

Technical Report:
**OGIresLPC: Diphone synthesizer using residual-excited
linear prediction**

Michael Macon

Andrew Cronk

Johan Wouters

Alex Kain

Center for Spoken Language Understanding
Oregon Graduate Institute of Science and Technology
P.O. Box 91000, Portland, OR 97291-1000
<http://www.cse.ogi.edu/CSLU/tts>

October 9, 1997

1 Introduction

OGIresLPC is a drop-in module for the Festival TTS system created by CSTR at the University of Edinburgh (<http://www.cstr.ed.ac.uk/projects/festival>). This version of OGIresLPC has been designed to work with Festival version 1.2.0, released September 1997. It should work with any version 1.2.x newer than this, and can possibly be made to work with other versions of Festival, but this would require some changes to the code and knowledge of Festival internals. It provides waveform synthesis of speech with reasonable quality, but has not been extensively optimized in any way. It is meant to serve as a simple baseline synthesizer in the CSLU Toolkit and for other experiments.

It includes the following:

- Residual LPC synthesizer
- Festival-format American English lexicon compiled from Moby and CMU lexicons.
- Four (4) American English voice group files, for the voices MWM, AEC, JPH, TLL.
- One Mexican Spanish group file (voice ABC), and miscellaneous Scheme functions for Spanish synthesis.

The module is distributed in several chunks:

1. OGIresLPC – the code, lexicon, and this short technical report describing the synthesizer.
2. `voice_mwm` – American male voice
3. `voice_aec` – American male voice
4. `voice_jph` – American male voice
5. `voice_tll` – American female voice
6. `voice_abc` – Mexican Spanish male voice

The user must first install the OGIresLPC package (and of course Festival itself) in order to use the other parts. Festival is available from CSTR at <http://www.cstr.ed.ac.uk/projects/festival.html>.

2 Installation

OGIresLPC basic module

(In each case, substitute for “1.0” the current version of OGIresLPC.)

1. Unpack the file `OGIresLPC-1.0.tar` in the directory *above* the festival installation directory `festival/`. This will unpack the synthesizer code and voice files into an existing festival installation.
2. Add the following line to the file `festival/src/modules/OTHER_DIRS`:

```
OGIresLPC
```

Create this file if it does not exist in your installation.

3. cd to the `festival/` directory and remake festival
4. Unpack the file `voice_mwm-1.0.tar` in the directory above `festival`.
5. When festival is restarted, the command (`voice_mwm_diphone`) should be available in Festival.
6. If you want MWM to be the default voice for Festival, add the line

```
(set! voice_default voice_mwm_diphone)
```

to a file called "siteinit.scm" (create it if not there) in the directory `festival/lib/`. This step is REQUIRED if you have not already installed any of the CSTR voices (like `rab_diphone`).

Optional additional voices

1. Unpack the tar file in the same directory as above.
2. Restart Festival.
3. When festival is restarted, the command (`voice_???_diphone`) should be available in Festival, where `???` are the voice initials as in the name of the tar file.

3 Using the Voices

When properly installed, you should be able to issue the festival commands

```
festival> (voice_???_diphone)
```

where `???` is one of `mwm`, `aec`, `jph`, `tll`, `abc`, depending on which of the optional modules has been installed. Subsequent synthesis will then use the selected voice.

Disclaimers

1. The Mexican Spanish voice uses some relatively unsophisticated synthesizer components. Components for intonation, duration, token-to-word rules, etc. are quite basic in comparison to the British and American English components in Festival. These Mexican Spanish synthesizer components were created in just a few days by Alejandro Barbosa of Univ. de las Americas (Puebla, Mexico) during a visit to OGI. The voice `voice_el_diphone` distributed by CSTR is much more comprehensive, and we hope to use some of these components for voice ABC in the future.
2. The voices AEC, JPH, and TLL were created by recording nonsense words and then *automatically* aligning phonetic labels to them using a speech recognizer. The diphone boundaries were not optimized in any way, but rather were chosen to be the mathematical center of each phone. This is, of course, far from optimal. We plan to release improved versions of these voices in the near future.

If you find this module useful for your own research projects, please cite our work in your publications.

4 License

This module and accompanying data are freely available for non-commercial use only. Please see the file “license.ogi” in the distribution for information on usage and redistribution, and for a DISCLAIMER OF ALL WARRANTIES.