TR-IT-0181

Detection of Creaky Voice in Speech Signals

Gregor Möhler

1996.5

ABSTRACT

Reliable F_0 extraction and pitch marking are essential for a good unit selection in concatenative speech synthesis. But natural speech is subject to irregularities. The phenomena is often described by the terms "creaky voice" or "laryngealization". The problem is that the fundamental frequency is hard to define in these parts of speech and extracting F_0 will often result in a F_0 contour jumping between different harmonics of the signal. But this is unacceptable for concatenative speech synthesis systems. We are therefor looking for a method that could detect sections of creaky voice in the speech database.

In this work a method has been developed that can detect irregularities in speech signals. It works on the basis of an F_0 algorithm (ADMF) which presents different candidates for F0 to a Recurrent Neural Network (RNN) classifier. The classifier is trained and tested on the female voices of a German Database (MÜSLI) with annotated creaky periods. This essentially very simple approach leads to 42% recognition rate in an open test. A program based on this RNN has been written that now can detect irregularities in a speech synthesis database.

Contents

1 Introduction

The extraction of fundamental frequency (F_0) has been an extensively studied subject. A large number of methods compete against each other for the smallest error rate. But even though very good results are achieved, using one of these advanced algorithms may in some cases lead to a f0 contour like the one showed in figure **. We discover a unsmooth pitch contour with a large number of jumps. We get these kinds of contours in speech signals with irregularities. They have in common that it is very hard to define a pitch. We may perceive a smooth contour but the signal (and the reference signal) show different results.

We call these kinds of irregularities *creaky voice* or *laryngealizations*. They are subject of this study. Their tendency to unsmooth F_0 contours causes problems for concatenative speech synthesis, because the selection process relies on a good pitch extraction. If creaky speech parts are not annotated as such we may select them for a specific target F_0 although they would be perceived at a different pitch.

When we want to adjust the pitch of a selected unit to the target value using signal processing methods like PSOLA we face the same problem. Additionally, PSOLA depends on a good pitch marking. Wrong pitch markers as they may occur in creaky speech parts lead to reduced synthesis quality.

Thus what we want to achieve is a annotation of laryngealized speech parts the database. This annotation may be used as an additional feature during the selection process. Then creaky signals in the synthesized speech would occur only in places where they were in the original (e.g. at the end of phrases).

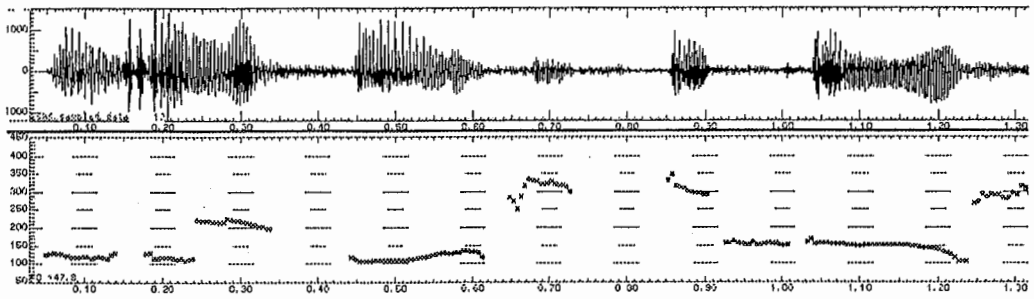

Figure 1: F_0 tracking for a creaky signal

2 About Laryngealizations

The phenomena that should be described in this section appears under a variety of terms: laryngealizations, creaky voice, vocal fry and others. It is a characteristic of the voice source and not of the vocal tract which becomes clear when we realize that the perception of sounds is not disturbed by creaky voice. The mechanisms of production are not yet fully understood. One explanation would be that the tension of the vocal cords becomes more lax for creaky periods of voicing.

Laryngealizations appear very often at syllable boundaries and towards the end of an prosodic phrase which would make it an attractive feature for speech recognition as well. Very often creaky voice is also used to express paralinguistic aspects like emotions.

There has been various attempts to classify the phenomena [6],[1]. Since the so-called MÜSLI database [1] is used throughout this work we will shortly describe this classification scheme here.

