

# **BOSS-IBB**

**Speech Synthesis Module Documentation  
for the BOSS Ibibio module**

Arne Bachmann

March 14, 2007 - June 10, 2007

---

## License

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

## Changes

**20070510 - R4** Little improvements to installation, corrections.

**20070419 - R3** New synthesis usage chapter, many additions and improvements.

**20070415 - R2** License added, index added.

**20070408 - R1** Initial version created.

## Acknowledgements

I'd like to thank all the persons that helped in creating the Ibibio module and its documentation: Stefan Breuer M.A. (Bonn, Germany), Prof. Eno-Abasi E. Urua (Uyo, Nigeria), Moses Ekpenyong (Uyo, Nigeria), Prof. Dafydd Gibbon (Bielefeld, Germany), Prof. Wolfgang Hess (Bonn, Germany). My sincere gratitude goes also to Nam C. Phamdo (Laurel, United States) and Axel Bußmann (Osnabrück, Germany) for providing me with additional source code and licensing it under the GNU public license.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Status of this document . . . . .	1
1.2	Purpose of this document . . . . .	1
1.3	Notes on formatting . . . . .	2
1.4	About BOSS . . . . .	2
1.5	About Ibibio . . . . .	2
<b>2</b>	<b>Installation</b>	<b>4</b>
2.1	Prerequisites . . . . .	4
2.2	Obtaining packages . . . . .	5
2.3	Installation . . . . .	6
<b>3</b>	<b>BOSS configuration file</b>	<b>7</b>
3.1	boss3conf.xml . . . . .	7
<b>4</b>	<b>Working with the tools</b>	<b>15</b>
4.1	Configuration . . . . .	15
4.2	General . . . . .	16
4.3	Corpus Preparation . . . . .	16
4.4	Intonation . . . . .	17
4.5	Prediction . . . . .	20
4.6	Database handling . . . . .	23
4.7	Evaluation . . . . .	25
4.8	Afterwork . . . . .	27
<b>5</b>	<b>Speech Synthesis with BOSS</b>	<b>28</b>
5.1	From the command line . . . . .	28
5.2	From a remote client . . . . .	28
5.3	Other options . . . . .	29
<b>6</b>	<b>File formats</b>	<b>31</b>
<b>7</b>	<b>Corpus</b>	<b>35</b>

*Contents*

---

7.1	Speaker and variety . . . . .	35
7.2	Stats . . . . .	35
7.3	Contained sentences . . . . .	36
	<b>Bibliography</b>	<b>39</b>

# 1

## Introduction

### 1.1 Status of this document

**T**HIS is revision 4 documentation for BOSS-IBB V0.1.0-1 (a BOSS 3.0 source code *adaption*) which is the predecessor of V0.2.0-1 (a BOSS 3.1 *module*), being under development. The document contains some descriptions on the software BOSS-IBB developed by the author in his master thesis. For contact and further information go to the BOSS website.

### 1.2 Purpose of this document

This document tries to provide the reader with background information about how to install, work with and extend or port the Ibibio module for BOSS. Beside the usage of all tools the reader will learn how the system works, and why certain programming steps were undertaken.

### 1.3 Notes on formatting

This document was created with the  $\text{\LaTeX} 2_{\epsilon}$  textprocessing system and the document class `scrbook` of the Koma-package (<http://www.komascript.de>).

Latin and German foreign words will be written in *slanted text* on first appearance. Special symbols, parameters and features will be written in a non-proportional `typewriter` font, single phonetic and other special symbols in **bold face**.

### 1.4 About BOSS

BOSS' development goes back to the large Verbmobil compound project, which was supported by the German Federal Ministry of Education, Science, Research and Technology (*Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie* BMBF). In Verbmobil many stakeholders from enterprises, universities and research centers were involved (Wahlster, 1997). Development for BOSS started in Dec. 1998 under the leadership of Karlheinz Stöber. Over time the whole system was redesigned and reimplemented with open and up-to-date data structures like XML (BOSS II since 2000). In Dec. 2005 BOSS 3 (development since Jul. 2004) was published on the internet at <http://sourceforge.net/projects/boss-synth> as open source<sup>1</sup>. Currently the release of v3.1 is under way. The whole project gains more and more stability, flexibility and versatility. If you would like to join the project, please don't hesitate to contact one of the project's members on Sourceforge or at the IKP.

### 1.5 About Ibibio

Ibibio is a Niger-Congo-language, spoken mainly in Akwa Ibom State and Cross River State in the southeastern part of Nigeria (Urua, 2001). For a general summary of Ibibio language characteristics, cf. Bachmann (2006) (master thesis in German) or some of the original sources (Urua, 2001, 2004; Gibbon et al., 2006; Essien, 1991). Ibibio is a register-terraced tone language, but tones aren't usually written in orthography. For the BOSS-IBB module, the BOSS client was

---

<sup>1</sup><http://www.opensource.org>

## *1 Introduction*

---

planned to do the linguistic preprocessing and provide this information (not yet developed). The main point in developing the Ibibio intonation module was to model the complex downstepping and downdrifting characteristics by a simple and adaptable mechanism. For tone symbols, **H,L,D,R,F** (high, low, downstepped high, rising, falling) will be used.

# 2

## Installation

### 2.1 Prerequisites

In order to properly install BOSS, the following software dependencies have to be satisfied:

- The GNU build tools: *m4*, *autoconf*, *automake* and *libtool*
- The MySQL database system, version 4.1 or higher (current version is 5.1)
- The Xerces-C XML parser library, version 2.4 or 2.5
- A recent version of Flex/Bison
- ESPS Speech tools for  $F_0$  extraction
- Some special libraries (e. g. Perl-compatible regular expressions)

Additionally to the standard BOSS dependencies the Ibibio module has the following requirements:



- Perl version 5 or higher
- Praat version 4 or higher
- XML::Simple (Perl parser package), install over CPAN (`perl -MCPAN -e shell then install XML::Simple`) or over PPM3 (`install XML-Simple`).
- Festival Speech Tools (Taylor et al., 1998; Black und Taylor, 1997), version 1.2.95 (for CART training)

## 2.2 Obtaining packages

### 2.2.1 Prerequisites

- **M4:** <http://www.gnu.org/software/m4/>
- **Autoconf:** <http://www.gnu.org/software/autoconf/>
- **Automake:** <http://www.gnu.org/software/automake/>
- **Libtool:** <http://www.gnu.org/software/libtool/>
- **MySQL:** <http://www.mysql.com>
- **Xerces-C:** <http://xml.apache.org/xerces-c/>
- **Flex/Bison:** <http://dinosaur.compilertools.net>
- **ESPS** <http://computing.ee.ethz.ch/sepp/esps-5.3.1-vj/>
- **Perl and Perl packages:** <http://cpan.org>.  
On most . . .nix systems perl should already be installed, under Windows OS you may want to install the ActiveState Perl: <http://www.activestate.com>
- **Praat:** <http://www.fon.hum.uva.nl/praat/>
- **Festival Speech tools:** <http://www.cstr.ed.ac.uk/projects/festival/>

### 2.2.2 Boss packages

- Basic german BOSS development package: `boss-3.x.0-1.i586.rpm`, [sourceforge.net/projects/boss-synth](http://sourceforge.net/projects/boss-synth)
- Ibibio module source code: `bossibb-0.1.0-2.tar.gz`, <http://sourceforge.net/projects/boss-synth>
- Ibibio MySQL precompiled database: `ibb_eno_database_0.1.0-1.tar.gz`

### 2.3 Installation

The standard BOSS distribution will usually be installed into `/usr/local/bin/` and its documentation into `/usr/share/doc/packages/`. You can change the install paths by augmenting the `-prefix=newpath` argument to the install script. Follow the install instructions for that distribution, available at <http://www.ikp.uni-bonn.de/dt/forsch/phonetik/boss/>. It contains also information about installing prepared ("dumped") MySQL database images.

