HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Engineering Physics
and Mathematics

Tom Bäckström

Development of an analysis environment for clinical analysis of continuous speech

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Espoo, January 4, 2001

Supervisor: Olli Simula, professor
Instructor: Paavo Alku, professor

Tekijä:	Tom Bäckström
Osasto:	Teknillisen fysiikan ja matematiikan osasto
Pääaine:	Informaatiotekniikka
Sivuaine:	Systeemi- ja operaatiotutkimus
Työn nimi:	Kliinisen tutkimusympäristön kehittäminen jatkuvan puheen analysointiin
Title in English:	Development of an analysis environment for clinical analysis of continuous speech
Professuurin koodi ja nimi:	Tik-61 Informaatiotekniikka
Työn valvoja:	Professori Olli Simula
Työn ohjaaja:	Professori Paavo Alku

Tiivistelmä:

Tässä työssä esitellään kliinisen puheentutkimusympäristön metodeita ja kehitystä erityisesti jatkuvan puheen analysointiin. Työn tarkoitus oli korvata edeltävä analoginen analysointiympäristö täysin digitaalisella järjestelmällä, käyttäen hyväksi nykyaikaisia signaalin digitaalisen käsittelyn menetelmiä. Järjestelmän käyttöliittymä tuli toteuttaa siten, että se soveltuu käytettäväksi sairaalaympäristössä.

Puhesignaalin analyysi on kolmevaiheinen muodostuen piirreirroituksesta, luokituksesta ja analyysistä. Näistä kaksi ensimmäistä muodostavat yhdessä olennaisesti klassisen hahmontunnistustehtävän. Piirreirroituksessa analysoidaan puheen energia, perustaajuus ja stationäärisyys. Näiden perusteella puhesignaali luokitellaan kolmeen luokkaan: Hiljaisuus, soinnillinen äänne tai soinniton äänne. Tämän luokituksen perusteella lasketaan puheesta edelleen useita erilaisia puheäänen laatuun liittyviä mitta-arvoja kuten perustaajuuden keskiarvo, kokonaispuheaika sekä soinnillisten äänteiden kokonaisaika.

Puheäänen analyysin kohderyhmänä oli sellaisia ammatteja edustavat ihmiset, joiden pääasiallinen työväline oli heidän oma äänensä. Siksi analyysissä keskityttiin erityisesti äänen väsymiseen liittyviin mitta-arvoihin.

Sivumäärä: 52	Avainsanat: puheenkäsittely, hahmontunnistus, signaalin digitaalinen käsittely, äänen väsyminen

Täytetään osastolla

Hyväksytty:	Kirjasto:

Author:	Tom Bäckström
Department:	Department of Engineering Physics and Mathematics
Major subject:	Information technology
Minor subject:	Systems and operation research

Title in English:	Development of an analysis environment for clinical analysis of continuous speech
Title in Finnish:	Kliinisen tutkimusympäristön kehittäminen jatkuvan puheen analysointiin
Chair:	Tik-61 Information technology
Supervisor:	Professor Olli Simula
Instructor:	Professor Paavo Alku

Abstract:

This work presents methods and development of a clinical analysis environment for analysis of continuous speech. The purpose of this work was to replace an antecedent analogical environment with a digital system, with the aid of modern methods of digital signal processing. The user interface was to be designed such that it is suitable for usage in a hospital environment.

The analysis of the speech signal was applied in three stages: Feature extraction, classification and analysis. The two first of which form together the setting of a classical pattern recognition problem. In the feature extraction stage, three signal properties were calculated. These properties are signal energy, fundamental frequency and stationarity. With the aid of these features, the signal was classified to three classes: Silence, voiced phoneme and unvoiced phoneme. Based on this classification vector, several voice quality measures were calculated. Some of the most important of these measures are average fundamental frequency, total time of speech and total time of voiced speech.

The focus group of this research was people who use their voices in their profession. The analysis was therefore concentrated around issues related to vocal loading.

Pages: 52	Keywords: speech analysis, pattern recognition, digital signal processing, vocal loading
Department fills	
Approved:	Library code:

Contents

Units and Abreviations

$\mathcal{N}(\bar{x}, \sigma)$	Normal probability distribution (Gaussian) with average $\bar{x}$ and variance σ
N	number of samples in signal
P	signal energy or power
r_x	autocorrelation vector of signal x
$\bar{x}$	average of x
$\chi^2(m)$	chi-square probability distribution with m degrees of freedom
ϵ	noise or residue signal
σ	standard deviation
σ^2	variation
dB	decibel
Gb	Giga bytes
Hz	Herz
Mb	Mega bytes
AIC	Akaike's Information Criterion
AD-converter	Analog to Digital converter
CIS	Laboratory of Computer and Information Science
DAT	Digital Audio Tape
DSP	Digital Signal Processing
F0	Fundamental frequency
GSM	Global System for Mobile Communications, also know as Group Speciale Mobile
HUT	Helsinki University of Technology
HUCH	Helsinki Unviersity Central Hospital
HNR	Harmonics to Noise Ratio
IAIF	Iterative Adaptive Inverse Filtering
LPC	Linear Predictive Coding
LTAS	Long Time Average Spectrum
OYS	Oulu University Hospital (Oulun Yliopistollinen Sairaala)
SNR	Signal to Noise Ratio
SOM	Self Organizing Map
SPL	Sound Pressure Level

Chapter 1

Introduction

In Oulu University Hospital (OYS), a speech analysis environment for the clinical speech analysis laboratory was developed in 1993 by Kari Haataja [1]. It is based on analogical measurement of voice quality parameters, which are transferred to a computer trough a multichannel A/D-card.

The purpose of the current work was to update and replace this analysis environment with a digital analysis environment. The new environment implementation would be based on modern signal processing techniques, which are made possible by the availability of high-speed processors and large memory capacity.

The project was initiated by the telecommunications operator Sonera in an effort to generate information about the voices of their call-center personnel, in order to improve their working environment and cut the amount of voice failure related sick leaves.

The project was funded by the Finnish Work Environment fund. The work for this thesis was done at the Laboratory of Acoustics and Audio Signal Processing, at the Helsinki University of Technology (HUT). The supervisor was Olli Simula from the Laboratory of Computer and Information Science (CIS) at HUT. The instructor and mentor was professor Paavo Alku from the Laboratory of Acoustics and Audio Signal Processing at HUT, whom I wish to thank for all the great help he offered to my work. An instructor on medical aspects of this work, and leader of the whole project was Professor Erkki Vilkman from the Helsinki University Central Hospital, whom I wish to thank especially for his encouraging support. Additional support on the medical side, and the invaluable evaluation and testing of the user interface was done by Laura Lehto, graduate student at Helsinki University Central Hospital.

Chapter 2

Background for this study

2.1 The human voice

The voice production system of humans is very complex. Still, voice usage, that is speaking, is a natural and important part of normal everyday life. The fundaments are clear, but because of the tight coupling of psychological, physical, physiological, acoustical and flow-mechanical aspects, there are still many unsolved problems. Therefore, a true understanding of the human voice production is still far away.

2.1.1 Voice production

The human voice is generated in the larynx. The driving force is pressure produced by the lungs, and the spectra of the excitation signal generated in the vocal folds, is shaped in the vocal tract [2]. Figure 2.1 illustrates the human voice production system.

This figure does not include any of the feedback mechanisms involved. All stages have some kind of built-in sensing mechanism of the process, which is fed back to the control system. The most important feedback is probably the sound output, which is processed by the hearing. Most of these feedback and control mechanisms are unconscious, and the conscious mind can only indirectly control the output.

It is essential to realise that each stage in the voice production flow chart is a biological system in itself. Each stage will therefore yield some fluctuations and temporal variation to the output of that stage. These variations will keep the output of the whole system constantly in a small fluctuation.

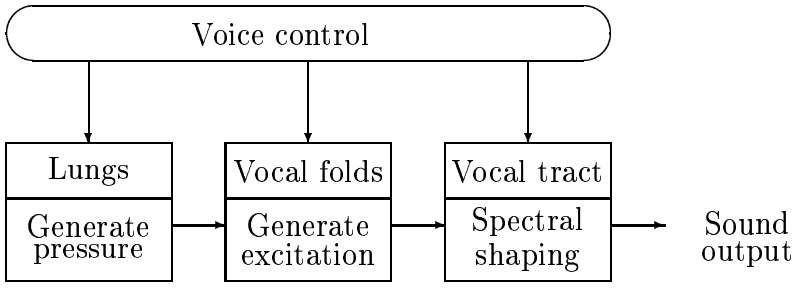

Figure 2.1: The human voice production system

2.1.2 Voiced and unvoiced sounds

Phonemes can be categorised in two classes based on if the vocal folds are oscillating or not. Phonemes in which the vocal folds are oscillating are called voiced sounds, and respectively, when no oscillation is present the sound is called unvoiced. Perceptually a voiced sound is defined as an utterance, which has a distinguishable fundamental frequency. Likewise, an unvoiced sound is a sound that does not have any distinguishable fundamental frequency.

In point of view of voice quality analysis, activity in the vocal folds is usually the most interesting part of speech. Because the vocal folds oscillate only during voiced sounds, the voiced sounds are therefore the most interesting part of speech for voice quality analysts. In addition, it has been shown that most western languages contain more voiced sounds than unvoiced [3]. It is therefore clear that emphasis should be laid more on voiced utterances than unvoiced. However, it is often difficult to strictly separate between voiced and unvoiced parts. Therefore, and in order to maintain generality, voiced and unvoiced parts are both included in the analysis.

2.1.3 Voice quality properties

The most common classical voice quality measures are jitter, shimmer, voice range and fundamental frequency. Definitions for jitter and shimmer are given below. These properties are usually measured in a laboratory environment. Jitter and shimmer are usually measured from a sustained phonation such as [a:], while voice range requires a longer test procedure. The problem with jitter, shimmer and fundamental frequency is that if the subject is

aware of what property is analysed, he can affect the result. The problem with Voice range -tests is that the test requires following some test procedure, and cannot be executed continuously.