There are six features used for the classification scheme. Each feature can take values from 1 to 3 or 1 to 4 showing the degree of irregularity. The features are:

1. number of glottal pulses in the creaky part (one, up to 3, many)
2. different kind of damping (normal, exponential, triangular, "unusual" damping)
3. amplitude compared to right and left context (lower, same, higher)
4. amplitude within the segment (regular, slightly irregular, diplophonic, break down of envelope)
5. F_0 compared to right and left context (regular, slightly irregular, subharmonics, extremely long periods or pauses)
6. F_0 within the segment (regular, slightly irregular, strong variations, periods not detectable)

Based on this features seven different classes (a-f,r) are assigned to the laryngealization discovered. Their main characteristics are sketched in table ** which is a simplified version of the one given in [1]. The rest class consists of all the creaky phenomena that could not be classified in the other categories.

A complete database of 1329 sentences had been labeled at the Universities of Erlangen and München / Germany according to the MÜSLI classification scheme. This database was available for the work described here. It consists of 30 minutes of speech by 1 male and 3 female speakers. 5% of the speech in the database had been labeled as laryngealized by 2 trained phoneticians.

type	number of pulses	damping	internal ampl. changes	F0 to context	F0 internal
aperiodicity (a)				irregular/ subharmonics	irregular
subharmonics (b)				subharmonics	regular
glottalization (c)	1-3			long periods	
diplophonia (d,e)			high		
damping (f)		high			
rest class (r)					

Table 1: Main characteristics of the MÜSLI classification scheme

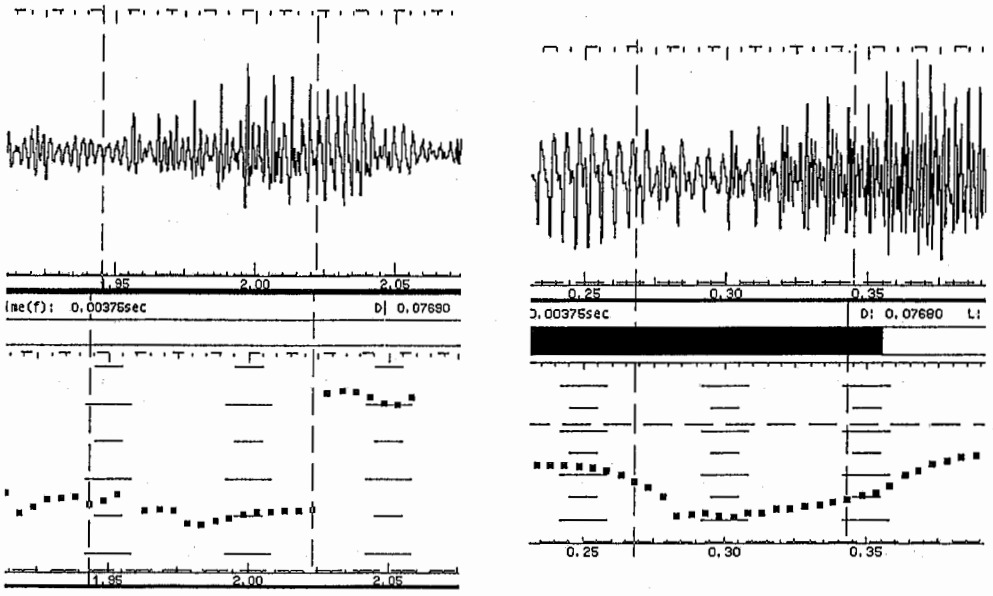

Figure 2: MÜSLI classes of aperiodicity, subharmonics

The figures ** to ** show speech waveforms and the corresponding F_0 contours (by the ESPS get_f0 program) for 6 MÜSLI classes. We see that some classes might be more critical for smooth F_0 tracking (aperiodicity, subharmonics, glottalization) while others might not necessarily cause jumps in the pitch contour (diplophonia, damping).

