale Sprachverarbeitung
der Christian-Albrechts-Universität zu Kiel
D-24098 Kiel
Germany
Contents

1 Introduction .......................... 1
  1.1 What is xassp? ......................... 1
  1.2 Manual structure ........................ 1
  1.3 Typographic conventions ............... 1

2 Getting started with xassp .......... 3
  2.1 Starting xassp ............................ 3
  2.2 Opening files ............................. 3
  2.3 Segmental labelling ...................... 4
      2.3.1 Fetching a label ..................... 5
      2.3.2 Playing the speech signal .......... 6
      2.3.3 Using the label list ................. 6
      2.3.4 Fetching and modifying a label ..... 6
      2.3.5 Modifying a label after moving it . 7
      2.3.6 Inserting a new label .............. 7
      2.3.7 Label syntax checks ................. 7
      2.3.8 Deleting an inserted label ........... 7
      2.3.9 Editing an inserted label .......... 7
      2.3.10 Moving a label ...................... 7
      2.3.11 Undoing recent changes .......... 7
      2.3.12 Jumping to a specific label ..... 7
      2.3.13 Saving the labels in a file ...... 8
  2.4 Prosodic labelling ................... 8
      2.4.1 Inserting a prosodic label ......... 8
      2.4.2 Editing a prosodic label .......... 9
      2.4.3 Deleting a prosodic label .......... 9
      2.4.4 Undoing recent changes .......... 9
  2.5 The auto-save feature ............... 9
  2.6 Analysing speech signals .......... 9

3 Using xassp .......................... 11
  3.1 xassp command line options .......... 11
      3.1.1 Examples .......................... 11
  3.2 User levels ............................ 11
  3.3 The xassp main dialog ................. 12
      3.3.1 File selection box .................. 13
      3.3.2 Data type buttons .................. 13
      3.3.3 Options for loading speech signal data in unsupported formats .... 13
      3.3.4 Align and link ....................... 14
      3.3.5 Data logging ......................... 14
      3.3.6 Configuration buttons ............... 14
      3.3.7 Additional buttons .................. 14
  3.4 Common xassp window properties ..... 14
      3.4.1 Elements of xassp windows ........ 15
### 3.4.2 Keyboard and Mouse .............................................. 16
### 3.4.3 Pop-up menus .................................................. 17
### 3.4.4 Window linking ................................................ 18
### 3.4.5 Data logging .................................................... 18
   - Choosing data for logging ..................................... 19
   - Log file format .................................................. 19
   - Adding comment to logged data ................................. 19
### 3.5 Specific `xaosp` window properties ................................ 20
#### 3.5.1 The speech signal window .................................... 20
   - Window menu .................................................... 20
   - Edit menu ...................................................... 20
#### 3.5.2 The label window ............................................. 20
   - Window menu .................................................... 21
   - Edit menu ...................................................... 21
   - Inserting a new label ......................................... 21
   - Moving a label ................................................ 22
   - Label syntax checks .......................................... 22
#### 3.5.3 The fundamental frequency window .......................... 22
   - Window menu .................................................... 23
   - Analysis options ............................................... 23
   - Manipulation of `fzero` contours and synthesis .......... 24
#### 3.5.4 The energy window ........................................... 25
   - Window menu .................................................... 25
   - Analysis options ............................................... 26
#### 3.5.5 The sonagram window ....................................... 26
   - Window menu .................................................... 26
   - Analysis options ............................................... 27
#### 3.5.6 The section window ......................................... 28
   - Window menu .................................................... 28
   - Analysis options ............................................... 29
#### 3.5.7 The palatogram window ..................................... 29
   - Window menu .................................................... 29
   - Display options ................................................ 30
#### 3.5.8 The articulogram window .................................. 30
   - Window menu .................................................... 30
   - Display options ................................................ 31
### 3.6 File handling ................................................... 32
#### 3.6.1 Speech signal ............................................... 32
   - KTH file format ................................................ 33
   - AIFF/AIFC file format ........................................ 33
   - RIFF-WAV file format .......................................... 33
   - CSL file format ................................................. 33
#### 3.6.2 Labels ......................................................... 33
#### 3.6.3 Fundamental frequency ..................................... 35
#### 3.6.4 Energy ......................................................... 36
#### 3.6.5 Palatogram .................................................... 36
#### 3.6.6 Articulogram ................................................ 36
### 3.7 Printing .......................................................... 36
#### 3.7.1 Modifying page options ...................................... 36
#### 3.7.2 Selecting windows ........................................... 37
#### 3.7.3 Choosing a print configuration ............................ 37
#### 3.7.4 Modifying window parameters .............................. 37
## CONTENTS

4 Configuring xassp 41
4.1 X Resources 41
4.2 User Configuration Files 43
  4.2.1 The display configuration file 43
  4.2.2 The label configuration file 44
  4.2.3 The font configuration file 45
4.3 Administrative configuration files 45
  4.3.1 The xassp users file 45

5 Analyses 47
  5.1 Energy analysis 47
  5.2 F0 or pitch analysis 47
  5.3 Formant analysis 48
  5.4 Spectral analysis 49

A Definition of signal processing terms 51

B Key and mouse button bindings 55

C X Resources 59
  5.1 Resources for the Core widget 59
  5.2 Resources for the XmPrimitive widget 59
  5.3 Resources for the XspMain widget 60
  5.4 Resources for the XspData widget 61
  5.5 Resources for the XspHscale widget 61
  5.6 Resources for the XspVscale widget 61
  5.7 Resources for the XspWave widget 62
  5.8 Resources for the XspLabels widget 62
  5.9 Resources for the XspFzero widget 62
  5.10 Resources for the XspEnerg widget 62
  5.11 Resources for the XspSonag widget 63
  5.12 Resources for the XspSection widget 63
  5.13 Resources for the XspPulate widget 64
  5.14 Resources for the XspArtgram widget 64

Bibliography 67

Index 68
Chapter 1

Introduction

1.1 What is xassp?

Xassp refers to the IPDS Advanced Speech Signal Processing tool, which operates under the X Window System. It can be thought of as an amalgamation of the MIX labelling program from the Department of Speech Communication and Music Acoustics of KTH Stockholm (Carlson and Granström 1986) and the general-purpose signal processing program ASSP developed at IPDS Kiel (Scheffers and Thon 1991).

Xassp has originally been developed for prosodic labelling (Kohler et al. 1995) and was later extended for segmental labelling. The various analysis options and signal displays of the current version, however, make it also useful for a wide range of different purposes. Analyses incorporated in xassp include energy, fundamental frequency \( (F_0) \), DFT and LPC spectrum (section) and spectrogram (sonagram). Next to the display of these data, versatile displays have been included for articulographic data (electro-palatograph and X-ray microbeam recordings).

1.2 Manual structure

This chapter (Chapter 1) contains a short description of xassp and this manual.

Chapter 2 shows how to use xassp for the most common tasks. It is intended to be a tutorial-like introduction to xassp for users at ipds.

Chapter 3 provides a complete description of the features of xassp. It explains the data types that xassp can handle and also shows different ways to create, load, display and manipulate them.

Chapter 4 deals with the numerous configuration possibilities of xassp. These include X resources as well as xassp’s own configuration files.

Chapter 5 explains the methods for analysing speech signals incorporated in xassp.

Appendix A contains definitions of the most important signal processing terms that are used throughout this manual.

Appendix B provides an overview of all key and mouse button bindings available in xassp.

Appendix C contains a reference list of X resources that have an influence on the behaviour of xassp.

1.3 Typographic conventions

When referring to keys or mouse buttons, the following conventions are used in this manual:

A single keystroke always appears as a single letter set in typewriter font, e.g. \( m \) means hitting the key labelled M. Special keys such as the \( Return \) key are written in bold face, e.g. \textbf{Return}. The mouse buttons are also treated as special keys: \textbf{LM} (left mouse button), \textbf{MM} (middle mouse button), \textbf{RM} (right mouse button).

Modifier keys (keys that must be held down while pressing another key) are written as \textless\textbf{Shift}\textgreater, \textless\textbf{Ctrl}\textgreater and \textless\textbf{Alt}\textgreater. \textless\textbf{Ctrl}\textgreater\textbf{RM} thus means pressing the right mouse button while holding down the Ctrl key.
Chapter 2

Getting started with xassp

This chapter should help you to get to know xassp and its basic principles. It describes how to perform the signal annotation tasks carried out at ipds (Kohler et al. 1992, Kohler et al. 1995), while Chapter 3 provides a more general description of the features of xassp. How xassp can be tuned to your personal preferences using configuration files and X-resources will be set out in Chapter 4.

2.1 Starting xassp

To start xassp simply type

```
  xassp
```

on the command line and press Return. If you want to do prosodic labelling, type

```
  xassp -u10
```

followed by Return to start xassp. For segmental labelling, you have to invoke xassp by typing

```
  xassp -u20
```

again followed by Return.

The numbers in the last two commands are the user levels required for prosodic (user level 10) or segmental (user level 20) labelling. Please refer to Section 3.2 for more information on user levels and to Section 3.1 for a complete description of xassp command line options.

2.2 Opening files

After starting xassp the main dialog box appears on the screen (see Figure 2.1). Now files can be selected for display in windows. There are several ways to do this. The first step is choosing the directory that contains the file of interest. This can be done either by typing the path in the text field labelled Filter or by selecting a directory from the Directories list. When typing the path remember to add a slash (/) at the end. To tell xassp to display the contents of the chosen directory you can either press Return in the Filter text field or click on the Filter button or double-click on the directory in the Directories list.

The next step is choosing the file to be loaded. You can either select the file from the Files list or type the file name in the Selection field.

Now you can actually load the file by clicking on the Load button. If xassp refuses to load the file, you have to specify the data type by selecting one of the buttons labelled Speech Signal, Labels, Fzero, Energy, Pulatoogram, and Articulogram. Then try to open the file again by pressing the Load button. If xassp still displays an error message, the file format is not recognised or you selected the wrong data type.

If the file you chose for display consists of raw speech signal data or data in an unsupported format, you can specify its properties by clicking on the File format options button. These are explained in Section 3.3.3.
If the Align and link button was selected when loading a file, all windows will be aligned and linked after the load operation is complete. For more information on links, see Section 3.4.4. Selecting this button only makes sense if you have already opened one or more xassp windows.

Instead of opening the files one after another you can use display configurations. To do this you select one of the files for loading, select one of the configuration buttons (they are located directly above the Selection field, see Figure 2.1) and press the Load button. The configuration Prosodic, for example, would open a speech signal window, a fundamental frequency window and a label window. The Segmental configuration opens a speech signal window, a sonagoram window and a label window. In both cases a speech signal file with the suffix .166,.r16,.shh,.16,.aif,.or,.wav and a label file with the same name but with the suffix .mix,.silh,.or,.sl must be present. The fundamental frequency and the sonagram are computed from the speech signal.

### 2.3 Segmental labelling

User level 20 in xassp has been especially tailored for segmental labelling following the ipds conventions (Kohler et al. 1995). Before you can begin with segmental labelling you must load a set of files including the speech signal file and the label file. It is also helpful to have a sonagram that is computed from the speech signal. You can load the files, do the analysis, align and link the windows by selecting the configuration Segmental as it is described in Section 2.2. If you use this configuration, the windows that appear on the screen should look similar to those shown in Figure 2.2.

When doing segmental labelling, labels are set at segment boundaries. The labels and the order in which they are to be placed are given (in the following the term label stack will be used for the given labels). The label order cannot be changed and labels that are taken from the label stack cannot be deleted. New labels can only be inserted if they are also marked as inserted with a – (minus) after the label prefix.
2.3. **SEGMENTAL LABELLING**

![Image of a spectrogram and waveforms](image1)

Figure 2.2: Display configuration for segmental labelling. Upper window: speech wave, middle window: sonagram, lower window: labels (this file has been partly segmented)

### 2.3.1 Fetching a label

To place the next label from the label stack, which is shown in the upper right corner of the label window, at a certain point in time, you first have to set the temporary marker onto this point (see Figure 2.3). This can be achieved by placing the cursor at this point and then pressing `<Shift>LM`. Then you have to move the cursor between the temporary marker and the end of the window. Pressing `<Shift>MM` moves the next label from the label stack to the point in time at which the temporary marker is set. This procedure of setting the temporary marker and fetching a label from the label stack is repeated until there are no more labels left on the stack. In this case the small box in the upper right corner that displays the next label disappears.

![Image of label placement](image2)

Figure 2.3: Label placement
2.3.2 Playing the speech signal

It is often difficult to determine segment boundaries without listening to the relevant stretch of the speech signal. You can repeatedly play the signal stretch between the temporary marker and the cursor with LM and shift the temporary marker (by pressing <Shift>LM) until you think that it is set on the right point. There are also shortcuts to play other parts of the speech signal (see Table B.1).

2.3.3 Using the label list

xassp is able to list the labels of the current file in a separate window. You can open this window by pressing <Alt>RM and selecting Show as Text from the pop-up menu that appears. The label list consists of two columns. The first one shows the label time in seconds. The second column contains the label names. The labels on the label stack normally have a time of 99,999 seconds. This can only be seen in the label list, they are not displayed in the normal label window. If you make changes to the labels in the label window, the label list will automatically be updated.

2.3.4 Fetching and modifying a label

Since the labels on the label stack do not always match the utterance that is to be labelled there are several possibilities to modify them. If you press <Shift>RM instead of <Shift>MM to fetch the label from the stack, you can edit the label before it is moved. In this case an Edit label dialog box is displayed (see Figure 2.4). In this dialog box you can edit the label in the text field that is located directly above the three buttons labelled OK, Cancel and Help. Pressing the OK button then moves the edited label onto the temporary marker, and pressing the Cancel button neither changes nor moves the label.

The most frequent changes that are made to labels taken from the stack are appending a minus sign to mark the label as deleted and inserting a percent sign to mark an uncertain segment boundary. These frequent modifications have been assigned directly to mouse buttons. You can fetch a label and append a minus sign by pressing <Alt>LM and fetch the label and insert a percent sign by pressing <Ctrl>LM.

2.3.5 Modifying a label after moving it

If you moved the label and then realise that you forgot to modify it, you can still edit it by moving the mouse pointer onto the label (the label is highlighted), pressing <Ctrl>RM and selecting Edit from the pop-up menu that appears. The Edit label dialog box is exactly the same as the one described above except that the label is not moved when you press the OK button.

Note that at user level 20, you are only allowed to make changes that are absolutely necessary, e.g. changing the label prefix from ## (word boundary) to \$ (word-internal).
2.3. SEGMENTAL LABELLING

2.3.6 Inserting a new label

If there is no label on the stack that matches the stretch of the speech signal that has to be labelled next, you can insert a new label at the temporary marker with $<\text{Ctrl}>\text{MM}$. This label must be marked as inserted by inserting a minus sign after the label prefix. If you forget to do this, xassp refuses to insert the label and shows a warning message.

2.3.7 Label syntax checks

The check that is done when inserting a label is only one of a number of label syntax checks that xassp performs whenever a label is edited, inserted or deleted. So, for instance, you are not allowed to insert invalid labels or to modify labels in such a way that they become invalid. A more detailed description of these checks can be found in Section 3.5.2.

2.3.8 Deleting an inserted label

If you accidentally inserted a label and want to delete it, you have to move the mouse pointer onto the label (the label is highlighted), press $<\text{Ctrl}>\text{RM}$ and select Delete from the pop-up menu that appears. You will not be able to delete any labels that were fetched from the label stack, since a check is performed before the label is deleted. If you try to do this anyway, a warning message is displayed and the label is not deleted.

2.3.9 Editing an inserted label

xassp refuses any attempt to edit an inserted label (at least if your user level is below 30). To edit an inserted label you therefore have to delete and re-insert it at the same point in time as described above.

