Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Semesterarbeit SA-2003.06

Paralinguistische Sprachtransformationen

Wintersemester 2002/2003
12. Feb. 2003

Jun Ma

Supervisors: Ulla Glavitsch
Dr. Beat Pfister

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Contents

1 Introduction 1

2 Paralinguistic transformation rules 3

2.1 Formant frequency transformation 3

2.1.1 The Bark Frequency Scale 3

2.1.2 Formant transformations 5

2.2 Fundamental frequency F0 transformations 6

2.3 Resonance bandwidth transformations 6

3 Method formant estimation and transformation 8

3.1 Introduction 8

3.1.1 Formant estimation with ESPS-formant 8

3.1.2 Formants, bandwidths and fundamental frequency transformation 9

3.1.3 Cascadefilter 10

3.1.4 LPC-Synthesis 10

3.2 Conclusion 11

4 Method Pole transformation 13

4.1 Introduction 13

4.1.1 LP-Filter 13

4.1.2 LPC-Analysis 14

4.1.3 Pole-Transformation 14

4.1.4 LPC-Synthesis 16

4.2 Conclusion 17

5 Conclusions and Outlook 19

Preface

This report contains the result of my semester work in WS2002/2003 in Computer Engineering and Networks Laboratory at Swiss Federal Institute of Technology. This work gives me an opportunity to get experience within the range of paralinguistic speech signal transformation. At the same time it is also a large challenge. It is really not easy to find out a correct method to reach the goal. Finally I have implemented two methods to investigate the transformation. Though both methods are not satisfied, but during this work I have learned a lot of knowledge and got a lot of happiness. Hereby I would like to thank my supervisor Ulla Glavitsch and Dr. Beat Pfister who always help me to solve the problem.

Jun MA

Abstract

To investigate the paralinguistic speech signal transformation for German, a study is being done in TIK at ETH Zürich. In my work German speech signals from 10 men and 10 women have been analyzed, using formant estimationtransformation and pole transformation, and resynthesized with the characteristic frequencies transformed according to the transformation rules from H.Traunmüller. Speech signals transformed this way in speaker age, in liveliness, and from male to female speaker developed to partially highly natural and from female to male developed to unsatisfied. For each method a conclusion has been listed. At the end of this report is a short conclusion and outlook.

1 Introduction

Paralinguistic speech signal transformation is defined as speaker classes transformation. The speaker classes are for example men, women, girls, and boys. A voluntarily transformation between these two classes is expected in this work. Paralinguistic speech signal transformation only transforms speech voice and holds the linguistic characteristics of speech voice, in this way, the sounds and words are not changed, so it is possible to transform the age, sex or the emotional involvement of speaker. About 15 years ago H.Traunmüller developed this transformation for Swedish. His attempt was very well and can be heard under www.ling.su.se/staff/hartmut/mainpul.htm.

A simple type of transformation rules has previously been shown to relate the characteristic frequencies of the same vowels that differ in paralinguistic quality. It is observed that these transformation hold not only for vowels but for any kind of speech segments.

The physical properties of speech sounds are known to vary as a function of paralinguistic factors such as the speaker's age, sex, vocal effort and emotional involvement. This variation concerns also formants F1 and F2 in vowels. Here the formants are the resonant frequencies of the vocal tract. We designate them as F1, F2, F3, and F4. Given constant paralinguistic circumstances, however, different vowels are distinguished almost exclusively by these two formants frequencies. Considering their paralinguistic variation, it is understood that the perception of vowel quality cannot be based on these formants frequencies as such. Nevertheless, it is obvious that ordinary speech signals contain invariant correlates to the phonetic quality of vowels. It has been suggested that a process of normalization of formant frequencies guided by other vowels produced by the same speaker under similar conditions might be in effect. Although it has been shown that perceived vowel quality is affected by a preceding context in this sense, this does not provide an exhaustive explanation of the phenomenon. As listeners we are able to judge the phonetic quality even of a single isolated vowel, no matter by whom it has been produced, given only that we can hear the signal clearly. Therefore it must be presumed that the speech signal contains properties informative of phonetic segmental quality, free from paralinguistic variation.