3 Classification by Recurrent Neural Networks

In the two sections following this one we will outline two approaches how creaky voice could be automatically detected. They differ in how the input parameters are derived from the speech signal. However, in both cases we

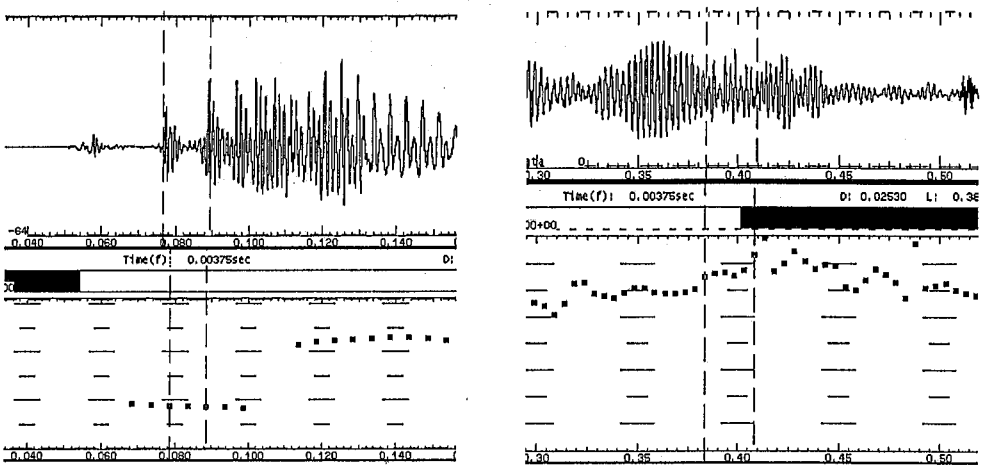

Figure 3: MÜSLI classes of glottalization and diplophonia

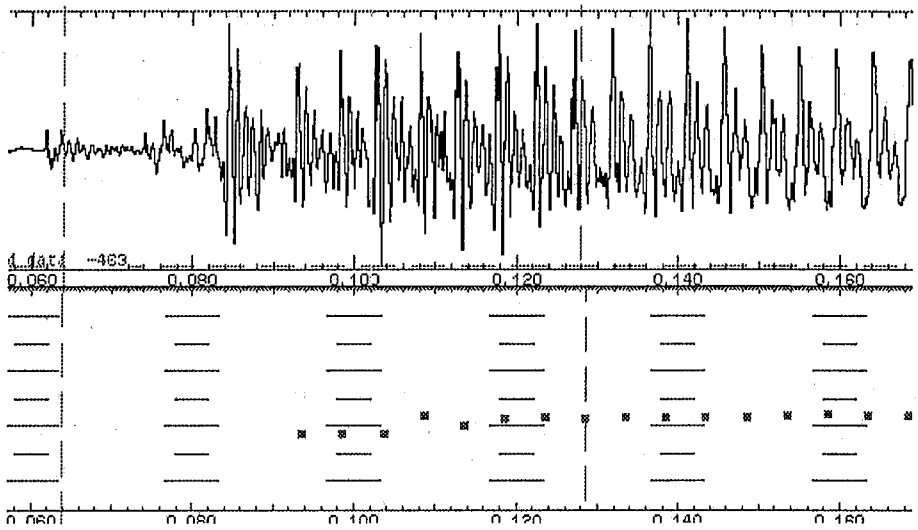

Figure 4: MÜSLI classes of damping

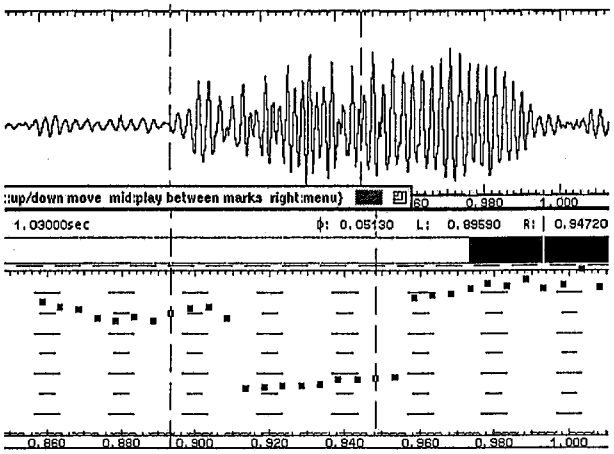

Figure 5: MÜSLI rest class

use a Recurrent Neural Network (RNN) to classify between non-creaky parts and creaky parts and between the different types of laryngealizations.