2.3.10 Moving a label

When you discover that a label was not set onto the right point you will want to move it onto a different one. This action is very similar to fetching a label from the label stack. You first have to set the temporary marker to the point in time onto which you would like to move the label. Then place the cursor between the temporary marker and the label that is to be moved. Note that only a label directly to the left or directly to the right of the temporary marker can be moved. The label order cannot be changed. By pressing $<\text{Shift}>\text{MM}$ you can now move the label onto the temporary marker. You can also use the shortcuts that are described above to edit the label before moving it or to modify it by inserting a percent sign, and so on.

2.3.11 Undoing recent changes

xassp remembers every change that you make (inserting, deleting, moving and editing labels). By pressing $<\text{Ctrl}>\text{RM}$ and then selecting Undo from the pop-up menu that appears you can undo the last change. Repeating this action undoes the last but one change, and so on. You can continue undoing until the first change that you made is reversed.

2.3.12 Jumping to a specific label

There are several nice features that make segmental labelling with xassp more user-friendly. If you want to label a file that has already been partly labelled, you can centre the display around to the last label that has been positioned by pressing $<\text{Ctrl}>\text{1}$ after loading the file.

$<\text{Alt}>g$ makes the Go to dialog box appear on the screen. In its Label text field you can type a label you wish to jump to. After pressing the Go button, the label you entered is searched for, starting from the current position of the beginning of the window and the first occurrence of the label. If the label has been found, the label window is scrolled so that the label is displayed in the centre of the window. If the label has not been found, a warning message will be displayed. Now you can jump to the next (previous) occurrence of the same label by pressing $<\text{Ctrl}>\text{3}$ ($<\text{Ctrl}>\text{b}$). If there is no next (previous) occurrence, a warning message will be displayed.
2.3.13 Saving the labels in a file

If you press the Exit button in the xassp main dialog box or select Close from the label window menu or Close all from the window menu of any open xassp window, you are prompted whether you want to save the data in case you made changes to them. Press Save to save the data and exit, press Do not save to exit xassp without saving any changes. Pressing Cancel neither saves the data nor exits xassp.

To save the data without exiting xassp press <Alt>R and select Save from the pop-up menu that appears. If an error occurs during the writing of the data, an error message is displayed on the screen.

You can save the labels under a different file name by pressing <Alt>R and selecting Save as from the pop-up menu. You can then type the new file name in the Selection text field or select a file from the Files list. Pressing the OK button saves the data to the specified file. If the file you selected already exists, you are prompted whether it should be overwritten.

2.4 Prosodic labelling

The first step for prosodic labelling is to load a set of files including a speech signal file and a label file. In most cases you will also need the fundamental frequency contour of the speech signal. If you select the configuration Prosodic as described in Section 2.2, the speech signal and label files are loaded, a fundamental frequency analysis of the speech signal is done and the corresponding windows are opened as shown in Figure 2.5).

As with segmental labelling, xassp has a user level especially tailored to facilitate the type of prosodic annotation carried out at ipds. Among other things this assumes a prior segmental annotation. The segmental labels are then used as the temporal points at which prosodic labels can be placed. Running xassp at user level 10 ensures that prosodic labels can be inserted and modified without affecting segmental information. Of course, as with segmental labelling all checking and restrictions on labels can be overridden if xassp is run at user level 30 or higher (see section 3.2).

When doing prosodic labelling new labels can only be set at points in time at which the segmental labels are already placed. There is therefore no need for a temporary marker. The prosodic labels are not fetched from a stack. You can insert any prosodic label at any point in time at which a segmental label is set.

2.4.1 Inserting a prosodic label

You can insert a prosodic label before or after a given label. To do this you move the mouse pointer onto the label before or after which the new label is to be inserted (the label will be highlighted), press <Ctrl>R
2.5. THE AUTO-SAVE FEATURE

and select Insert before or Insert after in the pop-up menu that appears. In the Insert label dialog box, which is shown in Figure 2.4, you can either type the label in the text field that is located directly above the row of buttons and then press Return to insert the label, or press one of the label buttons provided in the dialog box to insert the label it shows. The latter method is intended to be used for the most frequent labels. How the labels that are shown as buttons in the Insert label dialog box can be changed is described in Section 4.2.2.

Note that inserted prosodic labels, unlike segmental labels, do not require a – (minus) after the prefix and can be edited.

2.4.2 Editing a prosodic label

You can edit an inserted label by moving the mouse pointer onto the label (the label is highlighted), pressing <Ctrl>R and selecting Edit from the pop-up menu that appears. The Edit label dialog box is very similar to the Insert label dialog box. You can either edit the label in the text field, or press one of the buttons. If one of the buttons is pressed, the label is replaced by the one that is displayed on the button.

It is also possible to edit segmental labels. To do this you take the same actions as for editing prosodic labels. Note that the changes you can make to segmental labels are restricted.

2.4.3 Deleting a prosodic label

If you accidentally inserted a prosodic label, you can delete it by moving the mouse pointer onto the label (the label will be highlighted), pressing <Ctrl>R and then selecting Delete from the pop-up menu that appears. You will be asked whether you really want to delete the label. If you press the OK button, the label is deleted. If you press the Cancel button, nothing happens.

2.4.4 Undoing recent changes

By pressing <Ctrl>R you can undo the last change you made. You can repeat this action until the first change is reversed.

2.5 The auto-save feature

It is a well-known fact that computers tend to crash seconds before you intend to save an hour’s hard work to disk. Two features have therefore been included in xassp to reduce data loss. Both concern only label files and are activated when you start xassp with a user level of 10 or higher.

First of all, a back-up file is created when you load a label file. The name of this file is the same as that of the label file, but with .bak appended. A warning will be displayed if the back-up file cannot be created. The back-up file can, for instance, be used if the original label file has been destroyed or damaged.

Second, xassp verifies every minute whether you made changes to the labels since the last time you saved them. If this is the case, it will write the current label data to a temporary file. This file has the same name as the label file you are processing, but with %xassp% appended to it. If the program or the whole system crashes, this file will thus contain the label data of at most one minute before. The first thing you therefore have to do before starting xassp anew is to copy or move this file to the original one. If the auto-save file cannot be written, an error message box will be displayed. One of the buttons in this box allows you to disable auto-saving so that you don’t have to acknowledge the error message every minute. Beware that the error message may imply that you can also not save the data e.g., because you have no write permission in the file’s directory.

2.6 Analysing speech signals

Analyses incorporated in xassp are:

- fundamental frequency,
- energy,
• sonagram,
• section.

Refer to Chapter 5 for more information on these different analysis methods.

There are two different ways to perform an analysis of a speech signal and display the results. First, you can use the window menu item Analyse in the speech signal window. The second possibility is the definition of analyses in a configuration. The latter method is described in Section 4.2.1, while the former will be described in this section.

The first step in doing an analysis is to load the speech signal that you want to analyse as described in Section 2.2. Then press <Alt>RM in the speech signal window and select Analyse from the pop-up menu that appears. This opens the Analysis dialog box, in which you can select the analysis to be performed. Click on one or more of the buttons labelled Fzero, Energy, Labels, Sonagram and Section. If you select the Align and Link button, all xassp windows are aligned and linked after the analysis has been completed (see Sections 3.4.3 and 3.4.4 for a description of window alignment and links). If you marked a region of the speech signal by placing brackets (with F5, F6), you can choose whether to analyse the whole speech signal (select the All button) or only the marked region (select the Region button).

Now select the OK button to perform the analysis. The outcome of each analysis is presented in a separate window. If you select the Cancel button, no analysis is performed and no new windows are opened.

Note that you cannot modify any analysis parameters before the analysis is finished. Instead, you have to press <Alt>RM in the window that contains the results of the analysis whose parameters you would like to change and select Re-analyse from the pop-up menu. The analysis options dialog box will then appear on the screen in which you can modify the analysis parameters. After selecting the OK button, the speech signal will be re-analysed using the modified analysis parameters.

Descriptions of the analysis options dialog boxes are found in Sections 3.5.3 (fundamental frequency), 3.5.4 (energy), 3.5.5 (sonagram) and 3.5.6 (section). For more information on the different analysis methods and their parameters, see Chapter 5.

Since the analysis parameters can only be changed after the analysis has been done with default parameters, you can modify the default parameters. This can be achieved by editing the X resources used by xassp. For more information on X resources, see Section 4.1 and Appendix C.

You may have noticed that there is one button in the Analysis dialog box that has so far not been explained. It is the Labels button, which, if it is selected and you press OK, simply opens an empty label window. Beware that the format of the labels in this window is the MIX format. If you want to use the SAMPA format instead, the first thing you should do before setting any label is to select Save as from the window menu, press the button marked SAMPA in the file selection box that appears and select or define a file name. It is in general recommended that, when you create a label window using the Analyse option, as a next step you should define a file name for the data via the Save as sub-menu.
Chapter 3

Using xassp

3.1 xassp command line options

xassp can be invoked from the command line (shell) with the following syntax:

```bash
xassp [-c|--conf <dir>] [-u|--userlevel <user level>] [-h|--help]
```

Brackets enclose optional parameters; options that are separated by a vertical bar are equivalent.

With the option `-c` you can specify the local directory where xassp searches for its configuration files. The default directory is the subdirectory `.xassp` of your home directory.

The option `-u` lets you specify a user level that xassp should use during the session (see below). The default user level is taken from the file `/etc/xassp_users`, in which the maximum user level you are allowed is also specified. For more information on this file, see Section 4.3.1.

When the option `-h` is given, xassp displays a short description of the command line options and exits. In this case any additional options are ignored.

If you give an option on the command line that is not known to xassp, it displays a short message (illegal option) and ignores the option. If you forget to specify an argument to the `-c` and `-u` options, xassp prints a short message (option requires an argument) and ignores the option.

3.1.1 Examples

If you want xassp to show you a short help message describing the options, you simply type

```bash
xassp -h       or       xassp --help
```

If you want to start xassp with user level 20 to do segmental labelling, you should type

```bash
xassp -u 20     or       xassp --userlevel 20
```

Since all options shown above are enclosed in brackets and therefore are optional, you can simply type

```bash
xassp
```

to invoke xassp. The application is then started using your default user level and the default local configuration directory.

3.2 User levels

User levels restrict the use of xassp features excluding those that are not required for the job at hand. In this way modifications of speech signals can, for instance, be prevented when only label editing is required. The user levels needed to perform certain actions in xassp have been chosen on the basis of the tasks most often performed at ipds. It is important that you always choose the lowest user level that allows you to do your work. Table 3.1 displays a short overview to help you in the choice of the appropriate user level.
<table>
<thead>
<tr>
<th>User level</th>
<th>Restrictions/Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no data modification allowed</td>
</tr>
<tr>
<td>10</td>
<td>prosodic labelling</td>
</tr>
<tr>
<td>20</td>
<td>segmental labelling</td>
</tr>
<tr>
<td>30</td>
<td>unrestricted labelling</td>
</tr>
<tr>
<td>40</td>
<td>editing speech signals</td>
</tr>
</tbody>
</table>

Table 3.1: xassp user levels

At user level 0 all actions that involve modifications of any kind of data are blocked. If you choose a user level of 10 or higher, you can edit and delete labels, but at level 10 you may only insert a new one at a point in time where another label has already been set and you are not allowed to move labels to a different point in time. This user level is intended for prosodic labelling. To be able to insert new labels and to move existing ones, which is needed for segmental labelling, you need a user level of at least 20. User levels 10 and 20 also imply syntax and validity checks on label modification and insertion (see Section 3.5.2 for more information on label syntax checks). To suppress these checks, you should choose user level 30 or higher. With a user level of 40 or higher you are even allowed to edit speech signals and save them.

![xassp main dialog](image)

Figure 3.1: xassp main dialog

### 3.3 The xassp main dialog

The xassp main dialog (see Figure 3.1) contains

- the file selection box (Filter and Selection text fields, Directories and Files lists)
- data type buttons (Speech signal, Labels, Fzero, Energy, Palatogram and Articulogram)
- the Align and link and Data logging option buttons
- the File format options button
- several configuration buttons (for example Prosodic and Segmental),
- a row of action buttons (Load, Filter, Exit and About).

These items will be described in the following Sections.
3.3. THE XASSP MAIN DIALOG

3.3.1 File selection box

The file selection box lets you choose the directory and the actual file that you want to load.

To select the directory you use the Filter text field, the Directories list and the Filter button. You can either edit the directory that is given in the Filter text field and press Return or select the directory in the Directories list and then press the Filter button. Instead of pressing the Filter button it is also possible to double-click on the directory in the Directories list.

Now that you are in the right directory you can simply select the appropriate file from the Files list.

If you know the complete path to the file that you are interested in you can skip the whole procedure described above and just enter the path including the file name in the Selection text field.

Once you have selected a file you can press the Load button to load the file and display its contents in a separate window.

3.3.2 Data type buttons

When you load a single file, xassp determines the type of data in that file on the basis of the suffix (extension) of the file name (see Table 3.2). If it does not recognise the suffix or if the suffix does not correspond to the data type, you have to specify the data type by selecting one of the buttons labelled Speech signal, Labels, F0, Energy, Palatogram and Articulogram before pressing the Load button. Xassp then uses the type information provided when loading the file.

<table>
<thead>
<tr>
<th>Suffixes</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>.lbl .mix .sam .s1h .s1</td>
<td>labels</td>
</tr>
<tr>
<td>.pit .f0</td>
<td>F0</td>
</tr>
<tr>
<td>.rms .nrg</td>
<td>energy</td>
</tr>
<tr>
<td>.pal .epg</td>
<td>palatogram</td>
</tr>
<tr>
<td>.tsy .xyd</td>
<td>articulogram</td>
</tr>
</tbody>
</table>

Table 3.2: Assumed correspondence between file suffixes and data types

When loading files using a configuration, the data type buttons are ignored and the correspondence between data type and file suffixes is taken from the configuration definition (see Section 4.2.1 for more details).

![File Format Options](image)

Figure 3.2: File Format Options

3.3.3 Options for loading speech signal data in unsupported formats

By pressing the File format options button in the xassp main dialog, the dialog box shown in Figure 3.2 appears. Here you can specify the header size (bytes to skip), the bits per sample (8 or 16), the sampling rate, whether the file is mono or stereo, and the byte order. These options are used, if you load a file that contains raw speech signal data or has a format which is not supported by xassp. 
3.3.4 Align and link

Below the data type buttons you find the Align and link button. If this button is set, all xassp data windows are aligned and linked after loading some files. See Section 3.4.4 for a description of alignment and Section 3.4.4 for information on window links.

3.3.5 Data logging

By clicking the Data logging button, which is located next to the Align and link button, you can toggle data logging. This feature is explained in Section 3.4.5.

3.3.6 Configuration buttons

To make it easier for you to open several files at once, xassp provides display configuration buttons, which you find directly above the Selection field. Such a display configuration consists of a set of data types and associated suffixes that describe the files to be loaded. These configurations must be defined in the xassp configuration file (see Section 4.2.1).

If you want to open files using a pre-defined configuration, you just start with selecting one of the files to be loaded as described in Section 2.2. Xassp then determines which other files should also be opened by sequentially substituting the suffix of the selected file by those that are given in the configuration until the name matches that of an existing file. If the suffix of the selected file matches any of the suffixes in the configuration, that file will always be loaded. This ensures that, in case two files exist with names matching the configuration for the same data type (e.g., abc123.mix with labels in MIX format and abc123.s1b with the same labels but in SAMPA format), the desired one will be taken. Otherwise, the sequence of the suffixes in the configuration definition determines which of the two files will be loaded.

It is also possible to specify analyses in the configuration. These are then performed instead of loading a file.

All windows that are opened in this way are aligned and linked.

3.3.7 Additional buttons

There are two buttons that have not been mentioned yet:

The Exit button exits xassp. If any open xassp window contains data that have been modified, you are prompted whether to save the data before the window will be closed.

The About button gives brief information about xassp.