Jitter

Jitter is defined as short time frequency perturbation, that is, small changes in the length of consecutive pitch periods. Perceptually an increased amount of jitter is observed as hoarseness. A voice without jitter is perceived as a mechanical voice. A speaker can change the hoarseness of his voice, thus changing also the amount of jitter. Jitter is therefore a somewhat controversial measure. However, there is a lot of evidence that the amount of jitter changes when a voice is loaded.

Shimmer

Shimmer is closely related measure to jitter. Its definition is short time amplitude perturbation, that is, small changes in the amplitude of consecutive periods. Perceptually shimmer is observed like jitter, as a kind of hoarseness.

Voice range

The Voice range -measure tests the range of the fundamental frequency on different sound pressure levels. The most common way to test it is to measure the sound pressure level on maximal speaking (shouting) volume at different fundamental frequencies. The same procedure is repeated with minimal speaking volume. This procedure provides the total area of possible frequency and sound pressure combinations. The results are usually presented in a xy-graph showing the limits of the voice range. The interpretation of the voice range -graph requires thorough knowledge in phoniatrics.

Fundamental frequency

The average fundamental frequency (F0) of speech is another classical voice quality measure. It has been shown that the F0 rises when the voice is loaded. Measurement of the F0 can be done in a number of ways. There are several available hardware and software components, which can analyse the F0 of speech or sustained phonations.

2.1.4 Special properties of continuous speech

Voice quality analysis has traditionally been done foremost on sustained vowels. A sustained vowel is easy to analyse because it is long enough for most purposes, and relatively stationary. The analysis is therefore straightforward. But some researches claim that sustained vowels lack many of the properties that running speech has [4],[5]. This is one of the main reasons that the current work has been done to analyse continuous speech and not sustained vowels. Another important reason for using continuous speech is to make the recording situation as natural for the subjects as possible.

The most obvious differences between sustained vowels and continuous speech is that continuous speech is a combination of different utterances and phonemes, and they alternate in rapid succession. Where analysis of sustained vowels can use averaging over long temporal segments, continuous speech analysis has to be done in very short segments in order to include only one type of sound in each segment.

Many other voice parameters can also rapidly change in continuous speech, such as pitch, volume and sonority. These parameters are often connected with emotional and motivational connotations of the speech content [6], [7]. Short time changes of these parameters should preferably not affect the analysis results, but long time variations can be interesting.

2.2 Clinical voice quality research

In the field of clinical voice quality research, a lot of work has been done on pathological voices, but not so much on issue such as vocal loading. On the other hand, digital signal processing and voice transmission are very well known areas. The main interests driving the current work are research on effects of voice loading, and voice tiring symptoms. In this field, several studies have already been made in OYS, HUS, and HUT for example [8],[9].

2.3 Available voice analysis software

Under the very general header of "voice analysis software" a remarkably broad range of different software packages is available on the market. The available packages range from polygraph tests and astrological speech analysis, to clinical packages for phoniatrics, speech therapy and language study tools.

In the current thesis, the astrological speech analysis packages will be left aside. Almost as controversial as the astrological packages are the voice analysis based polygraph tests. Several commercial packages are available

such as Truster, Psychological Stress Evaluator (PSE), the Hagoth, the Mark II Voice Stress Analyzer (VSA), and the Computerized Voice Stress Analyzer (CVSA) [10],[11]. The same sources imply, however, that these polygraph tests are not even as trustworthy as regular polygraph tests, which reliability can be questioned.

In addition, some packages that lie closer to the current topic have been developed. Origin Data Realisation Limited offers a product called Talk Listen Analyser (TLA) for call-center monitoring. It is designed to analyse the customer service of telephone service personnel by monitoring dialogue speed and duration of telephone calls to the service center. For example, if the delay is long from a question by the customer to the answer of the call center agent, the software will report that the agent is slow to answer. In addition, statistics of call lengths are reported, as well as a multitude of talk/listen ratios [12].

2.3.1 Kay Elemetrics voice analysis products

The company Kay Elemetrics Corp. offers a whole range of clinical voice analysis products, both in hardware and software [13]. Perhaps the most interesting is the Multi-Dimensional Voice Program (MDVP) available for the Computerized Speech Lab (CSL). The MDVP analyses single vocalisations and provides over 20 different voice quality parameters, such as jitter and shimmer. The results are provided in a petal-diagram. On a side note should be mentioned that the usage of petal-diagrams has been greatly criticised in decision analysis literature because they are inherently biased.

However, the MDVP is a valuable tool for phoniatrics, although it analyses only single vocalisations. Kay Elemetrics provides also some other packages for formant, spectrogram and pitch analysis.

2.3.2 Praat

Praat is a program developed by Paul Boersma and David Weenink at the Institute of Phonetic Sciences of the University of Amsterdam, The Netherlands. It is a program designed specifically for analysis of continuous speech. The program includes functions like pitch, spectrogram and sound pressure analysis, as well as re-synthesis of speech and filtering. It has support for very long files (up to 2Gb) and has a wide range of graphical output formats [14].

The program is developed for the needs of phonetics, and as such, it is very convincing. The features available mostly overlap with features required

in the current project. However, as the program provides a multitude of features, the user-interface becomes a bit difficult to use. It is obvious, though, that the program applies robust and refined algorithms for all functions. For example, pitch extraction is done on a cycle-to-cycle basis, and provides therefore an as accurate as possible estimate of the fundamental frequency.

Chapter 3

Voice analysis methods

The essential methods in this work can be divided into three sections; Voice activity measures (also known as features), voice activity classification (also known as voice activity detection) and voice quality measures. The voice activity measures generate different features of the voice signal to the voice activity classification procedures. Each of the activity measures analyse a different property of the signal, and the classification methods aim to combine these in some intelligent way, to produce a classification to silent and speech segments of the signal.

Once the classification is done, the speech parts, both voiced and unvoiced, are analysed with different voice quality measures. These quality measures are given to the user for further analysis.

Figure 3.1 presents the hierarchy of methods.

Additional methods in this chapter are involved with more practical things, like optimisation of the code.

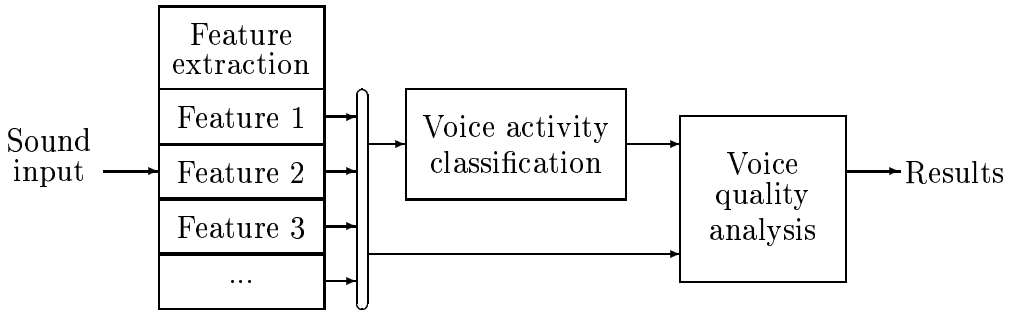

Figure 3.1: The hierarchy of analysis methods

3.1 General guidelines

In the following sections 3.2, 3.3 and 3.4, several different methods will be discussed. To be able to compare results between these methods, and to combine them as sub-methods of other techniques, some general guidelines have to be defined. Foremost, the signal has to be processed in frames because continuous processing requires too much time and memory. Therefore, a signal framing convention has to be chosen. In addition, some methods are by design most suitable for processing by frames, and converting them to continuous processing would be unnatural.

In this work, all the processing was chosen to be done in 20 ms non-overlapping frames. Non-overlapping frames were chosen for simplicity. The frame length 20 ms is a compromise between temporal and spatial accuracy. A shorter frame would allow changes that are more rapid in the temporal description, but then accuracy of the description would be compromised. Furthermore, the 20 ms frame length is an established convention in the literature and for example in GSM telephones [15].

3.2 Voice activity measures

This section will describe some measures usable in identifying voice activity. Identifying voice is a multifaceted problem, especially when operating in a noisy environment. No one feature can accurately and robustly distinguish between silence and voice. In addition, one should also carefully define what is meant by the term "voice". This definition will be left for later, but in order to maintain generality several different measures are considered, each tackling one feature of the signal.

At the classification stage of the analysis, it is preferable that all measures would be uncorrelated. This has to be taken into account for every measure designed. It is especially easy to include specific energy level limits in algorithms like the pitch track extraction.

3.2.1 Energy threshold

The energy threshold method is one of the earliest voice activity detection methods [16]. The signal energy P is calculated for each analysis window, and compared with a threshold value. The assumption is that a speech signal has a substantially higher energy then a noise signal. Therefore, low energy segments are labelled as noise.

Signal energy is defined as

$$P(i) = \sum_{\forall j \text{ in frame } i} x_j^2 \tag{3.1}$$

This method requires that the energy levels are stable, that is, the background noise has constant energy, and that the speech volume does not fluctuate too much. These constraints should be fulfilled in a controlled recording environment. However, it is sometimes difficult to identify the end of a speech segment when the voice fades out. To correctly identify the ending we would have to lower the threshold, but then we might too easily identify noise as speech.

An improvement to the energy threshold method would be a double threshold method, where the onset and offset of speech have different thresholds. This method would more easily correctly identify speech segments that fade out slowly. However, then we have to have two different free variables to set, which adds the complexity of the user-interface.

3.2.2 Zero crossings

One of the classical voice activity detection methods is the zero crossings rate method [16]. The zero crossing rate is defined as the number of times the signal changes sign per unit of time. The method is based on the assumption that the zero crossings rate is significantly smaller for a speech signal than for noise. The assumption holds for large SNR values.

The zero crossings rate is calculated for each analysis window, and a threshold value is set to identify voice activity. Windows with low zero crossing rates are labelled as speech and windows with high zero crossing rates as noise.

This method is very easy to calculate but not very accurate. Similar improvements as for the energy method can be developed for threshold handling, but the same problems appear here also.

3.2.3 Pitch track

A perceptual definition for voiced speech is that it is a spoken sound with a distinguishable pitch. Therefore, it can be assumed that if pitch can be identified from a speech signal, the speech signal is voiced.