The relations between phonetically identical vowels produced under different paralinguistic conditions could be described by simple transformation rules. All these relations could be interpreted as linear transformations of the characteristic frequencies on a logarithmic scale as well as on a tonotopic scale (Bark).

From the study of H. Traunmüller we know that transformation rules found to hold for vowels would apply not only to vowels but also to speech signals in general. Paralinguistic variations involve also certain variation, in addition to those in fundamental frequency F0, that in a traditional framework are ascribed to the voice source signal,i.e., to the shape of the glottal pulses. The transformation rules do not capture this type of paralinguistic variation. The results of the transformations may, accordingly, be expected to be deficient in naturalness.

The paralinguistic transformation bases on transformation of formants, their bandwidths and the fundamental frequency. The new speech signal is resynthesized with the characteristic frequencies transformed and it must be natural and clear. The determination of formants, bandwidths and fundamental frequency are the emphasis of this work.

2 Paralinguistic transformation rules

2.1 Formant frequency transformation

2.1.1 The Bark Frequency Scale

The Bark scale ranges from 1 to 24 Barks, corresponding to the first 24 critical bands of hearing. The published Bark band edges are given in Hertz as [0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500].

The published band centers in Hertz are [50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500, 13500]. These center frequencies and bandwidths are to be interpreted as samplings of a continuous variation in the frequency response of the ear to a sinusoid or narrowband noise process. That is, critical-band-shaped masking patterns should be seen as forming around specific stimuli in the ear rather than being associated with a specific fixed filter bank in the ear.

Note that since the Bark scale is defined only up to 15.5 kHz, the highest sampling rate for which the Bark scale is defined up to the Nyquist limit, without requiring extrapolation, is 31 kHz. The 25th Bark band certainly extends above 19 kHz (the sum of the 24th Bark band edge and the 23rd critical bandwidth), so that a sampling rate of 40 kHz is implicitly supported by the data. The researcher in Stanford University have extrapolated the Bark band-edges in their work, appending the values [20500, 27000] so that sampling rates up to 54 kHz are defined. While human hearing generally does not extend above 20 kHz, audio sampling rates as high as 48 kHz or higher are common in practice.

The Bark scale is defined above in terms of frequency in Hz versus Bark number. For computing optimal allpass transformations, it is preferable to optimize the allpass fit to the inverse of this map, i.e., Barks versus Hz, so that the mapping error will be measured in Barks rather than Hz.

There are several formulae that produce approximations to the bark/Hertz curve. However, Traunmüller's approximation (1) is the most suitable for speech analysis applications. Speech is rarely digitized at a rate greater than 16 kHz. This allows for analysis up to 8 kHz, a range within which Traunmller's approximation is easily the best performer despite being the simplest formula.

$$B = \frac{26.81}{1 + (1960/f)} - 0.53 \tag{1}$$

Where B is in Bark, f in Hertz.

Within the frequency range from 0.2 to 6.8 kHz, the values calculated with this equation deviate less than 0.05 Bark.

The figure (1) below is a plot of Hertz frequencies and their bark equivalents, according to Traunmüller's approximation. The crosses on the plot correspond to the standard rounded bark scale.