For the Ibibio module, the following steps are necessary:

- Extract the Ibibio database files from `ibb_eno_database_0.1.0-1.tar.gz` into `/var/lib/mysql`. That should create a new subdirectory called `boss_ibb_eno`.
- Change the user rights by `chown -R mysql boss_ibb_eno` to mysql database access.
- Log in to the MySQL shell as root and set the access rights by 

```
GRANT ALL on boss_ibb_eno.* to 'boss'@'boss-server'
identified by 'boss';
```

 properly.
- Configure the `boss3conf.xml` or `boss3conf_win.xml` file to your needs: Set paths, select options accordingly to your database (if changes are necessary). See chapter 3.

# 3

## BOSS configuration file

### 3.1 boss3conf.xml

#### 3.1.1 Introduction

THE following lists contain descriptions of all necessary parameters of the boss3conf.xml file. The configuration file is an XML-file with `CONFIGS` as root element and several `INVENTORY` structures containing all the configuration data. Thus many language databases can be configured in one file<sup>1</sup>. Each line contains some peculiar attributes:

- **Comment** contains a short description of the parameter
- **Type** is a boolean value: 0 means boolean in `Flag`, 1 means arbitrary value in `Value` (?)
- **Flag** the boolean value of that option

---

<sup>1</sup>Note, however, that not all tools support more than one inventory per configuration file. Especially the Ibibio tools don't work like that, due to the use of `XML::Simple`.

- **Name** Name of the configuration option
- **Value** designated value of the configuration option, if not a boolean (then look for value in "Flag" TODO)

### 3.1.2 New features for the Ibibio module

**usewords** This boolean differs between synthesis on syllable and word level. The Ibibio module was originally designed to work only on syllable level. But while developing it turned out that the german BOSS system wasn't able to start synthesis below word level. So words were introduced into Ibibio synthesis, although the reliability of the word boundaries were very low. Note that this switch can't be simply turned on or off, because larger portions of the source code and the database organization have to be changed in order for it to work. Check out `tools/tg2xml/tg2xml.pl`, `tools/blfxml2db/create-tables.sql` and `libraries/boss_unitselection/` for further information. Default: **1**

**order** Polynomial order for  $F_0$  interpolation. This is the central value used by all scripts for  $F_0$ -Stylization. The number is crucial for fitting accuracy between measured  $F_0$  data values and stylized codebook values used in unit selection and (not yet implemented) manipulation. Consider Bachmann (2006) before choosing another value. Note that the number of basic values or polynomial coefficients is always one higher than the order. Default: **4** (quartic)

**cbsize** Codebook size. Usually a power of two (e.g. 16, 32, 64, 128 ...). The value determines the accuracy of chosen polynomial shapes, so consider the size seriously and try different sizes. Default: **64**

**tonecart** The path to the tone template CART (classification tree, codebook predictor). The CART is created by the **wagon** speech tool with the following parameters:

```
wagon -desc tone.desc -data all.predf -o ibb_eno_tone.cart  
-stepwise -stop 10 > tone.res
```

Thus the description and data file are converted into the CART file.

### 3 BOSS configuration file

---

You may use `tools/cart/wagon.bat` script for creation of both tone and duration trees. The **stepwise** parameters reevaluates the parameter order after each tree split. The **stop** value forces at least 10 values per leaf, which guarantees some data reliability. Default:  
`.../data/ibb/eno/runtime/boss_sampa/ibb_eno_tone.cart`

**durcart** Duration CART tree file location. Similar to `tonecart`. Notice that this tree is a regression tree. Default: `.../data/ibb/eno/runtime/boss_sampa/ibb_eno_dur.cart`

**codebook** Location of the tone template (polynomials) codebook. This file is created by `tools/vectorq/vq.pl`. Default: `.../data/ibb/eno/runtime/boss_sampa/ibb_eno.cb`

**codebook2** Location of the tone classes template (second) codebook. This codebook's values are considered if the preselection algorithm switches down from tone templates to tone class templates. This codebook is created from the first and represents the same data in lower accuracy. Maybe it would be better to create the tone classes codebook from the original data and classify the codevector from the first codebook onto it. Default:  
`.../data/ibb/eno/runtime/boss_sampa/ibb_eno.class`

**audiocompression** A boolean flag for notifying BOSS of compressed wave file format. Currently not used. Default: **0**

**mbrolapho** Flag for turning MBROLA-style output on. Default: **0**

**mbrolamap** This file contains a mapping between Ibibio Sampa phone labels and MBROLA-style representation. Default: `.../data/ibb/eno/runtime/boss_sampa/mbrola_map.txt`

**ucw\_bv** Unit selection cost factor for basic value RMS deviation, used only on syllable level. Default: **1.0**

**ucw\_f0** Unit selection cost factor for  $F_0$  value deviation, used only on word and phone level. Default: **1.0**

- ucw\_tone** Unit selection tone cost factor for Ibibio surface tone deviation. Check out Bachmann (2006) for further information on this cost. Default: **1.0**
- ucw\_phpos** Unit selection cost factor for phone position in syllable. Check out Bachmann (2006) for further information on this cost. Default: **1.0**
- ucw\_dur** Unit selection cost factor for duration deviation. Default: **1.0**
- ucw\_phrase** Unit selection cost factor for phrase mode deviation. Check out Bachmann (2006) for further information on this cost. Default: **0.5**
- tcw\_f0** Transition cost factor for  $F_0$ -difference at unit boundaries. Default: **1.0**
- tcw\_lpc** Transition cost factor for LPC weighting. **Currently not used.** Default: **1.0**
- tcw\_melc** Transition cost factor for MFCC weighting. Default: **1.0**
- ucw\_bv\_norm** Unit cost *normalization* (denominator of the weight factor) for basic values RMS (root mean square of differences). Default: **50.0**
- ucw\_f0\_norm** Unit cost normalization for  $F_0$  cost. This value is set for the overall corpus mean  $F_0$ . Default: **238.745**
- ucw\_tone\_norm** Unit cost normalization. Cf. `ucw_tone`. Default: **1.0**
- ucw\_phpos\_norm** Unit cost phrase position normalization. Cf. `ucw_phpos`. Default: **1.0**
- ucw\_dur\_norm** Unit cost duration normalization. Cf. `ucw_dur`. Default: **60.0**
- ucw\_phrase\_norm** Unit cost phrase position normalization. Cf. `ucw_phpos`. Default: **1.0**
- tcw\_f0\_norm** Transition cost normalization for  $F_0$ . Default: **238.745**
- tcm\_melc\_norm** Transition cost normalization for MFCC weighting. Default: **26.0**
- ucw\_intonation** **Obsolete.** Default: **1.0**

**espsdir, espsext** Location and file extension of the raw ESPS-files. Those files were created by the "get\_f0" tool of the entropic package and contain usually many columns of 1 ms-spaced short-time sound information. In the German and Ibibio BOSS, the ESPS files were pruned to only 4 columns (estimated  $F_0$ , probable voicing, energy rms, ac\_peak) and in the Ibibio module only one frame per 10 ms was calculated. The script used for extracting those four columns lies in `/tools/esps2f0/fea_print.layout`. Default: `.../data/ibb/eno/inventory/esps` and **.f0\_ascii**

**f0dir, f0ext** Location and file extension of the f0-files. The files contain information about voicing for each 10 ms-frame. TO create \*.f0 files from \*.f0\_ascii files, you may use the `prune-esps.pl` script. The difference to the values in the ESPS-files is that in the f0-files frame are calculated from the beginning of the first to the end of the last sentence in the TextGrid. This leads to a maximum of  $\pm 5$ ms deviation of starting points which is negligible since  $F_0$ -values are calculated from a larger window than 10 ms anyway. Default: `.../data/ibb/eno/inventory/f0` and **.f0**

**f0sdir, f0sext** Location and file extension of the f0s-files. Create them by the `extract_syl_f0s.psc` Praat script. These files contain the  $F_0$ -values from the \*.f0-files, but tab-separated and one line for each syllable. Additional information is given in the first columns: line number (1..), syllable SAMPA label, syllable duration. The  $F_0$ -values are set to zero, if the frame ist unvoiced.