Tracking the pitch or the fundamental frequency F0 of the human voice is traditionally a very difficult task. In this work, several different approaches were studied. None of them proved to be very robust, effective and fulfilling the above criteria.

The possible energy and amplitude range of a valid speech signal is enormous. It would be tempting to include some criterion for excluding intervals with too low energy from the pitch track. However, in order to keep the measures uncorrelated energy thresholds should be avoided if possible.

Initial pitch approximation

The F0 of a voice can vary on a large range between different subjects. In speech, male voices can time to time be as low as 80 Hz, while female voices can go up to 400 Hz. However, as most algorithms are easily confused by harmonics, they need a robust approximation of the overall pitch level.

In this work, an autocorrelation method was chosen for the task. The autocorrelation for the whole length of the signal was approximated, and the most significant peak in a gender specific frequency interval was identified. This peak was assumed to be the average overall pitch. The frequency range is chosen by the user by selecting the gender of the subject before the analysis

starts. The frequency ranges for each gender (male, female or child) can also be adjusted by the user.

This method is robust and not easily confused by background noise, if the noise level is significantly less than speech. However, because the autocorrelation correlates to the amount of energy on each frequency, the emphasised syllables become more important than the others. Because emphasised syllables often have a higher pitch than the average pitch, the pitch estimate given by this method can be biased.

Time domain methods

The time domain method studied in this work is presented in [17]. It is based on extracting different features from the waveform. Consider the bandpass-filtered speech signal in figure 3.2. First, all peaks and valleys are searched from the signal. In the picture, these peaks and valleys are marked with a dot and a letter. Six feature vectors are generated from these as follows:

1. The peak magnitudes (points b, d, f, h and j).
2. The valley magnitudes (points a, c, e, g, i and k).
3. The differences between a peak and the following valley (point-pairs b-c, d-e, f-g, h-i and j-k).
4. The differences between a valley and the following peak (point-pairs a-b, c-d, e-f, g-h and i-j).
5. The magnitude differences between consecutive peaks (point-pairs b-d, d-f, f-h and h-j).
6. The magnitude differences between consecutive valleys (point-pairs a-c, c-e, e-g, g-i and i-k).

From each of these feature vectors, an impulse vector is generated such that the feature values are at the same places as the features in the original signal. The next step is to threshold the distance between features, and remove the impulses that are closer to each other than the highest allowed frequency. The features can also be filtered by magnitude, so that if the magnitude difference between consecutive impulses is too large, the smaller one is removed (for features 1-4).

The pitch track is then calculated as a median of the impulse distances for each feature. If the identified pitch is out of a predefined range, it was set to zero. The pitch track estimate vector generated by this method can be given either for each sample or as a median or average over a given frame. The temporal accuracy is thus a parameter for the function.

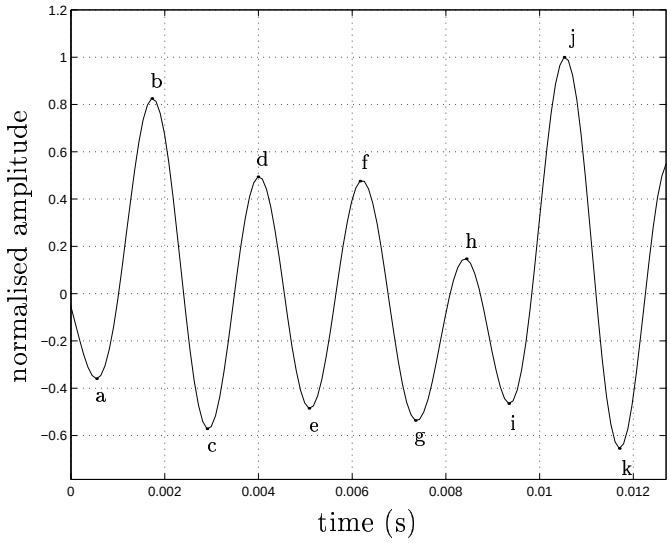

Figure 3.2: Wave features in a speech signal

Autocorrelation methods

In the same manner as the autocorrelation was used for the overall pitch level estimate, it can also be used for generating a pitch track. The autocorrelation is calculated frame by frame, and for each autocorrelation the highest peak is extracted from the valid region. The frequency corresponding to this peak is set as the pitch estimate in the frame.

A safeguard for invalid pitch identification was also applied to the algorithm. If the highest value in the valid range of the autocorrelation was found on the border of the valid range, it was assumed that the autocorrelation is higher just outside the valid range, and thus the pitch is outside the valid range. Then the pitch estimate was set to zero.

Comparison of methods and conclusions

Both pitch extraction methods gave roughly the same pitch estimates for voiced speech. For unvoiced speech and relative silence, the estimates are undefined and can vary a lot. However, comparing the results of the two methods, we do find some differences (figure 3.3). The most striking difference is that the feature extraction method seems to fail more often than the autocorrelation method. For example, at about 0.6 the feature extraction goes twice to zero, although there quite obviously is a voiced sound. At the same time, the autocorrelation method can successfully identify the pitch. Hereby it was concluded that the autocorrelation is the more robust method,

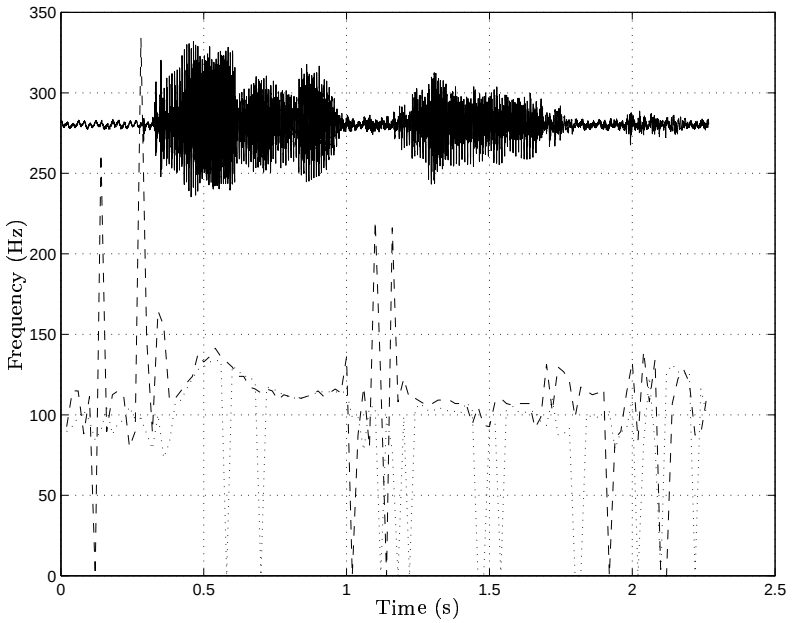

Figure 3.3: Comparison of pitch tracking algorithms. The wave feature extraction method is indicated with a dotted line '..', and the autocorellation method with a dashed line '−−'. The original speech signal is above.

giving results that are more consistent.

First and second order differences of the pitch track

The pitch track algorithms return steady curves for segments where a voiced speech is present and a random value elsewhere. It is therefore possible to measure the probability of presence of voiced speech from the estimated pitch track. A reasonable method for detecting steady curves would be the absolute value of the first difference d_1 of the pitch track vector p:

$$d_{1,i} = |p_i - p_{i+1}| \tag{3.2}$$

However, this method can give large values at the edges of a voiced segment because the pitch value for the frame just outside the voiced segment can differ a lot from pitch of the voiced segment. Therefore, a second order difference d_2 could give better results:

$$d_{2,i} = |p_{i-1} - 2p_i + p_{i+2}| \tag{3.3}$$

Setting a threshold for this difference a little above the average (non-zero) pitch value, would effectively identify single peaks in the pitch track. Then at least the worst parts of the pitch track could be ruled out.

Unfortunately, the first and second order differences use two or three consecutive pitch values. They thus use information from a 40 ms or, respectively 60 ms window. Therefore, the temporal accuracy is diminished. Sometimes, when a subject speaks rapidly, the voiced sounds are of the order 20 ms in length. At such times, the first and second order differences can have large values, and the sound would be determined as unvoiced, even if it would otherwise be a valid voiced utterance.

3.2.4 Information criterion - The Autocorrelation test

In order to determine if the input signal contains any information other than white noise, an autocorrelation test was taken to inspection [18]. The objective with this test is to assess the amount of information in the signal, and with a thresholding process, label a signal to speech and noise segments.

In a white noise process the autocorrelation of the signal is zero except for the origin:

$$r_\epsilon(\tau) = 0 \qquad \tau \neq 0 \tag{3.4}$$

The auto-covariance can be estimated with

$$\hat{r}_\epsilon(\tau) = \frac{1}{N}\sum_{t=1}^{N-\tau} \epsilon(t+\tau)\epsilon(t) \qquad (\tau \geq 0) \tag{3.5}$$

This estimate converges to r_ϵ as $N \to \infty$. However, the covariance is correlated to the energy of the signal, and must therefore be normalised. A reasonable normalised test quantity would be

$$x_\tau = \frac{\hat{r}_\epsilon(\tau)}{\hat{r}_\epsilon(0)} \tag{3.6}$$

Let us now define the auto-covariance vector

$$r = \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \epsilon(t-1) \\ \vdots \\ \epsilon(t-m) \end{pmatrix} \epsilon(t) = \begin{pmatrix} \hat{r}_\epsilon(1) \\ \vdots \\ \hat{r}_\epsilon(m) \end{pmatrix} \tag{3.7}$$

In the above expression, m is a user-defined constant expressing the order of the autocorrelation. When using the information criterion for model order verification, m expresses the model order. However, we are now not intrested in the model order. Therefore, the constant m should now be chose sufficiently large to provide an accurate description of the process.