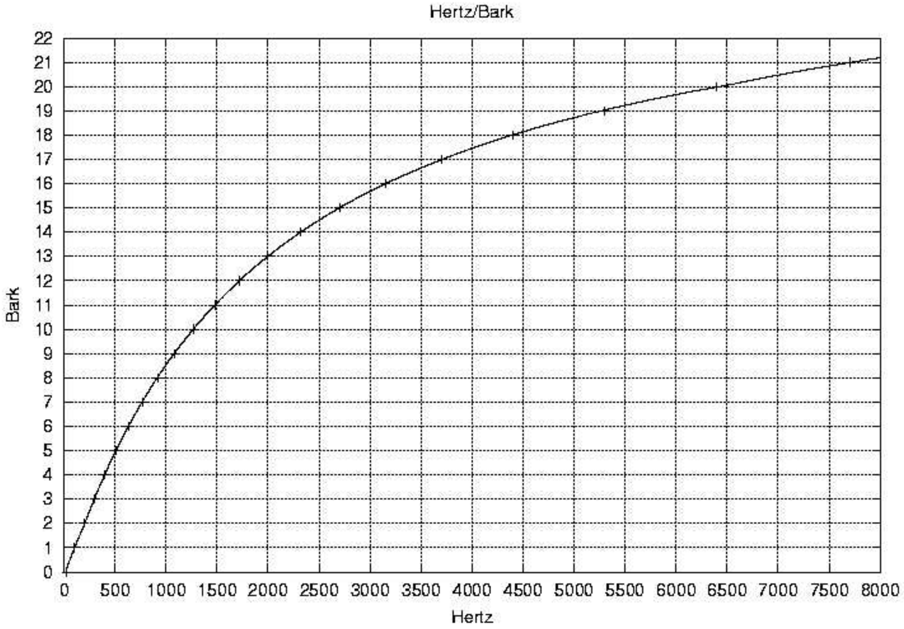

Figure 1: Barkcurve

The inverse transformation of equation (1) is follow:

$$F = \frac{1960}{26.81/(B + 0.53) - 1} \tag{2}$$

2.1.2 Formant transformations

Since equivalence in phonetic quality is defined by the perceiver, it may be preferable to adopt a genuinely perceptual point of view in describing the relation between phonetic quality and characteristic frequencies of vowels. We should account for the fact that listeners are able to decide on the phonetic quality of vowel sound even without prior exposure to any other vowels produced by the same speaker, and we would like to describe the phonetic quality of vowels in term of parameters that have the same values whenever the phonetic quality of vowels is the same. The tonotopic approach followed by Traunmüller may lead to this goal.

Fundamental to this approach is the tonotopic representation of speech sound spectra,. e.g., along the basilar membrane. The Bark scale is preferred to equivalent rectangular bandwidth (EBR) rate as a tentative measure of tonotopic position. EBR is a measure of spectral selectivity. Below 500 Hz, where the two scales are not proportional, auditory temporal resolution appears to contribute significantly to this selectivity.

The tonotopic approach suggests the same vowels produced by different speakers, with the same type of phonation, to share an invariant pattern and it suggests a simple relation to hold between the different speakers. According to this approach, the consequences of vocal tract size differences between speakers can be described very simply by a uniform tonotopic translation of the spectral envelope of the auditory pattern of excitation.

H. Traunmüller found an exact equation (3) for this translation between any speaker classes a and b.

$$Z_b = d0 + c * Z_a \tag{3}$$

In this equation, d0 is the intercept at $Z_a = 0$ Bark, and c is the slope of the regression line in a plot of Z_b vs. Z_a. Values for d0 abd c are listed in the Table below. Equation (3) can also be used to describe the relation between normally phonated and shouted or whispered vowels.

Z_a	Z_b	d0	c
men	boys,mature	0.43	1.030
men	boys,immature	1.5	0.976
men	children	2.52	1.011
woman	girls	0.36	0.984
woman	children	1.66	0.976
men	women	0.94	1.029
voiced	whispered(men)	1.78	0.882
voiced	whispered(women)	1.40	0.912

2.2 Fundamental frequency F0 transformations

Fundamental frequency F0 is shown to be subject to a similar kind of transformation when a speaker varies his degree of liveliness.

F0 is another paralinguistic variable, the degree of liveliness of speech, which can be seen as "prosodic explicitness". Acoustically, increased liveliness is reflected in increased F0-excursions towards higher frequencies, while the low frequency end of F0-range is not much affected. The width of the F0-range is affected by various attitudinal and emotional factors. The emotional continuum sad - happy is the clearest example. Emotionally depressed speakers produce speech with very little variation on F0. Increase liveliness, in general, reflects an excites emotional state if the speaker. The use of increased prosodic explicitness for the expression of a favorable attitude is clearly seen in speech directed to infants. The width of the F0-range also differentiates various modes of speech, such as conversation, reading aloud, and acting.

The relationship between the F0-contours of linguistically identical utterances produced with different degree of prosodic explicitness can also be described as a linear transformation on a tonotopic scale of frequency. With this reason we used the same translation equation (3).