**cbdir, cbext** Location and file extension of the codebook files. For each sentence a file is created. Each line represents a syllable and contains the categorization of its  $F_0$ -contour into the last created codebook and codebook class. The file has a header saying "cb" and "cbc", the last line must be "end". Every pause or syllable with unreliable values is categorized as **-1**. Default: `.../data/ibb/eno/inventory/cb` and **.cb**

**polydir, polyext** Location and file extension of the polynomial files. Every line represents a syllable. The first three tab-separated columns mean line number, SAMPA label and syllable duration. The next  $n$  columns contain the basic values for polynomial representation and the last  $n$  values the polynomial coefficients. The data is redundant but may be useful later on in the corpus,

for coefficients are easier to resynthesize with, but worse for distance (cost) measurements. Unsolvable polynomial systems are represented by  $2n$  tab-separated zeros. Default: `.../data/ibb/eno/inventory/poly` and **.poly**

**tgdir, tgext** Location and file extension of the Praat TextGrid files. Those files contain all original and additional segmentation and annotation data as received by Ms. Urua and extracted by scripts from the files. The tiers contain (from top to bottom): Sentences in broad transcription, phones and word boundaries in narrow transcription, surface tones, underlying tones, copy of tier 3, syllable boundaries and labels derived from tier 2 and 3, word boundaries with tones derived from tiers 2 and 3, narrow word annotation derived from tiers 2 and 7. In the third tier a #-sign shows, that there was a word boundary inside the syllable, which is bad for unit selection. The eighth tier contains additional annotation for unit selection: surrounding %-signs mean, that the "word"-segment really contains a word boundary. If not, it won't usually be used in unit selection. Default: `.../data/ibb/eno/inventory/textgrid` and **.TextGrid**

**xmlidir, xmlext** The location and file extension for the extracted XML files. The XML-files will be inserted into the database. The script `tools/tg2xml/tg2xml.pl` creates these files by calling `tg2xmlsub.pl` for each sentence.

**rawdir, rawext** The location and file extension for the raw sound files. Sound files in raw format don't contain a header. Instead they contain only sound samples. The default for BOSS is 16 kHz, 16 bit little-endian signed words with only one channel. Default: `.../data/ibb/eno/inventory/raw` and **.raw**

**pulsedir, pulseext** The location and file extension for the glottal pitchmark files. The files contain information about glottal cycles as extracted by the Praat-script `tools/bits/extract-pulses.psc`. Default: `.../data/ibb/eno/inventory/pulse` and **.PointProcess**

**zeropulsedir, zeropulseext** Information about zero crossings. **Obsolete.**

**compdir, compext** The location and file extension for compressed sound files. **Currently not used.** Default: `.../data/ibb/eno/inventory/cyclecomp` and **.ccmp**



**crawdir, crawext** The location and file extension for raw sound files before compression. **Obsolete**

**samplerate** Sample rate of the raw sound files in Hertz. Was taken into configuration when experimenting with wave compression. **Obsolete**. Default: 16000

**mapcycles** A file with glottal cycle information. Experimentally for wave compression. **Obsolete**. Default: `.../data/ibb/eno/map-cycles.dat`

**voicedcycle** This parameter is **obsolete**. `.../data/ibb/eno/voiced-segments.dat`

### 3.1.3 Old features that need explanation or underwent changes in value or meaning

**modules** The location and filename of the module order description. This file contains the names of all BOSS modules to be processed, e.g. `libbossintonationibb, libbossunitselectionibb, libbossconcat` (without library extension). Modules can be temporarily commented out by putting a #-sign in front of them, which will result in a warning at boss server startup. Default: `.../data/ibb/eno/runtime/boss_sampa/module_order.txt`

**cf** The file name addition for the unit selection cost library to be loaded on runtime. This results in using `cost_ibb.h` and `cost_ibb.cpp`???. Default: **ibb**

**wpsf** The location of the file defining word preselection. Default: `.../data/ibb/eno/runtime/boss_sampa/word_preselection.txt`

**spsf** The location of the file defining syllable preselection. Default: `.../data/ibb/eno/runtime/boss_sampa/syllable_preselection.txt`

**ppsf** The location of the file defining phone preselection. Default: `.../data/ibb/eno/runtime/boss_sampa/phone_preselection.txt`

**ccfile** The location of the file defining the phone context classes.

- no\_smooth** Flag to turn off segment boundary smoothing. Used by conman modules. Default: **0**
- adir, aext** Original parameters, in the Ibibio modules superceded by `rawdir`, `rawext`, but still used in some general (German) BOSS sources.
- db** The MySQL name of the database. Default: **boss\_ibb\_eno**
- fac\_dur** A factor for duration prediction, not used in Ibibio module??? Default: **1.0**
- fdir, fext** Location and file extension of the original ESPS-files. Those files contain all columns extracted by the Entropic-Tool `get_f0` for  $F_0$ -extraction. These files are converted by the `prune-esps.pl` tool, cf. `f0dir`. **Currently not used in Ibibio module.**
- list** The file list with all filenames (without file extensions) to process. Cf. chapter 4.

# 4

## Working with the tools

### 4.1 Configuration

**T**HE BOSS **configuration file** was build to speed up development as well as testing by putting often used parameters in one single location. But BOSS is a cross-platform system and needs to run on different operating systems. For the Ibibio tools there were special mechanisms installed to improve developing:

The **perl scripts** automatically detect the operating system (so far only Windows or Linux) in order to set the kind of path separator (/ or \) and select one of two configuration files (boss3conf.xml or boss3conf\_win.xml) from the user's current or home directory (in this order). The tools usually iterate over all or a part of all files in a directory (say, all \*.f0 files). This works over the `list` option in the configuration file: Its value is the location and name of a file that contains all raw filename to process (without extension). This file can be created by hand or using the `leave-one-out` tool (to be found in `/tools/bits`). The file name could be "files.list".

Note, however, that the perl scripts can handle only one inventory (corpus) per

configuration file due to the simple XML parsing.

## 4.2 General

From all the available files in the corpus, you can create but a selection, let's say for resynthesizing or testing purposes (leave-on-out method). You may use the tools `bits/loo1.pl` and `bits/loo10.pl` for that.

**loo1.pl** To create file lists that exclude only one sentence each, use this tool. The only parameter taken is the `list`-value from the BOSS configuration file. This "list"-file should contain *all* corpus files and is used as basis for the left-out file lists. When done, you may change the `list`-value in the configuration XML-file to one of the left-out files for further processing with other tools.

**loo10.pl** To create ten file lists with about one-tenth of all files, use this tool.

## 4.3 Corpus Preparation

The usual order of preparation is as follows:

1. Record a corpus
2. Annotate and segment the corpus, for example with Praat or Wavesurfer
3. Add missing tiers like syllable, word and word tone tiers by deriving already existing information. Scripts are `pre/convert/addsylys.psc` and `pre/convert/addwords.psc`.

**add-sylys.psc** Parameters include: From and to directory. Position of the tiers to use for data extraction, here: Sentence (1), phones (2), surface tone (3), underlying tone (4). Additionally an option exist to remove a possible fifth tier before processing the files. What the tool does:

- Change tone labels to simple one-letter format (rising LH → R, falling HL → F and downstepped high tone !H → D). Furthermore, sentence boundaries and empty intervals are converted to the "pause" symbol P.

- If there is a #-sign at the label boundaries, it is removed. Only in the middle of a label, the # is retained.

**add-words.psc** Has to be called after adding syllables, relies especially on tier 5 (syllable tones). Give source and destination directories. Three numbers are necessary: Tier for sentence, phone and tones. The script finds word boundaries by checking the phone tier labels for |, < and >. If there is a syllable boundary in the middle of a word, the word is marked with surrounding %-signs.