3.4 Common xassp window properties

xassp is able to display the following data types:

- Speech signal
- Labels
- Fundamental frequency
- Energy
- Sonagram
- Section
- Palatogram
- Articulogram (X-ray microbeam data)

To display each of these data types xassp uses windows with different properties. The main differences between the data windows are the way in which the data are displayed, the contents of the pop-up menus and the actions available via key and mouse button bindings. There are special displays for the Palatogram and Articulogram which will be described in their respective Sections (3.5.7 and 3.5.8). The other xassp windows have much in common, which will be described in this Section.
3.4. COMMON XASSP WINDOW PROPERTIES

![Image of a speech signal window](image)

Figure 3.3: xassp window containing a speech signal

### 3.4.1 Elements of xassp windows

As you can see in Figure 3.3, most xassp windows contain the following elements:

**Window title:** In the title area xassp displays the window number and the name of the file the data were loaded from. If this window contains analysed data, the name of the file from which the data were derived from is shown as well.

**Status line:** The status line provides information about the positions of the cursor and the brackets (see below). For displaying these values the same unit as for the horizontal scale is used. The cursor position in the speech signal window is displayed in seconds because the horizontal axis is a time scale. In the section window the cursor position is displayed in Hertz.

The contents of the status line vary from window to window because we try to show only the information that is useful for that particular data type. The speech signal and sonagram windows, for example, show, next to the time of the brackets (indicated by L and R) also the distance between the cursor and the brackets (marked by LC and RC) and between the brackets (labelled LR). The distances are given in milliseconds as most convenient unit for duration measurements. In the speech signal window the inverse of the distance between the left bracket and the cursor is also shown. This value (in Hz) can be used to measure the local fundamental frequency by positioning the left bracket at the beginning of a period and the cursor at the end.

In addition, the status line also shows the data value at the cursor position using the same unit as the vertical scale. The data value in the energy window, e.g. is displayed in dB. In the fzero and sonagram windows the vertical position of the cursor (in Hz) is also listed.

**Horizontal and vertical scales:** The scales allow you to associate the data plotted in the data area with values. In the energy window, for example, you can easily find the energy value at a certain point in time by locating this point on the horizontal (time-)scale and then reading the value on the vertical scale. An easier way is to move the cursor to the point in time you are interested in and then read out the data value shown in the status line.

**Data area:** In this part of the window the data are plotted.

**Cursor:** The cursor is a vertical bar that is drawn into the data area and always has the same horizontal position as the mouse pointer. The default colour of the cursor is blue.

**Brackets:** xassp provides two brackets to delimit a region. Brackets are displayed as vertical bars in the data area. The region that the brackets define can be used to play part of the speech signal, to analyse the region separately from the rest of the signal or to perform cut and paste operations in the speech signal window. Normally, brackets are set onto the next positive zero crossing to the left of the cursor, but this behaviour can be controlled by the user.

Brackets can be set with F5 (left bracket) and F6 (right bracket). To override the default setting of the brackets on zero-crossings you can use <Ctrl>F5 and <Ctrl>F6 to set the brackets on the exact cursor position. The brackets can be cleared with Return 1 or Del. They can be toggled between an active and inactive state with <Shift>F5 and <Shift>F6. The default colour of the left bracket is green, that of the right bracket red.

---

1Clearing the brackets with Return does not work on all machines
Temporary marker: The temporary marker can be thought of as a way of marking a certain point in time, where a label is to be placed. It is set with $<$Shift$>$-LM and is per default shown as a dashed yellow line.

In this version of xasp the temporary marker is always set on the exact cursor position. It can be moved onto the next zero crossing in the speech signal to the left by pressing F3 and to the next positive zero crossing to the right by hitting F4.

Scroll bar: Since the width of the data area is limited you cannot always see all data at the same time. The data area therefore only shows a part of the data. The scroll bar lets you choose which part to display.

Pop-up menus: Pressing $<$Alt$>$-RM makes a pop-up menu (window menu) appear on the screen. You can either hold the mouse button down, move the mouse pointer over the menu entry that you want to activate, and then release the mouse button, or you can click the mouse button, and then select the menu entry with the left or the right mouse button.

If you press $<$Ctrl$>$-RM in a window that contains modifiable data, a special pop-up menu providing edit functions will be displayed.

Not all of these elements make sense for all window types. Brackets, temporary marker and scrollbar are for example not available in the section window because it has no horizontal time scale.

3.4.2 Keyboard and Mouse

xasp windows are controlled by a combination of keyboard and mouse input.

When using the keyboard it is important that the window you want to control has the input focus. This is normally achieved by clicking into the title area of the window or, in some configurations, by simply placing the mouse pointer inside the window. The window that has the input focus usually has a darker border and title area than the other windows. Which action causes a window to get the input focus is determined by the way your the window manager has been configured. It is recommended to have to click on a window to give it input focus since this allows simultaneous control of two windows: one via the keyboard and another via the mouse.

There are a number of key bindings that are used in almost all windows for the same purpose. These are listed in Table B.1. Since the section window has no horizontal time scale only the bindings for the Esc key and the $<$Alt$>$-RM apply here.

The mouse buttons are mainly used to play parts of the speech signal. Refer to Table B.1 for a list of commonly used mouse bindings. Again, the commands for playing the speech signal are not available in the section window.

Appendix B provides a complete reference to the key and mouse button bindings defined in xasp.
3.4. COMMON XASSP WINDOW PROPERTIES

![Link windows dialog box](image)

Figure 3.5: The Link selection menu

### 3.4.3 Pop-up menus

As explained in Section 3.4.1 all xassp windows have a pop-up menu, which is called up by pressing `<Alt>RM` (see e.g., the pop-up menu of the speech signal window in Figure 3.4). The pop-up menus of most windows have many items in common, which will be described first. The window-specific menu items are listed in Section 3.5 which deals with the properties of the different windows.

**Zoom in:** (shortcut: ↓) By activating this menu item you increase the time resolution with which the data are displayed. This implies that the part of the data that is displayed becomes smaller. The point in time from which the data are displayed remains the same.

**Zoom out:** (shortcut: o) Same as *Zoom in* except that the time resolution is decreased and therefore the part of the data that is displayed becomes larger.

**Show all:** (shortcut: a) The resolution is adjusted so that all data are displayed at the same time.

**Show region:** (shortcut: b) The region defined by the two brackets becomes the part of the data that is displayed.

**Align:** Repositions all windows so that they are stacked on top of each other and aligned at the left hand side of the screen with the first window at the top of the screen. The order in which the windows are stacked is the same as the order in which they were opened. Gaps will occur when windows have been iconized. The section window is excluded from alignment.

**Link:** Opens the Link box in which you may link different features of the windows displayed on the screen (see Section 3.4.4 for more details).

**Unlink:** Removes all links from this window to other windows on the screen.

**Redraw:** Redraws the window contents in case parts of the data were accidentally not drawn.

**Print:** Displays the *Print* dialog. Please refer to Section 3.7 for more information on printing.

**Close:** (shortcut: Esc) Closes the current window. If you made changes to the data in this window, you will be prompted whether they should be saved or not.

**Close all:** Closes all windows.
3.4.4 Window linking

It has been mentioned before that it is possible to link windows. This is useful for easier handling of the different data displays associated with a particular speech signal, for instance when labelling the speech signal. There are several ways of linking windows. The first is shown in Section 3.3.6. In the file selection box different data types, e.g. speech signal and fixero, are selected. Then the button Align and link is pressed. The two windows will be displayed on the screen aligned and linked. By default all window features cited below are linked.

Another way of getting aligned and linked windows is to use one of the configuration buttons directly above the Selection field (see Figure 2.1). For example, Segmental, displays a speech wave, a sonagram and a label file in separate windows on the screen: these windows are automatically linked and aligned.

Finally, the menu item Link (see Figure 3.5) may be used in one of the windows. It has to be chosen if the windows have been called up separately without any configuration. You can also use this menu to unlink one or more window features. If you want all features to be unlinked, select the window menu item Unlink.

In the following list all possible links are explained.

Cursor: For several tasks it is necessary that the cursor is at the same point in time in each window. Especially for segmenting and labelling a speech signal it is very important that you can see where you are to correctly position a label.

Zoom: The temporal resolution is the same in all linked windows.  

Brackets: The brackets are set at the same point in time in linked windows.

Play: If a non-speech signal window is linked to a speech signal with a play link, you can use the key and mouse buttons listed in Table B.1 in the non-speech signal window to play the signal in the speech signal window.

Scrolling: The starting time for the data display is the same in all linked windows.

Temporary marker: The temporary marker is set at the same point in time in all linked windows.

3.4.5 Data logging

xassp provides the possibility to store label names, \( F_0 \)-values, energy values and formant and pellet data, associated with a certain point in time in a log file. To enable data logging you have to activate the Data logging button in the xassp main dialog. If logging is ON, a status window appears in the lower right corner of the screen (see Figure 3.6). It shows which file was selected as the xassp log file (the default is xassp.log in the directory xassp was started in). Pressing the Close button closes the status window and disables data logging. If you select the File button, you can change the xassp log file by picking a new file in the file selection box that is displayed. Once data logging is enabled, you can position the cursor at the point in time of interest and press \( w \) to write the data to the log file. The cursor will now briefly change its shape to a pencil to confirm that data have been logged. If this does not happen, check which window has the input focus because this determines the search for data to log. Also, some windows, such as the label, formant and pellet lists, do not receive the keyboard command to log data.

---

2Due to a construction error in xassp, the zoom link may initially not work properly for data with a sampling rate other than 16 kHz. A quick way around this bug is to zoom in, then out (simply type \( \text{z} \)).
3.4. COMMON XASSP WINDOW PROPERTIES

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>The point in time the logged data items are associated with</td>
</tr>
<tr>
<td>LABEL</td>
<td>The label that is associated with the segment containing the chosen point in time</td>
</tr>
<tr>
<td>F0</td>
<td>The $F_0$-value in Hz</td>
</tr>
<tr>
<td>RMS</td>
<td>The energy value in dB</td>
</tr>
<tr>
<td>$F_n^a$</td>
<td>The frequency of the $n$th formant in Hz</td>
</tr>
<tr>
<td>$B_n$</td>
<td>The bandwidth of the $n$th formant in Hz</td>
</tr>
<tr>
<td>$A_n$</td>
<td>The amplitude of the $n$th formant in dB</td>
</tr>
<tr>
<td>UL</td>
<td>The position and velocity of the pellet on the upper lip</td>
</tr>
<tr>
<td>LL</td>
<td>The position and velocity of the pellet on the lower lip</td>
</tr>
<tr>
<td>$T_n^a$</td>
<td>The position and velocity of the $n$th pellet on the tongue</td>
</tr>
<tr>
<td>MI</td>
<td>The position and velocity of the pellet on the incisor</td>
</tr>
<tr>
<td>MM</td>
<td>The position and velocity of the pellet on the molar</td>
</tr>
<tr>
<td>FILE</td>
<td>The file that is associated with the window in that you pressed $w$</td>
</tr>
</tbody>
</table>

*a* $n$ is to be substituted by an integral number

Table 3.3: Keywords used in xassp log files

Choosing data for logging

The data items that are to be logged can be chosen by opening the corresponding windows. If you want $F_0$-values to be written to the log file for example, you have to load an $F_0$-file with the same base name as the speech signal file, or do an $F_0$-analysis of the speech signal file. xassp always logs data items from those windows that belong to the same speech signal, i.e. if a speech signal window has the focus, and you press $w$, the values from all associated windows are written to the log file. If a non-speech signal window has the focus and $w$ is pressed, xassp checks whether it contains analysed data. If so, it searches the speech signal window that this window refers to and logs the values from all windows associated with this speech signal. Otherwise, it uses the base name of the file to determine which other windows contain associated data. In order to log formant or pellet data, the corresponding lists must have been opened (see Sections 3.5.6 and 3.5.8).

Log file format

xassp log files consist of lines that contain TAB-separated items. The first line of a log file contains the items XASSP and DATALOG. Each time you press $w$ a line is appended to the log file. The items in each line are paired, the first item of a pair is a keyword, the second is the associated value. The keywords that are used in xassp log files are listed in Table 3.3. Note that for the pellet data the value consists of a comma-separated list of the $X$- and $Y$-position in mm and arrival and interpolated velocities in mm/s.

An xassp log file could look like this:

```
XASSP   DATALOG
TIME 0.62625 LABEL  #U  F0  0  FILE g0/1a000.r16
TIME 0.9475 LABEL  #A:  F0  131  FILE g0/1a000.r16
TIME 1.09375 LABEL  #T-N F0  136  FILE g0/1a000.r16
TIME 1.31625 LABEL  #U  F0  0  FILE g0/1a000.r16
TIME 1.4575 LABEL  #N+ F0  127  FILE g0/1a000.r16
```

Adding comment to logged data

If you wish to add some comment to the logged data, press <Ctrl><w> rather than $w$. After the data have been collected, a text window will appear in which you can enter your comment. After pressing Return or the OK button, the line you entered will be written to the log file after the line with the data. As identifier, the line is preceded by the word COMMENT and a TAB. The text window will not appear when no data have been found that can be logged.
3.5 Specific xassp window properties

This Section describes the properties of the different data display windows.

3.5.1 The speech signal window

In the speech signal window the speech wave of a selected signal file is displayed. The key and mouse button bindings for this window consist of the common bindings listed in Table B.1 and the special ones given in Table B.2.

Window menu

In addition to the common items listed in Section 3.4.3, this pop-up menu contains the following special items (see Figure 3.4):

Go to: (shortcut: <Alt>+g) Opens the Go to dialog box which allows you to jump to a certain point in time or to a label.

Analyse: Different analyses of the speech signal are possible: fzero, labels, sonagram, energy and section. The Analysis dialog box which opens upon selecting this entry in the pop-up menu is shown in Figure 3.7. You can choose one or more data types. After pressing the OK button an analysis of the speech signal is performed for each selected data type, and the result is displayed in a separate window. If the Align and Link button was selected, all xassp windows are aligned on the screen and all new analysis windows will be linked to the speech signal window.

Furthermore, you can limit the analysis to the region defined by the two brackets by selecting the Region button. This is only possible if both brackets have been set.

There is no analysis routine that generates a label file from a speech signal file, so xassp opens an empty label file if Labels was selected in the Analysis dialog box (see Section 2.6 for more details).

Save: Saves the speech signal under the file name that is displayed in the title area of the window. If this file exists, it is overwritten without warning.

Save as: Opens a file selection box in which you can select a file name for the speech signal to be saved in. As can be seen in Figure 3.7, this box also allows you to change the format in which the signal is to be saved. The file formats you can choose from are: KTH, AIFF, RIFF-WAV and CSL. For further information on speech signal file formats see Section 3.6.1. If the the file you select already exists, you are prompted whether it should be overwritten.

Edit menu

Apart from playing and analysing the speech signal, it can also be edited. This is important in, e.g., the generation of speech stimuli. For this task you need at least user level 40 (see Section 3.2). The edit menu shown in Figure 3.4 is called up by pressing <Ctrl>+RM. The menu items are set out in the following list:

Cut: Cuts the region defined by the two brackets out of the speech signal and saves it in the cut buffer.

Copy: Copies the region marked by the brackets into the cut buffer.

Paste: Inserts the contents of the cut buffer at the position of the temporary marker. If the temporary marker has not been set, a new window is opened with the cut buffer contents.

3.5.2 The label window

The label window lets you assign labels to distinct points in time. This is used to mark the segment boundaries of an utterance or to describe its prosodic structure as in segmental and prosodic labelling, respectively.

Apart from the common key and mouse button bindings listed in Table B.1, there are very many additional bindings which are listed in Table B.3. As you can see, most of these special bindings are included to facilitate labelling according to the ipds conventions. A number of them are also of more general interest, however.
3.5. SPECIFIC XASSP WINDOW PROPERTIES

Figure 3.7: Analysis (left) and Save as selection dialog box (right)

Window menu
The window menu of the label window has the following additional entries:

Go to: (shortcut: <Alt>g)  Opens the Go to dialog box which allows you to jump to a certain point in time or to a label.