2.3 Resonance bandwidth transformations

To translate the resonance bandwidth between two speaker classes, the original- and transformed formants are needed. The equation (4) describes this relation between bandwidth and formant.

$$\frac{bw}{f} = \frac{bw\_tran}{f\_tran} = const \tag{4}$$

In this equation, bw, f are bandwidth and formant before transformation and bw_tran, f_tran are bandwidth and formant after transformation. The division of bw and formant must be constant for each formant and fundamental frequency F0.

3 Method formant estimation and transformation

3.1 Introduction

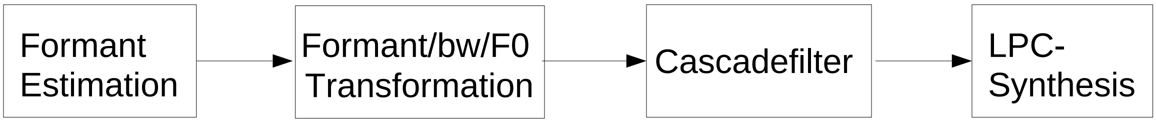

Figure 2: formant estimation

The first method has been shown in the form of a block diagram above. There are total 4 steps: formants estimation; formants, bandwidths, fundamental frequency transformation; Cascadefiler and LPC-Synthesis.

3.1.1 Formant estimation with ESPS-formant

In this Method we used software names ESPS-formant to determinate the formant frequencies and fundamental frequency. The original speech signal has to be at first divided into many frames. Here the frame size is 0.049ms and the frame shift is 0.01ms. ESPS-formant estimates speech formant trajectories, fundamental frequency and related information for each frame. In particular, for each frame of sampled data in this method, formant estimates 4 formant frequencies ($F1 \sim F4$), their formant bandwidths, pole frequencies corresponding to linear predictor coefficients, and voicing information (fundamental frequency, voiced/unvoiced decision, rms energy, first normalized autocorrelation, and the formant first reflection coefficient).

If only F0 analysis is desired, the new and better F0 estimation program get_f0 should be used, since it is faster, more accurate and more convenient to use. get_f0 processes data in stream mode, and so has no constraints on the length of the input data sequence (or file).

Dynamic programming is used to optimize F0 and formant trajectory estimates by imposing frequency continuity constraints. The formant frequencies are selected from candidates proposed by solving for the roots of the linear predictor polynomial computed periodically from the speech waveform. The local costs of all possible mappings of the complex roots to formant frequencies are computed at each frame based on the frequencies and bandwidths of the component formants for each mapping.

The input file infile is a sampled-data file—typically an ESPS FEA-SD file, though other formats are accepted as well (see get-feasd-recs). Formant produces various output files with the same file name body as infile (the name body result from removing the last of any extensions e.g., the name body of "foo.sd" is "foo"), but with different extensions. Voicing information is stored in a FEA file with extension ".f0", formants and bandwidths are stored in a FEA file with extension ".fb", and pole frequencies are stored in an ASCII file with extension ".pole". The advantage of ESPS- formant is easy to get all formants, their bandwidths, fundamental frequency and the gain.

After getting all formants, their bandwidths, fundamental frequency, a C program names get_f0_formants should be used. This program reads F0, prob. voice, gain, VoiThr, all formants and bandwidths from the given files and writes these values as binaries to an output file. The input file $< fb - file >$ is the result of the ESPS-formant command whereas $< f0 - file >$ was generated by ESPS_get_f0. The records of $< fb - file >$ contain an even number of doubles, i.e. 4 values for the formants and 4 values for the corresponding bandwidths. A record of $< f0 - file >$ always contains 4 doubles.

The output file of get_f0_formants must be read by a matlab function names read_f0_formants (filename). This function reads the data stored in 'filename' in an M*N-matrix F. Each row of matrix F contains the following elements: F0, prob_voice, gain, VoiThr,$F1 \sim F4$, $bw1 \sim bw4$. The number of rows N and the number of columns M of F is returned as well.