The RNN program has been developed at ATR-ITL by Mike Schuster [2] and consists of a training program, a test program and a C library for the classification. A RNN consists of states in an internal state neuron group and in an output group. The output group is fully connected both to the input layer and the internal state group holding the state of the last time step. In a bidirectional RNN there is one state group for each direction in time. The main advantage of RNNs are that only the information of the actual time must be presented to the net, while the history is internally captured by the state group. Considerations regarding window length as they occur when training MLPs are therefore not necessary. For more details about RNNs refer to [2].

4 Detection by Voice Source Parameters

In the dissertation by Wen Ding [3] a model is proposed how speech could be described by a parametric source and a time variant IIR filter for the vocal tract. This model is referred to as ARX model (autoregressive model with exogenous input) and could represented by the following equation:

$$\sum_{i=0}^{p} a_i(n)s(n-i) = \sum_{j=0}^{q} b_j(n)u(n-j) + \varepsilon(n) \qquad (1)$$

The differentiated source signal $u(n)$ is filtered by the IIR filter with time varying coefficients $a_i(n)$ and $b_i(n)$. The error signal $\varepsilon(n)$ is the signal part

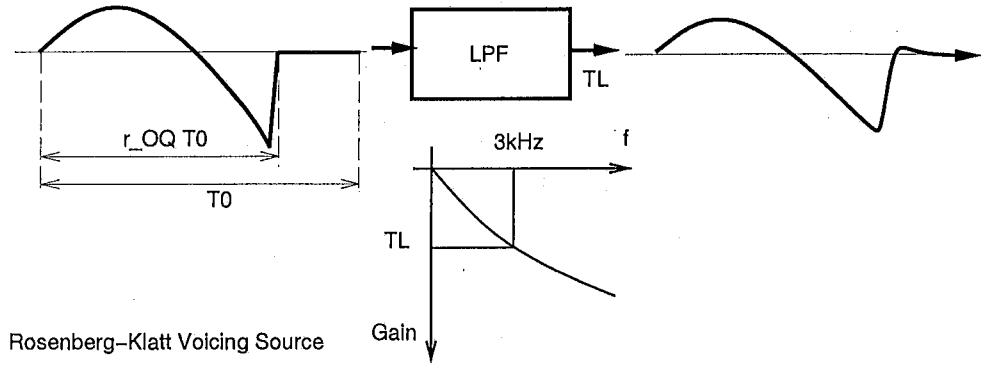

Figure 6: The Rosenberg-Klatt Voicing Source Model

that is not captured by the model. If we consider a linear (all pole) FIR filter then $\varepsilon(n)$ represents the so-called glottal noise. I should be higher in parts where the source model does not fully capture the waveform. We would therefore expect glottal noise to be a good cue for laryngealizations.

As a source model for the voiced parts the Rosenberg-Klatt Voicing Source Model is used. Figure ** shows its outline.

By applying the ARX model on speech signals we obtain the following 5 parameters: Fundamental Frequency F_0, Amplitude of Voicing AV, Open Quotient OQ and Glottal Noise GN. For the RNN classification task we added a binary parameter voicing V that was set to 0 in unvoiced and 1 in voiced parts of the signal. During the unvoiced periods all other parameter are kept constant and only V changed to 0.

We trained a RNN with 8 states in forward and backward direction each. The input and output vectors were sampled at 10ms. The net was trained to classify all 8 classes (non-laryngealized, 7 MÜSLI classes). The result of an open test is presented in table **:

We did not get any significant results for the classes we were particularly interested in (classes abc). Class f (damping) was recognized by 10% . But as we mentioned before this class is not really in the center of our interest because in most cases it does not disturb F_0 tracking.

So it finally turned out that the effect of laryngealization is not particularly well represented by the given parameters. Although no satisfying explanation could be given we may presume that the model can sufficiently follow the speech signal and the glottal noise does not significantly change during creaky periods.