### 4.4 Intonation

1. As described in 4, the  $F_0$ -values must be obtained. At the IKP we used the `get_f0` tool from the Entropic speech toolbox, that creates `fea_ascii` files (not available for purchase anymore, I think). You can use also Praat for  $F_0$ -extraction with `polyreg/extra_syl_f0s.psc`, but this script is used originally for extraction of  $F_0$ -values of resynthesized data values, see `polyreg`. Remember to take one value every 10 ms. After extracting all information by the tool.
2. Extract only four columns from those files (historical reason: The german BOSS module does it alike). Use `fea_print.layout` file in combination with the Entropic toolbox in order to achieve that. File extension should be `f0_ascii_`.
3. One more step is setting the second and fourth column to fixed values (1 and 0) by `esps2f0/prune-esps.pl`, this converts from `f0_ascii_` to `f0_ascii` files.
4. Fundamental frequencies are extracted for every sentence (`esps2f0/esps2f0.pl` and syllable-wise (`esps2f0/extract_syl_f0s.spc`) in a single file (this is a BOSS-wide convention). See chapter 3 for more info.
5. Extract glottal pulse information by `bits/extract-pulses.psc`

6. Convert syllable fundamental frequencies to polynomial shapes by `polyreg/`  
`polyreg.pl`. The polynomial configuration is kept in a separate file  
`poly.conf`. For mathematical background, cf. Stoer (2005).

**prune-esps.pl** Give one parameter: Directory to process. Takes `f0_ascii_-files`  
and extracts only 2 columns plus sets two columns to fixed values. The output  
files get the file extension `f0_ascii`.

**esps2f0.pl** The perl file obeys global BOSS settings, default from/to directories are  
`.../data/inventory/esps` and `.../data/inventory/f0`.

**extract-syl-f0s.psc** This tool iterates over *all* files in a directory; for each entry it  
takes a `*.TextGrid` and a `*.F0-file` and creates a `*.f0s` file (format description  
in 3). You have to set the syllable tier (default: 6). The script has a debug  
option that writes status information to the console/information window.

**extract-pulses.psc** Takes two directories, raw wave source and pulse destination.  
It creates a Praat Manipulation object and extracts the PointProcess object,  
writing it to the output directory (file extension `*.PointProcess`).

**polyreg.pl** Parameters taken from configuration file: `f0sdir`, `f0sext`,  
`polydir`, `polyext` and `list`. This wrapper script calls `polysylloop.pl`  
for every item in the list.

**polysylloop.pl** Takes three parameters: A `f0s` file to read from, a `poly` file to  
write to and the polynomial configuration file `poly.conf`. After creating  
pairs of timestamps and (interpolated) data values, the `polysylsub.pl`  
script is called.

**polysylsub.pl** This is the most importing routine. The source code was derived  
from a script by Axel Bußmann, found on the internet under [http://www.  
we.fh-osnabrueck.de/fbwe/vorlesung/edv2/node8.html](http://www.we.fh-osnabrueck.de/fbwe/vorlesung/edv2/node8.html), used with  
kind permission. The extended script tests for several data ranges: Find erro-  
neous data values at the begin and end of the data range (currently not used,  
since interpolation is turned on anyway and only voiced values are sent to  
the regression). Then the data value range is checked. Arbitrarily, only val-  
ues in the range of (50...1000) Hz are valid. Other values are set to zero.

Just in case, the script checks, if at least 2/3 of all data values are  $\neq 0$ . The script takes as parameters an input file, output file, and the polynomial configuration (order, mode, duration normalization, symmetry and logarithm).

**poly.conf** The configuration file is used to ease iterating the calls of the polynomial calculation script. The parameters contained in the tab-separated columns are:

- Order of the polynomial (1 = linear, 2 = quadratic, 3 = cubic, 4 = quartic, ...)
- Output mode (1 = polynomial coefficients, 2 = linear interpolation between data values (not used anymore), 3 = normalized equidistant basic values *and* polynomial coefficients (currently used method). This allows for saving both polynomial representations in the database.
- Duration normalization (not necessary in mode 3, currently not used). When set to 1, the duration (x-axis) is normalized to  $[0 \dots 1]$ .
- Symmetry (not necessary in mode 3, currently not used). Moves the data values to the left on the x-axis for better regression of odd-ordered polynomials. For each  $F_0$  value, a time- $F_0$  value pair is written to a temporary file (`polyreg.temp`). For symmetry = 0, **time** is the number of 10 ms frames (hence a value of 3 means 30 ms), calculated by  $x/dur/100$ . For symmetry = 1, time is  $(x/dur/100 * 2) - ((n - 1)/dur/100)$  (simplify this expression if needed!), where  $n$  is the number of data values.
- Logarithmic (currently not used). Calculates polynomials from the natural logarithms of the data values for better data compression. According to my statistic evaluation, this doesn't seem to lead to better accuracy (Bachmann, 2006).
- Filename
- Interpolation (0 = off, 1 = middle, 2 = whole). Default: 2. When set to 1, zero values at the syllable boundaries are set to the first or last reliable value. When set to 2, additionally missing values (zero values) in the middle of the data values are linearly interpolated between the first and last border values surrounding the zero values.

The basic values are distributed equidistantly over the whole syllable, so that  $b(0)$  is at the left syllable boundary and  $b(n)$  at the right. For  $n = 4$ ,  $b(2)$  stands in the middle of the syllable. The left boundary is set to 0 and the right to 1. This way, a time normalization is undertaken. On purpose, there is no  $F_0$ (-amplitude) normalization.

**extract\_syl\_f0s** This script takes a directory and a tier number. In the directory, for each `*.wav` file, the TextGrid is read and a Praat Pitch object is created (and backed up on disc). In the syllable tier given, every 10 ms a data value is extracted. The script tries to find and compensate data errors (if values differ more than factor 1.5). Then a `f0s` file is written.

### 4.5 Prediction

1. Create a codebook of prototypical polynomial  $F_0$  syllable shapes by vector quantization (VQ). This is done by calling `vectorq/collect-poly.pl`, `vectorq/vq.pl` and `vectorq/redistribute-poly.pl`.
2. Determine the mean fundamental frequency of the corpus/speaker by using the tool `bits/find-avg-f0.pl`
3. Create a selection of files from the corpus to work with. This is used by CART creation or by resynthesis. At the moment, this works very fussily by concatenating all necessary TextGrids and wavefiles to one big file each by performing the two-pass process of `cart/concat-textgrids.psc` before `cart/concat-part-two.psc` and `cart/concat-waves.psc`.
4. Extract all prediction values from the TextGrids by `cart/extract-params.psc`
5. Remove pauses from the parameters file by `cart/filter.pl` via `cart/filter.sh` or `cart/filter.bat`.
6. Create codebook (parameters) description files `tone.desc` and `dur.desc`. See wagon's documentation for file format or just check it out.
7. Determine codebook by calling the wagon tool via `cart/wagon.bat`.



8. Check accuracy of created codebooks by `cart/testcbc.pl` and `cart/testdur.pl`.
9. If wanting to create a codebook with exactly the original polynomial shapes for evaluation, take `vectorq/plain-cb.pl`.

**collect-poly.pl** This script collects all polynomials (`polydir`, `polyext`) for all files given in the `list` parameter of the BOSS configuration file. The collection is written to `all.poly`. If you give any parameter to the script, strange polynomials will be retained, otherwise excluded (if  $F_0 = 0$  and not  $-4000 \leq F_0 \leq +4000$ ).

**vq.pl** is a wrapper script to call the vector quantization (`vectorq/vectorq`) with standard parameters (output filename = `all(.poly)`, polynomial order = 4 (dimension:  $10 = (4+1) * 2$  for mode 3), number of vectors = 64, ignoring the last 5 columns (polynomial coefficients for mode 3)). A status report is created under `all.vq`. Then the created codebook is renamed to `all.poly` and thus treated as if it were a polynomial shape file (although it's a description of a n-dimensional vector room). The second vector quantization is called (with its name (`all2.poly`), dimension  $5 = 4+1$ , 16 destination codevectors and ignoring the first column (it contains only a line number)). At last the output files are renamed to useful descriptions: `all.cb`  $\rightarrow$  `ibb_eno.cb` and `all2.class`  $\rightarrow$  `ibb_eno.class`. Those two files need to reside in `.../data/ibb/eno/runtime/boss_sampa/` and are read by the intonation library.