3.1.2 Formants, bandwidths and fundamental frequency transformation

From chapter 3.1.1 we got formant frequencies, bandwidths and fundamental frequency. We used equation (1) to transform them in Bark then with help of equation (3) calculate the transformed formant frequencies and fundamental frequency, also in Bark. Finally we transform them into frequency using equation (2).

For the bandwidth transformation equation (4) is available.

3.1.3 Cascadefilter

A digital resonator is a second-order difference equation. The transfer function of a digital resonator has a sampled frequency response given by

$$T(f) = \frac{A}{(1 - BZ^{-1} - CZ^{-2})} \tag{5}$$

C = -exp (-2*PI*bw/fs);
B = 2*exp (-PI*bw/fs)*cos(2*PI*freq/fs);
A = 1-C-B;

where:
Z = exp(j*2*PI*f*T);
j: an imaginary number corresponding to the square root of -1;
fs: sampling frequency;
f: frequency in Hz and ranges from 0 to 5 KHz;
bw: bandwidths;
freq: formant.

To lead to this goal we wrote a matlab program: [b, a, cmax] = cascadecoeff (freq, bw, fs). This program computes the filter coefficients of the cascaded formant synthesizer with formant frequencies freq, bandwidths bw and sample rate. The filter coefficients are returned in b (nominator coefficients) and a (denominator coefficients). The index of the highest denominator coefficient is returned in cmax.

3.1.4 LPC-Synthesis

LPC is a method that codes and decodes a language. Each transformed frame has been restructured as a speech signal using LPC-Synthesis. The frame size and the frame shift are same as ESPS. A matlab program for this pitch-excited LPC synthesis is available from the exercise 6 of lecture SPV1:

[so, do, zo] = LPC-synthesis(G, P, i, A, ns, zi)

Where :
G: gain factor (RMS of prediction error) from ESPS-formant
P: pitch period (number of samples) from ESPS-formant
di: delay of 1st pitch pulse (from previous frame)
A: sythesis filter coeffs (A(1) = 1)
ns: number of samples to be synthesized
zi: filter state variables (initial)
so: synthesized speech signal
zo: filter state variables (final)
do: delay of 1st pitch pulse (for next frame)

At last we stored the transformed speech signal s in a outputfile using matlab program store_raw_signal_seg (s, outputfile, 'new'). The new signal can be heard using UNIX command S16play with the playing frequency 16 KHz. S16play sends all or a portion of one or more ESPS, SIGnal, NIST or header less sampled data files to a Sun Sparc dbri digital-to-analog converter.

3.2 Conclusion

The result of this method is partially good. Especially for speaker class men transformed into class women and into class boys. The transformed speech signal is clear and natural. However, class women transform into class men and class girls are very bad, the transformed speech signal is not clear. There are probably two reasons responding this result.

The first reason is the lack of the ESPS-formant. ESPS-formant estimates the formant frequencies and fundamental frequency by lower frequency very well, so the result of transformation from men into women and boys are satisfied. But obvious ESPS-formant cannot estimate the formant frequencies and fundamental frequency by higher frequency very well as by lower frequency. It results the bad quality of the transformation from women into men and girls.

The second reason is the lower order of the cascade filter we used. For example, if sampling frequency is 8 KHz, order 12 is needed. In this work the sampling frequency of original speech signal is 16 KHz, so order 20 24 of the cascade filter is expected. However, for 4 formants only cascad filter with order 8 is available.

4 Method Pole transformation

4.1 Introduction

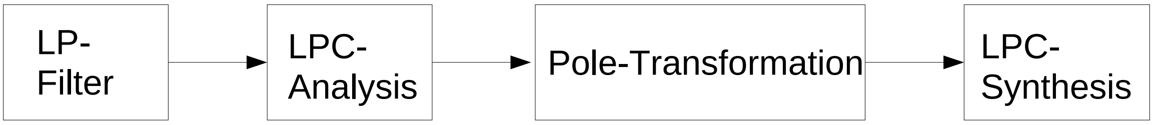

Figure 3: Pole transformation

From the failed method in chapter 3 we knew the exact estimation of formants and bandwidths are extreme important for paralinguistic transformation. To improve the exactitude of formants and bandwidths we decided to adopt H.Traunmüller's method. This method can be studied from his paper "Paralinguistic speech signal transformations". It has been also shown in the form of a block diagram above. There are total 4 steps: LP-Filter, LPC-Analysis, Pole-Transformation, LPC-Synthesis.