This result did not encourage us to make further studies based on the ARX model.

	classified as								
class	no-laryn	a	b	c	d	e	f	r	
no-laryn	12360	18	15	0	0	0	53	0	99.31%
a	396	9	11	0	0	0	11	2	2.10%
b	249	8	11	0	0	0	6	0	4.01%
c	108	2	1	0	0	0	0	0	0%
d	177	9	2	0	0	0	9	0	0%
e	79	1	0	0	0	0	0	0	0%
f	519	2	4	0	0	0	61	0	10.41%
r	385	10	12	0	0	0	3	0	0%

Table 2: Results for the parametric source parameters in an open test

5 Detection by the AMDF Algorithm

This approach has been motivated by the fact that laryngealizations cause problems in pitch detection (which was actually the reason why we started this work). Let's consider we have a pitch extraction algorithm and want to find a clear F_0 value for different parts of speech. In voiced speech this would be a more or less easy task. In laryngealized parts, however, we should face more problems in finding a single clear value. Several candidates may compete against each other. And finally in unvoiced speech we may not find any reasonable F_0 candidate at all. We want to use this fact to detect laryngealizations.

At ATR-HIP a pitch tracker had been developed by Alain de Cheveigné [4]. It is based on the principle of a Average Magnitude Difference Function (AMDF) between two windows. It has been shown that this algorithm has an error rate below commercial F_0 trackers (like ESPS `get_f0`). Despite this high performance it is a relatively simple method of pitch extraction where we can easily extract the parameters desired.

The AMDF algorithm is based on the difference taken between the samples lying in two windows. The first window is fix and the other one is moved up to a maximum lag (determined by the minimum frequency). The AMDF function is the following:

$$A_i(\tau) = \sum_{k=i}^{i+L} \mid s_k - s_{k-\tau} \mid \qquad (2)$$

In figure ** you can see the AMDF function for a voiced speech part.The harmonics of F_0 are represented by equally spaced minima in the AMDF function. The actual F_0 value is determined by the first minimum falling under a threshold. In [4] a threshold of 0.4 is proposed.

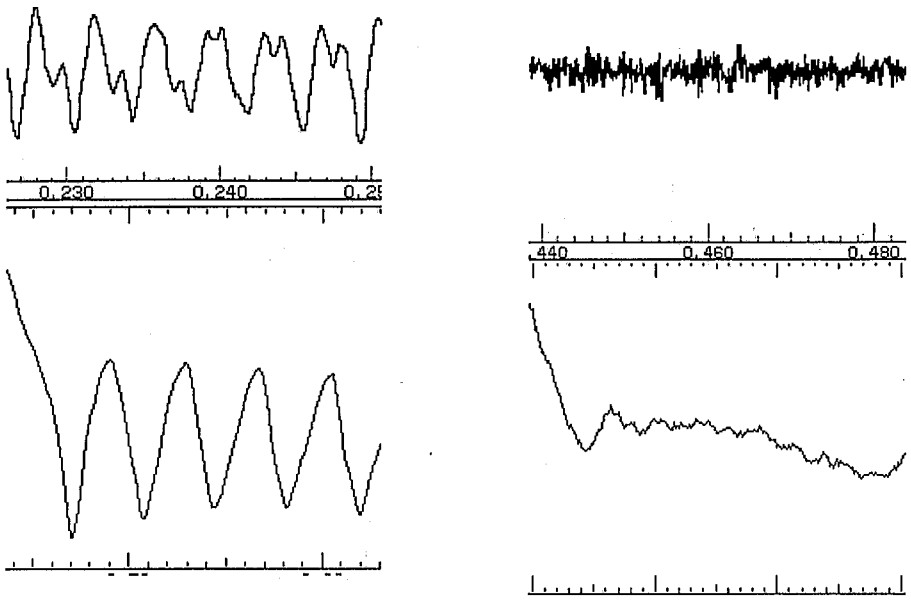

Figure 7: The AMDF function of a voiced and unvoiced speech part

Figure ** to ** shows the AMDF function for different kinds of excitation. We discover that laryngealized speech parts do show harmonic pattern but these patterns differ quite significantly from those of voiced speech parts. One observation is e.g. that in most cases the first minima is not the deepest one because it belongs to a super-harmonic.

By detecting several minima and their AMDF value we should therefore be able to detect creaky voice parts. There might be other possibilities for parameter extraction but we have chosen the following parameter set as input for the RNN:

We choose the deepest 4 minima in the AMDF function as F_0 candidates (see figure **). As proposed by [4] we take the first minimum below the threshold of 0.4 as the principal F_0 candidate. If, however, the AMDF value of the second minimum is more than 0.2 deeper than the principle F_0 candidate we will take the second one. This is to avoid the choice of a strong super-harmonic. The F_0 values of the other 3 candidates are related to the main F_0 value. The 4 F_0 values and the corresponding AMDF values are presented to the RNN. The principle F_0 value is z-scored over a file before training.