**redistribute-cb.pl** To write all codebook classification data (codevectors for currently used codebook) into `*.cb` files, use this script. It reads `ibb_eno.cb`, `all.class`, `ibb_eno.class` and uses the following parameters from the configuration: `polydir`, `polyext`, `cbdir`, `cbext` and `list` of course. The script classifies all codevectors from the main codebook into codevectors of the second codebook. Then it loops through all files given in the `list` option and writes the `cb` and `cbc` values. In parallel, it writes the same data to `all.cb_`. If any classification fails (for unknown reasons), the script writes an empty string instead of a number. Pauses are classified as `-1` for codevector and codevector class.

**find-avg-f0.pl** This script iterates over the complete file list, given in the configuration file. It reads all \*.f0 files (parameters `f0dir`, `f0ext`), averages over all non-zero  $F_0$ -values and prints out the result to the console.

**concat-textgrids.psc** This script needs as parameters the file list to process and the source directory from which to read (default: `.../data/ibb/eno/inventory/textgrid`). Additionally, all the tier numbers are given, that should be extracted and concatenated (without changing any duration or label information. Default: 1, 8, 6, 2, 7 and 5). All extracted data is written in one big temporary text file called `debug.dat` plus and information file `debug.info`, that are processed by the next script, because there were unsolvable Praat errors when trying to do the work in one pass (probably memory leaks or restrictions to the Praat object list or the like).

**concat-part-two.psc** Only the directory to write to is given. The script takes the temporary `debug.dat` and `debug.info` and creates a file called `all.TextGrid` in the destination dir. It removes the temporary files, if selected.

**concat-waves.psc** The parameters are the file list, from and to directories, if the script should read stereo files and if yes, if the right channel may be dropped. All files are concatenated into one big wavefile called `all.wav` in the destination directory. Meanwhile a temporary file list file `concat.tmp` is created and erased again.

**extract-params.psc** This file creates a tab-separated table file with all extracted parameters that are used in CART training. The output file name is `all.pref`. Necessary parameters are a concatenated Textgrid `all.TextGrid`, the concatenated codebook classification file with all syllable codevector `all.cb_` and the tiers of the different levels (1, 2, 3, 4, 6 and 7). There is also a logging option.

**wagon** This tool creates CARTs. The `wagon.bat` wrapper script creates tone template and duration prediction trees. Default: `ibb_eno_tone.cart` and `ibb_eno_dur.cart`.

**plain-cb.pl** This script reads a \*.poly file and writes a \*.cb and \*.class file of the same filename. It creates as many codevectors, as there are original syllable polynomial shapes. This is important if evaluating the polynomial regression accuracy only without the vector quantization.

## 4.6 Database handling

1. Convert all TextGrid files to XML files by `tg2xml/tg2xml.pl`.
2. Create preselection criteria by calling `bits/create-selections.*` or `bits/create-selections.bat`.
3. Optimize the boundary position by `optbounds/optbounds`. See original BOSS documentation.
4. Fill the XML files with  $F_0$ -values by `addf0/addf0`. See original BOSS documentation.
5. Fill the XML files with phone and tone context information by `addcontext/addcontext`.
6. Fill the XML files with boundary MFCC information by `melbounds/melbounds`. See original BOSS documentation.
7. Create a MySQL database and its tables. General information about the BOSS MySQL access can be found in the original BOSS documentation and in chapter 2. See tools `blfxml2db/create_tables.sql`, `drop_tables.sql` and `go.sql`.
8. Fill all tables with data from the XML files by calling `blfxml2db/main_ibb`.

**tg2xml.pl** This is a wrapper script using all information available so far to create very rich XML files to create the corpus/database from. Used parameters are `list`, `tgdir`, `tgext`, `cbdir`, `cbext`, `polydir`, `polyext`, `xmldir`, `xmltext`, `usewords`. The script class `tg2xmlsub.pl` for each file.

**tg2xmlsub.pl** This file creates an XML file in Latin-1 encoding (ISO-8859-1), since the system was developed under that charset. Change if necessary. Then

there are several subroutines to read the raw TextGrid file from the Perl script. Depending on the **usewords** option, word elements are written to the XML file, otherwise a direct jump to syllable level is undertaken. See note in chapter 3. Note: The sample frequency is hard coded here (16 kHz). Should be externalized to configuration file.

**create-selection.pl** has a described in the script, otherwise check out Bachmann (2006). The only parameter taken is an input file (e.g. phone.dat) and the script creates a file called `*._preselection.txt` that should be used by the BOSS preselection library. Script output is written to STDOUT, but can be redirected via pipe into a \*.key file. This file contains the information needed to create lookup keys for the MySQL database.

**addcontext** This tool is an extended version of the german `addcontext`. It uses `xmlmdir`, `xmllex`, `bext` and the `list` parameters from the configuration file. It adds tone and transcription information to neighbouring segments to the corpus xml files.

**create\_tables.sql** To create all database tables for current database style, change `create_tables.sql` to your needs and call it in MySQL (with the `source`-command). The directory name `blfxml2db` is an anachronism coming from german BOSS version, where a special BOSS label format (hence, BLF) is used). In Ibibio version we use the Praat TextGrids directly to convert to XML files, only the directory name was unchanged for historical reasons. Note, that missing tables columns will result in fatal errors when calling `main_ibb` or the BOSS server. It may result in errors that are difficult to understand. Always check your database status from MySQL command line or a similar tool.

**drop\_tables.sql** This script empties the database without destroying or removing it (it "drops" all tables).

**go.sql** Wrapper script for frequent use while developing: Calls `drop_tables.sql` first and then `create_tables.sql`.

**main\_ibb** Taken directly from the german version. Only the path names are changed to `xmlmdir`, `xmllex`.

## 4.7 Evaluation

- Calculate  $F_0$ -values of resynthesized sentences: This is non-trivial, since the unit selection may select any segments that fit its cost function. In the resynthesized sentence may have a different duration than the leave-one-out sentence from the corpus. Therefore, only the polynomial accuracy can be tested yet by `polyreg/resynth.psc` and `polyreg/rmse.pl`

`resynth.psc` for creating a resynthesis obeying the extracted polynomials, use this script. You can check for pure polynomial fitting accuracy by using `/tools/vectorq/plain-cb.pl` or using a special codebook (standard use). The parameters given to the script are:

- Polynomial configuration file. If this file is given, all other parameters covered by it are ignored and read from the file instead, including the filename. See chapter 6 for file format and in-depth description under `polysylsub.pl` in section 4.4.
- Polynomial order (*Polynomgrad*)
- Mode (*Modus*)
- Duration normalization (*Dauernormalisierung*)
- Symmetric translation (*Polynomsymmetrie*)
- Logarithmic output (*Logarithmisch*)
- Syllable tier (*Tier*), default **3**
- Tone tier (*TTier*), default **6**
- Show info before run (*Ask*), default **0**
- Draw syllable contours (*Draw*), default **0**
- Drawing range start (*fromSyllable*)
- Drawing range end (*toSyllable*)
- Debug switch *Debug*, default **0**. Currently not used anymore.

The file saves the Praat Manipulation and Pitch objects to disk for speed-up of later calls. This may consume a lot of disk space, so you may think of

switching it off. On the other hand you may save a lot of time when debugging.

The drawing option prints the resynthesized (normalized) syllable shapes into the Praat graphic window in three rows to 6 syllables each and saves it as an EPS file. Then it clears the output and continues for the next syllables (if any).