If the signal is white noise, then we should expect $\sqrt{N}r \to \mathcal{N}(0, P)$ where the covariance matrix is $P = \lim_{N\to\infty} E(Nrr^T)$. With some effort this can be evaluated to a more convenient form [18]

$$\frac{N}{\hat{r}_\epsilon^2(0)} \sum_{i=1}^{m} \hat{r}_\epsilon^2(i) = \frac{Nr^T r}{\hat{r}_\epsilon^2(0)} \xrightarrow{\text{dist.}} \chi^2(m) \tag{3.8}$$

It can also be shown that [18]

$$\sqrt{N}x_\tau = \sqrt{N}\frac{\hat{r}_\epsilon(\tau)}{\hat{r}_\epsilon(0)} \xrightarrow{\text{dist.}} \mathcal{N}(0, 1) \tag{3.9}$$

Before the test can be applied we still need to define the confidence level α for the random variable x as

$$\alpha = P\left(x > \chi_\alpha^2(m)\right) \tag{3.10}$$

In this context the random variable x is

$$x = \frac{Nr^T r}{\hat{r}_\epsilon^2(0)} \tag{3.11}$$

If our objective would be to recognise white noise, we can set a hypothesis H_0 that $\epsilon(t)$ is white noise. Then, at confidence level α, we have

$Nr^T r/\hat{r}_\epsilon^2(0) > \chi_\alpha^2(m)$ reject H_0
$Nr^T r/\hat{r}_\epsilon^2(0) \leq \chi_\alpha^2(m)$ accept H_0

Finally, collecting formulas 3.5 to 3.8 to one, we define our information criterion c as

$$c = \frac{Nr^T r}{\hat{r}_\epsilon^2(0)} = \frac{\left[\sum_{t=1}^{N} \begin{pmatrix} \epsilon(t-1) \\ \vdots \\ \epsilon(t-m) \end{pmatrix} \epsilon(t)\right]^2}{\sum_{t=1}^{N} \epsilon(t)^2} \tag{3.12}$$

Inversely, formula 3.12 can be used as a measure of the probability that a given signal contains some information other than white noise. If we would choose to search for a closed form probability function as a function of the confidence level α, we would need to generate an algebraic expression for formula 3.10. Then we still would need to choose the confidence level α. However, because P in formula 3.10 is an increasing function we can as well give a threshold for the information criterion c.

This gives us a unitless scalar measure of the amount of information in the signal. The amount of information can be used as a part of the voice

activity detection process. Due to the fact that the information criterion is unitless, its absolute value is not of interest to the user.

The thresholding methods considered in section 3.2.1 can identically be applied for the information criterion.

The information criterion is also known as the Akaike's Information Criterion (AIC), or the autocorrelation test.

3.2.5 Signal to Noise Ratio

A class of signal quality measures closely related to the Information criterion discussed in section 3.2.4, is the Signal to Noise Ratio (SNR). The motivation for the usage of SNR is similar to that of the Information criterion. A voiced sound, and indeed speech in all, is a, in some extent predictable signal and everything else is noise. Therefore, a high SNR value would indicate at least some kind of speech and probably a voiced sound.

Generally, the SNR is defined as

$$SNR = 10\log_{10}\frac{E_{signal}}{E_{noise}} = 10\log_{10}E_{signal} - 10\log_{10}E_{noise} \quad \text{[dB]} \qquad (3.13)$$

where E_{signal} is the signal energy and the noise energy is E_{noise}. This SNR definition does not specify which part of the total input signal is considered to be the desired signal. Thus, we have to define that separately.

The two most common definitions of the signal, in this context, are that the signal consists of the predictable components of the signal, or that the signal consists of the harmonic components of the signal. In the latter case, the SNR is often also called the Harmonics to Noise Ratio (HNR).

Linear prediction and the SNR

Linear predictive coding (LPC) is a well-known method of digital signal processing [19]. An SNR estimate can easily be calculated if the LPC is already known [5]. The assumption is that the part of the signal that can be modelled with LPC is the desired signal, and the rest is noise. Usually, it is easiest to calculate the noise energy, by making a noise filter, which cancels the predictable components.

Although the LPC method might seem as a good SNR estimator, it does present some problems. Namely, the calculation of the LPC requires essentially estimation of the autocorrelation matrix and inversion of that matrix. Both operations are computationally costly operations. However, in many applications, such as GSM-phones, it is already required to calculate the LPC in the speech coder. Then the calculation of the SNR is very easy as all the required data is available already [15].

Harmonics to Noise Ratio

Another possible assumption for SNR estimation is that the signal consists only of harmonic components, everything else is noise [20]. With different techniques, it is possible to estimate the Harmonics to Noise Ratio (HNR) either from the frequency spectrum or from the temporal signal after some transformations (e.g. zero-phase transformation). The frequency domain methods estimate the energy in the harmonic peaks. This requires searching of the peaks as well as identification of the peak widths. This operation requires a delicately tuned algorithm.

SNR and the Information criterion

The linear prediction method for SNR estimation relies mainly on information from the autocorrelation vector. The information criterion (autocorrelation test) uses the same information. They are thus, in effect, equivalent in view of the voice activity detection problem. However, the information criterion is easier to compute, as it doesn't require matrix inversion. The linear prediction method also requires some sort of filtering with a noise-generating filter. On the other hand, the SNR is a measure expressed in decibels, which is much easier to fathom than the unitless information criterion.

If we were able to scale the information criterion to same units as the SNR, that is decibels, the effort would certainly be worthwhile. Several different SNR and HNR measuring schemes have appeared in literature, but they all are more or less costly. Harnessing the information criterion to this task would provide for a new SNR measure, which is relatively simple to calculate. In addition, the mathematical background is well known, and there is a lot of available literature (see e.g. [21]). During this work some attempts to find such a suitable scaling were made, but with limited success. Intuitively it seems like the easiest approach would be to somehow measure the difference of formula 3.9 to the Normal distribution. This difference could perhaps then be scaled to the same units as the SNR.

3.3 Voice activity classification

Identifying speech and silence segments from a signal, that is, voice activity detection, is a typical case of a class of problems called pattern recognition problems. In short, the goal is to analyse the input signal, determine which parts are speech and which are silence, and return the identified segments as output.

So far, the goal has been to identify voice activity, but in order to be able to proceed, it is necessary to define more precisely what is to be identified from the input signal. The obvious choice would be to define two classes, speech and silence but alas, this approach is too simple. Some of the speech quality measures described in section 3.4 require identification of voiced segments, while other quality measures would prefer receiving all of the spoken segments. Therefore, the three classes sought for are voiced speech, unvoiced speech and silence.

3.3.1 Pattern recognition methods

Pattern recognition methods applicable to the present case can be roughly divided into two classes [22]: Knowledge based methods, and machine learning methods. The fundamental difference between these approaches is that knowledge based methods require that the application designer has, or can produce some knowledge of the problem. Based on this knowledge he designs some logical apparatus which (hopefully) solves the problem. On contrast, the machine learning methods do not require such a deep insight to the problem. However, these methods often require a model classification, a training set, with the aid of which the parameters are partly manually and partly automatically tuned for the particular problem.

Both these classes of methods usually require some other form of input data than just the raw input signal. These data, generally called features, measure some property of the input data. It is normally preferred that these features provide a complete description of the input data, but often this is not possible. If the description would be complete, the classification problem would be quite trivial.

3.3.2 Machine learning methods

Machine learning is most often associated with neural networks. These networks were originally designed to imitate the human brain, but soon a more general approach was adapted [23]. Today, the label "Neural networks" stands

for a wide variety of methods loosely related to the original concept. A common approach are the supervised learning methods. These methods start with a set of learning vectors, associated to a desired output. That is, there is a training data set, which is classified in advance to provide for the desired outputs. Then the general algorithm of machine learning methods is the following:

1. Initialise network
2. Calculate the output of the network for the next learning vector and compare it with the desired output.
3. Change the network in some way so that the network output would be closer to the desired output.
4. If there are more learning vectors, go to 2. Otherwise, go to 5.
5. If network output is close enough to the desired output for all learning vectors, then stop. Otherwise, go to 2.

The general problem with these methods is the generation of the training set. It can be difficult, expensive and time-consuming to generate a complete enough training set. The emphasis here should be laid on the words *complete enough*, because although it is hoped that the network would be able to generalise, it is still important to excite the system to all possible states in the learning process. This should ensure that the inherit "knowledge" of the system will cover all aspects of the process.

Another question of doubt is the convergence of the above algorithm. If the network converges to some point, is this result a local or global minimum? In this kind of very large non-linear optimisation problems, there generally is no way of knowing whether a minima is local or global [24]. We can usually not even know if the algorithm will converge to a local minimum. Practice has shown, though, that the algorithm does usually converge to some minima.

Self-organizing maps

To overcome the problem of training set generation, there exists another class of neural networks that are called unsupervised learning methods. One of these is the self-organizing maps (SOM) [25]. The SOM's are networks that try to automatically classify a pre-set number of different groups from a set of data [26], [23], [25].

The general idea of SOM's is to generate a mapping scheme from the input space to a representation space (the map) such that similar input

vectors are close in the representation space. This map can be generated by an iterative process. For each input vector, the closest map point is located. This map point, and some points in its neighbourhood are moved closer to the input vector. Therefore, similar input vectors will be mapped to map points close to each other. The iteration is continued until the map is stabilised. This process will generate a topology of the input space, which hopefully differentiates classes in the input space.

One of the most important network design considerations is choosing the number of neurons, or map points, in the map. Naturally, the number of neurons has to be large enough to describe all expected classes. In addition, it might be useful to include some extra neurons for any unexpected classes. However, increasing the amount of classes will make interpreting the map more difficult.

One difficulty lies now in the fact that it can be difficult to automatically identify classes from the map. There is now way to know in advance how the neurons will be distributed to the desired classes, or if the map is going to describe the desired classes or some other feature of the input space. Infact, there is no way to know if the map is going to converge to a stable topology.

In the present work, some attempts were made to design a SOM adapted to the voice activity classification problem. The results were quite good. For a ten-neuron network, the SOM clearly associated three or four classes to speech, some classes to various background noises, and the rest to silence. With this set-up, the network teaching process proceeded quite consistently on each trial. However, distinction between voiced and unvoiced sounds proved to be difficult. Labelling the neurons also proved to be difficult - almost as difficult as the original classification problem.