Show as text: Opens a dialog box containing a label list. In each line of the list the time in seconds and the associated label are displayed.

Label font: This menu item opens the Font selection dialog box (see Figure 3.8) that lists the fonts that are defined in the font configuration file (see Section 4.2.3). You can either choose a label font from the list, or you can type a valid X font name into the Selection field.

Save: Saves the labels to the file that is given in the title of the window. If this file exists, it is overwritten without warning.

Save as: You can save the file under a different file name. If the file you selected exists you are prompted whether to overwrite it.

Edit menu
With <Ctrl>RM you get the label edit menu (see Figure 3.8), which has the following interactive entries:

Insert before: Insert a new label before the highlighted label.

Insert after: Insert a new label after the highlighted label.

Delete: Delete the highlighted label.

Edit: Edit the highlighted label.

Inserting a new label
If a new label is to be inserted at a certain point in time, the temporary marker must first be set onto this point. Then press <Ctrl>MM to insert the label at the temporary marker.

If there already are labels at the point in time at which you want to insert a new one, you should use the edit menu items Insert before and Insert after.
Moving a label

To move a label to a different point in time, the temporary marker must first be set onto this point. Then place the cursor between the temporary marker and the label to be moved. If the label to be moved has not been set yet, you have to place the cursor after the last label of the file. To actually move the label press \(<\texttt{Shift}>\texttt{MM}\). To edit the label before moving press \(<\texttt{Shift}>\texttt{RM}\). You can automatically move the label and mark it as deleted (append a – (minus)) with \(<\texttt{Alt}>\texttt{LM}\), or mark it as deleted and edit it before moving with \(<\texttt{Alt}>\texttt{MM}\). An uncertainty marker can be added before moving the label with \(<\texttt{Ctrl}>\texttt{LM}\).

Label syntax checks

If xassp is run with a user level below 30, a syntax check is performed for each label that is inserted, edited or moved. If the label is not valid, an error message appears and the edit process is aborted.

The following additional checks are done when a label is:

- **deleted**: Only prosodic labels, inserted labels or labels with a \texttt{KP} prefix may be deleted.
- **inserted**: The inserted label must be a prosodic label, a label that is marked as inserted or a label with a \texttt{KP} prefix. Prosodic labels may not be inserted at the temporary marker.
- **edited**: Labels that are marked as inserted may not be edited. To change such a label you must delete and re-insert it.

You may not change the label class (prosodic or segmental), the canonical label or $\$\$ (word-internal) prefixes. You are not allowed to modify the lexical stress markers or the – (minus) that indicates that the label has been inserted.

3.5.3 The fundamental frequency window

In this window the fundamental frequency contour of a speech signal is displayed. The window is called up when you load a file with \(F_0\) data in the xassp main dialog. You can also let xassp analyse the contour either with the window configuration \texttt{Prosodic}, which gives you a speech signal, its fundamental frequency contour and labels, or directly by choosing \texttt{Fzero} in the \textit{Analysis} dialog box in the speech signal window (see Section 3.5.1). In each case a separate window appears for the fundamental frequency.

The special key and mouse button bindings for this window (see Table B.4) provide the interface to the synthesis and manipulation options described below.
3.5. SPECIFIC XASSP WINDOW PROPERTIES

Figure 3.9: Fzero window menu, Set range and analysis options dialog boxes

Window menu

As you can see in Figure 3.9, there are several special menu items for this window:

**Draw mode**: Allows you to choose between three draw modes of the $F_0$-values: circles, squares and lines.

**Logscale**: By selecting this menu item you can toggle between a logarithmic and a linear display scale.

**Set range**: In the Set range dialog box that opens when selecting this entry, you can specify the minimum and maximum $F_0$-value of the display. If the Compute button is pressed, the minimum and maximum values are taken from the data. Note that this only affects the display, not the analysis parameters.

**Keep** (shortcut: $<$Ctrl$>$k) Makes a copy of the current $F_0$-contour and displays it in a different style and colour (default: dark green circles). A previously kept contour will be overwritten by this action.

**Clear** (shortcut: $<$Ctrl$>$c) Removes the copy of the $F_0$-contour, created by a previous Keep command.

**Re-analyse**: Opens the Fzero analysis options dialog box (see below). This item is only present when the $F_0$ data were analysed by xassp rather than loaded from a file.

**Synthesize** (shortcut: $<$Ctrl$>$MM) Plays a version of the speech signal, synthesized with the displayed $F_0$-contour. This item will only be present when the $F_0$ data were analysed by xassp rather than loaded from a file.

**Save**: Saves the $F_0$ data in the file that is shown in the title of the window.

**Save as**: Opens a file selection box which allows you to save the $F_0$ data in a different file. If the file you selected exists you are prompted whether it should be overwritten.

**Save synthesis**: Lets you save synthesis results in a file. A warning will be displayed if there are no synthesis data available. The format of the file will be the same as the one containing the speech signal from which the data were generated.

Analysis options

In the Fzero analysis options dialog box (see Figure 3.9), which you get by selecting the Re-analyse item in the window menu, you can specify the following options:
CHAPTER 3. USING XASSP

All/Region  If both brackets have been set, you can choose whether to re-analyse the whole signal or only the part in the marked region. Re-analysing a region is particularly useful if you want to correct an octave error (see Section 5.2).

Frame shift (ns):  Specifies the frame shift for the analysis.

Maximum F0 (Hz):  Specifies the highest F0-value to be analysed.

Minimum F0 (Hz):  Specifies the lowest F0-value to be analysed.

Noise amplitude:  Specifies the minimum speech signal amplitude considered relevant for analysis.

Please refer to Section 5.2 for more information on F0-analysis and its parameters.

Manipulation of fzero contours and synthesis

As support for prosodic labelling, options have been included in the fzero window to interactively manipulate the F0-contour and to play a version of the speech signal, synthesized with the displayed F0-contour. Both options are only available if the F0-contour has been analysed by xassp, not when it has been loaded from file. Synthesis will be described first because it doesn’t require modification of the F0-contour.

To generate and play a synthesized version of the speech signal, you should first mark the stretch to be processed by setting the brackets. Then, with the mouse pointer in the data area of the fzero window, simply press <Ctrl>M. Xassp will now run an LPC analysis of the speech signal in the marked region, then re-synthesize the signal using the displayed F0-contour and play the results. Depending on the length of the stretch, there may be a short pause while the analysis and synthesis are performed.

The synthesized signal will be kept in memory until you either change the bracket interval or the F0-contour. Subsequent synthesize and play commands will therefore produce an instant replay of the synthesis results. You can also easily compare the original signal with the synthesized version by alternately pressing MM (original) and <Ctrl>M (synthesis). When synthesis data are available in memory, they may be saved in a file. For this you should press <Alt>R, select the menu item Save synthesis and then select or specify a file name.

When manipulating (modifying) an F0-contour, it is useful to also have the original contour on the screen for comparison. You can do this by selecting the item Keep in the window menu or using the shortcut <Ctrl>k. A copy of the original contour will now be drawn using a different colour and drawing style. Note that only one contour can be kept. To define a modified version of the F0-contour, you now have to set so-called turning points. For this you move the mouse pointer to the time/frequency position where you want to set such a point (the values are indicated in the status line of the fzero window) and then press F8. A circle with a plus sign in its centre will now appear at the position of the mouse pointer. Especially if you have selected a high temporal resolution, you may notice that it will be shifted horizontally to the centre time of the F0-frame nearest to the cursor.

Once you have set at least two turning points you can draw a contour through these points. To do this, you first have to mark the part of the contour to be modified by positioning the brackets around it. If you know press <Ctrl>d, you will see that a new contour is generated by connecting the turning points with straight lines. With aid of the synthesis option mentioned above, you may now listen to the manipulated contour.

Figure 3.10 illustrates the properties of the manipulation:

• Unvoiced parts in the original contour are not affected by the manipulation.

• Parts of the contour before the first turning point and after the last one in the marked interval remain unchanged.

• Turning points may be set in unvoiced stretches.

• Interpolation between turning points is linear on the display scale (in this case logarithmic).

• Turning points outside the marked interval are ignored.
3.5. SPECIFIC XASSP WINDOW PROPERTIES

Figure 3.10: Fzero manipulation: small circles: original $F_0$-contour; large circles: turning points; solid lines: interpolated contour

When modifying a contour, you may wish to change one or more turning points. If only the frequency value is to be changed, simply place the mouse pointer at the new position and press F8. Because only one turning point can be defined for an $F_0$ frame, the previous one will be replaced by the new one. To remove a turning point, position the mouse pointer on it and press F9; to remove all turning points, press <Ctrl>F9.

The manipulation option can also be used to correct analysis errors that cannot be corrected by fine-tuning of the analysis parameters (see Section 5.2). How to correct octave errors may be clear from the above description. To correct a voicing error, i.e. a stretch that is incorrectly classified as unvoiced by the analysis, you should first enclose that stretch in the brackets. Then, set turning points within that range at the desired $F_0$-values, ensuring that there are turning points as close as possible to the brackets, but within their range. If you now press <Ctrl>+<Shift>, the turning points within the range will be connected by straight lines, despite the fact that the range was classified as unvoiced.

3.5.4 The energy window

The energy course of a speech wave can be analysed by clicking on Energy in the Analysis dialog box called up via the pop-up menu of the speech signal window (see Section 3.5.1). The menu item Align and Link has to be chosen to display the energy window linked with the speech signal window on the screen (see Figure 3.11). At present, no special key or mouse button bindings have been defined for this window; only the common bindings (see Table B.1) apply.

Figure 3.11: Upper window: speech wave, lower window: analysed energy course

Window menu

The following special menu items exist in this window (see Figure 3.12):
Figure 3.12: Pop-up menu and analysis options of the energy window

**Draw mode:** You can choose between three draw modes of the energy values: circles, squares and lines.

**Re-analyse:** Opens the Energy analysis options dialog box (see below). This item is only present when the energy data were analysed on-line rather than loaded from a file.

**Save:** Saves the energy data in the file that is shown in the title of the window.

**Save as:** Allows you to select a file in which to save the energy data. If the file you selected exists you are prompted whether to overwrite it.

**Analysis options**

In the Energy analysis options dialog box (see Figure 3.12) you can change the following options:

- **All/Region** If both brackets have been set, you can choose whether to re-analyse the whole signal or only the part in the marked region.

- **Window function:** Lets you select the analysis window function.

- **Frame size (ms):** Specifies the frame size for analysis.

- **Frame shift (ms):** Specifies the analysis frame shift.

Please refer to Section 5.1 for a description of the energy analysis and to Appendix A for an explanation of the terms that are used in the analysis descriptions.

3.5.5 The sonagram window

One type of spectral analysis of the speech wave results in a sonagram. A sonagram is a time-frequency-amplitude representation of the speech signal. At present, no special key or mouse button bindings have been defined for the sonagram window; only the common bindings (see Table B.1) apply.

**Window menu**

The window menu of the sonagram window has only one additional item:

- **Re-analyse:** Selecting this menu entry opens the Sonagram analysis options dialog box (see Figure 3.13). The analysis parameters that can be changed in this dialog box are described below.
3.5. SPECIFIC XASSP WINDOW PROPERTIES

Fig. 3.13: Sonagram analysis options

**Analysis options**

In the *Sonagram analysis options* dialog box, which is opened by selecting the menu item *Re-analyse* in the pop-up menu, you can modify the following parameters:

**DFT/LPC spectrum**: The DFT spectrum with a bandwidth of about 300 Hz corresponds to a wide-band sonagram, one with a bandwidth of about 50 Hz to a narrow-band sonagram. The LPC spectrum shows the spectral envelope of the speech signal, enhancing the formant structure.

**Window function**: Lets you select the analysis window function.

**DFT bandwidth (Hz)**: Specifies the bandwidth of the DFT spectrum.

**LPC eff. length (ms)**: Specifies the effective length of the LPC analysis window.

**LPC order**: Specifies the order of the LPC analysis.

**Pre-emphasis**: Specifies the pre-emphasis applied to the speech signal before spectral analysis.

**Gain (dB)**: Specifies the gain applied in mapping the spectral levels to grey levels.

**Range (dB)**: Specifies the difference between the highest and lowest spectral level to be displayed.

**no. of FFT points**: Specifies the length of the FFT used in calculating the spectra; it also sets the vertical resolution to *sampling frequency/number of points* [Hz per pixel].

**Shift (ms)**: Specifies the frame shift for the analysis; it also sets the horizontal resolution to *shift* [ms per pixel].

Please refer to Section 5.4 and to Appendix A for further information on spectral analysis.
3.5.6 The section window

The section window shows the spectrum of the speech signal at the cursor position. It can be opened by selecting Section in the Analysis menu of the speech signal window, or by the shortcut s in that window.

The section window differs from those described above because it doesn’t have a time scale. The time to which it corresponds is shown in the status line. When this time is preceded by an F, the section has been frozen (de-coupled from the cursor position in the speech signal window). If in this case, you position the mouse pointer in the section window the cursor line in the speech signal window will be placed at the corresponding time. Because most of the common key and mouse button bindings do not apply for the section window, all bindings are listed in Table B.5.

Window menu

Because of the different nature of the section window, its pop-up menu (see Figure 3.14) contains only a few of the common ones. The special items are described below:

Follow/Freeze: (shortcut: F) Selecting this menu item toggles the updating of the section by cursor movement in the speech signal window.

Keep: (shortcut: k) Makes a copy of the section that is currently displayed. This copy is then shown in addition to the current section allowing you to compare sections at different points in time or with different analysis parameters.

Clear: (shortcut: C) Removes the last section that was kept with a Keep command.

Re-analyse: Opens the Section analysis options dialog box shown in Figure 3.14. The analysis parameters you can change here are explained below.

Formants: Shows the Formants options dialog box that lets you choose the display of a formant list in a separate window (see Figure 3.15) and the number of formants to display. If you invoke a formant list, the section display will automatically switch to an LPC spectrum.
Analysis options

By selecting the menu item Re-analyse the Section analysis options dialog box shown in Figure 3.14 appears. Here you can specify the following parameters:

**DFT/LPC spectrum:** The DFT spectrum shows the Fourier spectrum of the speech signal, while the LPC spectrum shows the spectral envelope.

**Window function:** Lets you choose the analysis window function.

**DFT bandwidth (Hz):** Specifies the bandwidth of the DFT spectrum.

**LPC eff. length (ms):** Specifies the effective length of the LPC analysis window.

**LPC order:** Specifies the order of the LPC analysis.

**Pre-emphasis:** Specifies the pre-emphasis applied to the speech signal before spectral analysis.

**Gain (dB):** Specifies the gain applied in determining the highest spectral level to display.

**Range (dB):** Specifies the difference between the highest and lowest spectral level to be displayed.

**no. of FFT points:** Specifies the length of the FFT used in calculating the spectra. If set to 0, the program automatically determines the optimal number of FFT points, given the analysis parameters and the width of the display window (the latter to obtain a smooth spectrum). This will also be shown in the option box by the word *auto* in the field for the number of FFT points.

**Average spectrum; ms to average:** If this button is selected, the average spectrum of a region is calculated. The length of this region is given in the *ms to average* text field. The cursor always defines the centre of the region. Averaging is only possible for the DFT spectrum. It can be simulated for the LPC spectrum by increasing the effective window length.

For further information on spectral analysis, see Section 5.4. The subject of formant analysis is covered in Section 5.3. Appendix A contains important terms that are used in the analysis descriptions.

### 3.5.7 The palatogram window

In connection with the development of a new method for synchronous recording of speech signals and electro-palatograph data (Scheffers and Thon 1999), a display for these data has been included in xassp.

**Window menu**

The following special menu items exist in this window (see Figure 3.17):

**Follow/Freeze:** (shortcut: F) Selecting this menu item toggles the updating of the palatogram display by cursor movement in the speech signal window.