4.1.1 LP-Filter

The original speech signal were passed through an anti-aliasing LP- Filter with order 19 and digitalized at a sampling frequency of 16 KHz. According to some preliminary experimentation from H.Traunmüller, the limiting frequency of the LP-Filter was lowered from 8 kHz to 6.3 Hz for females and to 5 KHz for male speakers, but the sampling frequency was kept at 16 KHz throughout.

To implement this LP-Filter, a Butterworth digital and analog filter has been used. In this work this filter designs an 30 order lowpass digital Butterworth filter and returns the filter coefficients in length 31 vectors B (numerator) and A (denominator). The coefficients are listed in descending powers of z. The cutoff frequency Wn must be $0.0 < Wn < 1.0$, with 1.0 corresponding to half the sample rate. The coefficients B and A from Butterworth filter will be used as inputs of filter(B,A,X). This filter(B,A,X) filters the data in vector X with the filter described by vectors A and B to create the filtered data Y. The filter is a "Direct Form II Transposed" implementation of the standard difference equation:

$$\begin{aligned} a(1) * y(n) &= b(1) * x(n) + b(2) * x(n-1) + ... + b(nb+1) * x(n-nb) \\ &\quad - a(2) * y(n-1) - ... - a(na+1) * y(n-na) \end{aligned}$$

If a(1) is not equal to 1, filter normalizes the filter coefficients by a(1).

4.1.2 LPC-Analysis

After the anti-aliasing LP-filter in 4.1.1 the speech signal thereafter subjected to LPC-Analysis with pitch detection. At first the original signal must be divided into a lot of frames. Here the frame size is 0.049ms and the frame shift is 0.01ms. Each frame must be then analyzed by LPC_Analysis. LPC_Analysis window the input signal using Hamming window then evaluates predictor coefficients using autolpc. For pitch detection a threshold value is desired. The outputs of LPC-Analysis are gain factor, pitch period and coefficient of inverse filter.

To implement this LPC-Analysis we used a matlab program from the exercise 5 of lecture SPV1:

[G,mx,P,A] = LPC_Analysis(s,p,Pmin,Pmax,vthr)

$Where:$
s: input signal
p: predictor order
Pmin: minimum pitch period
Pmax: maximum pitch period
vthr: voiced threshold in normalized autocorrelation
G: gain factor (RMS of prediction error)
mx: maximum of R(i), i = pmin...pmax
P: pitch period (number of samples)
A: coefficients of inverse filter

4.1.3 Pole-Transformation

This function transforms the poles of LPC filter according to the Paralinguistic transformations by H. Traunmller and computes a transformed polynomial. If the LPC filter has order 24, the number of pole should be 23. For each pole we calculate its absolute value and its angle as formant frequency. Poles within angles in the range [10.0 ... Sampling frequency/2] are transformed using equation (1), (3) and (2). All these poles are candidates of formants. If the angle of transformed pole is more than PI we set the angle PI for this pole while if the angle of transformed pole is negative we set the angle positive.