3.3.3 Knowledge-based methods

The development process of the voice activity measures described in section 3.2, gave quite a deep insight to the problem setting. The position was thus quite favourable for designing a knowledge-based classifier.

The first observation was that signal energy has a very strong correlation with the desired output (see figure 3.4). The first guess for an automatic threshold, that of the average energy of the signal, proved to be very good indeed. Only slight manual adjustment was needed. The signal energy provides a classifier between silence and speech, but nothing as means for separating voiced and unvoiced speech.

Now studying figure 3.4 we see clearly that when there is silence in the input signal, the signal energy is low, and goes below the threshold. At both utterances [s], the energy goes down, but stays above the sample threshold.

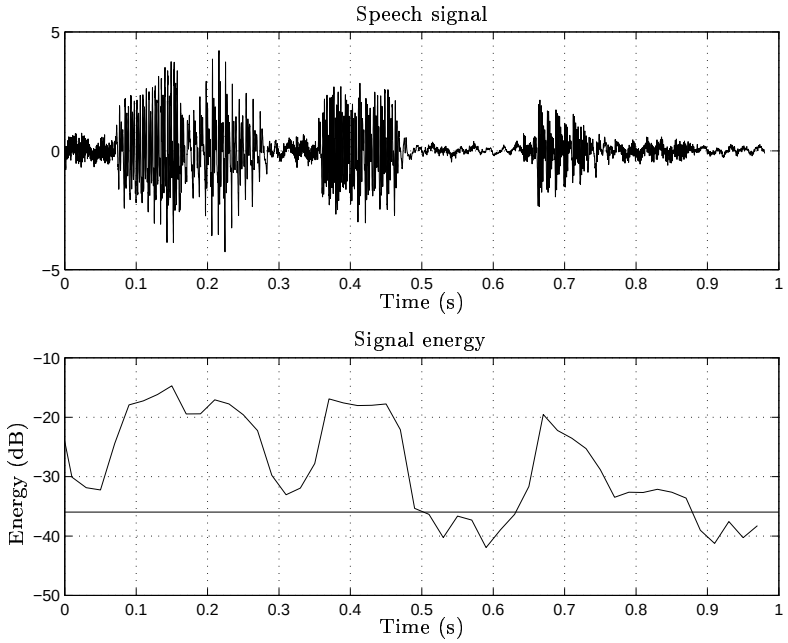

Figure 3.4: A sample speech signal [*siihen soittaa*], signal energy and a sample threshold.

It is also important to notice that the energy goes below the threshold in the middle of the word [*soittaa*] at the double [*t*]. Here, the double thresholding described in section 3.2.1, would obviously have helped.

The following task was to separate between voiced and unvoiced speech. Obviously, in voiced speech, there is very much energy in the fundamental frequency and its harmonics. Such a sound is also relatively stationary. Thus, the following measure used was the stationarity test, or the autocorrelation test.

In figure 3.5 a speech sample and its corresponding stationarity measure is presented. It is obvious that at both utterances [*s*], the stationarity drops significantly, and goes under the sample threshold. We should then classify these segments as unvoiced speech.

At the double [*t*] something curious happens. In what seems like silence in the interval 0.5 to 0.6, the stationarity still rises high. The same effect appears everywhere on silent segments. It was concluded that the background noise in the samples used is not white noise, but some other stationary sound.

The third measure included in this classifier was the pitch track. The algorithm of the pitch track was designed in such a way that if the algorithm could not find a valid F0 estimate, it would return zero in that frame. Therefore that frame does not contain a distinguishable pitch, and is thus

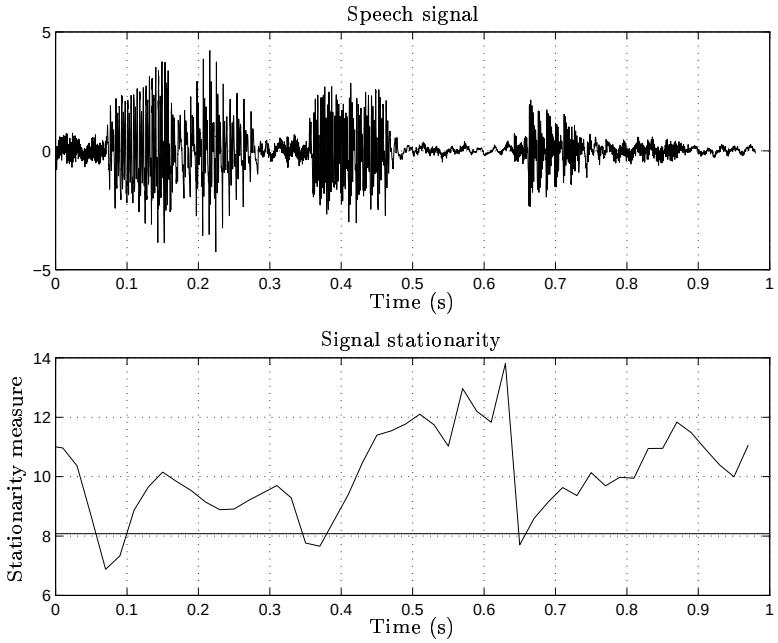

Figure 3.5: A sample speech signal [*siihen soittaa*], signal stationarity and a sample threshold.

not voiced speech. Hence, we can exclude all frames with zero pitch from the voiced segments.

The classification algorithm is thus in short:

1. Calculate pitch track, energy and stationarity for input signal.
2. Set thresholds for energy and stationarity (with the aid of a user interface).
3. Mark frames with energy above threshold as speech (voiced or unvoiced).
4. For all frames marked as speech, mark frames with stationarity above threshold and non-zero pitch as voiced, and others as unvoiced.

Although simple, the above-presented algorithm was found to be very effective. However, it requires some manual tuning for optimum performance. In the present case, this requirement was not a problem.

3.3.4 Post-processing methods

One problem often arising in classification problems is the segment discontinuities caused by misclassification in a single frame or short segments. Even though the rate of misclassification would be low, a single error will break a segment in two pieces. Consider the example in figure 3.6. After 169.1, the speech signal has a short break, which could be, for example, a stop consonant. The classifier labels this frame as silence, although it is a part of a word.

Similarly, at the end of the sample, the classifier identifies one frame as speech after the word, after 169.5. The correct classification would have been a single continuous segment, without either of these single deviations. The objective of misclassification detection, in this context, is to remove this kind of single misclassified frames. This can be achieved by searching for special sequences of frames, and modifying these frames in a predetermined way. A typical approach would be to replace all | 1 | 0 | 1 | sequences with | 1 | 1 | 1 |. Similarly all | 0 | 1 | 0 | sequences would be replaced with | 0 | 0 | 0 |.

In continuous speech, stop consonants such as [t], [k] and [p], especially in their non-aspirated flavours, produce short silent segments equal to the above mentioned misclassified frames. Such stops are typically shorter than 250ms in length. Therefore, all segment breaks shorter than 250ms could be labelled as unvoiced speech.

The dual of the speech segment breaks is the silence segment breaks. Typically, people seldom produce speech sounds shorter than 70ms. Therefore

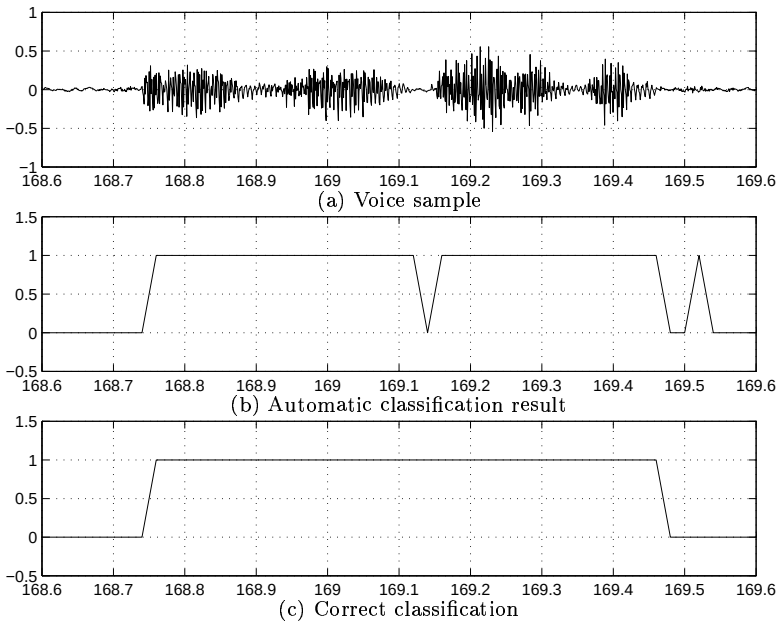

Figure 3.6: Example of segmentation discontinuities due to sparse misclassifications

any speech segments shorter than this should be labelled as silence. Both the shortest allowed speech segment and shortest allowed silent segment were set as parameters for the user.

In a practical application it is important to carefully choose the hierarchy of these segment break removal procedures. Consider the case presented in figure 3.7. In the original classification (3.7a) there are three speech segments, of which the two first are shorter than 70ms. All silent segments are shorter than 250ms.

If the too short speech segments are removed first as in figure 3.7b, only one speech segment remains. Likewise, if the too short silent segments are first removed (figure 3.7c), only on speech segment remains. Both classifications are wrong. In this case, the first speech segment, and the second silent segment are obviously misclassifications, and the correct classification would have been as in figure 3.7d.

Experiments show that the misclassifications prior to true speech segments appear more seldom than misclassifications in middle of a speech segment. Therefore, in the current work it was chosen to first remove too short silent segments and then the too short speech segments.