**Figure 3.16:** Palatogram display with a single and a multi-image configuration

**Attach:** Opens a dialog box in which you can select to which speech signal window on the screen the palatogram window should be attached. If the palatogram display is in follow mode, it will be centred around the time of the cursor in that speech signal window.

**Options:** Opens the Palatogram display options dialog box shown in Figure 3.17. The display parameters you can change here are explained below.

**Display options**

The Palatogram display options dialog box (see Figure 3.17) contains two groups of options: **FRAME** and **CONTACTS**.

The **FRAME** options allow you to specify the height and the width of the region in which a palate image is displayed and the margin around the region. In addition, you may specify the line width (thickness) of the box that is drawn around the palate image, corresponding to the cursor position. A thickness value of 0 means that no box will be drawn. All option values are in pixels.

The **CONTACTS** options let you to specify the way in which the contacts are drawn (circles or squares) and the size (diameter/width) of the contacts which were touched by the tongue (ON) and of those which were not touched (OFF). Sizes are in pixels.

**3.5.8 The articulogram window**

The articulogram window has been tailored to display the data from the University of Wisconsin X-ray microbeam speech production database (Westbury 1994).

**Window menu**

The following special menu items exist in this window (see Figure 3.19):

**Follow/Freeze:** (shortcut: f) Selecting this menu item toggles the updating of the articulogram display by cursor movement in the speech signal window.

**Attach:** Opens a dialog box in which you can select to which speech signal window on the screen the articulogram window should be attached. If the articulogram display is in follow mode, it will be centred around the time of the cursor in that speech signal window.

**Options:** Opens the Articulogram display options dialog box shown in Figure 3.19. The display parameters you can change here are explained below.
### 3.5. SPECIFIC XASSP WINDOW PROPERTIES

![Image: Pop-up menu and display options of the palatogram window](image1)

**Figure 3.17:** Pop-up menu and display options of the palatogram window

![Image: Articulogram display (left) and pellet list (right)](image2)

**Figure 3.18:** Articulogram display (left) and pellet list (right)

**List pellets:** Opens the *Pellet list* shown in Figure 3.18. For each of the pellets, this list shown the X- and Y-position, arrival velocity and estimated velocity for this position.

**Display options**

In the *Articulogram display options* dialog box (see Figure 3.19) you can change the following options: The *Articulogram display options* dialog box (see Figure 3.19) contains four groups of options: FRAME, LINE WIDTH OF SURFACES, SIZE OF PELLETS and COORDINATES.

The **FRAME** options allow you to specify the height and the width of the region in which an articulogram image is displayed and the margin around the region. In addition, you may specify the line width (thickness) of the box that is drawn around the image, corresponding to the cursor position. A thickness value of 0 means that no box will be drawn. All option values are in pixels.

The **LINE WIDTH OF SURFACES** options let you to define the width of the lines showing the pharyngeal wall and the hard palate (*Wall*) and of the tongue surface in pixels.

In the **SIZE OF PELLETS** group you can specify the diameter (in pixels) of the pellets on the lips, the teeth and the tongue. The size of the pellets on the tongue may be set to zero, in which case only the surface will be shown.

The **COORDINATES** options give access to the minimum and maximum values (in mm) of the coordinate system.
3.6 File handling

xassp can load data to be displayed from and save data to files. The data types that can be loaded and saved are

- Speech signal
- Fundamental frequency
- Energy
- Labels
- Palatogram
- Articulogram (X-ray microbeam data)

In the following sections the different file formats that are used in xassp are explained.

3.6.1 Speech signal

Xassp can handle speech signal files that have one of the formats that are explained in this Section.
3.6. FILE HANDLING

KTH file format

Files in KTH format consist of an ASCII header followed by binary data. The reason to use an ASCII header is that it can be viewed with a more or less command or a simple text editor such as vi.

The header of a KTH file normally looks like this:

```
head=1024<CR><LF>
file=samp<CR><LF>
data=int16<CR><LF>
msb=first<CR><LF>
chnans=1<CR><LF>
sftot=16000<CR><LF>
length=203722<CR><LF>
=<CR><LF><SUB><EOT>
<NUL>
<NUL>
```

The <CR> is a carriage return character, <LF> is a line feed character, and <NUL>, <SUB> and <EOT> are special ASCII characters. The dots indicate that the <NUL> characters are repeated until the end of the header is reached.

The header consists of simple keyword-value-pairs. The most interesting ones are head, which gives the header length in bytes, sftot, which gives the sampling frequency in samples per second, and msb, which gives the byte order.

AIFF/AIFC file format

The AIFF file format is mainly used on Apple Macintosh and SGI computers. It is a pure binary format with MSB-first byte-order. The AIFC file format is basically an extended version that allows compressed data. Since xassp cannot handle compressed data, files in AIFC format with uncompressed data can be loaded but will be saved in AIFF format.

RIFF-WAV file format

The WAV file format is used by Microsoft applications. It can be read by almost any application that plays or manipulates sounds. It is a pure binary format with MSB-last byte-order.

CSL file format

The CSL format is used by the Computerised Speech Lab, a combination of hard- and software for sampling and analysing speech data. It is a pure binary format.

3.6.2 Labels

xassp recognises two label file formats, MIX used internally at ipds and SAM, the label file format found on CD-ROMs#1–4 (IPDS 1994, 1995, 1996, 1997). A MIX file starts with a header, which contains the orthographic and the canonical form. The lines following the header consist of the identifier FR, the sample number, the label, the time in centiseconds and the time in seconds.

```
TEXT:
TIS010.
das wörde mir ganz gut passen . also <p> Dienstag ,
drei‘gister , Mittwoch , erster ? <{lum} ja , w}\’rd’
ich sagen . das k\’nnen wir ja dann so machen , dann
wér‘d’ ich da meine anderen Termine absa<T>
PHONET:
```
The time in centiseconds and the time in seconds are optional. Xassp also accepts the following file:

### TEXT:

```plaintext
GEF002.

<@> dann treffen wir uns um neun Uhr <#Raschenl> <Get{usch}>
<@> und <@> machen <@> <{hi}> bis maximal zwölf <Get{usch}>

### PHONET:

t: s: v: l: f: g: 
```

### CT 1

| FR  | 16000000 | #1k |
| FR  | 16000000 | #c: |
| FR  | 16000000 | #d |
| FR  | 16000000 | $N+ |
| FR  | 16000000 | $A |
| FR  | 16000000 | $#T |
| FR  | 16000000 | $#R |
| FR  | 16000000 | $#E |
| FR  | 16000000 | $#F |
| FR  | 16000000 | $#E0 |
| FR  | 16000000 | $#N |
| FR  | 16000000 | $#V |
| FR  | 16000000 | $#I:r+ |
| FR  | 16000000 | $#U |

This file is a so-called prototype file. It contains labels that have not yet been associated with specific points in time. These labels are then placed at the segment boundaries of the corresponding speech signal in the process of segmental labelling.

The SAM label file has a marginally different structure from that of MIX and the labels are in modified SAMPA (Wells et al. 1989). As with the MIX format, the SAM format contains an orthographic representation of the signal, a canonical transcription and individual lines containing time stamps and labels. In addition, following the canonical transcription, there is a variant transcription which represents a concatenation...
3.6. FILE HANDLING

tion of the labels. Lines containing the words oend, kend and hend serve to delimit the various chunks. The following is taken from the beginning of a SAM format label file:

g082a007.s1h
KAK007: <A> das k"onnen wir dann gleich dort machen .
ja , das ist ja praktisch . <P> und vielleicht
den Freitag vorher gleich<Z> <P> die Besprechung
da"fur machen .
oend
h: d a s+ k 9 n @ n+ v i: 6+ d a n+ q l 'aI C
d 'O 6 t m 'a x @ n . j 'a: , d a s+ Q I s t +
j a:+ p r 'a k t i s . p: Q U n t + f I l 'aI C t
d e: n+ f r 'a I t a: k f 'o: 6 #h "e: 6 g l 'aI C z:
p: d i:+ b @ s p r 'EC U N d a f 'y: 6 m 'a x @ n .
kend
c: h: %d - h a s+ k - h 9 n-m @ - n+ v i: 6-6+
d a n-N+ g l 'aI C d - h 'O 6 t m 'a x @ n . c:
j 'a: , d - h a s+ Q - I s t ++ j a:+
pr 'a k - h t - h I S . c: pr - h: Q - q U - n t ++
f I - l 'aI C t - h - z: d - h e: n+ f r 'a I t - h a: k - h
f 'o: 6 #h "e: 6 g - h l 'aI C z: p: d - h i:+
b @ s p r 'EC U N d - h a f 'y: 6 m 'a x @ - n - N .
hend
1 #c:
1 #h:
11985 ##d
12640 $-h
12823 $a
13412 $s+
15054 ##k
15715 $-h

3.6.3 Fundamental frequency

xassp knows only one file format for the fundamental frequency. It is an ASCII format. Each line consists of the time and the associated $F_0$- value separated by whitespaces (spaces, TABs, etc.). A part of a typical $F_0$ file looks like this:

::

0.840000 000.000000
0.850000 000.000000
0.860000 000.000000
0.870000 229.000473
0.880000 221.581589
0.890000 209.492630
0.900000 190.813736
0.910000 168.316666
0.920000 151.206406
0.930000 139.145554
0.940000 130.742325
0.950000 124.183624
0.960000 118.041351
0.970000 116.924339
0.980000 115.094978
xassp fundamental frequency files can optionally contain a header line, which must be the very first line of the file. It contains a file type identifier (XASSP_FZERO) and the sampling frequency.

3.6.4 Energy

The file format for energy data is exactly the same as for fundamental frequency, except for the header, in which the file type identifier XASSP_ENERGY is used instead of XASSP_FZERO.

3.6.5 Palatogram

The format of the palatogram data has been chosen such that it can very easily be processed further and, intrinsically carries information about the lay-out of the contacts on the artificial palate. It is a plain ASCII format. The first line contains the identification XASSP_PALATOGRAm, followed by the reference sampling rate in Hz and the frame duration in samples. The other lines consist of a series of fields, separated by tabulator stops. The first field gives the reference time of the scan of the electro-palatograph, the other the data per row of contacts on the palate. Each data field contains 0’s and 1’s separated by giving the state of the contacts in that particular row. The data recorded at ipds describe a top-view of the artificial palate. A line in such a file would look like this:

```
0.2898750 111111 11101111 11000011 10000001 1100000011 1000000001 10000000001
```

Data recorded with the ipds electro-palatograph and decoded with the program egpdecode will be in this format. When data from a Reading EPG2 or EPG3 or from a KAY Palatometer have been converted to this format, they can directly be displayed in xassp.

3.6.6 Articulogram

The CD-ROM distribution of the University of Wisconsin X-ray microbeam speech production database (Westbury 1994) contains the following data files:

- Speech signal (suffix)
- Accelerometer data (neck-wall vibration; suffix: .acc)
- Energy
- X-ray microbeam data (suffix: .xyd)

3.7 Printing

xassp provides the possibility to generate PostScript output from the data displayed on the screen. You can either configure the parameters for each window (position, size, font, etc.) yourself, or you can use configurations that define parameters and specify the layout for printing.

Choosing the window menu entry Print, which is present in all window menus, opens the Print dialog. Here you can choose, whether to send the PostScript output directly to the printer or to save it in a file. Further you can specify a print command and an output file in separate text fields.

Clicking the Select Windows button opens the Select Windows dialog box, where you can select windows, change their parameters and choose from different configurations, which are defined in the xassp print configuration file.

Clicking the Page Options button opens the Page Options dialog box, where you can change the page size, margins and the page orientation (landscape or portrait).

Press the Print button to generate the PostScript output. The Cancel button closes the dialog box.
3.7. PRINTING

3.7.1 Modifying page options

In the Page options dialog box you can modify the page orientation, page size and change the margins. Additionally, you can choose whether to generate EPS (Encapsulated PostScript) output, which can then, e.g., easily be included in \TeX\ documents.

You can choose between the page orientations Landscape and Portrait by selecting the corresponding toggle button in the Page orientation frame.

The page size can be changed either by selecting one of the pre-defined size from the Paper Size list or by selecting Custom in the Paper Size list and specifying the paper size in the Width and Height text fields (the values must be given in centimetres).

In the Margins frame you can specify the values for the left, right, top and bottom margins. The values must be given in centimetres. All position values in the option dialog boxes are relative to the margins.

3.7.2 Selecting windows

If you press the Print button in the Print dialog box without having selected any windows for printing, xassp automatically selects all open windows. To choose the windows that are to be printed yourself you have to open the Select windows dialog box and add the desired windows to the Selected windows list. To do this you select one or more windows from the Available windows list and click the Add button, which is located near the right edge of the dialog box. The selected windows are then moved from the Available windows list to the Selected windows list. You can remove windows from the list by selecting them in the Selected windows list and clicking the Remove button.

You can change the order of the selected windows by selecting one window in the Selected windows list and clicking the Up or the Down button to move the selected window up or down in the list. The window order in the Selected windows list is important if you let xassp compute the window positions, i.e. if you have selected a configuration.

3.7.3 Choosing a print configuration

Print configurations let you define parameters and layouts for printing so that you do not need to change the parameters yourself each time. You can choose a configuration by selecting a toggle button in the Configurations frame which is located in the Select Windows dialog box. Choosing a configuration changes the window parameters according to the specifications in the xassp print configuration file. You can manually change them afterwards. These changes are persistent as long as you do not select a different configuration.

If you choose the special configuration Default, which is not defined in the xassp print configuration file, all parameters are set to their default values.

3.7.4 Modifying window parameters

xassp lets you specify almost every parameter that can be defined in a configuration. To change the parameters for a certain window you have to select it in the Selected windows list and press the Options button if
you want to change the window parameters, the *Hscale options* button if you want to change the parameters of the horizontal scale for this window, or the *Vscale options* button if you want to change the parameters of the vertical scale for this window.

You can change the following parameters in the option dialog boxes:

- position (horizontal (X) and vertical (Y)),
- size (width and height),
- title,
- title font,
- whether to print a border,
- border width,
- line width (only Wave, Fzero, Energy and Label windows),
- line color or grey value (only Wave, Fzero, Energy and Label windows),
- label font (only Label window),
- print style (only Fzero and Energy windows).

The following parameters can be changed in the options dialog boxes for horizontal and vertical scales:

- font for tick labels,
- minimum space between two tick marks,
- margin between tick labels and tick marks.
Figure 3.21: Print output generated with xassp
Chapter 4

Configuring xassp

Since xassp is an application that runs under the X Window System, you can make use of the numerous possibilities of modifying X resources defined by xassp. With these you can control the appearance of xassp as well as parameters for the analyses xassp is capable of. You can also easily change key and mouse bindings.

Because of the complexity of xassp, X resources alone are not sufficient to control every aspect of xassp. Therefore, xassp supports three more configuration files to let the user define the buttons that are to appear in the xassp main dialog and the windows to be opened if one of the buttons is selected. Furthermore, you can specify default fonts to choose from, and frequent labels that are presented in the Edit dialog box in the label window.

4.1 X Resources

To be able to modify X resources, it is important that you understand how the xassp widgets are structured and which resources they define. In Figure 4.1 you can see what elements (widgets) a typical xassp window containing a speech signal consists of. Additionally, an xassp window is shown in Figure 4.2 with the designations used in Figure 4.1.

As you can see, every widget in an xassp window has a name and a class. If you refer to the widget class, all widgets belonging to this class are affected when changing some resource. By specifying the widget name, you are able to change resources for all widgets that are elements of the specified window type. The widget names consist of a prefix like wave and the actual name. Table 4.1 tells you which prefix to use for each window type.