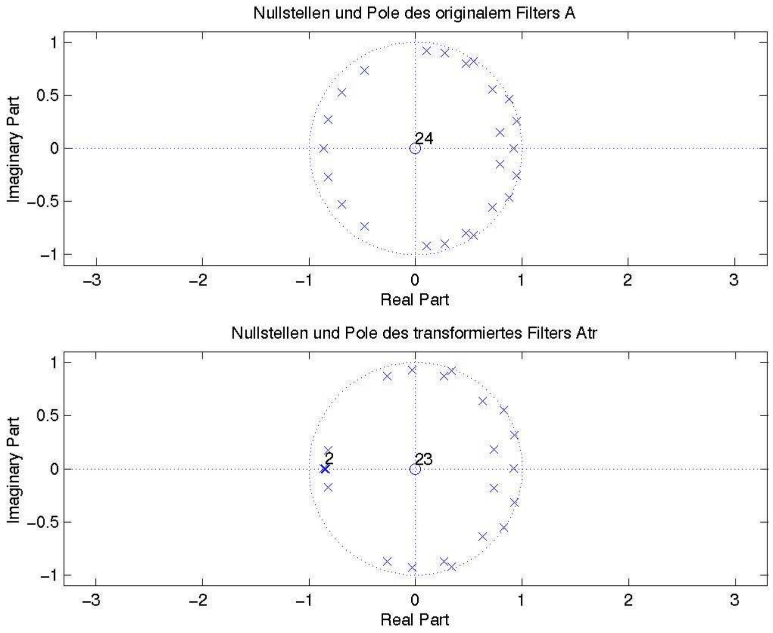

Figure 4: Pole

To calculate the absolute value we find the equation:

$$R\_tran = exp(R * f/f\_tran) \tag{6}$$

Where R and R_tran are the absolute value before and after transformation. f and f_tran are the formant before and after transformation.

For this purpose, a matlab program names transform-poles(A, type, fs) has been implemented:

[Atr] = transform-poles(A, type, fs)

Where :
A: coefficients of inverse filter from 4.1.2;
type: definition for speaker class;
1: man to women
2: women to men

3: men to boy, immature
4: men to boy, mature
5: women to girls
6: women to children
fs: Sampling frequency
Atr: coefficients of transformed polynomial

The Figure (4) above is a plot of Pole-Transformation according to transformations by H. Traunmüller. In the upper part of the figure is the pole before transformation. In the under part of the figure is the pole after transformation. As shown there are 2 transformed pole with angle PI.

4.1.4 LPC-Synthesis

Like the first method in chapter 3 we should also resynthesize the transformed frame using LPC-Synthesis. Here we use the same matlab program from the exercise 6 of lecture SPV1.

[so, do, zo] = LPC-synthesis(G, P, i, A, ns, zi)

where :
G: gain factor (RMS of prediction error)
P: pitch period (number of samples)
di: delay of 1st pitch pulse (from previous frame)
A: synthesis filter coeffs (A(1) = 1)
ns: number of samples to be synthesized
zi: filter state variables (initial)
so: synthesized speech signal
zo: filter state variables (final)
do: delay of 1st pitch pulse (for next frame)

However, the gain factor G and pitch period P must be equal the gain factor G and pitch period P from outputs of LPC-Analysis in 4.1.2.

Like method 1 at last we must store and play the transformed speech signal s using a matlab program store-raw-signal-seg (s, outputfile, 'new') and using S16play -f 16000 outputfile.

4.2 Conclusion

The result of this method is difficult to judge. Because of strong noise of transformed signal the speech is not very clearly to hear, no matter which speaker class has been transformed. There are probably some reasons responding this result.

The first reason is the higher amplitude of transformed signal.The figure (5) below is a plot of original speech signal and figure (6) is a plot of transformed speech signal. We can find obviously in the figure (6) the amplitude is so high that the signal cannot be correct resynthesized.