For male and female voices we have to choose different minimal and maximal frequency for the AMDF function to allow 3-4 minima to fall into the range between them. While we used 50 to 800 Hz in the female case, 25 to 400 Hz is appropriate for male voices. However, to keep the data consistent the net was only trained for female voices (3 of the 4 speakers in the MÜSLI database). The data for the male voice was not enough to train a second

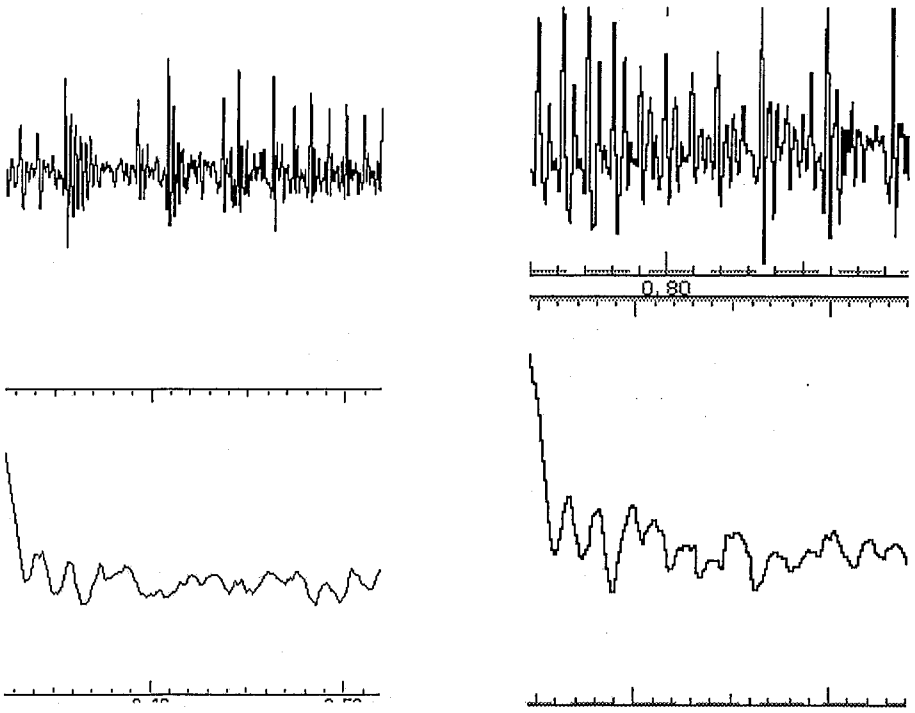

Figure 8: The AMDF function of a aperiodic and sub-harmonic creaky signal

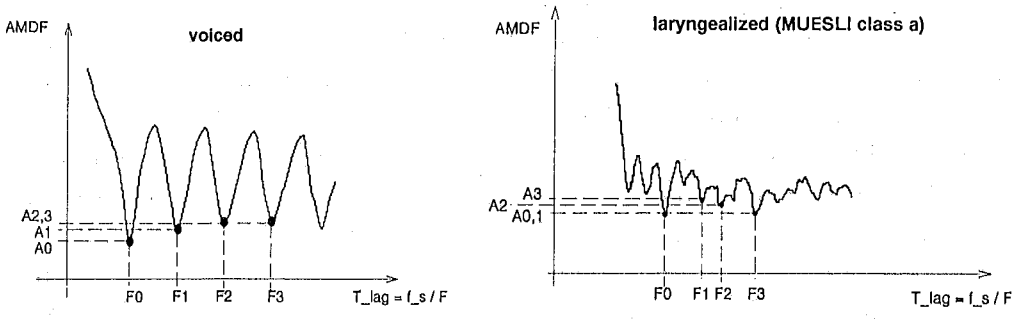

Figure 9: parameter extraction from the AMDF signal

	classified as								
class	no-laryn	a	b	c	d	e	f	r	
no-laryn	62717	71	44	10	16	0	0	9	99.76%
a	1022	170	39	4	2	0	1	2	13.71%
b	483	84	48	0	7	0	0	0	7.72%
c	370	18	0	5	0	0	0	0	1.27%
d	357	12	11	0	21	0	0	0	5.24%
e	38	4	0	0	2	0	0	0	0%
f	156	15	9	0	1	0	0	1	0%
r	667	55	25	1	5	0	0	0	0%

Table 3: Test results for the AMDF method (8 classes - female voices))

	classified as		
class	no-laryn	creaky	
no-laryn	16136	199	98.78
creaky	486	357	42.35

Table 4: Test results for the AMDF method (one compound class - female voices)

net (only 53 phrases of the male speaker contained laryngealizations of the classes a, b or r). But we will show that we can use the net trained for female voices also for male voices with a lower classification score.