X resources can be changed in the .Xdefaults file in your home directory. If the resource you would like to edit is not present in the file, you can copy it from the X application defaults file /usr/lib/X11/app-defaults/Xassp where the defaults for xassp resources are specified. The resources you can change and their possible values are listed in the manual pages for the xassp widgets, as well as in appendix C. It is also useful to have a look at the X manual page, especially the section called RESOURCES.

Each resource line in your .Xdefaults file changing an xassp resource normally has the following form:

<table>
<thead>
<tr>
<th>Window Type</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Signal</td>
<td>wave</td>
</tr>
<tr>
<td>Labels</td>
<td>labels</td>
</tr>
<tr>
<td>Fzero</td>
<td>fzero</td>
</tr>
<tr>
<td>Energy</td>
<td>energ</td>
</tr>
<tr>
<td>Section</td>
<td>section</td>
</tr>
<tr>
<td>Sonagram</td>
<td>sonag</td>
</tr>
<tr>
<td>Articulogram</td>
<td>artgram</td>
</tr>
<tr>
<td>Palatogram</td>
<td>palate</td>
</tr>
</tbody>
</table>

Table 4.1: Widget Name Prefixes
Figure 4.1: xassp window structure

```plaintext
Xassp*<widget name or class>.<resource>: <value>
```

If you want to modify the background colour of all widgets containing speech signals (widget class XspWave), you type:

```plaintext
Xassp*XspWave.background: white
```

If you want to change the font of the tick labels in the horizontal scale widget of all label windows (widget name labelsHorizontalScale), you need to add the following line:

```plaintext
Xassp*labelsHorizontalScale.labelFont: 9x14
```

To learn more about font specifications, you should once more consult the X manual page (section FONT NAMES).

Figure 4.2: xassp wave window
4.2. User Configuration Files

The main purpose of the configuration files described in the following sections is to cover configurable aspects of xasp that cannot be sensibly defined as resources.

xasp looks for configuration files in two places. The first is the default global configuration directory (/usr/local/lib/xasp), the second is the subdirectory .xasp of the user’s home directory. The files in the user’s home directory override any global defaults.

4.2.1 The display configuration file

The display configuration file is named config. Here you can specify the configuration buttons that appear in xasp main dialog, and which windows should be opened if one of them is selected. Furthermore, you can change global and configuration-specific defaults.

The file starts with a section in which you can specify values for global options. If you change options at this point, every window that you open in xasp is affected. An option consists of the option name and the value you want the option to take. At the moment you can specify yes or no. Refer to Table 4.2 for a list of valid options.

If you do not want the status lines to be displayed, you have to insert the following line at the beginning of the configuration file:

```
statusline no
```

After the global options section you can define an arbitrary number of configurations, i.e. groups of window types that can then be opened by selecting the corresponding configuration buttons. These definitions have the following form:

```
<identifier> {  
  [ <option> <value> ]  
  ...  
  [ <window type> {<suffixes>}, ]  
  ...  
}
```
The names to be used for the window types are identical to the prefixes listed in Table 4.1. The options that you specify affect only the window that is defined after the option line. If you specify multiple suffixes for a window type, you must separate them with commas. A suffix must always include a period. If you want the data of a certain window to be computed from the data of another, you have to specify an asterisk followed by the number of the window from which data are to be taken. In this case, the window number is the number of the window in this special configuration.

A simple configuration for segmental labelling could look like this:

```plaintext
Segmental {
  wave (.l16,.r16),
  sonag (*1),
  labels (.mix)
}
```

As you can see, the sonagram is computed using data from the first window in the configuration, which is the speech signal window.

If you want only one status line and one scroll bar, you can use the following:

```plaintext
Segmental {
  scrollbar no
  wave (.l16,.r16),
  statusline no
  scrollbar no
  sonag (*1),
  statusline no
  scrollbar yes
  smallscale yes
  labels (.mix)
}
```

In this configuration the scroll bar is only displayed in the label window, and the status line is only present in the speech signal window. Additionally, a small scale is created above the scroll bar in the label window.

### 4.2.2 The label configuration file

The labels configuration file is named `labels` and can be found in one of the default directories (global configuration directory or directory `.xassp` in the user’s home directory). It contains labels that are offered in the edit dialog box in the label window. Each line of the file can contain an arbitrary number of labels, which must be separated by TAB characters. Additional whitespaces (blanks, line breaks, etc.) are not allowed.

In the edit dialog box in the label window, each line in the configuration file is shown as a row of buttons that the user can select. This has the same effect as typing the label and pressing the OK button (see Section 3.5.2).

To make `xassp` create the buttons that are shown in Figure 2.4 your label configuration file must contain the following:

```plaintext
###-ma<TAB>$-ma
S-zi<TAB>$-"<TAB>$"'
###-k<TAB>$-p:<TAB>$-v:
#-k<TAB>$-l:<TAB>$-g:
#-s:<TAB>$-s:<TAB>$-q
```
4.3. ADMINISTRATIVE CONFIGURATION FILES

4.2.3 The font configuration file

The font configuration file is named fonts and resides in the global configuration directory or the subdirectory .xassp of the user’s home directory. It contains a list of fonts that the user can choose from in the Font Selection dialog box in the label window. It maps user-defined font names to X font names. Refer to the X manual page for information on X fonts.

Each line of the file contains the user-defined name and the X Window font name separated by a tab. It is important, that no additional whitespaces appear in the file.

To map the name default to a 24 point Courier font, you just add the following line:

```
default<TAB>-*courier-medium-r-*240-*
```

Be aware that a font that you give the name default is not automatically taken as a default font in the label window. If you want to change the default for this, you have to modify the corresponding resource (Xassp*XspLabels.labelFont).

4.3 Administrative configuration files

4.3.1 The xassp users file

The default path to this file is /etc/xassp_users. It contains a list of users that are allowed to use xassp. For each user it defines the default user level and the maximum user level.

Each line in this file has the following structure

```
<login name>:<default user level>:<maximum user level>
```

If, for example, the user with the login name Tux has the following entry

```
Tux:10:10
```

he or she receives a default user level of 10, which cannot be increased by any command line parameters. The user Tux can perform all tasks that are needed for prosodic labelling.

If this user wants to do segmental labelling as well, the entry will have to look like this:

```
Tux:10:20
```

This allows the user to increase the default user level to a maximum user level of 20, and so to perform all tasks that are necessary for segmental labelling.

To allow this user to work at all user levels, including those not yet defined, the entry should be changed to e.g.:

```
Tux:0:1000
```
Chapter 5

Analyses

This chapter gives an overview of the analyses included in xassp. The descriptions are kept brief for the general analyses, details of which can be found in speech signal processing textbooks like Markel and Gray (1976). The $F_0$-algorithm is described in more detail because it is a method developed at IPDS. Most signal processing terms used below are described in more detail in Appendix A. Those of you who are not too familiar with signal processing can find good introductions in Rosen and Howell (1991) and Ladefoged (1996). The latter book also includes chapters on Fourier analysis, digital filters and LPC analysis. Beware, however, that especially these chapters unfortunately contain many typing errors.

5.1 Energy analysis

For showing the course of the signal strength over time, an energy analysis can be performed. As a measure of energy the Root Mean Square (RMS) value is calculated. In formula:

$$s_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (w_n a_n)^2}$$

where $s$ is the sampled signal, $w$ the window function, and $N$ the frame size.

You are advised to use a non-rectangular window function and a window overlap of about 50% for a smooth curve. The window function is not very critical and is by default set to Hamming. Defaults for frame size and frame shift are 20 ms and 10 ms, respectively.

5.2 $F_0$ or pitch analysis

xassp contains a version of the Schäfer-Vincent periodicity detector (Schäfer-Vincent 1982,1983). This detector operates on local extrema (minima and maxima) in the speech signal as potential period markers. Triplets of extrema of the same type are evaluated to mark two adjacent periods (a period twin). If a twin is detected, its period durations are compared to those of previously detected twins. Adjacent or partly overlapping twins with comparable period durations are linked to chains (cf. $F_0$-tracks). If a chain exceeds a certain length (see point 2. below) and there exists no other, longer, chain with shorter period durations (this to suppress octave errors), the stretch covered by that chain is called voiced. For each sample within the stretch, an $F_0$ value is calculated on the basis of the period markers stored in the chain. The same is done for the samples within the twins that are subsequently appended to this chain. Samples between voiced stretches are called unvoiced and receive an $F_0$ value of 0. Note that more than one chain may exist at a time, but that only one of them may declare the stretch covered by it to be voiced. Chains die out when no more twins can be appended to them. At a final stage, the $F_0$ values are re-sampled at the frame rate. If at least half the samples within a frame have been declared voiced (have a positive $F_0$ value), the frame is assigned the mean of the positive $F_0$ values. Otherwise it obtains an $F_0$ value of 0.

The following properties of the algorithm should be kept in mind:
1. In somewhat difficult voiced stretches (irregularity in the initial part of a voiced stretch and voiced fricatives), the algorithm has a tendency to declare the stretch unvoiced.

2. Conditions for a chain to mark a voiced stretch include a minimum duration of the stretch of 30 ms. Also, the chain must contain at least three periods and three twins. Short stretches of voicing (e.g. in the vowel of ich) may therefore go unnoticed. If possible, select the analysis interval such that it begins and ends in an unvoiced stretch.

3. Creak may be declared voiced or unvoiced, depending on its regularity and duration.

4. The algorithm is optimised for speech signals and may therefore fail on some types of signals such as slightly noisy square waves.

User-definable parameters are \( \text{minFzero} \), \( \text{maxFzero} \), \( \text{noiseAmp} \), and \( \text{frameShift} \).

- \( \text{minFzero} \): the lowest \( F_0 \) to be evaluated (default 50 Hz). Decreasing \( \text{minFzero} \) slows down processing but has generally no adverse effect on the results. Increasing \( \text{minFzero} \) speeds up processing but has no effect on the results as long as it is lower than the actually lowest \( F_0 \) in the signal.

- \( \text{maxFzero} \): the highest \( F_0 \) to be evaluated (default 600 Hz). Decreasing \( \text{maxFzero} \) speeds up processing. However, since fewer potential periodicity markers will be evaluated, it may lead to voicing errors. Decreasing \( \text{maxFzero} \) may be necessary to suppress an octave error. Contrary to most other \( F_0 \) analyses, \( \text{maxFzero} \) should not be set to slightly above the correct \( F_0 \) but rather to slightly below its octave (i.e. the lowest erroneous value). Increasing \( \text{maxFzero} \) slows down processing, may in some cases give somewhat better results, but may also lead to octave errors.

- \( \text{noiseAmp} \): the amplitude threshold for an extremum to be considered a potential period marker (default 16 at 16 bit signal resolution). Extrema with amplitudes below this threshold are assumed to be due to background noise or weak unvoiced signals. Decreasing \( \text{noiseAmp} \) slows down processing and may in some cases lead to erroneous detection of periodicity in noisy stretches. Increasing \( \text{noiseAmp} \) speeds up processing but has no deteriorating effect on the results as long as it is lower than the actually lowest extremum in voiced stretches.

- \( \text{frameShift} \): determines the rate at which \( F_0 \) values are produced at the last stage, namely once per \( \text{frameShift} \) ms. The default value is 10 ms for proper alignment with other analysis data, such as energy contours. Decreasing \( \text{frameShift} \) may in extreme cases lead to gaps in the \( F_0 \) track. These may, for example, be caused by a phase shift in the transition from a vowel to a nasal. Note that \( \text{frameShift} \) may be reduced to one sample, because the \( F_0 \) values are determined at the sampling rate of the signal. Increasing \( \text{frameShift} \) mainly results in the voicing boundaries becoming increasingly inaccurate. Very short stretches that were declared unvoiced (e.g. in the occlusion of lenis plosives) may therefore disappear because the frames extending over such stretches now contain more voiced samples, resulting in the frames being declared voiced.

### 5.3 Formant analysis

If an LPC spectrum is selected in Section, a formant analysis may be invoked. In this analysis, local peaks are searched in the LPC spectrum. By parabolic interpolation through the three adjacent FFT points that define such a peak, the peak frequency, its bandwidth and amplitude are estimated. These values will be listed in a dialog box and may be logged in a file. How many peaks will be searched for, can be defined in the section pop-up menu under Formants. There, you will also find a toggle button for switching formant calculation on or off. The formant data will continuously be updated with a change in the spectrum in the section window.

When analysing formants, you are advised not to set the number of FFT points to auto (or 0) because this would result in the formant data becoming dependent on the width of the Section display window. The number of FFT points should rather be set such that a spectral resolution of between 20 and 50 Hz is obtained (e.g. 512 for a sampling rate of 16 kHz). In general, but specially if you are interested in formant amplitudes, you should set Pre-emphasis to 0 and choose an odd LPC order to have one coefficient free for modelling the overall spectral tilt (see Appendix A).
5.4 Spectral analysis

Underlying both Sonagram and Section are two spectral analyses, viz. calculation of the Fourier spectrum of the signal via the Discrete Fourier Transform (DFT) and calculation of the spectral envelope of the signal via an LPC analysis. In both cases, the signal is first pre-emphasised and multiplied by the window function specified. For the DFT spectrum, the size of the window follows from the specified bandwidth with a correction for the window function used. For the LPC spectrum, it follows from the specification of the effective length of the window and the window function (see Appendix A).

The DFT spectrum is then calculated in a straight-forward way using an FFT. For the LPC spectrum, the LPC coefficients are calculated from the windowed signal. From them, the inverse LPC filter is constructed and an FFT is applied to obtain the spectrum of the transfer function of this filter. In both cases, the FFT coefficients are converted to spectral power levels. The parameters gain and range determine how the power levels will be mapped on your screen: range defining the difference between the lowest and the highest level to be displayed; gain determining the highest level relative to an internal reference.

If both a sonagram and a section are displayed for a speech signal, setting the analysis parameters for the section to exactly the same values as those for the sonagram will result in a true cross-section of the sonagram.
Appendix A

Definition of signal processing terms

**Bandwidth**: The bandwidth parameter determines which details will be resolved in a spectral analysis. Generally, there is a trade-off between the bandwidth (frequency resolution) and the temporal resolution. The better the frequency resolution (smaller bandwidth) the worse the temporal resolution. For display purposes such as sonagrams and sections, we take the product of the bandwidth and the effective window length to be 1. Thus a bandwidth of 400 Hz corresponds to a temporal resolution of 2.5 ms. In a frequency detection task, such as a formant or an $F_0$-analysis, it is more appropriate to use a product value of 2. In order to analyse $F_0$ values down to 50 Hz, for example, the effective window length in a frequency-domain analysis must thus be at least 40 ms.

Typical values for a sonagram are about 300 Hz (wide-band analysis in which the harmonics are not resolved) and 50 Hz (narrow-band analysis showing the harmonics as horizontal bars). Note that for very high-pitched voices, a 300 Hz bandwidth may still be too narrow and need to be increased to remove harmonics. There is a lower limit to the bandwidth, determined by the window function and the number of FFT points. If you need a smaller bandwidth than this limit, choose a less compact window function (e.g. a Hamming window) or increase the number of FFT points.

**Discrete Fourier Transform (DFT)**: As the name implies, the DFT is the Fourier transform for discrete signals such as sampled speech signals. Usually, a short-term DFT is calculated. For $N$ input samples, the DFT delivers $N$ (complex) Fourier coefficients. Thus, the frequency resolution of the DFT equals the sampling frequency divided by $N$. For example, a 512 point DFT at a sampling rate of 16 kHz has a frequency resolution of about 31 Hz. The DFT is usually calculated using the Fast Fourier Transform (FFT). Note that calculating a short-term DFT violates the requirements for applying a Fourier transform. The signal must therefore be multiplied by a tapering window function to reduce spectral distortion due to this violation.

**Effective Window Length**: The application of a (non-rectangular) window function, puts more emphasis on the signal near the centre of an analysis window than near the edges. Hence, the length of the window is effectively reduced. There are several definitions of the effective length of a window, depending on the application. Unless specified differently, xassp uses the equal bandwidth criterion. That is: the effective length of a window is the length of a rectangular window whose transfer function has the same bandwidth. The reduction factor varies between about 1.3 for a triangular window and 2 for the most compact Blackman and Gaussian windows (see Harris (1978) for more details). The bandwidth definition used in this case is the Equivalent Noise Bandwidth.