In the above example, we can see that short segments are more probably errors than long segments. That would suggest that it would be possible

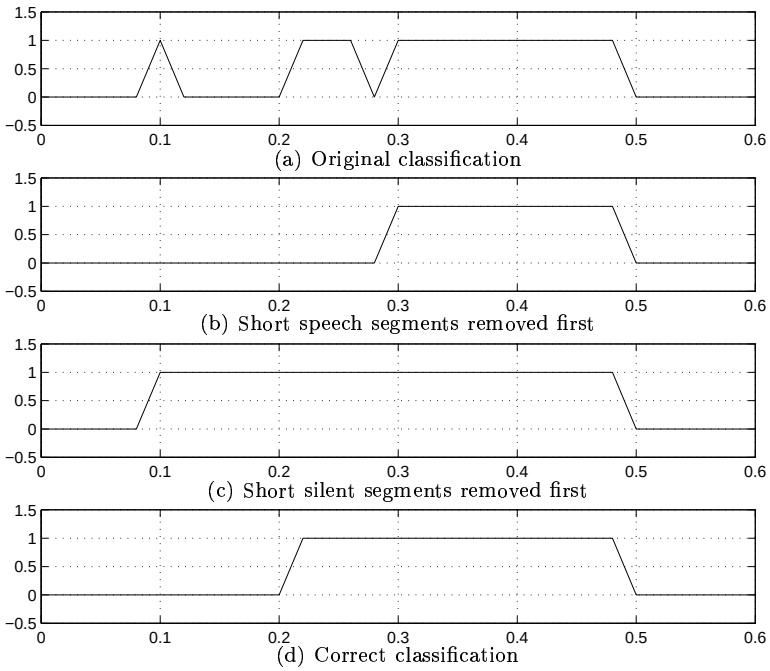

Figure 3.7: Example of differences due to order changes in misclassification removal

to build a more intelligent segment break removal algorithm, by starting with the shortest segments and continuing up to the segment length limit values. Most efficiently this would probably be done by some kind of a search-tree, with the segments at the leaves. Segment removal would then mean combining the current segment and its two adjoining segments to one segment. [27]

3.4 Voice quality measures

When the signal has been classified to speech and non-speech segments, it is possible to proceed with the objective to analyse the speech. Several different measures were analysed and developed to this task.

3.4.1 Speech to Silence Ratio

A basic and very intuitive measure of speech analysis would be the total time of speech in the signal t_{speech}. To give this measure a form uncorrelated with the total signal length t_{total}, we define the speech to silence ratio as

$$SSR = \frac{t_{speech}}{t_{silence}} = \frac{t_{speech}}{t_{total} - t_{speech}} \tag{3.14}$$

where the silence time is $t_{silence} = t_{total} - t_{speech}$.

In the current study, we have two definitions for speech, voiced sounds and all speech sounds. Therefore, we can apply this Speech to Silence Ratio similarly for voiced speech. In this work, this ratio will be called Voiced to Silence Ratio. defined as

$$VSR = \frac{t_{voiced}}{t_{silence} + t_{unvoiced}} = \frac{t_{voiced}}{t_{total} - t_{voiced}} \tag{3.15}$$

In addition, the Unvoiced to Voiced Ratio might be interesting:

$$UVR = \frac{t_{unvoiced}}{t_{voiced}} \tag{3.16}$$

3.4.2 Signal to Noise Ratio

Another basic signal quality measure is the Signal to Noise Ratio (SNR) [28]. The SNR is defined as

$$SNR = 10log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) dB \tag{3.17}$$

Where σ_x^2 is the input signal variance representing the signal power and σ_e^2 is the noise variance representing noise power. In the present case, speech and silence energies were calculated from the average power in the corresponding segments, as labeled by the classifier. Similarly, the Voiced to Noise Ratio was defined as average voiced power to average noise power ratio. From this definition, the unvoiced power was left out on purpose, as it is by definition neither unvoiced speech nor noise.

Both of the measures described above are more useful as measures of recording quality, as of voice quality. This was still deemed to be a useful measure as it could indicate changes in the recording conditions, or classification problems.

In section 3.2.5 SNR measures were discussed in view of voice activity detection. Of these SNR measures, some could be calculated frame by frame, instead of for the whole signal (as presented above). Such a SNR measure would be interesting in analysing voice quality. It could describe qualities similar perceptual properties such as hoarseness. Especially if the Information criterion could be harnessed to SNR estimation, it would provide an effective continuous SNR estimate.

3.4.3 Fundamental frequency and pitch track

According to several publications [2], [6], [7], [8], [29], the fundamental frequency (F0) plays an important role in detection of different voice qualities. These qualities include both emotional qualities as well as qualities related to voice disorders. It is therefore useful to include F0 and pitch track analysis into this study.

Statistical F0 description

The most common statistical properties for any signal are mean, standard deviation (or variance), maximum, minimum and median. All these properties are interesting descriptors for F0 statistics, too.

95% F0 range

The F0 estimates and the pitch track sometimes generate clearly erroneous values. These errors are distributed unpredictably along the pitch track, and the erroneous values are close to random values. The errors will therefore present a bias for the statistical descriptions of the F0. To overcome this problem, a 95% F0 range filter was applied. The idea is the following: Excluding 5% of the most extreme values, that is, the highest 2,5% and lowest 2,5% of the F0 estimates, will (hopefully) remove the worst F0 estimate errors.

For this new pitch track, the same statistical properties were calculated as for the total pitch track.

3.4.4 Number of periods

When the vocal folds oscillate, they will collide in each other once in each period. These collisions will probably inflict some stress on the tissue. As a measure of this stress, the number of collisions, or the number of periods can be estimated.

The pitch track has already been estimated, and the voiced speech segments likewise. We can then estimate the number of periods by

$$N = \sum_i l f_I \tag{3.18}$$

where N is the number of periods, l is frame length and f_i the average frequency in frame i. The summation goes over all voiced frames. Defining a vertical vector $\mathbf{v}$ which is one for all frames with voiced speech, and zero elsewhere, we have

$$N = l\mathbf{v}^\mathbf{T}\mathbf{f} \tag{3.19}$$

Where $\mathbf{f}$ is the vertical vector consisting of all f_i's.

It should be clear that even though the true value of number of periods N is a natural number $\in \mathbb{N}$, this estimate will be a fractional number $\in \mathbb{Q}$. This is not a problem. The estimate is at least as good as the pitch estimate, because no new approximations are done at this stage. Furthermore, the summation will average the variance present in the pitch track, so the total variance will decrease. Still, any bias in the pitch track will be transferred to the number of periods -estimate.

3.4.5 Segment length analysis

The lengths of segments provide some interesting information of speech. Statistical analysis of the lengths can provide information about the speaking habits of the subject as well as speaking conditions. The lengths of following segments were extracted:

- speech (both voiced and unvoiced) segments
- voiced segments
- unvoiced segments
- all silent segments
- silent segments shorter than 2 seconds (< 2s)
- silent segments longer than 2 seconds (> 2s)

For all these some statistical analysis were performed. Namely, the mean, standard deviation, median, minimum and maximum values were calculated, and a histogram was plotted. The interpretation for these length measures is mostly clear, but the meaning of the two sub-classes of silent segments requires some more explanation. The reasoning is that silent segments longer than two seconds are not just pauses in speech, but more like stopping the speech and then, after a while, starting all over again. In other words, in continuous speech speakers sometimes have pauses between sentences or phrases, but these will normally not be longer than 2 seconds. Therefore, an analysis of all silent segments shorter than 2 seconds will provide information about the speaking habits of the subject. Pauses longer than 2 seconds will similarly tell more about the speaking conditions.

Some comments on the validity of these measures are necessary. The process for generating the segment lengths is long and complex. Each step has some error distributions and furthermore there are steps with user intervention, too (i.e. thresholding). To keep results between different recordings comparable, each step in the measurement, thresholding and classification has to be done exactly in the same way.

In addition, the recording quality and the speech "quality" will affect the results. That is, if the noise level increases, or the sonority of the voice decreases, the identified speech segments (both voiced and unvoiced) will be shorter. This is a property of the measures used before classification. Due to all these possible error sources, these measures should be used with care.

3.4.6 Spectral properties

The spectrum of the speech signal contains some usable voice quality features [30]. These voice quality features describe the shape of the spectrum in some (preferably) robust manner. A central issue in this analysis is the spectral shaping effect of the vocal tract. The vocal tract shapes the spectrum, in an unpredictable way, if the phonation is not known. Therefore, we have to include the effect of the vocal tract to all stages of the analysis.

Temporal length of analysis window

Before we can discuss the spectral properties, we must define what is meant by a spectrum. The main issue is the choice between a short time spectrum or a long time average spectrum (LTAS). In this context, the length of a short time spectrum window would be 20 ms, as was chosen as the common window length in section 3.1. Then the short time spectrum will be strongly correlated with the phonation. When it is desirable to avoid the correlation to

the phonation, then the LTAS is sometimes used. The LTAS is the average of the spectrum for the whole speech signal. Background noises can be difficult when generating a LTAS, because they can bring properties to the spectrum that do not exist in the speech. Therefore, background noises should be excluded from the speech signal in some manner.

The α-criteria

A criteria describing the spectral tilt, is the α-criteria. It is defined as the ratio between energy below 1kHz and above 1kHz. The α-criteria is one of the simplest spectral properties, and in its simplicity, quite robust. For a short time spectrum the phonation will have a significant effect on the α-criteria, but in the current work, the α-criteria could be averaged over the all the speech segments, thus averaging the effects of phonations. Still, this approach avoids most of the difficulties with LTAS's and background noise. In addition, averaging the α-criteria over time, instead of averaging the spectrum, also diminishes the emphasis of strong phonations. That is, in the LTAS α-criteria, the emphasised vowels will have a strong shaping effect on the spectrum, where as other vowels have a smaller shaping effect. This brings some bias to the criteria. Averaging the α-criteria over time, will treat all phonations with equal importance, thus giving a more realistic estimate over the underlying system.

Other LTAS-criteria

In his article, Kitzing also mentions several other LTAS-criteria than the α-criteria [30]. Apart from the α-criteria, the most important LTAS-criteria are:

- The fundamental frequency F_0 and the level of the fundamental frequency L_0.
- The first formant F_1, and its level L_1.
- The minimum point F_{min} between F_0 and F_1, and its level L_{min}.
- The ratio between energy in ranges 0.3-0.8kHz and 1.5-3.0kHz.
- Median of the spectral energy.

These criteria have to be calculated from a LTAS, because the phonation will too often hide the formant positions, thus disrupting the result for short time spectra.