**rmse.pl** For evaluation purposes, the script calculates some difference measures between an original `f0s` and an resynthesized `polyf0s` file that underwent polynomial regression. The measures calculated are:

- Root mean square error RMSE:

$$RMSE_{F_0} = \sqrt{\frac{1}{N} \sum_{i=1}^N [F_{0\text{ synthesized}}(i) - F_{0\text{ original}}(i)]^2} \quad (4.1)$$

- Mean absolute error MAE (Rao und Yegnanarayana, 2004):

$$MAE(x, y) = \frac{1}{N} \sum_{i=1}^N |(x_i - y_i)| \quad (4.2)$$

- Mahalanobis distance for (x vs. y) and (y vs. x), defined as

$$md_{xy} = \frac{\sum |(x_i - y_i)|}{N * \sqrt{\frac{\sum x^2 - ((\sum x)^2 / N)}{(N-1)}}} \quad (4.3)$$

- Correlation coefficients (Sun, 2002):

$$r = \frac{SAQ_{xy}}{\sqrt{SAQ_x * SAQ_y}} \quad (4.4)$$

$$\gamma = \frac{\sum_i |(x_i - \bar{x})| * |(y_i - \bar{y})|}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \quad (4.5)$$

- Additionally the script calculates, if the agreement between the two compared intonation shapes is significant (with an  $\alpha$  depending on the sample size).

## 4.8 Afterwork

- Create a printable codebook (used in the appendix of my master thesis) by `cart/prune-cart.pl`
- Draw differences between original and resynthesized intonation curve by `polyreg/draw-diff.psc`
- Prepare a codebook for  $\text{\LaTeX}2_{\epsilon}$  integration by `vectorq/cb2tex.pl`
- Draw all codevectors to a range of graphic files by `gfx/drawCodebook.psc`.

# 5

## Speech Synthesis with BOSS

### 5.1 From the command line

**T**O start the synthesis process, there are two possibilities: Call the server locally or from a remote client. The communication between client and server is over a special protocol and XML in any case.

1. Start a server instance and connect to it via the localhost loopback-adapter from another console/terminal on the same machine

```
boss_server -conf \emph{boss3conf.xml} -file \  
emph{inputfile} -xml \emph{outputfile}
```

2. Start the server by giving it the necessary files on the command-line directly

### 5.2 From a remote client

To do.



### 5.2.1 Web interface

To do.

### 5.2.2 Java client

To do.

## 5.3 Other options

### 5.3.1 MBROLA

You can force BOSS\_IBB to produce MBROLA-compatible output files. In order to do that, you need

- a map file to convert the corpus' SAMPA phone symbols to MBROLA labels. This file can be found under `.../data/ibb/eno/runtime/boss_sampa/mbrola_map.txt`
- a MBROLA database that supports all phone symbols that your map file creates. Otherwise MBROLA may stop synthesis at the first unrecognized item. Sometimes it's possible to use databases of other languages, e. g. English or German.
- The unit selection module `libraries/boss_intonation/boss_intonation.cpp` of the Ibibio BOSS module provides a method called `loadMbrolaMap()`. It reads the file given under `mbrolamap` in the BOSS configuration file if the flag `mbrolapho` is set to **1**. The map is a STL-style map and uses the same datatype as all unit feature sets (`FMap`). The intonation module looks for a command-line option `-pho filename` and uses it for writing the MBROLA output to<sup>1</sup>. Otherwise the output file is named `boss.pho`. The method then interpolates the phone fundamental frequencies between the polynomial basic values. This should be sufficiently accurate for our purposes since the MBROLA synthesis engine must decide how to interpolate (linearly or polynomial and so on). The problem so far

---

<sup>1</sup>This doesn't work yet. Somehow the commandline parameters are not parsed here.

is, that the unit selection hasn't taken place when the intonation module is called. This leads to unequal syllable durations of MBROLA and BOSS synthesized outputs.

# 6

## File formats

### **cart**

See **wagon** documentation for this lisp-style file or check it out yourself.

### **cb**

- Tab-separated columns with header: **cb**, **cbc**
- Items: Integer numbers, codevectors  $-1 \dots N - 1$  with  $N$  codebook size. **-1** means pause
- Last line marked as **end**

### **eps**

This is the *encapsulated postscript* file format used by Praat and  $\text{\LaTeX} 2_{\epsilon}$ .

### **ibb\_eno.cb**

- Tab-separated columns without header
- Line numbers as first column 0...
- Items:  $n + 1$  float values for polynomial basic values of the prototype  $F_0$  shapes, with  $n$  polynomial order
- Last line marked as **end**

### **f0\_ascii**

- Four tab-separated columns without header
- Items: Single precision real numbers

### **f0**

- Tab-separated columns with header: frame, f0
- Items: Integer frame number 0..., single precision float  $F_0$  or zero

### **f0s**

- Tab-separated columns without header
- Line numbers as first column 1...
- Items: String syllable SAMPA label, double precision float syllable duration, single precision float  $F_0$  values for every 10 ms frame, last column always **end**
- Last line marked as **end**

### **mbrola\_map.txt**

This file contains the mapping information from SAMPA symbols onto MBROLA symbols. Use \*, if MBROLA doesn't support any label.

- Tab-separated columns without header
- Items: String Ibibio SAMPA phone label, String MBROLA phone label

### **poly**

- Tab-separated columns without header
- Line numbers as first column 1 . . .
- Items: String syllable SAMPA label,  $n + 1$  double precision float polynomial basic values,  $n + 1$  double precision float polynomial coefficients with  $n$  polynomial order
- Last line marked as **end**

### **polyf0s**

This format is the same as the `f0s` files, but represents the syllable-wise  $F_0$  values after polynomial recalculation.

### **poly.conf**

This file resides usually in `/tools/polyreg/`<sup>1</sup>.

- Tab-separated columns with header: `order, mode, durnorm, symm, log, filename, interpolate`
- Items: `[1..10]` order, `[1..3]` output mode, `[0..1]` duration normalization, `[0..1]` symmetry translation, `[0..1]` logarithmic output, `String filename`, `[0..2]` interpolation/smoothing mode

See `polysyllloop.pl` under 4.4 for in-depth information on the parameters.

---

<sup>1</sup>To do: Make it reside in a unix/linux standard directory like `~`.

## PointProcess

See Praat manual for further details.

- First line File type = "ooTextFile short"
- Second line "PointProcess"
- Third line empty
- Fourth and fifth lines start and end time in seconds
- Sixth line number of entries
- Items:  $n$  double precision time stamps

# 7 Corpus

## 7.1 Speaker and variety

**T**HE Ibibio corpus used so far was spoken by Prof. Eno-Abasi E. Urua, University of Uyo, Uyo, Nigeria. It's a variety spoken in Uruan local government area of Akwa Ibom State of Nigeria (the capital of Akwa Ibom state is Uyo). There are many other Ibibio dialects spoken throughout the country, for example Ibiono, Uruan, Uyo, Iman, Ikot Abasi/Mkpat Enin, Ikono (Urua, 2004) and Enyong, Central Ibibio, Itak, Nsit (Gordon, 2005).

## 7.2 Stats

There are 165 sentences in the corpus. Up till now, only 94 of them are segmented and annotated. Used in the synthesis are sentences 21–70, 91–120 and 131–134.