**Equivalent Noise Bandwidth (ENB)**: The Equivalent Noise Bandwidth (also called Equivalent Rectangular Bandwidth or ERB) of a window function equals the integral over frequency of the power spectrum of the window function, divided by the maximum level of the power spectrum. The term Equivalent Rectangular Bandwidth refers to the fact that this bandwidth is the same as the width of an (ideal) rectangular filter with the same peak power gain and the same integral over frequency of its power spectrum (see Figure A.1). The term Equivalent Noise Bandwidth refers to the fact that, if the window function were treated as a filter, it would output the same noise power as the (ideal) rectangular filter when both are excited by the same white noise (see Harris (1978) for more details).
Fast Fourier Transform (FFT): The FFT is a computer algorithm for calculating a DFT. It exploits symmetry properties in the calculation of the Fourier coefficients that occur when the number of coefficients (generally called the number of points) is a power of 2, to drastically reduce computation effort. This reduction is such, that even on modern computers and despite the rather strong restrictions on the number of points, the FFT is nearly always used to calculate a DFT.

Frame: Since the speech signal is by nature non-stationary, most analyses are short-term. This means that they process only a short signal stretch at a time. During this stretch (or frame as it is usually called) the signal is assumed to be quasi-stationary. Processing parameters involved are the frame size, the frame shift and, for most analyses, a window function and an effective window length.

Frame shift: In a frame-based analysis, the analysis window is shifted along the signal in discrete, fixed steps. In other words, the global signal features are analysed at a constant rate (the frame rate). In order not to leave parts of the signal unanalysed, a certain overlap should exist between adjacent frames. In general, an overlap of 50% or more is recommended, hence the default value of 10 ms for a frame size of 20 to 25 ms. Smaller values (larger overlap) will result in smoother contours at the cost of higher data rates. Larger values may result in data loss, especially for very compact windows like the higher-order Blackman and Gaussian.

As resource and in menus, frame shift is entered in milliseconds as the most convenient unit. Internally, however, it is set to the nearest integral number of samples. Therefore, the frame shift used may differ somewhat from the value you specified, especially for small values at low sampling frequencies.

Frame size: The frame size or length of the analysis window determines the frequency resolution of the analysis. The larger the frame size, the better the spectral resolution. However, a large frame size will also result in smearing of fast transients such as plosives. In other words, the larger the frame size, the worse the temporal resolution.

For analysis of global features, such as LPC, formants or energy, a generally recommended compromise is a frame size of 20 to 25 ms. For very high-pitched voices a smaller frame size may be used to get a better temporal resolution. For very low-pitched voices a larger frame size may be necessary to reduce variation in the analysis results of voiced stretches due to the relative position of the frame with respect to the pitch period.

As resource and in menus, frame size is entered in milliseconds as the most convenient unit. Internally, however, it is set to the nearest integral number of samples. Therefore, the frame size used may differ somewhat from the value you specified, especially for small values at low sampling frequencies.

Gain: In sonagrams and sections the gain parameter is used to shift the display range upwards or downwards. A positive gain value shifts the sonagram more towards black and the spectrum in a section upwards. A negative value more to white and down, respectively. Note that gain does not influence
the spectral levels themselves, but only which range of levels will be displayed. Gain is expressed in dB relative to an internally calculated estimate of the maximum spectral level. The default value of 0 dB should give a reasonable plot, provided the signal values span about the full dynamic range of the A/D converter. The latter may be achieved by setting autogain on in your config.

**Linear Predictive Coding (LPC):** LPC has basically been developed for data compression in signal coding. In speech research, it is typically used for estimating the spectral envelope of the speech signal, which is then assumed to represent the transfer function of the vocal tract. Underlying LPC is the assumption that a sample value can be predicted on the basis of a linear combination of a number of previous sample values. In formula:

\[ s_n = \sum_{m=1}^{M} a_m s_{n-m}, \quad \text{with } M \text{ the prediction order} \]

Since, in the analysis, the sample value \( s_n \) is known, the prediction error

\[ e_n = |s_n - \hat{s}_n| \]

can be calculated for the set of prediction coefficients \( a_1 \ldots a_M \). For each frame, the LPC analysis determines the prediction coefficients in such a way that the energy of the error signal within that frame is minimal. The prediction coefficients then form the coefficients of a digital filter that optimally flattens the signal spectrum. In other words, it will have anti-resonances where the signal spectrum has resonances. The inverse of this filter can therefore be used to estimate the spectral envelope of the signal. The LPC analysis method used in xassp is called the autocorrelation method and uses the Durbin recursion to determine the prediction coefficients on the basis of autocorrelation coefficients (for more details, see e.g., Makhoul 1975 or Markel and Gray 1976).

**LPC order:** The LPC order (or number of coefficients) determines how closely the inverse LPC spectrum models that of the signal. Since, in speech analysis, we are mainly interested in the spectral envelope, the optimal order relates to the length of the vocal tract of the speaker and the sampling frequency used (see Markel and Gray 1976). The rule of the thumb is: sampling frequency in kHz + 2 or 3 for male voices and about 10% fewer for female voices. Thus, for a sampling frequency of 16 kHz: 18 or 19 for a male voice and 16 to 17 for a female voice. Note that two coefficients are needed to model one resonance (formant). Therefore, in the above example for a male voice, maximally 9 formants can be modelled, which corresponds to the rule of the thumb: 1 formant per kHz bandwidth. The extra formant follows from a potential nasal formant or the coefficients may be used to compensate for an anti-resonance. The odd coefficient may be included to model a very global aspect of the spectrum. If, for example, no pre-emphasis is applied prior to the analysis, this coefficient can model the overall spectral tilt.

If the LPC order is chosen too low, only the strongest resonances will be modelled and may undergo a frequency shift. If the order is chosen too high, strong harmonics will be treated as resonances.

**Pre-emphasis:** Pre-emphasis is used to change the overall spectral tilt of the speech signal. It implies a first-order filter with the transfer function:

\[ H(z) = 1 + uz^{-1}, \quad \text{with } -1 \leq u \leq 1. \]

A positive value of \( u \) amplifies the lower frequencies, while a negative value amplifies the higher frequencies. Typically, \( u \) is chosen near -1 (e.g., -0.95) to compensate the -6 dB/oct slope of the spectrum of voiced speech. Such a compensation is necessary in a sonagram in order to show the higher formants. Although a positive value should be used for unvoiced speech (which usually has a positive spectral tilt), the fixed negative value compensates for the lower amplitudes of such stretches. Note that some signal processing packages use a definition with a different sign for \( u \).
**Range:** The range parameter sets the difference between the highest and the lowest spectral level to be displayed in a sonagram or a section. For a sonagram, the default value is 50 dB. Choosing a smaller value may result in components of interest becoming invisible. Choosing a higher value will make the picture less distinct, due to the limited resolving power of grey levels. Moreover, too high a range may lead to side-lobes due to the window function becoming visible. For a Section, the range may be set larger because the visual resolution is far better and side-lobes are easier to recognise as such. The default value here is 80 dB, also because no pre-emphasis is applied.

**Window function:** In a frame-based analysis, the signal is implicitly assumed to be zero outside the frame. This is equivalent to multiplying the signal with a window function which equals 1 within the frame and 0 outside it. The *sharp edges* of this function cause distortions of the signal spectrum. The two main effects are:

- a broadening of the frequency components (width of the main lobe)
- the occurrence of spurious components (side-lobes)

The second effect is the most serious one, because, for a rectangular window, the maximum level of the side-lobes is only 13 dB below that of the main lobe. Therefore, the side-lobes of a single strong signal component may mask other components completely or suggest energy at frequencies where there are in fact no signal components. In order to reduce these side-lobes, a *tapered window* function (a function that gradually approaches 0 at its ends) is usually applied prior to the analysis. For any window function there is a trade-off between the width of the main lobe and the maximum level of the side-lobes: The lower the relative level of the side-lobes, the wider the main lobe. In sonagrams and sections, the resulting increase in the analysis bandwidth is compensated for. For more details on window functions and their figures of merit, see Harris (1978). In xassp, a large number of window functions are available. These include the *classical* triangular (Bartlett), hanning (or Hann) and Hamming windows, and two groups, viz. the Blackman and Gaussian windows. For most analyses, the Hamming window with a highest side-lobe level of -43 dB is a good choice. For the Sonagram, the minimum 3-term Blackman-Harris window has been chosen because of its even lower side-lobe level of -67 dB.
Appendix B

Key and mouse button bindings

This Appendix lists the key and mouse button bindings (shortcuts to specific actions) defined in the version of xassp described in this manual.

Table B.1 lists the bindings, common to those xassp windows that have a regular horizontal time scale. Most of these windows have some specific additional bindings, which are listed in separated tables. All bindings for the other windows (section, palatogram and articulogram) are given in their respective tables (B.5, B.6 and B.7).

<table>
<thead>
<tr>
<th>Key/Mouse button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Alt&gt;RM</td>
<td>post window menu</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;RM</td>
<td>post edit menu</td>
</tr>
<tr>
<td>Esc</td>
<td>close window</td>
</tr>
<tr>
<td>&lt;Shift&gt;LM</td>
<td>set temporary marker to cursor position</td>
</tr>
<tr>
<td>F3</td>
<td>move temporary marker to next zero crossing on the left</td>
</tr>
<tr>
<td>F4</td>
<td>move temporary marker to next positive zero crossing on the right</td>
</tr>
<tr>
<td>Backspace</td>
<td>clear temporary marker</td>
</tr>
<tr>
<td>F5</td>
<td>set left bracket to positive zero crossing left of the cursor</td>
</tr>
<tr>
<td>F6</td>
<td>set right bracket to positive zero crossing left of the cursor</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;F5</td>
<td>set left bracket to cursor position</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;F6</td>
<td>set right bracket to cursor position</td>
</tr>
<tr>
<td>&lt;Shift&gt;F5</td>
<td>toggle left bracket</td>
</tr>
<tr>
<td>&lt;Shift&gt;F6</td>
<td>toggle right bracket</td>
</tr>
<tr>
<td>Return &amp; Del</td>
<td>clear brackets*</td>
</tr>
<tr>
<td>Space &amp; p</td>
<td>play signal from start to end or stop playing</td>
</tr>
<tr>
<td>LM &amp; l</td>
<td>play from (temporary marker/left bracket/begin of window)b to cursor</td>
</tr>
<tr>
<td>MM &amp; m</td>
<td>play from (left bracket/begin of window) to (right bracket/end of window)</td>
</tr>
<tr>
<td>RM &amp; r</td>
<td>play from cursor to (right bracket/end of window)</td>
</tr>
<tr>
<td>&gt; &amp; i</td>
<td>horizontal zoom in</td>
</tr>
<tr>
<td>&lt; &amp; o</td>
<td>horizontal zoom out</td>
</tr>
<tr>
<td>a</td>
<td>horizontal zoom full out (show all)</td>
</tr>
<tr>
<td>b</td>
<td>display range between brackets (show region)</td>
</tr>
<tr>
<td>w</td>
<td>log data to file (only when data logging is enabled)</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;w</td>
<td>log data to file and add a comment line</td>
</tr>
</tbody>
</table>

*Return does not work on all machines
*b.xassp takes the first item from the slash-separated lists that is present when determining what to play. If a temporary marker is present and you press the left mouse button, this marker determines the begin of the region to be played. If the temporary marker is not present, but the left bracket is, then the region starts at the position of the left bracket. If neither marker has been set playing starts at the begin of the window.

Table B.1: Commonly used key and mouse button bindings in xassp
### APPENDIX B. KEY AND MOUSE BUTTON BINDINGS

#### Table B.2: Additional key bindings in the Speech signal window

<table>
<thead>
<tr>
<th>Key/Mouse button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>attach a section window</td>
</tr>
<tr>
<td>CursorUp</td>
<td>increase vertical resolution</td>
</tr>
<tr>
<td>CursorDown</td>
<td>decrease vertical resolution</td>
</tr>
<tr>
<td>Home</td>
<td>restore the original vertical resolution</td>
</tr>
</tbody>
</table>

#### Table B.3: Additional key and mouse button bindings in the Label window

<table>
<thead>
<tr>
<th>Key/Mouse button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CursorRight</td>
<td>move temporary marker to the next label on the right</td>
</tr>
<tr>
<td>emergence of the label last searched for</td>
<td>jump to previous occurrence of the label last searched for</td>
</tr>
<tr>
<td>Move label at the temporary marker</td>
<td>go to the last label that was set</td>
</tr>
<tr>
<td>Move label at the temporary marker</td>
<td>move label at the temporary marker and edit</td>
</tr>
<tr>
<td>Move label at the temporary marker</td>
<td>move label at the temporary marker and mark as deleted</td>
</tr>
<tr>
<td>Move label at the temporary marker</td>
<td>move label at the temporary marker and mark as uncertain</td>
</tr>
<tr>
<td>Insert a new label at the temporary marker</td>
<td>insert aspiration or creak label at temporary marker (after a plosive or glottal stop, respectively)</td>
</tr>
<tr>
<td>Insert a new label at the temporary marker</td>
<td>insert lengthening label at temporary marker</td>
</tr>
</tbody>
</table>

*Which label will be moved depends on the position of the cursor: If the cursor is to the left of the temporary marker, the label to the left will be taken, otherwise the label to the right.

#### Table B.4: Additional key and mouse button bindings in the Fzero window

<table>
<thead>
<tr>
<th>Key/Mouse button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Ctrl&gt;k</td>
<td>keep current $F_0$-contour</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;c</td>
<td>clear kept $F_0$-contour</td>
</tr>
<tr>
<td>F8</td>
<td>set/replace turning point at cursor position</td>
</tr>
<tr>
<td>F9</td>
<td>clear turning point at cursor position</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;F9</td>
<td>clear all turning points</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;d</td>
<td>interpolate voiced $F_0$-values between turning points</td>
</tr>
<tr>
<td>&lt;Ctrl&gt; &lt;Shift&gt;d</td>
<td>interpolate all $F_0$-values between turning points</td>
</tr>
<tr>
<td>&lt;Ctrl&gt; MM &amp; &lt;Ctrl&gt;m</td>
<td>synthesize with displayed $F_0$-contour and play</td>
</tr>
</tbody>
</table>

*Interpolation is linear on the display scale.
*Only between turning points in the interval between the brackets. No action if the brackets have not both been set.
*Only the interval between the brackets

#### Table B.5: Key and mouse button bindings in the section window

<table>
<thead>
<tr>
<th>Key/Mouse button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Alt&gt;RM</td>
<td>post window menu</td>
</tr>
<tr>
<td>Esc</td>
<td>close window</td>
</tr>
<tr>
<td>f</td>
<td>toggle update of section with cursor movement in the speech signal window</td>
</tr>
<tr>
<td>k</td>
<td>keep current section</td>
</tr>
<tr>
<td>c</td>
<td>clear last kept section</td>
</tr>
<tr>
<td>w</td>
<td>log data to file (only when data logging is enabled)</td>
</tr>
<tr>
<td>&lt;Ctrl&gt;w</td>
<td>log data to file and add a comment line</td>
</tr>
</tbody>
</table>
### Key/Mouse button | Action
--- | ---
<Alt>RM | post window menu
Esc | close window
f | toggle update of palatogram with cursor movement in the speech signal window
CursorLeft & l | scroll palatogram one image to the left
CursorRight & r | scroll palatogram one image to the right
<Alt>LM & c | centre palatogram around image under the mouse pointer
MM & p | synchronous play of speech signal and display of palatogram images
<Alt>MM | as above, but play at reduced rate (75%)
<Ctrl>MM | as above, but play at reduced rate (50%)

*Interval between brackets for single image display; Image range for multi image display.