3.4.7 Inverse filtering

As it was mentioned before, the vocal folds are the most interesting object when analysing voice-loading effects. Unfortunately, the possible methods of recording vocal fold movements are limited. The vocal folds are located in such a place that recording their movement is difficult. The sound radiating from the lips can not either directly describe the vocal fold movement, because the vocal tract filters the sound signal. However, there are methods designed to remove the effect of the vocal tract from a sound signal. The Iterative Adaptive Inverse Filtering (IAIF) is such a procedure [31].

The IAIF computes a model of the vocal tract transfer function. Then the effect of the vocal tract is cancelled by inverse filtering, applied with an iterative procedure. The result is then an estimate of the glottal flow. This glottal flow is directly correlated with the vocal fold movement.

Glottal flow properties

There are several interesting properties available in the glottal flow signal, for example fundamental frequency (F0), sound pressure level (SPL), sub-glottal pressure, negative peak amplitude of differentiated flow, AC amplitude of glottal flow and many more. Many of these features have been found to correlate with different loading effects [32]. In addition, the skewing of the pressure pulse and other properties of the shape of the pressure pulse has been found to correlate with different loading effects.

3.5 Performance considerations and optimisation

3.5.1 Matlab functions

The initial assumption was that all calculations are done off-line, so that real-time calculation requirements would not constrain the algorithms. The first approach for the off-line implementation was to take in the data as one block, do the required calculations and return a resultant vector. However, when the input data was large, this approach presented serious problems. Every so often, the required amount of temporary variables would be a multiple of the input data size. That is, for a five minute mono recording with 11025 kHz sampling rate, the input vector would require 25Mb of RAM memory. Assuming that six temporary variables would be needed (for example pitch tracking by feature extraction in section 3.2.3), the temporary variables would require 150Mb of RAM memory. On top of these, the operating system and the other software components require some 200Mb of memory, which totals to about 400Mb. It is clear that with the available hardware this was far too much. Although the operating system did allow such memory usage through swapping to the hard disk, it increased time consumption roughly by a factor of ten.

The first solution to this problem was to divide the processing to smaller time frames (i.e. 20ms to 5 s). This makes the algorithms far more complex, as many functions need information about signal continuity. However, with some effort it was possible, though it increases somewhat the number of operations required. The total processing time was still decreased.

The computing time for a five-minute sample presented also other problems. After the memory usage optimisation, the computing time was still about ten minutes. Studying the Matlab profile reports showed that almost half of the running time was spent in function 'xcorr' which calculates correlation matrixes, and in this case especially autocorrelation matrixes. A simple trick replacing the 'xcorr' function with the inverse Fourier transformation of the absolute value of the power spectrum proved to be significantly faster. This method uses one tenth of the amount of floating point operations and half the time of function 'xcorr'.

Some additional optimisation was done in a fairly standard ways:

- All unnecessary operations were moved outside all loops.
- Loops and if-else constructions were replaced with Matlab's built-in vector and matrix operations as far as possible.

– Most final functions were compiled to Matlab's own binary format (i.e. Mex-files).

The two last points are very easily justified by the fact that especially the loops and if-else constructions are very slow when executed in the standard .m-file format. Compiling the code gives some improvement, but especially the built-in matrix functions are often especially effective.

3.5.2 LabVIEW and the LabVIEW-Matlab interface

LabVIEW is a graphical programming environment, where the ease and visualisation of programming has been emphasised. The compromise is, however, that the program execution is slow. Therefore, all functions that require complex calculations were implemented in Matlab. Again, this does present some new problems.

In the Windows operating system, the data-exchange interface between LabVIEW and Matlab is implemented through the ActiveX-library, which, in our case, presents some constraints. Two of these constraints should be especially emphasised, namely, memory usage for larger files and programming errors (also known as 'bugs') in the interface. The memory usage of the ActiveX-library interface was surprisingly large, that is, transferring a 25Mb wave-file through the interface required more than 90Mb of memory. This kind of memory consumption was deemed unnecessary and unacceptable due to speed considerations.

However, the other problem with ActiveX was far more worrisome. When transferring large files through the interface, some obvious programming errors in the interface were found. That is, the interface apparently had some memory overflow problems, as quite regularly the computer would crash or otherwise behave strangely after, and during the transfer. The most worrisome part of this is, firstly, that documentation for ActiveX is, both incomplete and very complex. Secondly, the source code for LabVIEW, Matlab and ActiveX is unavailable. Thirdly, even if the source of the problem would be found, it is quite unlikely that the corresponding vendor would have a fast correction for it.

Both problems were bypassed by transferring the data not through the ActiveX-library but by saving the file on the hard-drive in LabVIEW, and re-reading it in Matlab. This workaround was found to be about 30 seconds faster and used 90Mb less memory for a 25Mb wave-file.

Chapter 4

Application environment

4.1 The recording environment

The sound recordings used in this work were recorded at the Sonera Telephone service unit in the city of Kuusamo. The recordings were executed in a relatively silent room separate from the other workers. The environment was otherwise similar to their normal working offices. The subjects wore a headset microphone (model AKG CK 97-0) which held the microphone on constant distance of XX cm from the mouth. The signal was recorded onto a DAT recording device (model Sony DAT TCD-D3).

The subjects were personnel from the Sonera Telephone service unit, consisting of 10 males and 30 females. A subset of these, two males and two females were used for application development and evaluation.

4.2 Analysis hardware

4.2.1 The computer

The requirements for the computer hardware are fairly loose. The only important thing is that it is computer which can run Microsoft Windows 95, 98, NT4, or later. The amount of memory has a crucial impact on speed - anything above 128Mb should be enough. The question of computer speed is still mostly a question of users patience. With the Pentium III 350Mhz on which all software development was done in this project, the speed was tolerable. The analysing time for a five-minute recording is around five minutes. If the computing capacity of the processor is lower or higher, the computing time will sink or rise accordingly.

Storing capacity for wave-files and result files should not be a problem for

modern computers. Only one wave-file (of about 25Mb) is stored at a time, and the intermediate results are stored in files of about 0.5Mb each. At the current trend, with harddrives being generally of order 5Gb, these file sizes do not induce any problems whatsoever.

4.2.2 The audio input card

The audio input card, also known as the sound card, or AD-converter, is the hardware interface between analog audio equipment, such as a microphone or a tape recorder, and the computer. The card converts the analog sound signal to a digital form. The requirements or the selection criteria for the audio card set for this project were the following:

- Flat amplitude response on the range from 10 Hz to the Nyquist frequency.
- Signal to noise ratio and harmonic distortion as small as possible.
- At least two analog input channels.
- At least two digital input channels.
- Precision at least 16 bits.
- Sampling frequencies available between 8 kHz and 44.1 kHz.

In order to find the audio card fulfilling the above criteria and was the least expensive, a large amount of specifications of different cards was reviewed. The chosen card was Turtle Beach Montego II Plus.

With this equipment, it was also possible to transfer data from DAT-recordings to the computer directly in a digital form. As the sounds are usually not recorded on-site with the analysis equipment, some data copying is always involved. This possibility of making digital copies is therefore preferable to all other options.

4.3 Software

4.3.1 User interface

The original and most important requirement for the whole project was that the system would be easy enough to use in a hospital environment. This short statement is a big constraint from the point of view of programming. An easy user interface is never easy to do.

The software development package chosen for the task was LabVIEW by National Instruments. It is a program specially designed for development of user interfaces for measuring instruments. It has its own graphical programming environment and language. LabVIEW is widely used in package in industry in-house applications and has a broad range of free applications available on the Internet.

4.3.2 Mathematical programming

In mathematics and engineering for example in the academic world, the defacto standard of mathematical computing is the program package Matlab by MathWorks Inc. Several similar programs are available, but they are mostly just worse copies or imitations of Matlab. Therefore, if a mathematical programming component outside LabVIEW was needed, Matlab would be the choice. First, LabVIEW was inspected if its capabilities would suffice.

LabVIEW has a broad collection of mathematical instruments. They can be categorised as follows:

- Graphical flow-chart like components with basic mathematical operations ranging from summation and multiplication, to integration, basic matrix operations and some simple digital signal processing functions.
- For a bit more complex equations, a flow-chart object for mathematical equations.
- An add-in program package HiQ, for complex mathematical programming.
- An interface for communication with mathematical programs trough the ActiveX-library. Available only in the Windows operating system.

In our case, the two first points are far too simple. Some experiments were conducted to determine the efficiency of the functions, but the performance was far off the mark. Functions that for example Matlab performs in an instant, could take tens of seconds in LabVIEW.

The choice between HiQ and Matlab was less clear. For the author, Matlab was a familiar language, and some functions were already available. Most of Matlab's properties, good as well as bad ones, were well known. In contrast, HiQ would have required a lot of studying before mastered on a sufficient level. In addition, there was some doubt that HiQ's performance and capabilities would not be of the same level as Matlab's. Therefore, Matlab was chosen for mathematical computations.

Chapter 5

Experimental results

5.1 Classifier performance

First experiments with the relatively simple classifier developed, proved that the classifier is surprisingly effective. When the speech signal is visually and auditory evaluated, and compared with the classifier result, the result is most satisfactory. Speech and silence are recognised in general very well. Distinction between voiced and unvoiced segments has a bit more errors, but is still most of the time correct.

The most common problems are rapid changes from voiced phonemes to unvoiced and back to voiced. If the true unvoiced segment is short, and not in the middle of a frame, the autocorrelation test will see the ending of the voiced segment and the start of the new voiced segment, but will not pay attention to the unvoiced segment. This is probably caused by the fact that the unvoiced phoneme usually has less energy than the voiced phoneme and the autocorrelation will thus emphasise the voiced segment.

Another problem is the accuracy of the beginnings and endings of speech segments. The energy level usually rises and descends slowly, and it is difficult to decide where to set the segment border. This problem would be relaxed with the hierarchic segment unification scheme presented in section 3.3.4, but that is probably only a partial solution. To further improve the segment border accuracy, some new information is needed. This information could be for example, some other feature of the speech signal (e.g. spectral property), or shorter or overlapping analysis frames. The frames cannot be too much shorter, because then the frames would begin to see individual signal periods. On the other hand, using overlapping frames would significantly increase processing time.