### 7.3 Contained sentences

- 21: /ke abaja uduak edijak utom umijakpo ake ukara nno mbonowo/  
 22: /njobio owo nseabasi ette/  
 23: /ke emum umiaj ndito akwa ibom edueehe mbet ufok utom emi/  
 24: /ke ekpesuk odo eerumme ebo ke ibaan esoj itak afara unamijakpo/  
 25: /ke idaha esukk enam mbeejeidem ebara edikoono akpa iduok ndito ufokjwet ukpeepjakpo mbara ibok mbakara mme nsajidem ndoj/  
 26: /dokta desi wiidism/  
 27: /mbok idakka do inno ke edinim utom emi no akpeene ntom/  
 28: /nniehe idotepen ke idem owo emi atoojoke utom omo je jakara ke esit/  
 29: /ekamba awetjwet ke ufokutom enamma jakpo ebara isoj mme uboo ufok aketaj ami/  
 30: /epe akedo arwaan ete joo esien abaaha ke ufok utom usuan mbuk ake ekebe ndise ujo/  
 31: /editwak nda ke mben usuj/  
 32: /amaakwooro obo no ikot abasi ejaak uwem ommo esim abasi ke ubok mbaak ebo edidij abasi/  
 33: /owo ibarake ikaj ikeene daja udok ikot ojooj/  
 34: /eeboppo ke idit keetekeet anaasime jakpo siben ufon isok ibaan/  
 35: /ekamba ase jakpo mbara kooberatif defedopmen bajk/  
 36: /mbaak ommo ekeme ediwu nda ke usuj ofono/  
 37: /ikiduojoj ke ija/  
 38: /no epe je ikot utom omo eewot njun jakem ommo ojoojoj/  
 39: /mbon unam nduuj ebene ewot ebo ke otu bidion owo ke afit unadot eedoojoke akpoikpoi uweene/  
 40: /amaanam ediojoj ke esai mme itie adiduuj udooj itaita ke ijiip mme owo mi ke sted njun/  
 41: /je ndito ete ebe ommo ke ini mme ebe ommo ekpaaha ekpoj ommo/  
 42: /eje ikisitreke editiik idem ke ini ke ini/  
 43: /ke imo itie nte asasawrek/  
 44: /obooj fikt attah amaakoom mbon ufokutom emi/  
 45: /ete joo akenofa amaabo no mbon se ibo mfon ami etim ekama okuk ami/  
 46: /mfon jems ej ake ufokutom usuan mbuk ake ekebe ndise ujo/  
 47: /ommo emaedoj idem ommo okpoho ke nnuun nte idiojoj edidiana keet/  
 48: /nsinia okon amaadian obo ke ufon ntuno odo ke adikpuro owo no akappa ofon/  
 49: /emaesin ebeeje ebo no ukara eno mmoto boos/  
 50: /anam no ntakejin esim idem ommo ke edinam mme ido ammiikpinaaha nte ommo enam/  
 51: /njun iminum ite ke epe obo nduok odudu/  
 52: /ojooj okoom mme ajam urua odo ke edimaana ntooj utom adiboo mme ufok urua odo ke idem ommo/  
 53: /enam nduuj ese me mme edinam se eduak ediofon die je ibaan/  
 54: /mbokoo inuj odo mbokoo edat/  
 55: /kristian kuujeeme ke ini ajwan/  
 56: /odooho komred emma eseoke/  
 57: /etak osuh kraiss amaajem itimme jakpoj utom epe akediihe/  
 58: /ekenimme ke umuahia/  
 59: /amaakop iduoesit abaja unoo ake boomb akesiine ke dekos/  
 60: /daja ekenooj mme jakpoon owo ke ufok utom usuan mbuk ake ekebe ndise okuk udeeme jakpodero ufok/  
 61: /ufok abasi odo ekesooboke ke isua itiaaba akeboijoke do/  
 62: /ojooj akot obo ke eesooj mme mbet akotte obo ke owojwaan odo ndo oto ke ujumme omo/  
 63: /aakpenamma no mme owo esifiik unen ibaan enieehe ke ntak edido owojwaan/  
 64: /saja saja ono item ke sains mme teknoroji/  
 65: /muusoppooj idem udakka ke mmi/



## 7 Corpus

---

- 66:** /enɔ mbio efiommoke ibaan uk̄peep ηk̄pɔ ke editre mme uto eduuwem ntom/  
**67:** /ibaaha owo se ik̄pikikan inək ejɛ ism ke itie utaj ikɔ/  
**68:** /ejɔɔɔ enam nduuɔ ekɪt ke enam mme uk̄puuɔ ηk̄pɔ/  
**69:** /ke andidiito ikan inam nɔ akaisaj asɪɔɔɔ ɔfɔɔ ufɔhɔ/  
**70:** /ekɪt ebo ke ek̄peme ibaan je ubɔik̄pa eedəɔɔɔɔɔ ke mme itie ekɔɔ ajwanake/  
**91:** /ediɔɔɔɔɔ ibio ibio nte akruwasan aasiak mme itie edibɔ mmoɔɔ esajukwak akanna ikie itiokeet/  
**92:** /ufɔk fedraad sekriteriad ke ujo amaabuuro eti eti ke idaha mbon napɛp ke ik̄pehe ujo/  
**93:** /abasi mbɔk jaaja ubon owo/  
**94:** /mma teedma ηwokedi/  
**95:** /ejauwɔt afɪt se akenam ke ndibe ajwa ajwa/  
**96:** /ideppedep/  
**97:** /ke ini ekeɔɔɔ ɔmmɔ mme mbuumo ke abaɔa itie odo/  
**98:** /ke mme mfina ake mme ibaan ufɔkndɔ ajaamaana afeere/  
**99:** /amaadido se ekokonno eda ekɪt koo/  
**100:** /abaaha ke ekamba ufɔkɔɔwɛt ntaifɔk nnamdi asikiwe ke ɔka/  
**101:** /ndion ke mbre adiiɔwam ke utɔ nneeɔɔɔɔɔ ami/  
**102:** /ke ntak inem mmm je ukɔt emi ɔmmɔ ekedatta nte ak̄pan ufan ɔmmɔ/  
**103:** /k̄pa ntro ηkɔ ke akaisaj anenekke atuut ɔfɔɔ ufɔhɔ ɔfɔk idem ɔmɔ/  
**104:** /ebo jak ukara enam mbet ebaɔa mme eduuwem ake miijakka ibaan ekop inemuwem je nsɔɔɔɔɔ/  
**105:** /tɔɔɔɔ ke ηkanika itiaita ubahausen mbaak ediɔɔɔɔ mme mme anamutom kɔfinen enam ujo eekenɔɔɔ/  
**106:** /mmuudiaha unɔ usan/  
**107:** /mme itie akeododo ke sted/  
**108:** /mma duisa ubɔɔɔ amaadian do obo ke ηketi esɛɔɔɔ odo ηk̄pɔ uto ake ɔsɔbɔke eti eti/  
**109:** /okop ujukkɔ abaɔa utom akaahaiso ke usɔɔ itu mbon usɔ mme arosukwu/  
**110:** /fɔn ido jem ηk̄pɔ mfɛn dia/  
**111:** /unek unek je uɔɔɔɔ ηk̄pɔ ke mme ufɔk udia uwem emi ekedidakka kwa efaak kwa efaak ke iduɔ/  
**112:** /ejɛ amaasua ɔnɔ ubiɔɔ utire utom/  
**113:** /akedooho ηkanika abee itiaba/  
**114:** /edet itohoakwa/  
**115:** /ekebaaha ke ndopnda ke ηk̄pɔ nte isua iba mme akanna odo/  
**116:** /ndion ek̄pak ufɔkmbet ukara sted ebo nɔ enam mbet/  
**117:** /ɔɔɔɔ ataj obo nɔ afɪt mbon ηka odo edian ubɔk emenne utom sednet/  
**118:** /kuufere ke idomo mme ke ukuut/  
**119:** /nsikak eduɔk amaataɔ ke adijeeɔɔ mbon abasi mbo nɔ esm ubɔk ke ek̄pat/  
**120:** /iibohoke ke iwuot afɪt ibaan/  
**131:** /itie unam spɔɔt/  
**132:** /ejɛ amaano ufɔk uman ake ufɔk abasi odo beed nnaa anieehe sibriɔ duopeba/  
**133:** /ejɛ amaabo ke ufɔkɔɔwɛt odo osiooɔɔ mme oku abasi ke mme nsio nsio ufɔk abasi mme ik̄pehe/  
**134:** /amaataɔ obo ke udɔɔ adik̄pɔɔɔ ajɔɔ ufɔk abasi odo ikitooho ndausɔɔ ake sibiɔt abasi/

### 7.3.1 Test sentences

This is the test set 8, when using the split-into-ten-sets tool in `l0010.pl`:

- 121:** /ana ekop nsɔɔɔɔɔ odo ɔɔhɔ ɔɔhɔ ke ifuuro uwem/  
**122:** /ekærɪku odo akwa owo eto/  
**123:** /mmoɔɔ idim ukana amaasidɛɔɔɔ/  
**124:** /usioudat usioudat/  
**125:** /ke ukara ebiooro adism uɔɔwam nnoɔ ufɔkɔɔwɛt uk̄peep mme ɔkwɔɔɔɔ ke mme utom ηkɔt adidɔɔɔ ufɔkɔwɛt

odo/

**126:** /eṗe ikifreke ediwakka ufɔk abasi ŋkeboṅ akam ŋṗuṅ ŋkɔm abasi ke uṅwam je ndausuṅ ɔmo/

**127:** /ete edet nduoṗoi/

**128:** /aṅwaan mme ebe emi emaewɔt ima ke adikpikke kek ndɔ nno keet eken/

**129:** /enam no ibaan enie nnnnnoidem/

**130:** /rattawu/

### 7.3.2 More stats

For testing purposes, I used the leave-on-out corpus 8, which excludes the test sentences given above. This results in 84 remaining sentences, 728 words, 1750 syllables and 3446 phones plus 244 pauses on all levels. The corpus' length is 9'46", of which 4'32" are pauses.

## Bibliography

- Bachmann, A. (2006): *Ein quantitatives Tonmodell für Ibibio. Entwicklung eines Prädiktionsmoduls für das BOSS-Sprachsynthesesystem*. Magisterarbeit, Universität Bonn.
- Black, A. W.; P. A. Taylor (1997): *The Festival Speech Synthesis System: System documentation*. University of Edinburgh, Scotland, UK: Human Communication Research Centre,  
<http://www.cstr.ed.ac.uk/projects/festival.html> (2006-08-11).
- Essien, O. E. (1991): "The nature of tenses in African languages: a case study of the morphemes and their variants." In: *Archiv Orientalni*, Bd. 59, 1–11.
- Gibbon, D.; E.-A. Urua; M. Ekpenyong (2006): "Problems and solutions in African tone language Text-To-Speech." In: *ISCA Workshop on Multilingual Speech and Language Processing (MULTILING 2006)*, Stellenbosch, South Africa: Center for Language and Speech Technology, Stellenbosch University, paper 014.
- Gordon, R. G., Hg. (2005): *Ethnologue: Languages of the World, Fifteenth edition*. Dallas, Texas: SIL International,  
<http://www.ethnologue.com> (2006-08-11).
- Rao, K. S.; B. Yegnanarayana (2004): "Intonation modeling for Indian languages."
- Stoer, J. (2005): *Numerische Mathematik 1. Eine Einführung - unter Berücksichtigung von Vorlesungen von F.L. Bauer*. Berlin: Springer, neunte Auflage.
- Sun, X. (2002): " $F_0$  generation for speech synthesis using a multitier approach."
- Taylor, P. A.; A. W. Black; R. Caley (1998): "The architecture of the festival speech synthesis system." In: *The Third ESCA Workshop in Speech Synthesis*, Jenolan Caves, Australia, 147–151,  
<http://www.cstr.ed.ac.uk/projects/festival/> (2006-08-11).
- Urua, E.-A. E. (2001): "The tone system of Ibibio." In: *Typology of African Prosodic Systems Workshop*.

## Bibliography

---

- Urua, E.-A. E. (2004): *Ibibio*, Nr. 34/1 in Journal of the international phonetic association, International phonetic association, Kap. Ibibio. 105–109.
- Wahlster, W. (1997): “VERBMOBIL: Erkennung, Analyse, Transfer, Generierung und Synthese von Spontansprache.” *Informatik 97 – Informatik als Innovation-motor Heidelberg* , 215–224  
<http://www.dfki.de/~wahlster> (2006-08-11).

# Index

<b>A</b>	
ActiveState .....	5
add-syls.psc .....	16
add-words.psc .....	17
Akwa Ibom State .....	2
audiocompression .....	9
Autoconf .....	5
Automake .....	5
<b>B</b>	
Bison .....	4, 5
BOSS	
about .....	2
German .....	6
installation .....	6
packages .....	6
boss3conf.xml .....	7
<b>C</b>	
cart	
file format .....	31
cb	
file format .....	31
ibb_eno.cb .....	32
cbdir .....	11
cbext .....	11
cbsize .....	8
ccfile .....	13
cf .....	13
codebook .....	9
codebook2 .....	9
collect-poly.pl .....	21
concat-textgrids.psc .....	22
concat-waves.psc .....	22
Corpus .....	35
create-selection.pl .....	24
create-tables.sql .....	8
create_tables.sql .....	24
Cross River State .....	2
<b>D</b>	
data structures .....	2
db .....	14
drop_tables.sql .....	24
durcart .....	9
<b>E</b>	
eps	
file format .....	31
ESPS .....	4, 5
esps2f0.pl .....	18
espsdir .....	11
espsext .....	11
Evaluation .....	25
extract-params.psc .....	22
extract-pulses.psc .....	18
extract_syl_f0s .....	20
<b>F</b>	
f0_ascii .....	11
file format .....	32
f0dir .....	11
f0ext .....	11
f0s	
file format .....	32
f0sdir .....	11
f0sext .....	11
F0	

## INDEX

---

- file format ..... 32
- fac\_dur ..... 14
- fdir ..... 14
- Festival Speech Tools ..... 5
- Festival Speech tools ..... 5
- fext ..... 14
- File formats
  - .PointProcess ..... 34
  - .cart ..... 31
  - .cb ..... 31
  - .eps ..... 31
  - .f0 ..... 32
  - .f0\_ascii ..... 32
  - .f0s ..... 32
  - .poly ..... 33
  - .polyf0s ..... 33
  - mbrols\_map.txt ..... 33
- Flex ..... 4, 5
  
- G**
- GNU ..... 4
- go.sql ..... 24
  
- I**
- Ibibio
  - about ..... 2
  - module source code ..... 6
  - MySQL database ..... 6
- Install
  - BOSS ..... 6
  
- L**
- Libtool ..... 5
- list ..... 14
- loo1.pl ..... 16
- loo10.pl ..... 16
  
- M**
- M4 ..... 5
- main\_ibb ..... 24
- MBROLA
  - configuration switch ..... 9
  
- file format ..... 33
- map file ..... 9
- synthesis by ..... 29
- modules ..... 13
- MySQL ..... 4, 5
  
- N**
- Niger-Congo ..... 2
- no\_smooth ..... 14
  
- O**
- order ..... 8
  
- P**
- package
  - for BOSS ..... 6
- Perl ..... 5
- plain-cb.pl ..... 23
- PointProcess
  - file format ..... 34
- poly
  - file format ..... 33
- poly.conf ..... 19
- polydir ..... 11
- polyext ..... 11
- polyf0s
  - file format ..... 33
- polyreg.pl ..... 18
- polysylloop.pl ..... 18
- polysylsub.pl ..... 18
- ppsf ..... 13
- Praat ..... 5
- prune-esps.pl ..... 18
- pulsedir ..... 12
- pulsext ..... 12
  
- Q**
- quartic ..... 8
  
- R**
- rawdir ..... 12
- rawext ..... 12

## INDEX

---

redistribute-cb.pl ..... 21

### S

samplerate ..... 13

#### Simple

    Perl package ..... 5

spsf ..... 13

### T

tcw ..... 10

tg2xml.pl ..... 23

tg2xmlsub.pl ..... 23

tgdir ..... 12

tgext ..... 12

tonecart ..... 8

#### tones

    symbols for ..... 3

tools ..... 15

    configuration ..... 15

### U

ucw ..... 9

usewords ..... 8

### V

vq.pl ..... 21

### W

wagon ..... 22

wpsf ..... 13

### X

Xerces-C ..... 4, 5

XML::Simple ..... 5

xmdir ..... 12

xmlxt ..... 12