**Table B.6: Key and mouse button bindings in the Palatogram window**

### Key/Mouse button | Action
--- | ---
<Alt>RM | post window menu
Esc | close window
f | toggle update of articulogram with cursor movement in the speech signal window
CursorLeft & l | scroll articulogram one image to the left
CursorRight & r | scroll articulogram one image to the right
<Alt>LM & c | centre articulogram around image under the mouse pointer
MM & p | synchronous play of speech signal and display of articulogram images
<Alt>MM | as above, but play at reduced rate (75%)
<Ctrl>MM | as above, but play at reduced rate (50%)
<Ctrl>W | log data to file (only when data logging is enabled)

*Interval between brackets for single image display; Image range for multi image display.

**Table B.7: Key and mouse button bindings in the Articulogram window**
Appendix C

X Resources

C.1 Resources for the Core widget

The Core widget is defined by the Intrinsics toolkit. Because many of its resources are seldom needed by the user, only the most frequently used ones will be listed. For a complete list see the Core manual page, which is normally included in common Motif distributions.

- **background**: Specifies the background colour for the widget.
- **backgroundPixmap**: Specifies a pixmap for tiling the background. The first tile is placed at the upper left corner of the widget’s window.
- **borderColor**: Specifies the colour of the border in a pixel value.
- **borderPixmap**: Specifies a pixmap to be used for tiling the border. The first tile is placed at the upper left corner of the border.
- **borderWidth**: Specifies the width of the border that surrounds the widget’s window on all four sides. The width is specified in pixels. A width of 0 (zero) means that no border shows.
- **height**: Specifies the inside height (excluding the border) of the widget’s window.
- **translations**: Points to a translations list. A translations list is a list of events and actions that are to be performed when the events occur.
- **width**: Specifies the inside width (excluding the border) of the widget’s window.
- **x**: Specifies the x-coordinate of the upper left outside corner of the widget’s window. The value is relative to the upper left inside corner of the parent window.
- **y**: Specifies the y-coordinate of the upper left outside corner of the widget’s window. The value is relative to the upper left inside corner of the parent window.

C.2 Resources for the XmPrimitive widget

The XmPrimitive widget class is defined by the Motif toolkit. Since many of its resources are seldom used, only the most important ones are listed. For a complete list see the XmPrimitive manual page.

- **bottomShadowColor**: Specifies the colour to use to draw the bottom and right sides of the border shadow.
- **bottomShadowPixmap**: Specifies the pixmap to use to draw the bottom and right sides of the border shadow.
- **foreground**: Specifies the foreground drawing colour used by Primitive widgets.
- **highlightColor**: Specifies the colour of the highlighting rectangle.
**highlightOnEnter**: Specifies whether the highlighting rectangle is drawn when the cursor moves into the widget. The default is False.

**highlightPixmap**: Specifies the pixmap used to draw the highlighting rectangle.

**highlightThickness**: Specifies the thickness of the highlighting rectangle.

**shadowThickness**: Specifies the size of the drawn border shadow.

**topShadowColor**: Specifies the colour to use to draw the top and left sides of the border shadow.

**topShadowPixmap**: Specifies the pixmap to use to draw the top and left sides of the border shadow.

### C.3 Resources for the XspMain widget

**cursorColor**: Specifies the cursor colour.

**dataColor**: Specifies the color that is used to draw the data.

**dataMargin**: Specifies the amount of empty space that is added to the left and the right of the displayed data.

**gridColor**: Specifies the colour of the grid or zero line.

**gridLength**: Specifies the length of a single grid line.

**gridSpace**: Specifies the horizontal space between grid lines

**leftBracketColor**: Specifies the colour of the left bracket.

**maximum**: Specifies the maximum data value for display.
minimum: Specifies the minimum data value for display.
pixPerSpl: Specifies the resolution used to display the data in pixels per sample.
rightBracketColor: Specifies the colour of the right bracket.
showCursor: Specifies whether a cursor should be drawn.
showGrid: Specifies whether or not to draw a horizontal grid.
sp1PerSec: Specifies the sampling frequency Hz.
tempMarkColor: Specifies the colour of the temporary marker.
tempMarkLineStyle: Specifies the line style of the temporary marker. This can be solid, onoffdash, or doubledash.
zeroLine: Specifies whether or not to draw a zero line.

C.4 Resources for the XspData widget

labelLinesColor: Specifies the colour of the label lines.
scrollbarIncrement: Specifies the fraction of the window width that the data should be scrolled if the scroll bar arrows are used.
scrollbarPageSizeIncrement: Specifies the fraction of the window width that the data should be scrolled if the user clicks inside the scroll bar.
setOnZeroCrossing: Specifies whether the brackets should automatically be set on the next zero crossing to the left of the cursor.
showLabelLines: Specifies whether label lines from a linked label window should be displayed.
valueUnit: Specifies the value to be used in the status line value label.

C.5 Resources for the XspHscale widget

labelFactor: Specifies a factor for multiplication of the scale labels.
labelFont: Specifies the font to be used for the tick labels.
labelMargin: Specifies the margin between tick labels and tick marks.
labelUnit: Specifies a unit that is appended to each label.
relTicLength: Specifies the length of the shorter tick marks in percent of the length of the longer ones.
ticHeight: Specifies the length of the longer tick marks.

C.6 Resources for the XspVscale widget

factor: Specifies the factor for multiplication of each label.
labelColor: Specifies the colour of the scale labels.
labelFont: Specifies the scale label font.
labelMargin: Specifies the margin between labels and tick marks.
precision: Specifies the number of digits to appear after the decimal-point.
ticColor: Specifies the colour of the scale tick marks.
unit: Specifies the unit to be appended to each scale label.
C.7 Resources for the *XspWave* widget

The *XspWave* widget class defines no new resources.

C.8 Resources for the *XspLabels* widget

- **bottomMargin**: Specifies the space in pixels between window bottom and labels.
- **highlightColor**: Specifies the colour to highlight labels with.
- **horizontalSpacing**: Specifies the space in pixels to be left between the label and the vertical label line.
- **labelFont**: Specifies the label font.
- **truncate**: Specifies whether labels that do not fit into the window are truncated, or other labels are overwritten if there is no space left.
- **verticalSpacing**: Specifies the vertical space in pixels between stacked labels.

C.9 Resources for the *XspFzero* widget

- **drawMode**: Specifies how to draw the data values. This can be lines, circles, or rectangles.
- **f0Precision**: Specifies the precision in Hz with which to show F0 values.
- **frameShift**: Specifies the shift of the analysis window in ms.
- **keepColor**: Specifies the colour of the kept F0 contour.
- **keepDrawMode**: Specifies how to draw the kept data values. This can be lines, circles, or rectangles.
- **logarithmic**: Specifies whether to use a logarithmic vertical scale.
- **maxFzero**: Specifies the highest F0 value (Hz) to be analyzed.
- **minFzero**: Specifies the lowest F0 value (Hz) to be analyzed.
- **noiseAmp**: Specifies the minimum speech signal amplitude considered relevant for analysis.
- **pointColor**: Specifies the colour of the turning points.
- **selectColor**: Specifies the colour of the selected turning point.

C.10 Resources for the *XspEnerg* widget

- **drawMode**: Specifies how to draw the data values. This can be lines, circles, or rectangles.
- **frameShift**: Specifies the frame shift.
- **frameSize**: Specifies the frame size.
- **wfunction**: Specifies the window function.
C.11  Resources for the XspSonag widget

**bandwidth:** Specifies the bandwidth of the DFT spectrum.

**gain:** Specifies the gain in dB.

**lpclength:** Specifies the effective length of the LPC analysis window.

**lpemode:** Specifies whether to compute an LPC (True) or DFT (False) spectrum.

**lpcoorder:** Specifies the order of the LPC analysis.

**numFFT:** Specifies the number of FFT points.

**numLevels:** Specifies the number of grey levels to be used.

**preemphasis:** Specifies the pre-emphasis.

**range:** Specifies the range in dB.

**shift:** Specifies the shift of the analysis window.

**wfunction:** Specifies the analysis window function.

C.12  Resources for the XspSection widget

**average:** Specifies whether to compute an average spectrum.

**avrLength:** Specifies the length of the signal part the average spectrum should be computed from.

**bandwidth:** Specifies the bandwidth of the DFT spectrum.

**follow:** Specifies whether or not the spectrum should follow the cursor position in the speech signal

**gain:** Specifies the gain in dB.

**listFormants:** Specifies whether formants should be analyzed and displayed

**lpclength:** Specifies the effective length of the LPC analysis window.

**lpemode:** Specifies whether to compute an LPC (True) or DFT (False) spectrum.

**lpcoorder:** Specifies the order of the LPC analysis.

**numFFT:** Specifies the number of FFT points.

**numFormants:** Specifies the number of formants to analyze

**preemphasis:** Specifies the pre-emphasis.

**range:** Specifies the range in dB.

**sectionColors:** This is a comma-separated list of colours that are used if more than one section is displayed.

**wfunction:** Specifies the analysis window function.
C.13 Resources for the XspPalate widget

dataStyle: Specifies the shape of the contacts. This can be ‘circles’ or ‘squares’.
followColor: Specifies the color of the palate frame in follow mode.
frameHeight: Specifies the height of the palate frame.
frameMargin: Specifies the margin between the palate frames and the window borders.
frameThickness: Specifies the thickness of the palate frame.
frameWidth: Specifies the width of the palate frame.
frozenColor: Specifies the color of the palate frame in frozen mode.
scaleFont: Specifies the font to be used for the time text.
sizeOFF: Specifies the size of the contacts when ‘OFF’.
sizeON: Specifies the size of the contacts when ‘ON’.

C.14 Resources for the XspArtgram widget

axesColor: Specifies the color of the coordinate system.
followColor: Specifies the color of the articulogram frame in follow mode.
frameHeight: Specifies the height of the articulogram frame.
frameMargin: Specifies the margin between the articulogram frames and the window borders.
frameThickness: Specifies the thickness of the articulogram frame.
frameWidth: Specifies the width of the articulogram frame.
frozenColor: Specifies the color of the articulogram frame in frozen mode.
lipsColor: Specifies the color of the pellets on the lips.
lipSize: Specifies the size of the pellets on the lips.
listPellets: Specifies whether or not to show a list with numerical pellet data.
maxXmm: Specifies the maximum X coordinate in mm.
maxYmm: Specifies the maximum Y coordinate in mm.
minXmm: Specifies the minimum X coordinate in mm.
minYmm: Specifies the minimum Y coordinate in mm.
scaleFont: Specifies the font to be used for the time text.
showAxes: Specifies whether or not to draw a coordinate system.
showTime: Specifies whether or not to print the time of the articulogram.
teethColor: Specifies the color of the pellets on the teeth.
teethSize: Specifies the size of the pellets on the teeth.
tongueColor: Specifies the color of the tongue surface.
tonguePelletColor: Specifies the color of the pellets on the tongue.
**tonguePelletSize**: Specifies the size of the pellets on the tongue.

**tongueThickness**: Specifies the thickness of the tongue surface.

**wallColor**: Specifies the color of the palate and pharyngeal wall.

**wallThickness**: Specifies the thickness of the palate and pharyngeal wall.
Bibliography


Index

aligning windows, 17
analyses, 47
analysis, 9–10, 20
   energy -, 47
   formant -, 48
   fundamental frequency -, 47–48
   - options, 10
   pitch -, 47–48
   redoing an -, 10
   spectral -, 49
analysis options
   energy -, 26
   fundamental frequency -, 23–24
   section -, 29
   sonogram -, 27
articulogram
   display options, 31
   key and mouse button bindings, 55
   window, 30–31
auto-save, 9
back-up
   - of label file, 9
bandwidth, 51
brackets, 15
buttons
   configuration -, 4
closing a window, 17
closing all windows, 17
command line
   starting xassp from the -, 3
command line options, 3, 11
configuration, 41
   - files, 43
   - buttons, 14
   display - file, 43–44
   fuft - file, 45
   label - file, 44
   local - directory, 11
cursor, 15
data
   - type, 3
   - type buttons, 13
data logging, 18–19
   - button, 14
data types, 14
DFT, 51
display configuration buttons, 14
display configuration file, 43–44
display options
   articulogram -, 31
   palatogram -, 30
effective window length, 51
energy
   analysis options, 26
   window, 25–26
equivalent noise bandwidth, 51
exiting xassp, 14
FFT, 52
file
   articulogram s, 36
   back-up of label -, 9
   changing the format of a speech -, 20
   configuration -, s, 43
   display configuration -s, 43–44
   energy -s, 36
   font configuration -, 45
   - format options, 13
   - formats, 32–36
   fundamental frequency -s, 35–36
   - handling, 32–36
   label -s, 33–35
   label configuration -, 44
   loading a -, 3
   log -, 19
   palatogram -s, 36
   saving labels to -, 8
   saving speech signals to -, 20
   saving speech signals to a different -, 20
   - selection box, 13
   speech signal -s, 32–33
   unsupported - format, 13
   users -, 45
font configuration file, 45
fourier transform, 51, 52
frame, 52
frame shift, 52
frame size, 52
fundamental frequency
   analysis options, 23

68
key and mouse button bindings, 55
manipulation of - contours, 24–25
synthesis with - contours, 24–25
window, 22–25

gain, 52

input focus, 16

jumping to point in time, 20, 21

key bindings
  articulogram, 55
  common, 55
  fzero, 55
  labels, 55
  palatogram, 55
  section, 55
  speech signal, 55

label
- configuration file, 44
  automatic saving of -s, 9
  back-up of - file, 9
  deleting a - , 7, 9
  fetching a -, 5
  fetching and modifying a -, 6
  inserting a -, 7, 8, 21
  jumping to a -, 7, 20, 21
  list, 6
  modifying a -, 6, 7, 9
  moving a -, 7, 22
  opening an empty - window, 10
  saving -s, 8
- syntax checks, 7, 22
undoing - changes, 7, 9

label window, 20–22

labelling
  prosodic -, 8–9
  segmental -, 4–8

labels
  key and mouse button bindings, 55
linear predictive coding, 53
linking windows, 17–18
log file format, 19
logging, 18–19
LPC, 53
LPC order, 53

main dialog, 3, 12
menus, 16–17
mouse button bindings
  articulogram, 55
  common, 55
  fzero, 55
  labels, 55
  palatogram, 55

section, 55
speech signal, 55

palatogram
  display options, 30
  key and mouse button bindings, 55
  window, 29–30
playing a speech signal, 6
pre-emphasis, 53
print options, 17
printing, 36
prosodic labelling, 8–9
range, 54
raw speech signal data, 13
redrawing windows, 17
region, 15
resources, 41, 59

saving
  automatic - of labels, 9
  - labels, 8
scale, 15
scroll bar, 16
section
  analysis options, 29
  key and mouse button bindings, 55
  window, 28–29
segmental labelling, 4–8
sonagram
  analysis options, 27
  window, 26–27
speech signal
  editing a -, 20
  key and mouse button bindings, 55
  playing a -, 6
  window, 20
status line, 15
synthesis
  - with fzero contours, 24–25

temporary marker, 16
typographic conventions, 1

undoing changes, 7, 9
unlinking windows, 17
user levels, 11–12
users file, 45

widget
  - class, 41
  - name, 41
  - structure, 41
window
  aligning -s, 17
  articulogram -, 30–31
  closing a -, 17
closing all -s, 17
- elements, 15
energy -, 25–26
fundamental frequency -, 22–25
label -, 20–22
linking -s, 17–18
- menus, 17
palatogram -, 29–30
redrawing -s, 17
section -, 28–29
sonagram -, 26–27
speech signal -, 20
- title, 15
unlinking -s, 17
window function
- for energy analysis, 47
window function, 54
window manager, 16

X resources, 41, 59

zero crossing
set brackets on -s, 15
set temporary marker on -s, 16
zooming, 17