5.2 Accuracy of measures

5.2.1 Pitch extraction

The most important measure of the current software was pitch. It is therefore important to verify that the extracted F0's are correct.

Pitch extraction performance was manually verified by comparing the reciprocal of the manually measured period length with the pitch estimate, on several occasions. The values calculated by the software agreed well with the manual measurements. However, the values do not match exactly. The manual measurements are done by measuring the period length - two analysis points - while the autocorrelation based pitch estimation does a continuous calculation over a frame. The autocorrelation will thus give an estimate based on each point in the frame, while the manual estimate is based on the distance between two features of the wave. The autocorrelation uses therefore a larger set of data and should be more accurate. However, proving the accuracy would be tedious.

Transitions from voiced to unvoiced, or vice versa, proved to be difficult. For voiced phonemes the pitch estimate is defined, while for unvoiced it can give random values. The theory does not give the outcome *a priori* in the region between, or the transition from voiced to unvoiced. Studying pitch extraction results in comparison to the original signal, shows that these transitions are handled relatively well. The median filtering over five consecutive frames most often stabilises the value in the transition regions.

A common problem in pitch analysis is the harmonic frequencies and octave jumps. Sometimes the first harmonic frequency is dominant in energy over the fundamental frequency, so that the harmonic is, falsely, identified as the fundamental frequency. For autocorrelation tests, the problem is inverse. When there is a large correlation at a distance T, there should also be large correlation at $2T$. This distance $2T$ represents half the fundamental frequency. If the frequency limits are too narrow, this half frequency correlation is identified as the dominant correlation. On the other hand, using wide frequency limits will increase the possibility of other estimation errors.

Some experiments with the current software show that it indeed is difficult to choose good frequency limits. Generally, the limits were too narrow. Only when the limits were chosen far larger than the frequencies that were expected, did the pitch extractor give correct octave values.

5.2.2 Segment length measures

In section 5.1 the classifier performance was discussed. The segment length is very closely related to the same problems. It should be emphasised that the segment length measures are especially sensitive to manual selection of the thresholds.

Acquiring exact verification of the segment lengths is difficult. Only a visual comparison of the original signal and the segmentation is possible. However, the sensitivity can easily be shown by analysing the same data with small changes in the thresholds.

5.2.3 Other measures

Verifying performance of the other measures is difficult. Measures such as the alpha-ratio cannot easily be measured manually from the original signal. Therefore, the results were checked only to see that they are in the expected range.

5.3 Execution speed

A large part of the development effort was related to optimisation of execution speed. The amount of data is very large, and the calculations complex, so expectations should not be set too high.

The main reason why execution speed is an issue, is usability. The faster the program, the more comfortable it is to use.

Experiments conducted on the current software showed that the classification and analysis of a sound sample would take about one second per recorded second. That is, analysing a five-minute recording will take about five minutes. Copying the original recording to the computer will also take five minutes. The manual thresholding will also increase the total analysis time, so that the total analysis time for a five-minute recording is about 15 minutes.

Chapter 6

Conclusions and views on further development

This paper presents methods and development of an analysis environment for voice quality analysis of continuous speech. The main objective of this project was to create a speech analysis environment suitable for usage in hospital environment. This objective was fulfilled. The software works as it was supposed to do and is easy to use. However, many points that could be improved are still left.

One of the primary points of improvement is the post-processing of classification results. That is, the hierarchical segment unification discussed in section 3.3.4. This process would increase the percentage of correctly classified frames, and would relax the requirements on correct thresholding, thus improving the accuracy of segment length analysis.

The double thresholding scheme discussed in section 3.2.1, is another method which could improve the classification reliability. Some kind of automatic threshold generation could be a justified choice in order to avoid more user intervention. The offset threshold could be combine with some logic, for example that a silent segment is not started before three consecutive frames below the threshold are found. Alternatively, the lower threshold could be calculated by some simple arithmetic using the average of all values below the higher threshold, and the value of the higher threshold as reference levels.

One possibly minor improvement would be the option of storing all wavefiles on some medium for permanent storage. For example, the software could provide support for storage on writable compact disks (CD-R). This would make it easier to repeat the analysis later, for example when some new features are implemented in the software.

In view of speech quality research as a whole, the most interesting possibility of the current work is the relation of the autocorrelation test to Signal

to Noise Ratio. Finding an algebraic relation between these would require some work. Nevertheless, if the relation is found, it would provide an effective new SNR measure. Some experiments were done already, and they show a clear relation, when a single sinusoid is analysed in additive white noise. However, a mixture of several sinusoids did not give as satisfactory results.

For this particular project, perhaps the most interesting development direction is inverse filtering and the many new measures that it offers. Inclusion of inverse filtering in the current software would probably require a different user interface than the one provided so far. Implementing such an interface is easy to do, but it requires a lot of work. Some thought should also be given to the choice of on which parts of the signal the inverse filtering should be applied. Inverse filtering background noise might give unpredictable results. However, filtering only speech segments brings the issue of how to concantate speech frames.

Concluding, the current objective was reached, but at the same time, many new possibilities were found. Some additional tests are still required to verify the reliability of the software for different subjects, but so far, no indications have been evident that different subjects would present any new problems.

Bibliography

[1] Haataja K. *Diplomityö: "Kliinisen puheentutkimuslaboratorion puheenanalysointiympäristö"*. Oulun Yliopisto; 1993

[2] Titze IR. *Principles of Voice Production*. Englewood Cliffs, NJ: Prenitce-Hall; 1994.

[3] Catford JC. *Fundamental problems in phonetics*. Edinburgh University Press, Edinburgh; 1977.

[4] Klingholz F. "Acoustic recognition of voice disorders: A comparative study of running speech *versus* sustained vowels", *J. Acoust. Soc. Am.*, 1990;87(5):2218-2224.

[5] Qi Y, Hillman RE, Milstein C. (1999) "The estiamtion of signal-to-noise ratio in continuous speech for disodered voices". *J. Acoust. Soc. Am.*. 1999;105(4):2532-2535.

[6] Leinonen L, Hiltunen T, Linnankoski I, Laakso M-L. "Expression of emotional-motivational connotations with a one-word utterance", *J. Acoust. Soc. Am.*, 1997;102(3):1853-1863.

[7] Scherer KR. "Expression of Emotion in Voice and Music". *Journal of Voice*. 1995;9(3):235-248.

[8] Vilkman E, et al. (1997) "Loading Changes in Time-Based Parameters of Glottal Flow Waveforms in Different Ergonomic Conditions", *Folia Phoniatrica et Logopaedica*, 1997;49:247-263

[9] Vilkman E., et al., (1998), "Ergonomic conditions and voice", Logopedics, phoniatrics, vocology, 23, pp. 11-19

[10] The American Polygraph Association Homepage [online], [cited 30th November 2000], URL:http://www.polygraph.org

[11] Trustech Ltd. Homepage [online], [cited 30th November 2000], URL:http://www.truster.com

[12] Origin Data Realisation Limited Homepage [online], [cited December 1st, 2000], URL:http://www.origin-data.co.uk

[13] Kay Elemetrics Corp. Homepage [online], [cited November 30th, 2000], last modified November 21st, 2000, URL:http://www.kayelemetrics.com

[14] Boersma P. *Praat: Doing phonetics by computer* [online], [cited November 30th, 2000], URL:http://www.praat.org

[15] Steele R. *Mobile Radio Communications*. Chapter 8. Pentech Press, London, 1995.

[16] Tanyer SG, Özer H. "Voice Activity Detection in Nonstationary Noise". *IEEE Trans. Speech and Audio Processing*. 2000;8(4):478-482.

[17] Rabiner LR, Schafer RW, *Digital Processing of Speech Signals*, Prentice-Hall; 1978

[18] Söderström T, Stoica P. *System Identification*. Prentice Hall International, University Press, Cambridge, UK, 1989. Ch.11:423-426

[19] Makhoul J. "Linear Prediction: A Tutorial Review", *Proceedings of the IEEE*, 1975;63(4):561-580.

[20] Qi Y, Hillman EH. (1997) "Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals", *J. Acoust. Soc. Am.*, 1997;102(1):537-543

[21] Bozdogan H. "Akaike's Information Criterion and Recent Developments in Information Complexity", *Journal of Mathematical Psychology*. 2000;44:62-91.

[22] Schalkoff, RJ. *Pattern Recognition: Statistical, Structural and Neural Approaches*. New York:Jon Wiley & Sons; 1992

[23] Haykin S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. New Jersey:Prentice-Hall; 1998

[24] Bazaraa MS, Sherali HD, Shetty CM. *Nonlinear programming. Theory and algorithms*, Second edition, New York: John Wiley & Sons, Inc.; 1993.

[25] Kohonen T. *Self-Organizing Maps*, Springer Verlag; 1995.

[26] Demuth H. et al, *Neural Networks Toolbox User's Guide*, Fifth printing, Natick MA:The MathWorks Inc.; 1998

[27] Gonzalez RC, Woods RE. *Digital Image Processing*, New York:Addison-Wesley; Chapter 7, pp. 413-443, 1993

[28] Mitra, S.K., *Digital Signal Processing, A Computer-Based Approach*, McGraw-Hill, New York, Chapter 9, pp. 591-594, 1998

[29] Lauri R-L. "Effects of Prolonged Oral Reading on Time-Based Glottal Flow Waveform Parameters with Special Reference to Gender Differences", *Folia Phoniatrica et Logopaedica*, 1997;49:234-246.

[30] Kitzing P. "LTAS criteria pertinent to the measurement of voice quality" *Journal of Phonetics*. 1986;14:477-482.

[31] Alku P. *An Automatic Inverse Filtering Method for Analysis of Glottal Waveforms*; acad diss Helsinki University of Technology, Helsinki, 1992.

[32] Vilkman E., et al, (1999), "Effect of Prolonged Oral Reading on F0, SPL, Subglottal Pressure and Amplitude Characterisitcs of Glottal Flow Waveforms", *Journal of Voice*, 1999;13(2):303-315

[33] Rossing T.D., *The Science of Sound*, 2nd ed., Addison-Wesley, New York, Part IV, 1990

Index