Training the RNN in 10ms frames and 8 states in forward and backward direction each lead to the classification result (open test) presented in table **.

These results were very promising especially because the interesting classes a (aperiodicity) and b (sub-harmonics) received a significant score. We can also discover that in about 12% of the cases class r (rest class) was classified as class a or b. This suggests a similarity of this class to the classes a and b. In a following experiment we created a compound class 'creaky' consisting of the MÜSLI classes a, b and r and trained the net. The result is shown in table ***.

Thus laryngealizations can be spotted in the signal in 42% of the cases. We tried to get better results for more RNN states, z-scoring all parameters (instead of only the principal F_0 value) and by extracting data in a window of 200ms around the creaky parts (prior probability of creaky voice 15% instead of 5%). But none of these experiments showed better results.

	classified as		
class	no-laryn	creaky	
no-laryn	8844	356	96.13
creaky	428	164	27.70

Table 5: Test results for the network on the male voice

When we use now this net to classify laryngealizations in the male voice we obtain the results showed in table **. There is a loss of performance but nevertheless almost 1 out of 3 laryngealizations could be detected. Also the misclassification of not laryngealized parts is higher. But looking at the results revealed that most of these errors were unvoiced periods classified as creaky. Taking voicing information into account would solve most of this problem.

6 The Laryngealization Detection Program creaks

The RNN package provides a C-library for classification purposes. We use this library to write a program that would give us the probability of laryngealization for each frame of a specific speech file.

This code of program is based on the original AMDF algorithm by Alain de Cheveigné [4]. But instead of detecting the first minimum under the threshold the 4 deepest minima are searched and then passed to the RNN. As it was the case for the training a principle F_0 candidate is determined among the 4 minima and the other 3 F_0 values are related to this one. The data of the principle F_0 candidate is z-scored.

The output of the program contains 3 column: First the time in seconds, than the principle F_0 candidate in Hz and finally the probability of creaky voice. No voicing decision is made. An F_0 value of 0 is given out if no minimum could be located in the AMDF function.

The program has the following parameters most of which come from the original AMDF program. The defaults are given in brackets.

- -i input filename
- -train prints out the data instead of passing it to the classification network. This option is used if data for a training is required.
- -rnn the classification RNN filename (creaks.rnn)

- -hi Hz, highest expected frequency (800- good for female voices) should be set to 400 for *male voices*
- -lo Hz, lowest expected frequency (50 - good for female voices) should be set to 25 for *male voices*
- -step samples, frame interval (160)
- -wsize samples, integration window size (640)
- -wpow size of power window (.01)
- -fuzz fuzz factor (0.05)
- -thresh keep first amdf min below this thresh (0.4)
- -skip samples, interval between diff calcs. (4)
- -smooth samples, waveform smoothing (8)

7 Results and Further Improvements

We found a way to detect creaky voice in speech signals using parameters derived from an AMDF pitch tracker. The classification rate is 42% for the 3 female speakers (for which the net was trained) I.e. almost half of the laryngealized sections in the test set of female voice could be detected. When we applied this net to the male voice the result was 28%.

In this work we have chosen a very simple parameter extraction of the AMDF signal. It turned out to be appropriate. But it is definitely sub-optimal and we may find a better parametric representation of the AMDF signal. Or we may down-sample the AMDF signal and pass it directly to the net.

The MÜSLI system is an serious attempt to classify and label irregular speech. But we could not expect the classes to be perfect. Especially when we compare the work that has been done in labeling creaky voice compared to labeling databases for speech recognition we could come to the conclusion that even with better parametrisation we may not achieve very high recognition results. Considering this, a classification of 42% might be already a very good result.