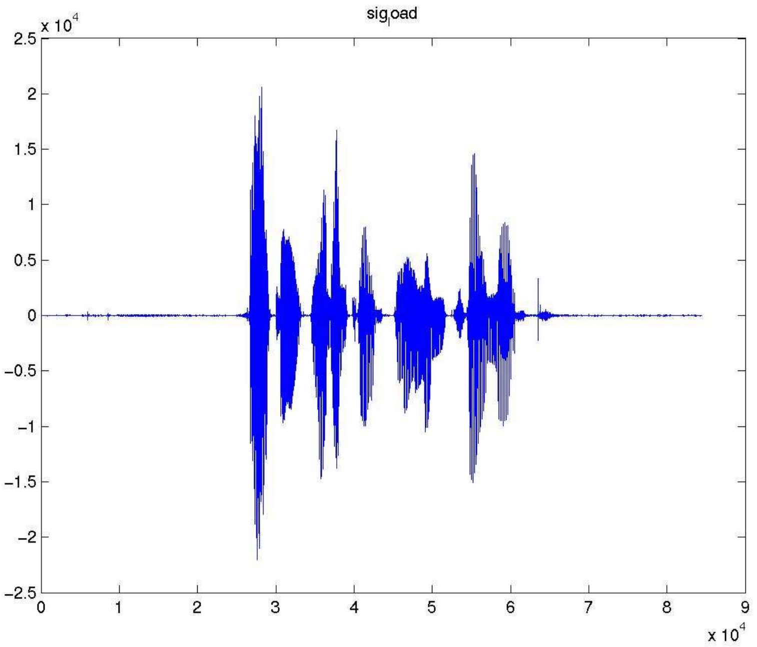

Figure 5: original Amplitute

The second reason causes the Pole-Transformation. We know, if the angle of transformed pole is more than PI we set the angle PI for this pole. From some experiences we found the pole with angle PI has very big energy. It maybe causes the strong noise.

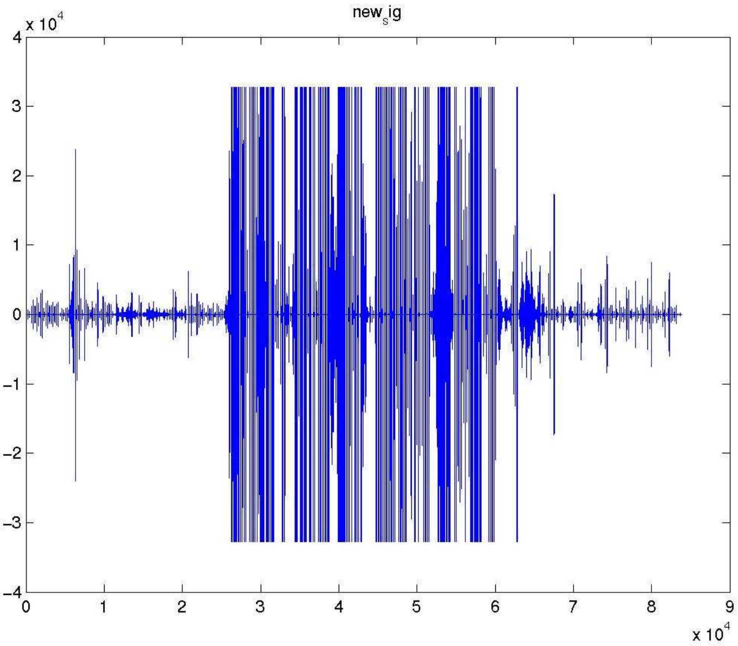

Figure 6: transformed Amplitute

5 Conclusions and Outlook

In this work I have implemented 2 methods and judged their advantage and disadvantage. The quality of both methods is not excellent comparing with H. Traunmller's experiments. The first method is though easy to implement and the result is partially good, but considering the lack of ESPS for formants estimation, this method is impossible to be compensated and must be given up. The result of second method is through not satisfied, however, considering of the lacking know-how about H. Traunmller's method, this method must be continued to explore in the future. In addition, more different speech samples must be transformed und judged.

References

[1] Cook,Cunningham,Pulleyblank,Schrijver*Combinatorial Optimization*, JOHN WILEY SONS,INC

[2] Colin R Reeves*Modern Heuristic Techniques For Combinatorial Problems* , Blacktwell Scientific Publications

[3] B.Pfister und H.-P.Hutter. *Sprachverarbeitung I.* Vorlesungsskript für das Wintersemester 2000/2001, Department Elektrotechnik, ETH Zürich, 2000

[4] Traunmüller, H.Branderud, P.Bigestan *Paralinguistic speech signal transformations.*, Technical report, Phonetic Experiment Research, Institute of Linguistics, University of Stockholm (PERILUS), December 1989.

[5] Dennis H.Klatt *Software for a casacde/parallel formant synthesizer*, MIT, Cambridge, Massachusetts 02139

[6] Julius O.smith, Jonathan S.Abel *The Bark Frequency Scale* , Center for Computer Research in Music and Acoustics (CCRMA), Stanford University