The creak detection program `creaks` has been written to annotate creaky voice in a speech synthesis database. In could easily be run over the whole database adding laryngealization labels where creaky voice is detected. Then in a second step this new feature can be added to the unit selection features. We would expect more natural speech since creaky signal parts are chosen

where they occur in natural speech. Furthermore no or only moderate modification by algorithms like PSOLA should be done in creaky speech parts. This again would increase synthesis quality.

Finally we could confirm the good performance of the AMDF pitch tracker. Especially for low energy speech parts where other pitch trackers often showed unreliable results the AMDF tracker mostly found the "right track". The only thing to be done to get a fully working pitch tracker is adding a voiced/unvoiced decision.

8 Scripts and Stuff

The following scripts might be useful for later applications. For more information about how to use them refer to the comments in the scripts themselves. The scripts are found in the directory `/homes/gregor/perl/`:

- `toglab.pl` perl script that converts various kinds of label formats (ESPS no time print, time in s or ms, ...) to a time synchronous format with the columns as follows:

  `time/sec param1 param2 param3 ....` (in equal time steps)

- `glab2sd.pl` converts the time synchronous label file (first column) to an single channel ESPS sd-file

- `glab2esps.pl` converts the time synchronous label file to an multi-field ESPS file. The field names can be added in the command line. If they are skipped param0, param1, ... is chosen. If 'f0' is specified as field name a 4 column time synchronous label file is converted to an ESPS f0-file.

- `glab_range.pl` extracts a specific time range from a time synchronous label file

- `f0esps2glab.pl` converts an ESPS f0-file to a time synchronous label file.

- `show_muesli.pl` displays a number of files from directories specified in the script and displays them in xwaves. Take this script as workbench (as I did) and modify it to your desires. In its current state it would display a MÜSLI speech file, the appropriate ESPS F_0 file with the MÜSLI phone and laryngealization labels. Then it calculates the probability of laryngealization with the program `creaks` and displays it.

- `muesli_sel_files.pl`. This script was used to create a list of files for training and testing of the RNN. The number of files in the training set and in the test set is specified in the command line as well as the number of project. The script then takes the specified number of filenames out of a filename list given in the script and creates the 4 lists m*i*`.list_in`, m*i*`.list_out`, t*i*`.list_in`, t*i*`.list_in`. The training projects start with the letter `m` and the test project with `t`. *i* is the project number.

Acknowledgments

Throughout this work I received help and enrichment by a large number of people whom I would like to sincerely thank:

Norio Higuchi, for giving me the opportunity to come to ATR and work in his department, allowing me to do research in this well equipped laboratory,

Nick Campbell who supervised my work, always lent an open ear to my questions and ideas, and enlarged my understanding of many of the problems of speech synthesis,

Mike Schuster, for giving me his Recurrent Neural Network program and deepening my understanding of speech recognition and neural networks throughout long discussions,

Andreas Kiessling and Anton Batliner (Nürnberg/München - Germany) for their MÜSLI database,

Chieko Koshima and Chie Kimura for organizing this stay so well,

and all the others that made my time at ATR really worth coming.

References

[1] Batliner et. al. MÜSLI: A Classification Scheme For Laryngalizations. *ESCA Workshop on Prosody 1993.* Working Papers 41, Dept of Linguistics and Phonetics, Lund, Sweden.

[2] Schuster, Mike. Learning out of Time Series With an Extended Recurrent Neural Net. To appear in *Proceedings of the Neural Network Workshop for Signal Processing 96*, Kyoto.

[3] Ding, Wen. Joint Estimation of Voice Source and Vocal Tract Parameters Based on an ARX Speech Production Model. PhD Theses, Department of Electrical and Electronic Engineering, Utsunomiya University, Japan, 1996.

[4] De Cheveigné, Alain. Speech Fundamental Frequency Estimation. ATR/HIP Technical Report No. TR-H-195, 1996

[5] Giebner, Marcus. Bestimmung der Anregungsart von Sprache mit einem HMM/NN-Hybridverfahren. Diplomarbeit im Fach Informatik. Lehrstuhl für Mustererkennung, Universität Erlangen Nürnberg, 1995.

[6] Huber, Dieter. Aspects of the Communicative Function of Voice in Text intonation. PhD thesis, Calmers University Göteburg/Lund, 1988